

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова  
праця на правах рукопису

**ДУБАС ЮРІЙ ВІТАЛІЙОВИЧ**

УДК 004(477): 378.091.214:378.091.39

**ДИСЕРТАЦІЯ**

**Рекомендаційна система формування індивідуальних навчальних  
траєкторій студентів ІТ-спеціальностей**

124 – «Системний аналіз»

12 – «Інформаційні технології»

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,  
результатів і текстів інших авторів мають посилання на відповідне  
джерело.

Дубас Ю.В.

Науковий керівник:

Кунанець Наталія Едуардівна

доктор наук із соціальних комунікацій, професор

Львів – 2023

## АНОТАЦІЯ

Дубас Ю.В. Рекомендаційна система формування індивідуальних навчальних траєкторій студентів ІТ-спеціальностей – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 124 «Системний аналіз» - Національний університет «Львівська політехніка», Міністерство освіти і науки України, Львів, 2023.

**Зміст анотації.** Дисертація описує побудову системи, математичних моделей та програмних засобів надання рекомендацій при формуванні індивідуальної освітньої траєкторії студентами ІТ-спеціальностей.

Розвиток освіти дедалі частіше сприймається як умова і передумова якісного розвитку економіки та соціальної сфери. Людський капітал у сучасній економіці грає дедалі більшу роль. Саме цим обґрунтовується збільшення бюджетних вкладень у систему освіти. Зростає складність людських відносин, що потребує нового рівня соціалізації молодого покоління. Підвищується складність і самої системи освіти, що висуває нові вимоги до педагогічних працівників, їх кваліфікації, і навіть до якості управління цією сферою.

Так як питання індивідуальної освіти є одним з рушіїв розвитку сучасної освіти та суспільства в цілому, створюється високий попит не лише на реформування освітніх закладів, а й засобів їх підтримки, зокрема, програмне забезпечення та інформаційні технології в цілому.

Інтеграція інформаційних технологій у освітні процеси, перш за все, вирізняється унікальністю підходу та різноманітністю імплементацій програмного забезпечення. Зважаючи також, що однією з численних переваг використання інформаційних технологій є можливість адаптації методів такого навчання до індивідуальних особливостей студентів, а також гнучкість і постійний розвиток технологій, можна стверджувати про такий підхід як

оптимальний при дослідженні освітніх процесів та реалізації методів їх підтримки.

Наукова новизна отриманих результатів полягає у:

- Вирішення проблеми технологічного супроводу процесу формування індивідуальної освітньої траєкторії для студентів ІТ-спеціальностей;
- Розробці комплексу методів аналізу даних для отримання інформації з веб-сайтів ІТ-вакансій та веб-сайтів освітнього призначення;
- Створенні моделей визначення семантичної близькості на основі методів математичної логіки;
- Розробці моделі зменшення розмірності даних при аналізі вмісту текстових документів;
- Створенні моделі визначення кращої альтернативи на основі методу аналізу ієрархій;
- Побудові системи для надання рекомендацій при формуванні індивідуальної освітньої траєкторії.

У **першому розділі** був здійснений аналіз інформації та джерел, що стосуються рекомендаційних систем. Фокус дослідження систем надання рекомендацій спрямовувався на розгляд їх особливостей, переваг та недоліків, а також на предмет використання існуючих технологій реалізації. Результати показали, що рекомендаційні алгоритми та метрики є надто складними та ресурсозатратними у повсякденному використанні, особливо для освітніх закладів. Більше того, точність наданих рекомендацій прямо залежить від об'ємів опрацьованих даних, що унеможлиблює використання алгоритмів при обмеженій інформації. Додатково був розглянутий процес формування ІОТ в межах Національного університету «Львівська політехніка», а також інформаційний супровід цього процесу і існуюче програмне забезпечення.

Аналіз інших досліджень з реалізованими системи для формування ІОТ виявив відсутність деяких необхідних функцій у створених раніше рішеннях. Таким чином, було обґрунтовано створення нової системи для надання рекомендацій.

У **другому розділі** визначені агенти процесу формування ІОТ та взаємозв'язки між ними, були наведені відповідні моделі та представлено процес формування рекомендацій у контексті системи. Даний процес складається з наступних кроків:

- Визначення профорієнтації студента;
- Збір та аналіз даних з освітніх веб-сайтів та веб-сайтів ІТ-вакансій;
- Визначення семантичної близькості між даними, що були отримані в результаті парсингу веб-сайтів ІТ-вакансій;
- Додаткове ранжування семантично близьких дисциплін за методом аналізу ієрархій;
- Зменшення розмірності даних, що отримані в результаті парсингу освітніх веб-сайтів;
- Формування списку дисциплін для рекомендацій.

Початкові кроки процесу формування рекомендацій полягають у визначенні профорієнтації студента (за методикою Дж. Холланда), визначення зв'язку з ІТ-вакансіями та подальший підбір ключових слів на основі знайдених поєднань. У дослідженні також розглянуто поняття онтології та можливість семантичних операцій з ціллю знаходження дисциплін, котрі найкраще підходили би студенту. Проте, використання онтології було обмежене до отримання ключових слів компонентів вибіркового блоку з науково-освітніх програм.

Наступним кроком є проведення ранжування з відібраних дисциплін на фоні обмеженості використання методів для роботи з семантикою. Таке ранжування дозволило би залучити у процес формування ІОТ експертів, стейкхолдерів та викладачів. Відповідно, для вирішення проблеми був обраний метод аналізу ієрархій (MAI). Його суть полягає у підтримці прийняття рішення за допомогою ієрархічної композиції завдання та рейтингування альтернативних рішень.

Останнім кроком перед кінцевим формуванням рекомендацій є проведення лінійної апроксимації багатовимірних даних з використанням методу головних компонент (PCA). Даний метод часто використовується для аналізу великих наборів даних, що містять велику кількість вимірів на одне спостереження. У системі цей метод застосовується для масиву загальноуніверситетських дисциплін.

**Третій розділ** описує практичне використання раніше описаних моделей та методів математичної логіки у експериментальних обчисленнях. Зокрема, здійснений детальний опис технології аналізу даних, що використовувалася у дисертаційному дослідженні.

Технологія аналізу даних у системі формування рекомендацій базується на використанні комплексу методів парсингу даних з освітніх веб-сайтів та веб-сайтів ІТ-вакансій. У роботі описані особливості, переваги та недоліки технології парсингу, а також відповідні методи, що використовуються у системі: *similartext* та *substr\_count* (мова веб-програмування PHP).

Для перевірки коректності роботи використаної методики був використаний метод Левенштейна. Експериментальні обчислення з випадковими вхідними даними показали, що хоча в деяких випадках використання алгоритму Левенштейна може надати більшу точність, проте час на його виконання зростає у прогресії  $O(N^3)$ , де  $N$  – довжина довшого рядка з

двох порівнювальних. Методи веб-програмування, що націлені на роботу з рядками натомість, використовують значно менше ресурсів системи і можуть здійснювати пошуку у текстових документах з значно меншими затратами часу.

Додатково були проведені експериментальні обчислення з МАІ та методом головних компонент. Ранжування дисциплін дозволило включити оцінку експертів та виділити саме ту дисципліну серед альтернатив, яка найкраще підходить студенту. Застосування методу головних компонент дозволило зменшити розмірність даних інформаційного масиву загальноуніверситетських дисциплін та визначити дисципліни з найбільшим значенням (на основі ознак) і включити їх у рекомендації для формування ІОТ студента.

У **четвертому розділі** описана технічна реалізація моделей та методів, що висвітлювалися у попередніх розділах засобами мови веб-програмування *PHP*. Детально описана «подорож» користувача у системі від початку його авторизації чи реєстрації до отримання рекомендацій та завершення роботи.

Зокрема, основний фокус розділу зроблений на архітектурі системи та її роботі завдяки модульності.

Практична реалізація система представлена у вигляді веб-сайта. У розділі також наведені скріншоти з користувацьким інтерфейсом та покровою експлуатацією веб-застосунку користувачем.

**Ключові слова:** рекомендаційна система, індивідуальна освітня траєкторія, індивідуальний навчальний план, інформаційно-комунікаційні технології, парсинг, онтологія, семантична близькість, метод аналізу ієрархій, метод головних компонент

**Список публікацій здобувача:**

**Наукові праці, в яких опубліковані основні наукові результати дисертації:**

*Публікації у наукових фахових виданнях України:*

1. Дубас Ю. В., Кунанець Н. Е. Застосування методу багатовимірної середньої при формуванні індивідуальних навчальних траєкторій студентів ІТ-спеціальностей. Вісник Національного університету “Львівська політехніка”. Серія: Інформаційні системи та мережі. 2020. Вип. 7. С. 70–77.
2. Dubas Y., Kunanets N. Definition of professional vector in educational systems of training future IT specialists Вісник Національного університету “Львівська політехніка”. Серія: Інформаційні системи та мережі. 2021. Вип. 9. С. 142–150.
3. Dubas Y., Kunanets N. Use of ontology in information system to assist in the formation of an individual educational trajectory of students. Вісник Хмельницького національного університету. Серія: Технічні науки. 2023. № 1 (317). С. 9–17.

*Публікації у наукових виданнях, які входять до міжнародних наукометричних баз:*

1. Dubas Y., Mudrokha V., Pasichnyk V. Information system for the formation of individual educational trajectories of students of IT specialties. CEUR Workshop Proceedings. 2021. Vol. 2851: Proc. of the 2nd intern. workshop IT project management ITPM 2021, Slavsko, Ukraine, Febr. 16-18, 2021. Vol. 1. P. 184–194.

*Публікації у наукових фахових виданнях інших держав:*

1. Dubas Y., Kunanets N. The role of the analytic hierarchy process in the formation of a student’s individual educational trajectory. Manažérska Informatika. 2022. R. 1, č. 2.

## **Наукові праці, які засвідчують апробацію матеріалів дисертації:**

1. Dubas Y., Kunanets N. The role of the analytic hierarchy process in the formation of a student's individual educational trajectory. *Manažérska Informatika*. 2022. R. 1, č. 2.

### **ABSTRACT**

Dubas Y.V. Recommendation system for the formation of individual educational trajectories of IT-specialties students – Qualifying scientific work as a manuscript.

A thesis on the degree of Candidate of Science (PhD) on specialty 124 "System Analysis" - National University "Lviv Polytechnic", Ministry of Education and Science of Ukraine, Lviv, 2023.

**Content of the abstract.** The dissertation describes the construction of a system, mathematical models and software tools for providing recommendations in the formation of an individual educational trajectory by students of IT specialties.

The development of education is increasingly perceived as a condition and prerequisite for the qualitative development of the economy and social sphere. Human capital is playing an increasingly important role in the modern economy. This is the reason for the increase in budget investments in the education system. The complexity of human relations is growing, which requires a new level of socialization of the younger generation. The complexity of the education system itself is increasing, which puts forward new requirements for teachers, their qualifications, and even for the quality of management in this area.

Since the issue of individual education is one of the drivers of the development of modern education and society as a whole, there is a high demand not only for reforming educational institutions, but also for the means of their support, in particular, software and information technology in general.



The integration of information technologies into educational processes, first of all, is distinguished by the uniqueness of the approach and the variety of software implementations. Also, taking into account that one of the numerous advantages of using information technology is the ability to adapt the methods of such teaching to the individual characteristics of students, as well as the flexibility and constant development of technologies, it can be argued that this approach is optimal in the study of educational processes and the implementation of methods to support them.

The scientific novelty of the obtained results is as follows:

- Solving the problem of technological support for the process of forming an individual educational trajectory for students of IT specialties;
- Development of a set of data analysis methods for obtaining information from IT job websites and educational websites;
- Creation of models for determining semantic proximity based on the methods of mathematical logic;
- Development of a model for reducing the dimensionality of data in the analysis of the content of text documents;
- Creation of a model for determining the best alternative based on the method of hierarchy analysis;
- Building a system to provide recommendations in the formation of an individual educational trajectory.

In the **first section**, an analysis of information and sources related to recommendation systems was carried out. The focus of the study of recommendation systems was aimed at considering their features, advantages and disadvantages, as well as the use of existing implementation technologies. The results showed that recommendation algorithms and metrics are too complex and resource-intensive in everyday use, especially for educational institutions. Moreover, the accuracy of the

recommendations provided It directly depends on the amount of data processed, which makes it impossible to use algorithms with limited information. In addition, the process of IOT formation within Lviv Polytechnic National University, as well as information support for this process and existing software were considered. Analysis of other studies with implemented systems for the formation of IOT revealed the lack of some necessary functions in previously created solutions. Thus, the creation of a new system for providing recommendations was justified.

In the **second section**, the agents of the IOT formation process and the relationships between them are identified, the relevant models are given and the process of forming recommendations in the context of the system is presented. This process consists of the following steps:

- Determination of the student's career guidance;
- Collecting and analyzing data from educational websites and IT job websites;
- Determination of semantic proximity between the data obtained as a result of parsing IT job websites;
- Additional ranking of semantically similar disciplines by the method of hierarchy analysis;
- Reducing the dimensionality of data obtained as a result of parsing educational websites;
- Formation of a list of disciplines for recommendations.

The initial steps of the recommendation process are to determine the student's career guidance (according to the method of J. Holland), determine the connection with IT vacancies and the subsequent selection of keywords based on the found combinations. The study also examines the concept of ontology and the possibility of semantic operations in order to find disciplines that would best suit the student.

However, the use of ontology was limited to obtaining the keywords of the components of the sample block from scientific and educational programs.

The next step is to rank the selected disciplines against the background of limited use of methods for working with semantics. Such a ranking would make it possible to involve experts, stakeholders and teachers in the process of IOT formation. Accordingly, to solve the problem, the method of hierarchy analysis (MAI) was chosen. Its essence is to support decision-making through the hierarchical composition of the task and the rating of alternative solutions.

The last step before the final formation of recommendations is to carry out a linear approximation of multidimensional data using the principal component method (*PCA*). This method is often used to analyze large datasets containing a large number of measurements per observation. In the system, this method is used for an array of university-wide disciplines.

**The third section** describes the practical use of the previously described models and methods of mathematical logic in experimental calculations. In particular, a detailed description of the data analysis technology used in the dissertation research is carried out.

The technology of data analysis in the recommendation generation system is based on the use of a set of methods for parsing data from educational websites and IT job websites. The paper describes the features, advantages and disadvantages of parsing technology, as well as the corresponding methods used in the system: *similartext* and *substr\_count* (*PHP* web programming language).

To check the correctness of the technique used, the Levenshtein method was used. Experimental calculations with random inputs have shown that although in some cases the use of the Levenshtein algorithm can provide greater accuracy, the time to execute it increases in progression  $O(N^3)$ , where  $N$  is the length of the longer string of the two comparatives. Web programming methods that focus on

working with strings instead use significantly fewer system resources and can search text documents with much less time.

In addition, experimental calculations were carried out using MAI and the principal component method. The ranking of disciplines made it possible to include the assessment of experts and highlight exactly the discipline among the alternatives that best suits the student. The use of the principal component method made it possible to reduce the dimensionality of the data of the information array of university disciplines and to identify the disciplines with the highest value (based on features) and include them in the recommendations for the formation of the student's IOT.

The **fourth section** describes the technical implementation of the models and methods covered in the previous sections by means of the *PHP* web programming language. The user's "journey" in the system from the beginning of his authorization or registration to receiving recommendations and completing work is described in detail.

In particular, the main focus of the section is on the architecture of the system and its operation due to modularity. The practical implementation of the system is presented in the form of a website. The section also contains screenshots of the user interface and the user's exploitation of the web application.

**Keywords:** recommendation system, individual educational trajectory, individual curriculum, information and communication technologies, parsing, ontology, semantic proximity, method of analysis of hierarchies, method of principal components

#### **References:**

**Scientific works in which the main scientific results of the dissertation are published:**

*Publications in scientific journals of Ukraine:*

1. Dubas Y. V., Kunanets N. E. Application of the multivariate average method in the formation of individual educational trajectories of students of IT specialties. Bulletin of the Lviv Polytechnic National University. Series: Information systems and networks. 2020. Issue 7. P. 70–77.
2. Dubas Y., Kunanets N. Definition of professional vector in educational systems of training future IT specialists Bulletin of the National University "Lviv Polytechnic". Series: Information systems and networks. 2021. Issue 9. P. 142–150.
3. Dubas Y., Kunanets N. Use of ontology in information system to assist in the formation of an individual educational trajectory of students. Bulletin of the Khmelnytskyi National University. Series: Technical sciences. 2023. No. 1 (317). P. 9–17.

*Publications in scientific journals included in international scientometric databases:*

1. Dubas Y., Mudrokha V., Pasichnyk V. Information system for the formation of individual educational trajectories of students of IT specialties. CEUR Workshop Proceedings. 2021. Vol. 2851: Proc. of the 2nd intern. workshop IT project management ITPM 2021, Slavsko, Ukraine, Febr. 16-18, 2021. Vol. 1. R. 184–194.

*Publications in scientific journals of other countries:*

1. Dubas Y., Kunanets N. The role of the analytical hierarchy process in the formation of a student's individual educational trajectory. Manažérska Informatika. 2022. R. 1, ch. 2.

**Scientific works certifying the approval of the dissertation materials:**

1. Dubas Y. Inclusion of Disciplines by Using Analytic Hierarchy Process (AHP) / Yurii Dubas, Natalia Kunanets //Computational Linguistics and Intelligent Systems. Proceedings of the 6th International Conference on

COLINS 2022. Volume II: Workshop. Gliwice, Poland, May 12-13, 2022,  
pp. 39-49 (ISSN 2523-4013)

## ЗМІСТ

ВСТУП.....	21
Розділ I. Аналіз інформаційних технологій для формування рекомендацій.....	29
1.1 Характеристика рекомендаційних системи.....	29
1.2 Проблема формування індивідуальної освітньої траєкторії та її технологічний супровід.....	41
1.2.1 Процес формування індивідуальної освітньої траєкторії.....	41
1.2.2 Аналіз технологічного супроводу освітніх процесів.....	44
1.3 Аналіз моделей та методів для формування рекомендацій навчальних дисциплін з їх подальшим включення у ІОТ.....	49
1.4 Постановка задачі дослідження.....	54
Висновки до розділу I.....	55
Розділ II. Моделі Рекомендаційної системи.....	57
2.1 Формальна модель системи формування індивідуальної освітньої траєкторії.....	57
2.2 Модель формування рекомендацій навчальних дисциплін для включення у ІОТ.....	60
2.3 Модель аналізу даних.....	63
2.3.1 Модель визначення профорієнтації за методикою Голланда.....	66
2.3.2 Модель процесу визначення семантичної близькості.....	69
2.4 Модель процесу ранжування дисциплін з використанням методу аналізу ієрархій.....	72

2.5	Методи оцінки розмірності та ефективності системи.....	79
	Висновки до розділу II.....	84
Розділ III. Методи комплексного аналізу.....		86
3.1	Метод аналізу даних з веб-сайтів ІТ-вакансій.....	86
3.1.1	Аналіз даних ІТ-вакансій.....	94
3.2	Знаходження семантичної близькості на основі методів роботи з даними.....	98
3.2.1	Перевірка роботи алгоритму знаходження семантичної близькості.....	101
3.3	Застосування методу головних компонент.....	108
3.4	Експериментальні обчислення за методом аналізу ієрархій для ранжування дисциплін.....	117
	Висновки до розділу III.....	126
Розділ IV. Програмні особливості рекомендаційної системи.....		128
4.1	Функціональне призначення.....	128
4.2	Структурно-функціональна модель веб-застосунку.....	131
4.3	Особливості програмної реалізації.....	135
4.3.1	Особливості реалізації бази даних.....	140
4.3.2	Опис реалізації користувацького інтерфейсу.....	144
4.3.3	Технічні характеристики.....	149
	Висновки до розділу IV.....	150
ВИСНОВКИ.....		151
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....		153



## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ІОТ – індивідуальна освітня траєкторія;

ІНП – індивідуальний навчальний план;

ІКТ – інформаційно-комунікаційні технології;

ІС – інформаційна система;

МАІ – метод аналізу ієрархій;

РСА – метод головних компонент;

БД – база даних.

## ВСТУП

**Актуальність теми.** Розвиток освіти дедалі частіше сприймається як умова і передумова якісного розвитку економіки та соціальної сфери. Людський капітал у сучасній економіці грає дедалі більшу роль. Саме цим обґрунтовується збільшення бюджетних вкладень у систему освіти. Зростає складність людських відносин, що потребує нового рівня соціалізації молодого покоління. Підвищується складність і самої системи освіти, що висуває нові вимоги до педагогічних працівників, їх кваліфікації, і навіть до якості управління цією сферою. Водночас здобуття освіти потребує все більших державних та приватних витрат. У бідних країнах та таких, що розвиваються, зазначене зростання зумовлене збільшенням охоплення дітей та молоді освітою (вона стає більш доступною), у розвинених країнах — зростанням якості та різноманітності освітніх траєкторій, індивідуалізацією освітніх програм.

Зокрема, поняття Індивідуальної освіти в Україні, як такого, не існувало і лише нещодавно був прийнятий відповідний закон. Визнання права здобувачів освіти на індивідуальну освітню траєкторію є одним із прогресивних нововведень Закону «Про освіту» (2017), що передбачає «персональний шлях реалізації особистісного потенціалу здобувача освіти, що формується з урахуванням його здібностей, інтересів, потреб, мотивації, можливостей і досвіду, ґрунтується на виборі здобувачем освіти видів, форм і темпу здобуття освіти, суб'єктів освітньої діяльності та запропонованих ними освітніх програм, навчальних дисциплін і рівня їх складності, методів і засобів навчання» [1].

Такий закон був необхідним, адже в наш час пріоритетною ціллю стає розвиток компетентних фахівців. Необхідними якостями для таких фахівців є здатність до саморозвитку та самореалізації. Забезпечення цих факторів є можливим з орієнтацією на освітні процеси, що передбачають врахування можливостей кожного індивіда, їх потреб та свободи вибору. Таким чином, в наш час кожен здобувач освіти може сподіватись на отримання освіти та вибір індивідуальної освітньої траєкторії (ІОТ).

Питання інтеграції ІОТ в освіту також розглядав І. Каньковський у дослідженні [2]. Він вивів залежність, котра дозволяє стверджувати, що навіть при однакових знаннях фундаментальних освітніх об'єктів освітні продукти різних учнів або студентів різні, оскільки засвоєні ними види діяльності й рівень їхнього розвитку відрізняються. Завдяки цій залежності стає зрозумілим необхідність і неминучість введення в процес навчання індивідуальної освітньої траєкторії [2].

Так як питання індивідуальної освіти є одним з рушіїв розвитку сучасного освіти та суспільства в цілому, створюється високий попит не лише на реформування освітніх закладів, а й засобів їх підтримки, зокрема, програмне забезпечення та інформаційні технології в цілому.

Використанням інформаційних технологій в закладах освіти зараз уже нікого не здивуєш і це не є чимось абсолютно новим чи революційним. Проте це зовсім не зменшує потенціал використання таких засобів, особливо на фоні його постійного збільшення та попиту, що тільки зароджується в Україні [3]. Особливу увагу необхідно звернути на мережеві технології чи засоби Інтернет, які зазвичай використовуються з метою пошуку інформації. Надання

здобувачу освіти можливість використання таких технологій також не можна назвати новаторським підходом, адже це активно здійснюється на протязі останнього десятиліття, якщо не більше. Проте інтеграція інформаційних технологій у освітні процеси, перш за все, вирізняється унікальністю підходу та різноманітністю імплементацій програмного забезпечення. Зважаючи також, що однією з численних переваг використання інформаційних технологій є можливість адаптації методів такого навчання до індивідуальних особливостей студентів, а також гнучкість і постійний розвиток технологій, можна стверджувати про такий підхід як оптимальний при дослідженні освітніх процесів та реалізації методів їх підтримки.

**Зв'язок роботи з науковими програмами, планами, темами.** Тема дисертації відповідає науковому напрямку «Дослідження, розробка та впровадження Internet-застосувань, інтелектуальних розподілених інформаційних технологій та систем на основі ресурсів баз даних, проведення оцінки наявних технологій та знань з метою прискорення процесів формування сучасного інформаційного суспільства» та Держбюджетній темі «Методи та засоби нейронечіткого управління групою мобільних робототехнічних платформ» Національного університету «Львівська політехніка».

**Мета і завдання дослідження.** Метою дисертаційної роботи є розроблення рекомендаційної системи для формування рекомендацій з їх подальшим включенням у індивідуальну освітню траєкторію студентів ІТ-спеціальностей.

Мета дисертаційної роботи ставить у необхідність виконання таких завдань:

- аналіз особливостей, недоліків та переваг наявних інформаційних технологій та систем, що використовуються для формування рекомендацій з їх подальшим включенням у ІОТ;
- побудова методу визначення схильностей до професії ІТ галузі;
- вдосконалити метод аналізу даних з веб-сайтів ІТ-вакансій.
- розробити метод відбору значущих чинників при побудові індивідуальної освітньої траєкторії на основі даних визначення семантичної близькості текстових документів;
- розробленні інформаційної технології аналізу даних шляхом зменшення їх розмірності з втратою найменшої кількості інформації;
- створити підхід до ранжування рекомендаційних елементів з використання методу аналізу ієрархій;
- розроблення рекомендаційної системи та апробування отриманих рекомендацій.

**Об’єкт дослідження** – процеси інформаційно-аналітичного супроводу процесів прийняття рішень при відборі дисциплін для формування індивідуальної освітньої траєкторії

**Предмет дослідження** – моделі, методи та інформаційна технологія аналізу неоднорідних даних для відбору дисциплін на основі порівняння коротких вибірок даних.

**Методи дослідження.** Для розв’язання поставлених в дисертаційній роботі завдань використані наступні методи:

- Для формального опису рекомендаційної системи - методи сформовані на основі теорії графів та системного аналізу;

- Для визначення професійних схильностей – методи математичної логіки;
- Для отримання інформації про ІТ-вакансії на ринку праці – методи інформаційного аналізу даних та парсингу;
- Для визначення семантичної близькості – семантичні методи та бібліотеки мови веб-програмування *PHP*;
- Для зменшення розмірності даних без значних втрат інформації – метод головних компонент (*PCA*);
- Для ранжування дисциплін – метод аналізу ієрархій.

**Наукова новизна одержаних результатів.** Наукова новизна дисертаційної роботи полягає у наступному:

*Вперше створено:*

- інформаційну технологію, що використовує метод головних компонент для зменшення розмірності даних при аналізі вмісту текстових документів;

*Удосконалено:*

- методи визначення професійних схильностей студентів з використанням методів математичної логіки;
- інформаційну технологію аналізу даних з освітніх веб-сайтів та веб-сайтів ІТ вакансій;
- метод аналізу даних шляхом визначення семантичної близькості текстових документів;
- модель процесу визначення кращої альтернативи серед дисциплін для включення у ІОТ на основі методу аналізу ієрархій.

*Побудовано:*

- рекомендаційну систему для формування індивідуальної освітньої траєкторії для студентів ІТ-спеціальностей.

**Практичне значення одержаних результатів:** у ході дослідження було розроблено рекомендаційну систему, яка використовує веб-технології для формування рекомендацій з можливістю їх подальшого включення у ІОТ. Розроблена система може використовуватися для надання персоналізованих рекомендацій користувачам щодо вибору дисциплін при формування ІОТ.

Результати досліджень впроваджені з використанням серверних технологій, скриптової мови програмування та сервісу хостингу у вигляді веб-сайту і можуть бути використаними будь-якими відвідувачами мережі інтернет.

**Особистий внесок здобувача.** Усі наукові результати дисертаційної роботи отримані автором самостійно. У друкованих працях, опублікованих у співавторстві, внесок здобувача наступний: використання методу багатовимірної середньої у процесі формування ІОТ [4]; визначення професійного вектора у освітній системі при підготовці ІТ-спеціалістів [5]; розроблення інформаційної системи при формування ІОТ для студентів ІТ-спеціальностей [6]; використання онтології в інформаційній системі для формування ІОТ студентів [7]; визначення ролі МАІ у процесі формування ІОТ студента [8].

**Апробація результатів дисертації.** Основні наукові та практичні результати роботи доповідались та обговорювались на міжнародних наукових та науково-практичних конференціях, зокрема:

- Proceedings of the 2nd International Workshop IT Project Management (ITPM) 2021, Slavske, Ukraine, Febr. 16-18, 2021. Vol. 1.
- Computational Linguistics and Intelligent Systems. Proceedings of the 6th International Conference on COLINS 2022. Volume II: Workshop. Gliwice, Poland, May 12-13, 2022

**Структура та обсяг дисертації.** Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 78 найменувань та двох додатків. Загальний обсяг дисертації становить 148 сторінок, а також 48 рисунків та 13 таблиць.



# **РОЗДІЛ I. АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ**

У першому розділі проведено аналіз актуальних наукових досліджень та інформаційних технологій розроблення рекомендаційних систем та формування індивідуальних освітніх траєкторій. Здійснено постановку завдання дослідження, а також подано загальну характеристику рекомендаційної системи та проаналізовано існуючі моделі для формування індивідуальних освітніх траєкторій.

## **1.1 Характеристика рекомендаційних системи.**

В дисертаційному дослідженні в якості інформаційної системи для формування рекомендацій навчальних дисциплін було обрано рекомендаційну систему. Системи даного типу здійснюють фільтрацію інформації з метою визначення об'єктів, яким користувач може надати перевагу базуючись на його уподобаннях.

Рекомендаційні системи виникли відносно недавно. 1990 року технологію вперше згадав [9] шведський учений Юссі Карлгрен, описавши її як «цифрову книжкову полицю». Ця робота лягла основою його майбутніх досліджень.

У 2000 роках алгоритми рекомендацій почали проникати у сферу електронної комерції. Одним з піонерів у цій галузі є онлайн-ритейлер Amazon. У 2006 році компанія Netflix, яка займалася на той час прокатом DVD-дисків

за підпискою, запустила конкурс на найкращий рекомендаційний алгоритм [10] із призовим фондом у \$1 млн. Для його отримання незалежним розробникам необхідно було покращити точність алгоритму рекомендацій на 10%. 2009 року приз вручили команді BellKor's Pragmatic Chaos. У 2010 роках рекомендаційні системи з'явилися у соціальних мережах. На сьогодні більшість популярних платформ відмовилися від використання хронологічної стрічки на користь алгоритмічної.

Рекомендаційна система це підклас систем фільтрації інформації, які надають пропозиції щодо елементів, які найбільше стосуються конкретного користувача. [11][12] Як правило, пропозиції стосуються різних процесів прийняття рішень, наприклад, який продукт купити, яку музику слухати чи які онлайн-новини читати [11]. Рекомендаційні системи особливо корисні, коли людині потрібно вибрати товар із потенційно величезної кількості товарів, які може запропонувати послуга. [11][13]

Рекомендації також можна впроваджувати і у різні канали комунікації. Вони можуть бути представлені у вигляді вітрини на головній сторінці веб-сайту, у кошику, на картці товару або в пошуку. Алгоритми активно використовують і у email-маркетингу: на їх основі формуються ланцюжки листів, які радять нові, додаткові товари, або схожі на ті, які користувач вже придбав.

Системи рекомендацій часто застосовуються у різноманітних онлайн ресурсах, зокрема таких, що надають користувачу персоналізовані рекомендації серед товарів чи послуг. Наприклад, для створення попиту при покупці товару необхідно проаналізувати поведінку користувача при

відвідуванні веб-ресурсу, дізнатися яким товаром він цікавиться, що переглядає, як довго затримувався на сторінках, чи переглядав акційні пропозиції, тощо. Подібний аналіз даних вимагає потужного технологічного супроводу, що здатен забезпечити виконання різнопланових пошукових завдань. У пошуку присутній запит, контекст (уподобання користувача і його характеристики), а також елементи, які необхідно ранжувати. У рекомендаціях все відбувається аналогічно, проте під запитом розуміється сам користувач та його вподобання.

Зокрема, можна провести паралелі з пошуковими алгоритмами, адже це сукупність правил, відповідно до яких пошукові системи оцінюють релевантність сторінок запитам та будують пошукову видачу. Основні принципи таких алгоритмі зображені на рисунку 1.1. Алгоритми ранжирування всіх комерційних пошукових систем є засекреченими. Тим не менш, з досвіду роботи оптимізаторів та їх багаторічної практики відомо, що алгоритми використовують деякий набір параметрів та факторів для перевірки веб-сторінок та визначення їх релевантності для користувача при пошуку.

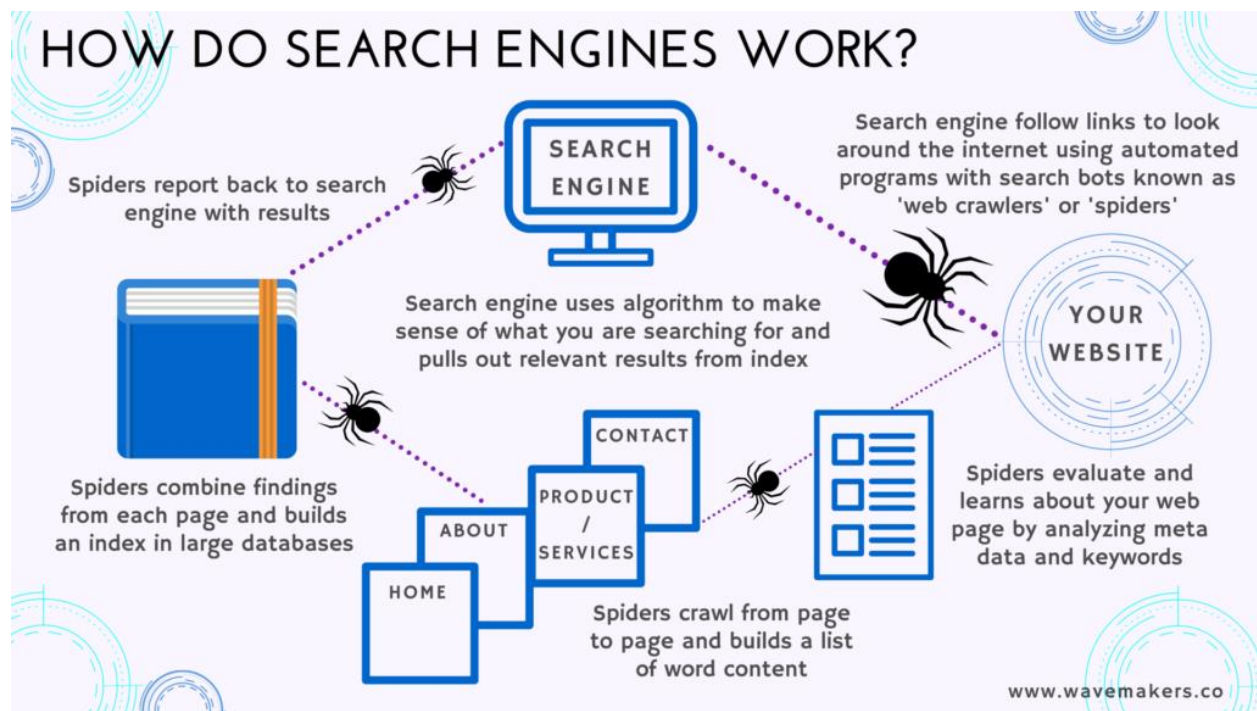


Рис. 1.1. Принципи роботи пошукових алгоритмів [14].

Пошукові алгоритми за принципом дії можна поділити на дві групи:

- Алгоритми прямої дії. Щоб відповісти на запит користувача, алгоритми прямої дії перебирають усі документи, що зберігаються в індексі пошукової системи. Такі алгоритми формують максимально релевантну видачу, але вимагають величезних обчислювальних ресурсів та є дуже часозатратними. Наразі великі пошукові системи їх не використовують, або використовують вкрай обмежено.
- Алгоритми інвертованого (зворотного) індексу. Зворотний (інвертований) індекс – це структура даних, де для кожного терміну перераховані посилання на документи, що його містять. Алгоритм пошуку за

зворотним індексом звертається до такої бази та одразу отримує повний список релевантних документів.

Пошукові алгоритми іноді застосовуються і у системах рекомендацій, наприклад, алгоритми пошуку подібних об'єктів. Таке використання є доцільним, адже загалом рекомендаційні алгоритми працюють схожим чином при оцінці релевантності контенту для користувача, хоча і не завжди використовують явні методи для збору даних, наприклад, пошуковий запит користувача.

Таким чином, рекомендаційні системи можна вважати зручною альтернативою системам, що використовують алгоритми пошуку, або навіть доповненням до таких систем, оскільки вони дозволяють виявити об'єкти не лише завдяки взаємодії користувача з системою (запити), а й аналізуючи історію пошуку, перегляду чи покупок. Також варто відмітити, що рекомендаційні системи доволі часто використовують пошукові машини для індексації незвичайних даних.

Існує кілька типів моделей рекомендаційних систем, які використовуються для прогнозування та надання персоналізованих рекомендацій користувачам. Ось декілька з них:

Колаборативний фільтр: Ця модель рекомендацій використовує спільність між користувачами або предметами для здійснення рекомендацій. Існують два типи колаборативного фільтру: на основі користувачів і на основі предметів. Модель на основі користувачів рекомендує предмети, що сподобалися подібним користувачам, тоді як модель на основі предметів рекомендує користувачам подібні предмети до тих, які вони вже сподобалися.

Змішана модель: Цей підхід поєднує кілька методів рекомендаційних систем для досягнення кращих результатів. Він може комбінувати колаборативний фільтр з контентним фільтром, що аналізує характеристики або описи предметів, або використовувати гібридні моделі, які поєднують різні алгоритми рекомендаційних систем.

Контентний фільтр: Ця модель рекомендаційних систем використовує характеристики або описи предметів та враховує особисті інтереси користувачів для рекомендацій. Наприклад, якщо користувач сподобався фільм з певним жанром, система може рекомендувати інші фільми того ж жанру. На рис. 1.2 і 1.3 проілюстрована різниця між двома найпопулярнішими підходами до реалізації фільтрування у рекомендаційних системах та наведений приклад *user-item* матриці.

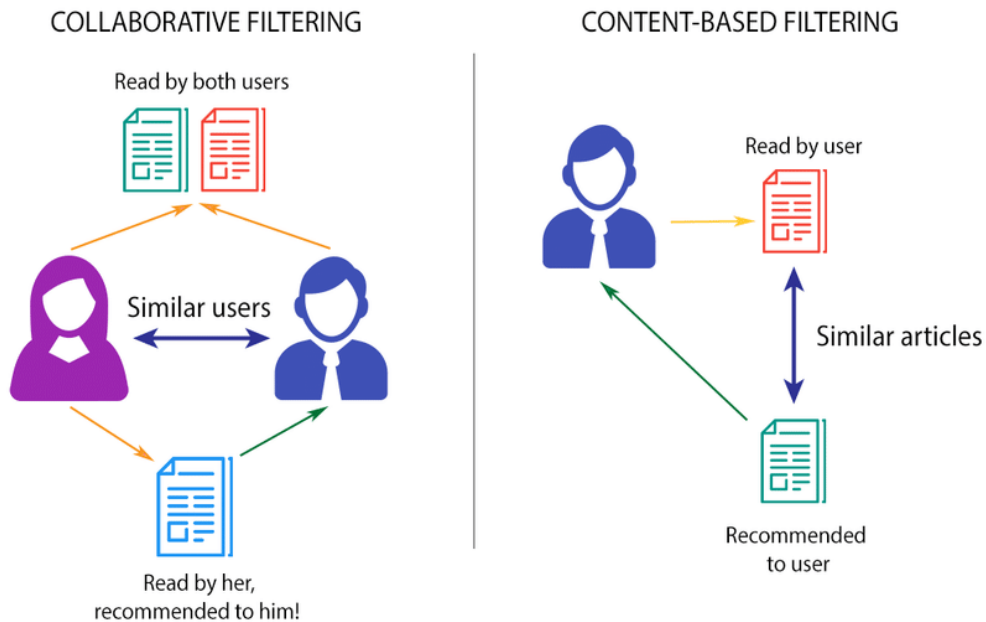


Рис. 1.2. Різниця між видами фільтрації у рекомендаційних системах [15]

	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$
$u_1$	1	1	1	0	1
$u_2$	0	1	1	0	0
$u_3$	0	1	1	1	0
$u_4$	1	0	1	1	1

← items →

↑ users  
↓

(a) user-item matrix

Рис. 1.3. Приклад user-item матриці [16].

Модель на основі популярності: Цей підхід використовує відомості про популярність предметів для рекомендацій. Наприклад, система може рекомендувати користувачам популярні пісні або книги

Глибинне навчання. Глибинне навчання (англ. deep learning) - це галузь машинного навчання, яка використовує нейромережі з багатьма шарами (глибокими моделями) для вирішення складних завдань обробки і аналізу даних. Глибинне навчання дозволяє комп'ютерам "вчитися" шляхом самостійного виявлення корисних ознак або представлень з великих обсягів даних.

Основними перевагами моделі на основі глибинного навчання є:

Автоматичне виявлення ознак: Глибинні нейронні мережі можуть самостійно виявляти корисні ознаки з вихідних даних без необхідності ручного вибору або інженерії ознак. Це дозволяє моделям глибинного навчання працювати з різноманітними типами даних і вирішувати складні завдання.

Обробка великих обсягів даних: Глибинні нейронні мережі можуть ефективно працювати з великими обсягами даних. Завдяки своїй глибині і

структурі вони можуть виявляти складні залежності і патерни в даних, що забезпечує високу якість прогнозування та аналізу.

Висока точність: Глибинні нейронні мережі, завдяки своїй глибокій структурі, можуть досягати високої точності в багатьох завданнях, таких як класифікація зображень, розпізнавання мови, машинний переклад, рекомендаційні системи та інші.

Розширена гнучкість: Глибинне навчання дозволяє будувати складні моделі з різними типами шарів і функцій активації, що дає можливість моделювати складні залежності і адаптуватися до різноманітних завдань і типів даних.

Також, варто відмітити гібридні методи роботи з даними, котрі набувають значної популярності у останній час за рахунок використання інструментів машинного навчання і методів штучного інтелекту. Тим не менш, складнощі, з якими доводиться стикатися при побудові рекомендаційних систем, лише частково схожі на проблеми суміжних областей машинного навчання. Такі системи багатоскладові, до того ж потребують постійного оновлення та покращення. Деякі складнощі зображені на рисунку 1.4 і проаналізовані далі.



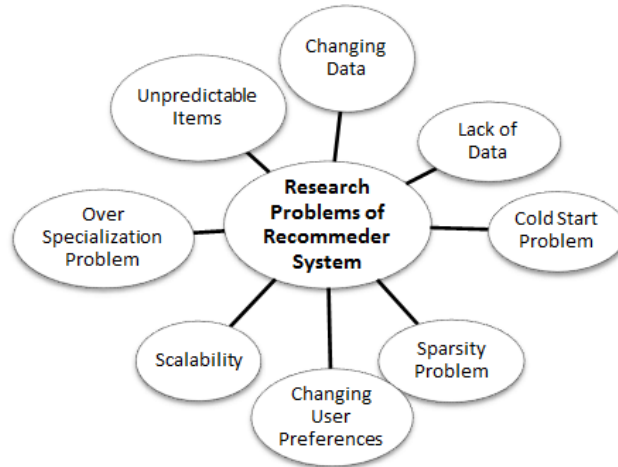


Рис. 1.4. Проблеми рекомендаційних систем [17].

У системах рекомендацій присутні і свої проблеми, що пов'язані, здебільшого, з даними та методами роботи з ними.

Зокрема, серед найрозповсюдженіших проблем систем рекомендацій можна виділити *недостатню кількість даних*.

Мова йде водночас як і про проблеми “холодного старту”, коли у системи на старті немає достатньо даних для надання хоч якихось рекомендацій, так і про більш загальні проблеми з вибірками та кількістю даних для опрацювання. Зрозуміло, що чим більшою кількістю даних володіє система, тим більша вибірка і ширший спектр рекомендацій для користувача. Як показано на рис. 1.5, будь-якій “хорошій” рекомендаційній системі необхідно спочатку отримати дані про певну предметну область, далі опрацювати їх і проаналізувати дані про поведінку користувача, і лише тоді сформулювати рекомендації на основі отриманих та проаналізованих даних. Відповідно, створюється пропорційна залежність від кількості даних (або ж кількості проаналізованих користувачів) до якості наданих рекомендацій.

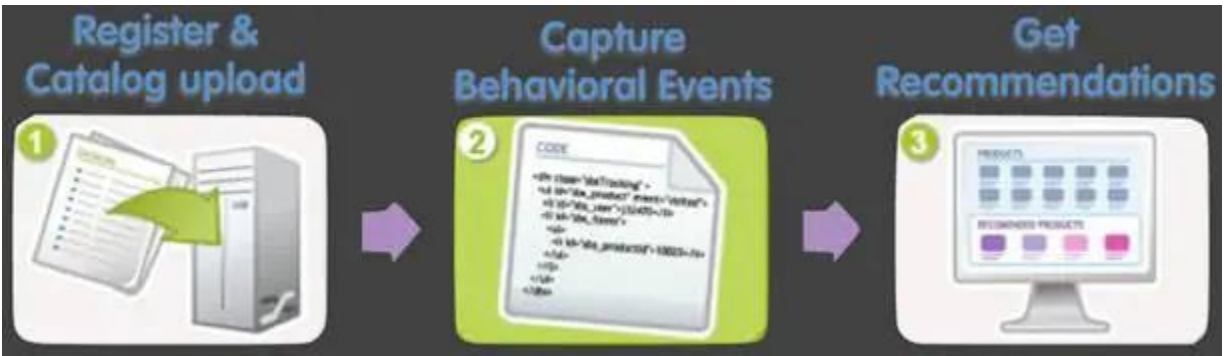


Рис. 1.5. Принцип роботи “хорошої” рекомендаційної системи [18].

Посилаючись на статтю [19], а саме на фразу “попередня поведінка [користувачів] не є хорошим інструментом, тому що тенденції постійно змінюються” переходимо до другої поширеної проблеми систем надання рекомендацій - *мінливості даних*.

Дана проблема присутня повсякчас, адже прогрес не стоїть на місці і продукт, який міг би зацікавити користувача учора вже не буде актуальним сьогодні і повністю буде забутий завтра. Багато систем рекомендацій збирають дані користувача або використовують минулі моделі його поведінки, проте аналіз такого типу (на поведінкових моделях минулого) не є ефективним інструментом для отримання якісних та хороших рекомендацій, тому що тенденції та інтереси користувачів постійно змінюються, що створює додаткові складнощі для рекомендаційних систем - вчасна реакція на мінливе середовище даних.

Знову посилаючись на [19] можна зробити висновок, що “рекомендації товарів не працюють, тому що в моді просто забагато атрибутів продукту, і кожен атрибут (ціна, колір, стиль, тканина, бренд тощо) має різний рівень важливості в різний час для того самого споживача”. Однак соціальні рекомендації можуть створити шлях для вирішення даної проблеми.

Наступним важливим викликом для систем рекомендацій є *зміна уподобань користувача* та узгодження з уподобаннями користувача, що швидко змінюються. Наміри користувачів щодо перегляду та пошуку певного продукту може відрізнятись у різні проміжки часу, тому системи рекомендацій, які повністю базуються на вподобаннях користувача, можуть формувати неправильні та недостатньо точні рекомендації. З моніторингу робіт існуючих систем можна навести приклад [20], скажімо, система оцінила поведінку користувача, яка була абсолютно випадковою і пошук здійснювався по непов'язаним категоріям товарів, для наступного користувача такі рекомендації (на основі попередніх елементів) будуть нерелевантними, тому питання узгодження з уподобаннями користувачів, які часто змінюються, є одним з найважливіших питань у системі рекомендацій [20].

Ще однією проблемою є *масштабованість* системи.

Масштабованість - це властивість системи, яка визначає, чи зможе система витримати навантаження при збільшенні її розмірів. [21] Наприклад, у випадку систем рекомендацій, масштабованість можна розуміти як ситуацію, коли система рекомендацій працює бездоганно у випадку кількох десятків чи сотень користувачів, але коли кількість користувачів зростає до тисячі, десятків чи навіть сотні тисяч, вона перестає справно працювати і не витримує такого навантаження. Коли система стикається з проблемами та навантаженнями, які ніколи не тестувалися і яких ніколи не виникало, вона починає працювати повільно чи, у критичних випадках, повністю зупиняти свою роботу.

Проблеми масштабованості розділяють на дві частини: масштабованість апаратного забезпечення та масштабованість програмного забезпечення. При цьому, апаратне забезпечення повністю залежить від програмного забезпечення системи, яке націлене на вирішення проблеми. Зазвичай такі

проблеми вирішуються при заміні процесора, оперативної пам'яті чи зміні конфігурацій серверу.

Проблеми програмного забезпечення полягають у алгоритмах та методиках, що використовуються у системі надання рекомендацій. При зростаючій практиці нових методик чи, навіть, алгоритмів на основі штучного інтелекту чи машинного навчання, проблеми даного типу, хоч і є доволі розповсюдженими, мають потужні ресурси для їх вирішення.

Таким чином, рекомендації можна розуміти як щось на кшталт самоздійснюваного пророцтва. Коли надаються рекомендації, користувач дивиться на щось із рекомендованого, і система намагається ще краще зрозуміти (передбачати), що саме він подивиться. Таким чином, користувач постійно оточений рекомендаціями і дуже рідко покидає цю зону.

Розглянувши особливості та проблеми рекомендаційних систем постає важливе питання у доцільності їх використання та імплементації при вирішенні проблем дисертаційного дослідження. Питання в більшості стосується складності у опрацюванні даних та підбору ефективних методів взаємодії з ними. Необхідність використання складних метрик також зумовлена потенційними вимогами розроблюваної інформаційної системи та процесу формування рекомендацій для індивідуальної освітньої траєкторії. Тим не менш, рекомендаційні системи існують різних видів і не всі вони стійко базуються на алгоритмах колаборації чи фільтрації контенту. Гібридні системи передбачають використання технологій, метрик та методів як можуть бути зовсім не пов'язані з ідеєю «передбачування» бажань користувача. Навіть без використання просунутих алгоритмів будь-яку інформацію для

користувача можна вважати рекомендацією. Особливо, якщо ця інформація побудована на особистісних уподобаннях.

Відповідно, для цілей дисертаційного дослідження необхідно розглянути технології, які ґрунтуються не лише на рекомендаціях, а і суміжних з ними. Такими можна вважати інформаційні системи які здійснюються аналіз даних та формують висновки на основі проведеного аналізу.

## **1.2 Проблема формування індивідуальної освітньої траєкторії та її технологічний супровід**

### **1.2.1 Процес формування індивідуальної освітньої траєкторії**

Згідно з інформацією, що розміщена на веб-сайті Національного університету «Львівська політехніка»:

Структура освітньо-наукової програми передбачає можливість для формування індивідуальної освітньої траєкторії, зокрема через індивідуальний вибір здобувачами вищої освіти навчальних дисциплін в обсязі, передбаченому законодавством. Процедура вибору здобувачами вищої освіти індивідуальної освітньої траєкторії регламентована «Положенням про організацію навчального процесу» [22], «Положенням про формування та реалізацію індивідуальних навчальних планів студентів» [23] та «Порядком вибору студентами навчальних дисциплін» [24]

Формування індивідуальної освітньої траєкторії відображається в індивідуальних навчальних планах аспірантів та передбачає можливість індивідуального вибору ними навчальних дисциплін у межах, передбачених

відповідною освітньою програмою та робочим навчальним планом (в обсязі, що становить не менш як 25 % загальної кількості кредитів ЄКТС, передбачених для певного рівня вищої освіти), з дотриманням послідовності їх вивчення відповідно до структурно-логічної схеми підготовки фахівця. Індивідуальна освітня траєкторія формалізується у індивідуальному навчальному плані аспіранта [25].

Індивідуальний навчальний план (ІНП) студента – це документ, за яким студент навчається упродовж семестру. ІНП студента містить інформацію про перелік навчальних дисциплін, обсяги навчального навантаження із усіх видів навчальної роботи та форми семестрового контролю [26].

Вибіркові навчальні дисципліни, внесені до ІНП студента стають для нього обов'язковими до вивчення. Вибіркові навчальні дисципліни студент може обирати:

- а) Для першого (бакалаврського) рівня вищої освіти – на 2, 3 і 4 курсах;
- б) Для другого (магістерського) рівня вищої освіти, - як правило, у 2 і 3 семестрах.

ІНП студента формується автоматизовано в електронному вигляді після вибору студентом навчальних дисциплін та відображається в його електронному кабінеті до початку нового навчального семестру [27].

Зважаючи на вище наведено інформацію, процес надання рекомендацій для включення у індивідуальну освітню траєкторію не є одноразовим, а, скорше, ітеративним, адже траєкторія формується упродовж всього процесу здобування вищої освіти студентом. Процес формування траєкторії вважається лінійним і відповідно, його складність зростає у такому ж порядку.

Таким чином, ми розуміємо, що кожного навчального року ІОТ студента буде доповнюватися і розширюватися відповідно до індивідуального

навчального плану. Процеси доповнення і розширення ІОТ вважатимемо як процеси інтеграції даних.

Інтеграція даних включає об'єднання даних, що знаходяться у різних джерелах, та надання даних користувачам в уніфікованому вигляді. Цей процес стає суттєвим як у комерційних завданнях (коли двом схожим компаніям необхідно об'єднати їх бази даних), так і в наукових. Роль інтеграції даних зростає, коли збільшується обсяг та необхідність спільного використання даних.

Інтеграція різних даних є одним із ключових аспектів результатів досліджень щодо формування індивідуальних освітніх траєкторій. Це означає збір та поєднання різноманітних даних про студента з різних джерел з метою забезпечення більш повної та об'єктивної картини потреб їхнього академічного розвитку. Основні аспекти інтеграції різних даних включають:

- Академічні дані: Це включає оцінки, рейтинги тощо, які стосуються контенту навчальних дисциплін. Інтеграція цих даних дозволяє отримати повну картину про складові вибіркового блоку освітніх програм.
- Інтереси та здібності: Інформація про інтереси, хобі, здібності та цілі студентів також може бути інтегрована у процес формування індивідуальної навчальної траєкторії. Це допомагає враховувати індивідуальні потреби та мотивацію студентів під час планування їхнього навчання.
- Дані про розвиток: Інтеграція даних про розвиток студентів, включаючи когнітивні здібності, соціальні навички, емоційний і психологічний стан, є важливим аспектом формування індивідуальних освітніх траєкторій. Ці дані надають більш повну

картину про стан та потреби учня, що допомагає забезпечити індивідуалізоване навчання.

Когнітивні здібності включають інтелектуальні здібності, розумові процеси та навички, які впливають на сприйняття, мислення, пам'ять та розв'язання проблем. Інтеграція даних про когнітивні здібності допомагає враховувати індивідуальні особливості учнів при плануванні навчального матеріалу та методів навчання.

Соціальні навички включають вміння спілкування, співпраці, розв'язання конфліктів, лідерства та інші соціально-емоційні аспекти. Інтеграція даних про соціальні навички допомагає враховувати соціальну адаптацію учнів, їхню взаємодію з оточуючими та сприяє розвитку міжособистісних навичок.

Емоційний і психологічний стан включає інформацію про емоційну стабільність, мотивацію, самооцінку, стресостійкість та інші аспекти емоційного благополуччя учнів. Інтеграція даних про емоційний і психологічний стан допомагає зрозуміти вплив емоційного стану на навчання та допомагає розробляти підходи, що підтримують психологічних якостей.

Таким чином, враховуючи поетапність процесу формування траєкторії, а також необхідність у інтегрування різних даних, формується необхідність у технологічному супроводі у вигляді інформаційної системи. Оцінювання розмірності та ефективності системи вимагає використання відповідних аналітичних методів та було здійснене у другому і третьому розділі дисертаційного дослідження.

### **1.2.2 Аналіз технологічного супроводу освітніх процесів.**



Згідно з Національною стратегією розвитку освіти в Україні на 2012-2021 роки основним напрямом розвитку освітнього простору є впровадження (учнецентрированої) концепції індивідуалізованого навчання [28]. Реалізація індивідуалізованого навчання забезпечується визначенням індивідуальної освітньої траєкторій і поетапним проходженням маршруту [29], що дозволяє її реалізувати. Педагогічні аспекти створення й реалізації ІОТ (маршруту) на різних етапах навчання, у вивченні окремих дисциплін, організації навчання у навчальних закладах вищої освіти розглянуті у працях великої кількості дослідників (Н. Суртаєвої, Є. Остропольської, В. Черненко, І. Костікової, Л. Ткаченко, О. Власенка та ін.)

Завдяки потужному розвитку інформаційних технологій сьогодення все частішого використання набуває комплекс інформаційно-комунікаційних технологій (ІКТ). Зокрема, якщо розглядати ці технології через призму розвитку освіти, можна стверджувати, що:

- Освіта сьогодення все частіше перестає бути залежною від фізичного розташування суб'єктів;
- Онлайн ресурси та їх різноманітність створює унікальні можливості для додаткового навчання у позаурочний час;
- З двох попередніх тверджень можна сформулювати і третє: контроль над своєю освітою тепер знаходиться у руках самого студента; він сам здатний обирати час чи місце для ініціації навчального процесу.

Згідно з публікацією [30] суттєві зміни в освітньому процесі значною мірою зумовлені новою парадигми з впровадженням ІКТ в освіту. Деякі з цих змін зображені у таблиці 1.1

**Таблиця 1.1**

**Зміни в системі освіти, зумовлені впровадженням ІКТ [30]**

	<b>Традиційна модель</b>	<b>Нова модель</b>
<i>Роль учителя</i>	Експерт	Партнер
<i>Процес навчання</i>	У центрі процесу - вчитель	У центрі процесу - студент
<i>Критерії успішності</i>	Демонстрація наявного рівня знань, умінь, навичок	Демонстрація покращення рівня особистих знань, умінь, навичок
<i>Тип знань</i>	Отримання, накопичення, відтворення	Інтерпретація, пояснення
<i>Методи контролю знань</i>	Тестування	Практико-орієнтовані завдання
<i>Освітня парадигма</i>	Змістово-орієнтована; орієнтована на педагога	Процесно-орієнтована; орієнтована на учня/студента
<i>Форма навчальної роботи, що переважає</i>	Самостійна робота	Групова робота

Дана таблиця демонструє зміни в освітній парадигмі, а саме перехід від “учитель-орієнтованого” до “студент-орієнтованого” підходу. Концептуально зміни в новій парадигмі обумовлені переходом від змістово-орієнтованого підходу до процесно-орієнтованого, коли навчання розуміється не лише як отримання, накопичення та відтворення знань, а як формування здатності до їх осмислення, пояснення та інтерпретації [31]. Тобто, здобуття освіти вважається не кінцевим продуктом, а, скоріше, динамічним процесом, що націлений на залучення студентів до виконання завдань аутентичного характеру, що безпосередньо пов’язані з реальними життєвими і професійними ситуаціями [31].

Таким чином, освітній процес з залученням ІКТ підпорядкований безпосередньо творчому розвитку особистості кожного студента. Це дає

суттєві переваги над традиційним підходом, де фокус створюється лише на формування компетенцій.

Розглядаючи проблему формування індивідуальної освітньої траєкторії, неможливо бути певним у виборі технологічного рішення без попереднього аналізу. Системи надання рекомендацій хоч і забезпечують хороші умови, по своїй суті, є надскладними системами і вимагають величезних ресурсів для постійної та точної роботи. Використання суміжних систем збору і аналізу даних створює гідну конкуренцію системам рекомендацій на підґрунті простоти та економії ресурсів, що витікає у дешевшу підтримку та обслуговування, якщо мова йде про автоматизовані інформаційні системи.

Тим не менш, описаний раніше підхід до формування ІОТ вимагає не лише активної взаємодії студента зі своїми освітніми компетенціями, а й забезпечення гідного супроводу від освітнього закладу, використовуючи інформаційні технології. Під інформаційно-освітніми технологіями розуміється сукупність інформаційно-комп'ютерних засобів та способів, що використовуються як домінуючі в освітніх технологіях і сприяють досягненню запланованих цілей навчання та виховання.

Щоб вирішити цю проблему, Е. де Кортє пропонує більше зосередитися на компетенціях, які розширюють можливості учнів. Оскільки учні не є пасивними одержувачами інформації, конструктивні компетенції стимулюватимуть їх активне залучення до процесів набуття знань і навичок; компетенції саморегуляції сприятимуть моніторингу поточного навчального процесу шляхом надання власного зворотного зв'язку та збереження концентрації та мотивації; компетенції співпраці направлять учнів від

«одиначної» діяльності до «розподіленої» [32]. Відповідно до цього підходу інші дослідники пропонують запровадити стійкі управлінські механізми, здатні усунути антропологічний та гуманітарний дефіцит і сприяти «особистісній дії в освіті людини» [33].

Це особливо актуально, коли мова йде про сучасні освітні ресурси. Справа в тому, що більшість освітніх інструментів хоч і надають базовий рівень підтримки абітурієнтам, насправді є доволі обмеженими в функціональному та технологічному планах, що, зазвичай, зумовлено нестачею інвестицій у розвиток інформаційних технологій даної сфери, та низьким рівнем зацікавленості у цьому ж розвитку. Ця проблема, звісно, є відкритою для активних дискусій, так як ми є жителі інформаційної ери, і технологічний розвиток освіти, все ж таки, активно підтримується, особливо на фоні мінливої ситуації у світі, що регулярно створює перешкоди традиційним методам навчання. Існують десятки, якщо не сотні інформаційних ресурсів та засобів для їх розробки чи використання, проте далеко не всі є легкодоступними чи здатними надати гідну підтримку [34].

Безумовно, будь-яке програмне забезпечення чи онлайн застосунок залежить від інформації з освітніх закладів. Існує безліч реалізованих рішень які використовують різноманітні методи аналізу та інтеграції даних. Такі інструменти, зазвичай, або послуговуються своїми базами даних, які створені на основі відкритих джерел у мережі інтернет, або ж напряму залежать від закладів освіти та відповідними ресурсами, які їх представляють. Залежно від типу реалізації, має місце інтеграція та аналіз даних, які потім пропонуються користувачеві.

Отже, будь-який інструмент для формування індивідуальних освітніх траєкторій повинен бути пов'язаний з освітніми закладами. Під категорію таких інструментів найчастіше підпадають інформаційні системи, завдяки продуктивності при використанні різноманітних методів аналізу даних.

Ряд дослідників описували та обґрунтовували використання сучасних інформаційних технологій у освіті. Проведений аналіз показав, що сучасні інформаційні технології відкривають учням доступ до нетрадиційних джерел інформації, підвищують ефективність самостійної роботи, дають цілком нові можливості для творчості, знаходження та закріплення професійних навичок [35].

Додатково варто зазначити, що використання таких систем у сферах освіти набуває все більшого значення у наш час. Це зумовлено актуальністю програмного забезпечення, яке привносить потужні інструменти автоматизації при організації навчальних процесів студентів у системах освіти, що знаходяться у процесі трансформації. Це також дозволяє сформувати нову інформаційну ментальність для всіх зацікавлених сторін, як для студентів і викладачів, так і для компаній чи бізнесів.

### **1.3 Аналіз моделей та методів для формування рекомендацій навчальних дисциплін з їх подальшим включення у ІОТ.**

Перш ніж розглядати створення власного технологічного супроводу процесу формування ІОТ необхідно розглянути не лише освітні ресурси, що опосередковано намагаються вирішити проблему (зокрема, проблему професійної спрямованості), а й існуючі технологічні імплементації, що безпосередньо стосуються процесу формування індивідуальної освітньої траєкторії. Зокрема, можна звернути увагу на дослідження, цілком яких була

розробка інформаційної системи з метою формування індивідуальної освітньої траєкторії. Такі технологічні напрацювання описуються в роботах С. Шарова, А. Богданова та Щ. Данила.

Перша інформаційна система (ІС), яку ми розглянемо, описана у статті [36]. Процес формування траєкторії описується як такий, що передбачає аналіз навчального плану спеціальності, ознайомлення з анотаціями дисциплін, які входять до навчального плану, а також аналіз компетентностей, які формуються в процесі вивчення конкретних дисциплін [36]. Складність процесу формування індивідуальної траєкторії обґрунтовує використання інформаційних систем для досягнення результатів.

При розробці інформаційної системи автори ставили перед собою низку завдань, серед яких найважливіші:

- Створення проекту структури інформаційної системи;
- Вибір інструментів для розробки інформаційної системи;
- Завантаження офіційної інформації про навчальні програми, користувачів, тощо в інформаційну систему.

До найважливіших функцій ІС належать наступні:

- Створення, редагування та затвердження навчального плану за певною освітньою програмою;
- Вибір студентом дисциплін за вибором з переліку курсів за вибором навчального плану;
- Формування учнем індивідуального навчального плану.

Також, при розробці ІС були використані такі інформаційні технології, як мова програмування *PHP* та база даних *MySQL*. Створена система була

реалізована у вигляді інтерфейсної програми з повним функціоналом згідно з описаною методологією (рисунок 1.6).

The screenshot displays a web interface with a dark header bar containing navigation links: 'Головна', 'Навчальний план', 'Отримати інформацію', 'Зворотній зв'язок', and 'Вихід'. Below the header, there are four distinct form panels arranged in a 2x2 grid. Each panel has a title and a 'Додати' (Add) button. The top-left panel, 'Додати спеціальність', includes fields for 'Ступінь вищої освіти', 'Галузь знань', 'Спеціальність', 'Академічна кваліфікація', and 'Професійна кваліфікація'. The top-right panel, 'Додати студента', includes fields for 'ФІО студента', 'Спеціальність', 'Факультет', 'Рік початку навчання' (with '2016' selected), 'Термін навчання', 'П'ятю років', 'Логін', and 'Пароль'. The bottom-left panel, 'Додати викладача', includes fields for 'ФІО викладача' and 'Кафедра' (with 'Математики' selected). The bottom-right panel, 'Додати предмет', includes fields for 'Шифр', 'Назва', and 'Кафедра' (with 'Математики' selected).

*Рис. 1.6. Приклад інтерфейсу ІС С.Шарова.*

Інакший приклад реалізації інформаційного забезпечення описаний у статті [37], де був проведений огляд західних праць та робіт, які досліджували інструменти для підвищення рівня підготовки спеціалістів та сприянню мобільної освіти. В результаті було визначено, що найбільш доцільним є використання Європейських напрацювань у сфері технологій інформаційного забезпечення при розробці власної системи.

Таким чином, розроблена у цьому прикладі система інформаційного забезпечення призначена для допомоги особам, що бажають здобути освіту для працевлаштування, у створенні індивідуальної освітньої траєкторії, з можливістю аналізу їх поточних знань і навичок і співставленні їх з потребами

ринку праці [37]. Система реалізована з використанням веб-технологій, таких як мова програмування *PHP* та *Javascript*, а дизайн розроблений за допомогою фреймворку *Bootstrap* (рисунок 1.7).

Особливістю системи є її модульність та відкритість, що дозволяє в подальшому розширювати її або тимчасово використовувати модулі інших систем [37]. Особливу увагу при реалізації автор приділив до зв'язку між модулями. Для цього навіть був реалізований окремий модуль зв'язку, котрий пов'язує усі функціональні модулі системи з головним модулем.

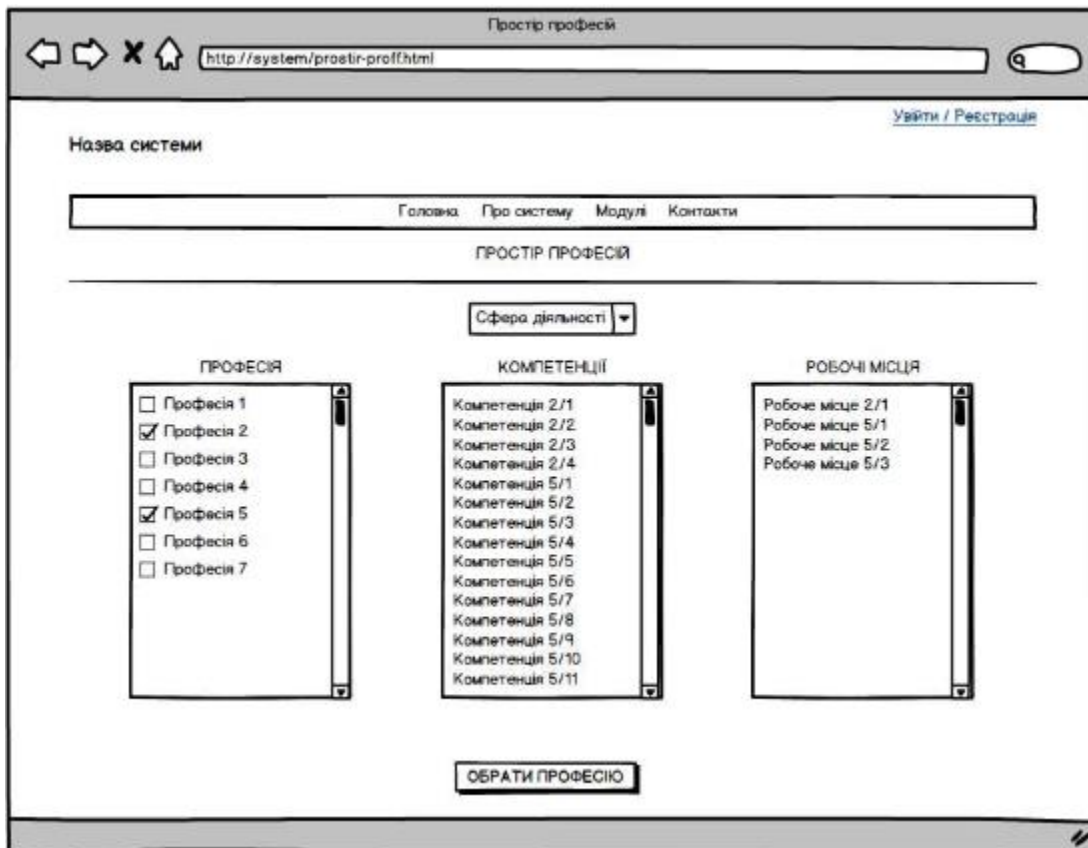


Рис. 1.7. Приклад інтерфейсу ІС.



Аналіз цих робіт показав, що хоч розроблені ІС і виконують поставлені завдання (формування індивідуальних траєкторій), проте, жодна з них не дає будь-яких рекомендацій при виборі дисциплін та формуванні траєкторій. Усі реалізовані системи лише оптимізували процес збору та подачі інформації користувачу. Таким чином, користувач при користуванні такою системою постає перед вибором серед десятків чи навіть навчальних програм та дисциплін без будь-якої допомоги при їх формуванні. Також, серед розглянутих рекомендаційних систем не було помітно використання онтологічного підходу, що дозволило би отримати точніші результати.

Остання розглянута робота [38] описує рекомендаційну систему для персоналізації навчального шляху. В ході розроблення системи були використані різноманітні веб-технології, такі як *JavaScript*, *AngularJS*, *Bootstrap*. Для зберігання даних була використана технологія *NoSQL* баз даних — *MongoDB*. Розроблена рекомендаційна система використовує гібридну фільтрацію з більш активним використанням фільтрації контенту та менш активним - колаборативної фільтрації. Алгоритм полягає у аналізі уподобань та інтересів студентів і пошуку відповідних навчальних матеріалів. Пошук контенту відбувається різними способами, які виділені в окремі функції. Ці функції використовують вбудовану логіку визначення рекомендаційної ваги, що представлена у вигляді графу переходів. В результаті, рекомендації для користувача показуються списком з двадцяти навчальних матеріалів. Користувач також має можливість дізнатися, чому йому порекомендували саме цю дисципліну і, відповідно, відмовитися від рекомендацій та отримати їх ще раз.

Дана система хоч і використовує веб-технології, що надає їй певні переваги перед іншими реалізаціями, проте використані алгоритми для визначення та надання рекомендацій є далеким від необхідного для вирішення проблеми формування ІОТ. Вхідними даними для виконання аналізу є уподобаннях студентів та їх інтереси, що є мізерною долею інформації у процесі формування ІОТ. Такий обмежений аналіз не може бути ефективним і навряд чи зможе надати точні результати.

Тим не менш, реалізація системи, що надає рекомендації є найбільш близьким та ефективним способом вирішення проблеми. Рекомендаційна система дозволить не лише отримувати та зберігати інформацію, а й аналізувати та надавати рекомендації при виборі дисциплін. Існуючі алгоритми рекомендаційних систем дозволили би враховувати численні фактори при формуванні ІОТ, проте і значно збільшили би складність з інтеграцією та аналізом даних. Поширені алгоритми систем надання рекомендацій використовують величезні масиви даних для аналізу і, в контексті формування ІОТ, вимагали би велику кількість інформації про користувачів (студентів), наприклад персональні профілі чи відгуки тих, хто навчається чи завершив навчання за ІОТ. Для уникнення цих проблем і обмеження отримуваної інформації про студента, було вирішено використати інші алгоритми, які забезпечили би гнучкість та простоту для надання рекомендацій. Це дозволить формувати рекомендації на основі професійних схильностей студентів та оцінки експертів, стейкхолдерів, викладачів.

#### **1.4 Постановка задачі дослідження.**

Головною метою формування сучасного технологічного рішення для формування рекомендацій є задоволення потреб абітурієнтів та здобувачів освіти при формування власної індивідуальної траєкторії.

Аналіз досліджень дозволив зрозуміти особливості та проблеми моделювання інформаційно-технологічних рішень при формуванні індивідуальної освітньої траєкторії.

Описана ситуація формує актуальну наукову задачу: забезпечити користувача постійним доступом до інформаційного інструменту, що здатен сформувати рекомендації до включення у ІОТ на основі персональних якостей студента ІТ-спеціальності.

### **Висновки до розділу 1**

1. У даному розділі подано загальні особливості та характеристики рекомендаційних систем. Наведено існуючі методи формування, моделі, описано поширені проблеми, плюси та мінуси реалізації. У якості висновку наведено обмеженість та складність при використанні класичних алгоритмів та методів формування рекомендацій.

2. Детально описано з наведенням джерел процес формування індивідуальної траєкторії у межах Національного університету “Львівська політехніка”.

3. Здійснений огляд та аналіз наукових праць та досліджень у сфері технологій інформаційного забезпечення, що націлені на формування індивідуальної освітньої траєкторії. Аналіз дозволив зрозуміти, що, хоч на сьогоднішній день і існують компетентні систем для забезпечення процесу формування ІОТ, в більшості вони не використовують веб-технології та методи аналізу даних в повній мірі з можливістю надання рекомендацій.

Отже, було доведена актуальність наукового завдання з розроблення рекомендаційної системи для формування індивідуальної освітньої траєкторії студентів ІТ-спеціальностей.

## РОЗДІЛ ІІ. МОДЕЛІ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

Другий розділ присвячений характеристиці та опису рекомендаційної системи, яка генерує рекомендації навчальних дисциплін для їх включення у ІОТ. Також у цьому розділі були розроблені та представлені моделі рекомендаційної системи, зокрема модель формування рекомендацій, модель визначення профорієнтації, модель процесу визначення семантичної близькості та модель ранжування дисциплін з використанням МАІ.

### 2.1 Формальна модель системи формування індивідуальної освітньої траєкторії

Формально модель процесу формування індивідуальної освітньої траєкторії можна представити у вигляді трьохелементного кортежу:

$$Tuple = \{T_1, T_2, T_3\},$$

де  $T_1 = \{t_{11}, t_{12}, \dots, t_{1n}\}$  – множина суб'єктів (агентів), що беруть участь у формуванні траєкторії. В ролі агентів виступають здобувачі освіти і студенти, а також викладачі, стейкхолдери та експерти;  $T_2 = \{t_{21}, t_{22}, \dots, t_{2n}\}$  – множина властивостей агентів (професійні схильності та персональні уподобання, професійність та можливість оцінки, тощо),  $T_3 = \{t_{31}, t_{32}, \dots, t_{3n}\}$  – множина відношень між агентами (професійні відносини, обмін оцінками чи рекомендаціями, тощо), що дозволяє сформувати ІОТ.

Процес формування ІОТ загалом базується на роботі та взаємодії її агентів і, відповідно, формуванні індивідуального навчального плану. Логіку

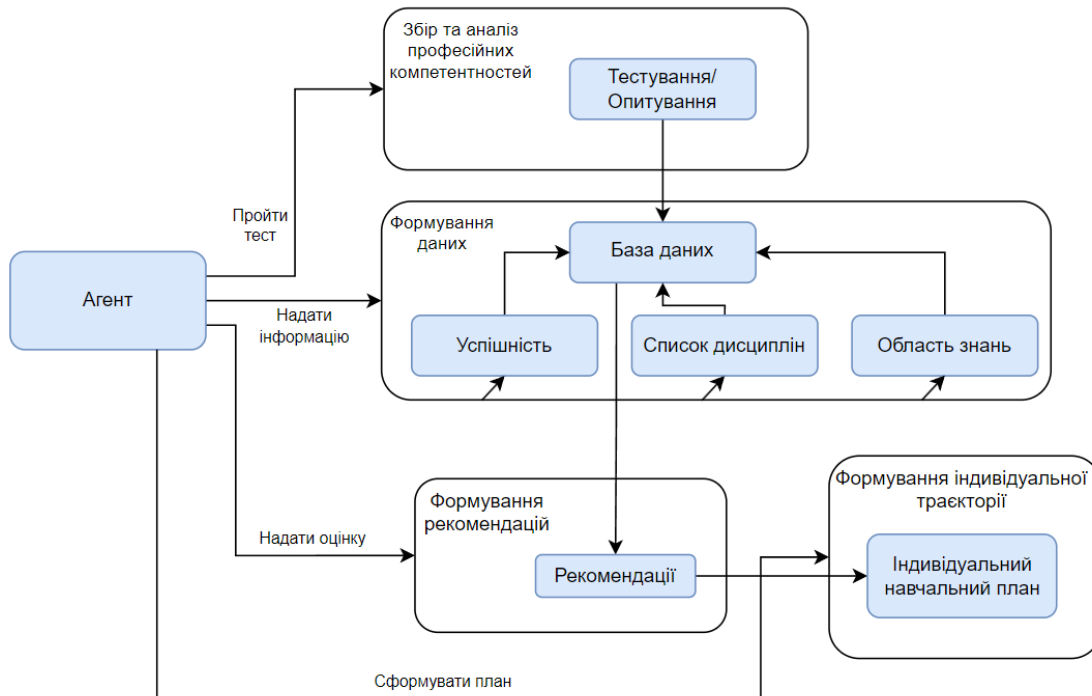
поведінки агентів зображено у вигляді графу переходів станів на рис. 2.1. Такий граф зображує процес формування ІОТ в цілому:

1. Збір та аналіз професійних компетентностей - здобувачі освіти або чинні студенти проходять опитування та профорієнтаційні тести для визначення своєї мотивації, мети та персональних уподобань. Додатково здійснюється аналіз вимог до вакансій професій та узгодження компетентностей від стейкхолдерів;

2. Формування та надання інформації (бази даних) - викладачі у вищих освітніх закладах зв'язують компетентності з результатами навчання, а також визначають область знань і дисципліни для подальшого формування індивідуального навчального плану.

3. Формування рекомендацій - експерти, стейкхолдери та викладачі надають свою оцінку, рекомендації та корекцію щодо формування індивідуального навчального плану студентом.

4. Формування індивідуального навчального плану - останнім етапом є безпосередньо формування навчального плану для подальшого його використання у індивідуальній траєкторії використовуючи попередні напрацювання.



*Рис. 2.1. Граф переходів станів агентів рекомендаційної системи.*

Взаємодія агентів (рис. 2.2) може відбуватися за участю додаткових інструментів - інформаційних систем. Одним з описаних раніше пунктів формування ІОТ і, відповідно, ціллю дисертаційного дослідження, є формування рекомендацій, що передбачає використання різноманітних методів інтеграції та аналізу даних. Зрозуміло, що такі методи повністю залежать від програмного забезпечення і їх реалізація, в першу чергу, базується на детально розроблених моделях, які необхідні для подальшого проведення системного аналізу. Для досягнення цілей дослідження достатньо буде і однієї моделі формування рекомендацій, на основі якої буде здійснена практична реалізація.

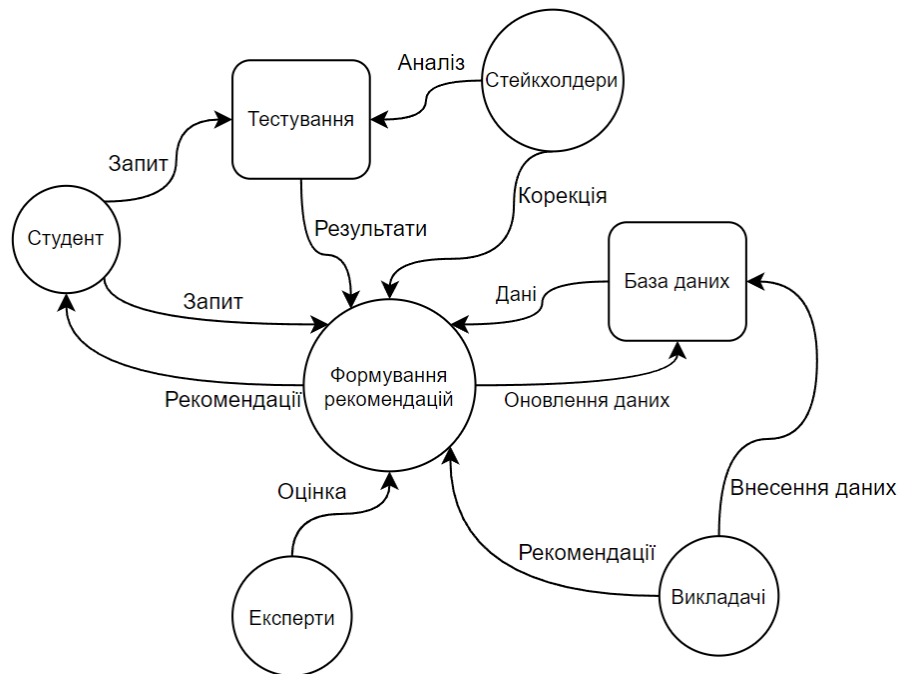


Рис. 2.2. Взаємодія агентів у рекомендаційній системі.

## 2.2 Модель формування рекомендацій навчальних дисциплін для включення у ІОТ

Процес формування рекомендацій для їх подальшого включення у ІОТ вимагає поетапного системного аналізу для виявлення проблем та особливостей. Потенційно, робота буде вестися з величезними об'ємами інформації, що потребує врахування усіх критеріїв, передумов та особливостей і, відповідно, складатиметься з наступних етапів:

**Етап 1.** Визначення професійних нахилів. Для інтеграції і аналізу даних про користувача, необхідно запропонувати йому пройти низку тестів для визначення профорієнтації. Профорієнтацію можна розуміти як систему заходів, що націлені на самопізнання для вибору професії або корективи професійної траєкторії. Отримані результати тестів представлятимуть



професійні схильності користувача. Зокрема, у дисертаційному дослідженні використовується тест Холланда на профорієнтацію.

**Етап 2.** Парсинг даних та пошук семантично близьких дисциплін. Рекомендації формуються по дисциплінам, які найбільше підходять до включення у ІОТ студента. Для більшого охопту ринку праці та отримання актуальних даних здійснюється парсинг ІТ-вакансій з метою пошуку таких вакансій, які би були семантично близькі до компонентів вибіркового блоку у освітньо-професійних програмах. Враховуючи попередньо отримані дані на першому етапі, можна здійснити пошук саме тих ІТ-вакансій які відповідатимуть професійним схильностям студента. Визначення семантичної близькості відбувається на основі знаходження відповідності між ключовими словами дисциплін та описовим текстом ІТ-вакансій.

**Етап 3.** Ранжування дисциплін за МАІ. Серед семантично близьких дисциплін необхідно вибрати ту, яка також би задовільняла експертну оцінку при формуванні ІОТ студента. Для таких цілей був використаний метод аналізу ієрархій (МАІ), де за альтернативи приймаються попередньо знайдені дисципліни.

**Етап 4.** Метод головних компонент. Окрім вибірових компонент з освітньо-професійної програми існують також загальноуніверситетські дисципліни (мова йде про каталог навчальних дисципліни Національного університету «Львівська політехніка» [39]). Їхня кількість є значно більшою ніж у кожній освітньо-професійній програмі, та й використання МАІ є недоцільним через різко зростаючу складність. Відповідно, для зменшення такої вибірки даних використаємо метод головних компонент. Даний метод дозволяє по визначених характеристиках кожної дисципліни зменшити їх вибірку та обрати одну для включення у ІОТ.

**Етап 5.** Формування рекомендацій. На основі попередніх обчислень відбувається формування рекомендацій у вигляді списку дисциплін для включення у ІОТ.

Усі етапи процесу формування рекомендацій представлені на рисунку 2.3 у вигляді UML «Діаграми діяльності».

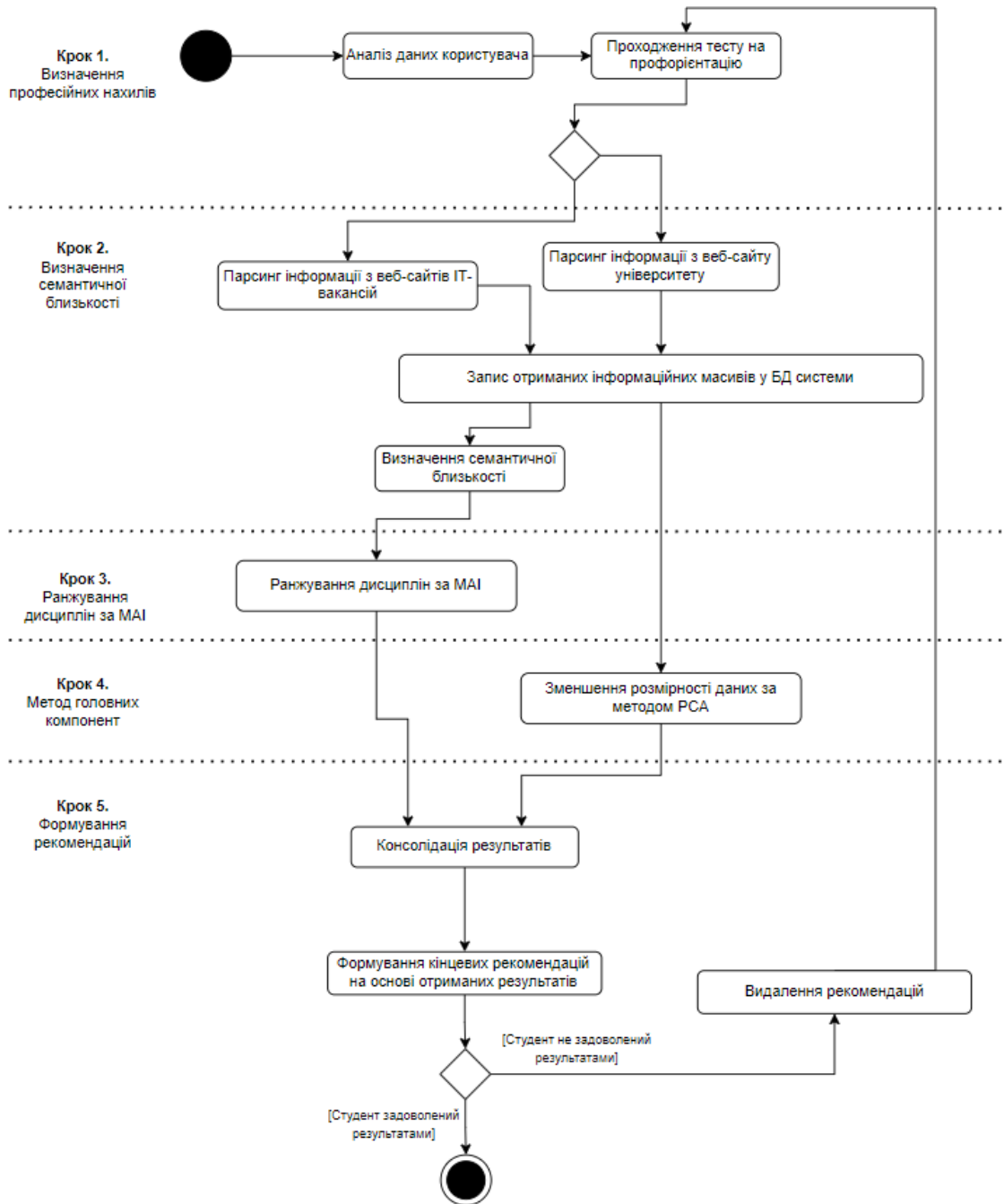


Рис. 2.3. Діаграма діяльності рекомендаційної системи формування рекомендацій.

## 2.3 Модель аналізу даних

Як було розглянуто у попередньому розділі, основа ідея усіх алгоритмів у рекомендаційних системах, полягає у аналізі зв'язків користувачів і матеріалу (контенту), який їх цікавить чи може зацікавити. Зазвичай, використовуються як складні, гібридні методи (або навіть з використанням технології машинного навчання), так і прості, з контентною чи колаборативною фільтрацією. Тим не менш, використання таких методів для формування рекомендацій з їх включенням у ІОТ не є повністю доцільним, зважаючи на попередньо описаний процес формування індивідуального навчального плану. Для формування нових рекомендацій необхідно брати до уваги не лише досвід інших студентів, чи статистику обраних для включення дисциплін, а й низку інших характеристик, таких як:

- Індивідуальні знання та здібності студента, його мотивацію чи поставлені цілі, а також особисті уподобання;
- Формування вимог до вакансій професій та аналіз даних на ринку праці (веб-сайти з вакансіями);
- При формуванні ІОТ обов'язковим є врахування оцінки та думки експертів чи стейкхолдерів (або навіть викладачів), а також проведення коригування навчального плану на фінальних етапах його створення.

Таким чином, використання алгоритмів рекомендаційних систем, обґрунтовано, можна вважати нездалим і позбавленим будь-якої практичної користі. Відповідно, для кожного етапу формування рекомендацій будемо використовувати унікальний підхід з різноманітними методами та алгоритмами.

Першим і базовим етапом у процесі формування ІОТ та рекомендацій є визначення професійних компетентностей студента та їх формування зв'язку з навчальними дисциплінами. Цей процес, очевидно, потребує здійснення

аналізу вимог до професійних вакансій і, відповідно, використання інструментів збору та аналізу даних.

Таким чином, для фіксації мети, мотивації та уподобань студента, необхідно здійснити тести, які також дозволять визначити професійні компетентності. Необхідний тест повинен включати сучасні методи оцінювання професійних здібностей та компетентностей студента, зокрема - опитувальник професійної спрямованості (ОПС) Дж. Голланда [40] Тест Дж. Голланда є вкрай зручним та визнаним не лише на заході, а й у всьому світі. В основі тесту є методика професора США Джона Голланда, що полягає у визначенні ступеня зв'язку особистості з сферою професійної діяльності, до якої у неї є природні нахили.

Джон Генрі Голланд є одним із перших вчених, які почали вивчати складні системи та нелінійну науку; відомий як батько генетичних алгоритмів. Здобув докторський ступінь у галузі психології (1952) та ступінь магістра (1947) з Університету Міннесоти [41].

Холланд зробив кар'єру на вивченні сфері праці та її дослідження заклали основу для галузі профорієнтації. Його новаторська теорія каже: "...якби люди були обізнані про свої типи особистості: технічні, дослідницькі, художні, соціальні, промислові, підприємницькі, звичайні - тоді вони були б щасливішими за працівників..." [41].

"Holland codes", найчастіше згадуються як RIASEC коди, складають загальні професійні теми як у сильних, так і надсильних оцінках. Їх створення приписують роботі професора Дж. Холланда. Широко прийнята теорія

Холланда про тематичні коди RIASEC виникла з безлічі життєвих дослідів: його роботи з навчання в аспірантурі з Дж. Д. Дарлі (одним з перших експертів з сильної оцінки в Університеті Мінесоти), його досвіду консультування з питань кар'єри та часу, проведеного в армії під час війни (Другу світову він пережив як класифікаційний інтерв'юер). В інтерв'ю Холланд зазначив, що одним із його найбільших розчарувань було використання масивних систем класифікації занять, які були надто великими, щоб їх можна було запам'ятати або навіть використати. Це спонукало його створити спрощену систему класифікації занять [42].

У 1959 Холланд опублікував першу статтю своєї теоретичної основи, заснованої на його дослідженнях ключів сильної оцінки і основних роботах, проведених такими експертами, як Стронг, Гот, Лорен, Вайнштейн і Фойер [42]. На основі свого дослідження Холланд у 1959 році виділив шість основних професійних областей.

### **2.3.1 Модель визначення профорієнтації за методикою Голланда.**

Практична реалізація методики Голланда полягає у використанні так званих «Кодів Голланда» RIASEC. По своїй суті, це система класифікації робочих стилів діяльності за категоріями професій, кластерами інтересів або областями функціонування особистості [43]. Управління зайнятості та професійної підготовки Міністерства праці США використовує у своїй роботі модель RIASEC (шестифакторну модель опису стилів роботи, яким надають перевагу, зображена на рис. 2.4). Їхнє поєднання утворює близько сімсот двадцяти індивідуальних особистісних робочих стилів. Згідно з RIASEC виділяють шість основних стилів робочого функціонування особистості:

- Реалістичний (R) – спрямованість на створення матеріальних речей, обслуговування технічних пристроїв та забезпечення технологічних процесів.

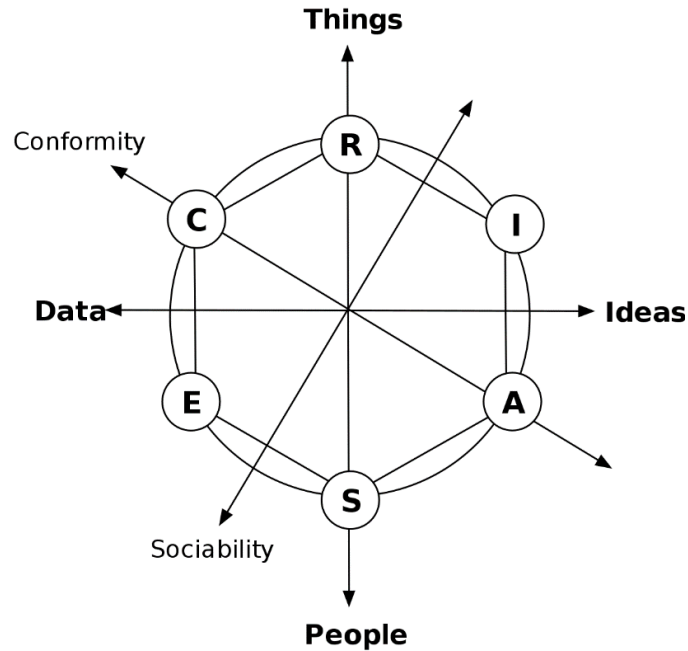
- Інтелектуальний (I) – до цього типу належать будь-які професії, що потребують підвищених вимог до інтелекту, аналітичних здібностей, раціоналізму чи вміння тверезо мислити.

- Артистичний (A) – до цього типу належать креативні, з незвичайним поглядом на життя люди, які намагаються не залежати від думки оточуючих. Дуже часто вирізняються високою чутливістю, емоційністю, емпатією.

- Соціальний (S) – для цього типу належать будь-які професії, що пов'язані із соціальним середовищем: навчання, виховання, лікування, обслуговування, консультування, догляд чи допомога.

- Підприємницький (E) – професії даного типу зазвичай пов'язані з ризиком, необхідністю постійно приймати самостійні вольові рішення, вимагають наявності організаторських здібностей, розвинених комунікативних навичок та великого запасу сил.

- Конвенційний (C) – тісний зв'язок з опрацюванням та систематизацією інформації у вигляді чисел, формул, тексту, а також веденням документації.



*Рис. 2.4. Модель опису стилів роботи, яким надають перевагу.*

Тест є набором питань, кожне з яких це можливість описати речі, які міг чи хотів би робити користувач у своїй потенційній професійній діяльності. Після проходження тесту користувачу присвоюється домінуючі стилі особистості, що записуються кодом з одної, двох або трьох літер: I, SA, IAR і т.д.

Результати тесту записуються у профіль користувача і можуть бути представлені як множина характеристик  $X$ :

$$X = \{x_1, x_2, x_3\},$$

де кожен елемент  $x$  набуватиме значення коду Голланда. Таке представлення результатів зумовлене адаптацією профорієнтаційного тесту під роботу рекомендаційної системи. Кожній професійній характеристиці



притаманні неоднозначності – тобто, коли після проходження тесту користувачу присвоюється не один тип особистості, а одразу два або і три. Такі випадки не є винятковими і зумовлені методикою професійного тесту Голланда. Відповідно, для забезпечення подальшої семантичної роботи з професійними характеристиками, необхідно консолідувати інформацію для зручного доступу до них.

Хоча тест Голланда і дозволяє дізнатися рекомендовану професію, проте він не враховує особливості ІТ-вакансій на сучасному ринку праці України. Недостатньо буде отримати рекомендовану професію на основі результатів лише цього тесту, адже вона, скорше за все, не буде пов'язана з ІТ-спеціальностями. Для вирішення цієї проблеми необхідно провести аналіз даних ринку праці, зокрема, веб-сайтів з ІТ-вакансіями. Більше детально аналіз даних описано у третьому розділі.

### **2.3.2 Модель процесу визначення семантичної близькості.**

Для створення зв'язку між професійними компетентностями студента та навчальними дисциплінами необхідно визначити, які саме дисципліни відповідають професійним схильностям (тобто, можуть бути рекомендованими). Такий процес найкраще проводити на основі порівняння ключових слів, що описують дисципліну, та ключових слів, що описують професійні вакансії, які підходять студенту. Необхідно також врахувати тренди на ринку праці, зібрати вимоги стейкхолдерів та проаналізувати відповідні ІТ-вакансії, для отримання повноцінної ідеї щодо професійної компетентності студента. Так як цей процес включатиме семантичні операції, називатимемо його процесом визначення семантичної близькості.

Використання даного методу зумовлене необхідністю у залученні в процес дисертаційного дослідження онтологічного підходу, щодо рекомендації навчальних дисциплін. Під поняттям онтології мається на увазі знання, що формально відображені на базі концептуалізації. У свою чергу, концептуалізація у контексті онтології – це опис множини об’єктів і понять, знань про них і зв’язків між ними [44]. Відповідно, онтологія складається з термів, понять чи концептів, що організовані в таксономію атрибутів та правил виведення. В основі онтології також задана певна предметна область (ПО) та відповідний їй інформаційний словник для користування. Формально, модель онтології представляють у вигляді наступного кортежу:

$$O = \{A, V, R\},$$

де  $A$  – скінченна множина понять (концептів, термів) ПО, яку задає онтологія  $O$ ;  $V : A \rightarrow A$  – скінченна множина відношень між концептами (поняттями, термами) заданої ПО;  $R$  – скінченна множина функцій інтерпретації (аксіоматизація, обмеження), заданих на концептах чи відношеннях онтології  $O$ . Зазначимо, що природним обмеженням, яке накладається на множина  $A$ , є її скінченність і непустота,  $V$  і  $R$  мають бути скінченними множинами [45-47].

Онтологію також можна представити у вигляді графу, де під дугами вказують напрями відношень між концептами ПО котрі, відповідно, є вершинами графу. Онтологія з якою працює система, представляє собою дисципліни та ключові слова, які її описують. Усі дисципліни отримані з освітньо-професійних програм Національного університету «Львівська

політехніка» ІТ напряму. Відповідно, ПО онтології вважатимемо вибіркові компоненти (дисципліни) з конкретної освітньо-професійної програми.

Під семантичною близькістю розуміється подібність пар об'єктів, а не семантичний зв'язок між ними, коли об'єкти різної природи володіють певним зв'язком. Семантична близькість об'єктів включає у собі безліч аспектів схожості, а вибір тих чи інших критеріїв для оцінки близькості в кожному окремому випадку створює складне і комплексне завдання, що залежить від цілей дослідження. Відповідно, міри близькості онтологічних термів використовують різноманітні семантичні характеристики порівнювальних термів, а також їх властивості та розміщення в онтологічних ієрархіях (атрибути і відношення між іншими термами).

Зокрема, у даній роботі [48] описані міри семантичної близькості відношень усередині онтології. Робота описує, що близькість двох термів оцінюється по положенню вершин, які відповідають цим термам у ієрархічних онтологічних структурах. Найпростіша міра близькості такого роду заснована на довжині найкоротшого шляху, що обчислюється кількістю вершин (або ребер) на шляху між двома відповідними вершинами таксономії [49], враховуючи глибину її ієрархії [50]. Тобто, чим менша довжина шляху між вершинами, тим вона менша.

Тим не менш, застосування міри близькості у рекомендаційній системі є неможливим, адже завдання полягає у визначенні семантичної близькості між елементом онтології та зовнішнім семантичним поняттям (словом). Відповідно, ми не можемо використати напрацювання згаданої наукової роботи.

Натомість, можна виокремити елементи онтології і представити їх у вигляді контекстної множини. Таким чином, необхідність у онтологічних методах та операціях зводиться до мінімуму. Це дозволяє використовувати загальні та легко доступні методи роботи з семантикою для досягнення поставлених цілей.

Таку контекстну множину можна задати у вигляді наступного кортежу:

$$C = \{c_1, c_2, \dots, c_n\},$$

Де  $n$  – кількість усіх ключових слів дисциплін. Завдяки відношенню між концептами ПО онтології, кожне ключове слово належить до своєї дисципліни, що можна легко відслідкувати для відповідної побудови рекомендацій.

Володіючи контекстною множиною усіх необхідних термів, з'являється можливість здійснювати безпосередньо семантичний пошук серед попередньо знайдених ІТ-вакансій, зокрема, нас цікавить опис такої вакансії, котрий зазвичай містить семантично близькі слова у розділі «вимоги до кандидата».

Практична реалізація алгоритму знаходження семантичної близькості описана у наступному розділі.

## **2.4 Модель процесу ранжування дисциплін з використанням методу аналізу ієрархій**

У роботі з рекомендаціями часто можуть виникнути неоднозначні випадки, коли варіанти, запропоновані користувачу, вважаються надто схожими і закладними для диференціювання. Рекомендовані елементи, котрими у нашому випадку є навчальні дисципліни, можуть створювати ситуацію невпевненості у виборі користувачем одного варіанту серед інших альтернатив. Наприклад, при рекомендуванні двох чи трьох схожих дисциплін, користувачу залишається лиш здійснити свій власний вибір, котрий може вплинути на його подальший освітній процес.

Додаткова складність створюється і через особливості формування індивідуальної траєкторії, адже самостійний вибір студента може виявитися помилковим у майбутньому. Зокрема, має місце парадокс, який полягає у занадто поспішному рішенні і створенні ситуації, коли студент забажатиме змінити дисципліни, бо вони йому не підходять.

Якщо класифікувати описану проблему як складне рішення, створюється можливість для використання математичних методів підтримки прийняття рішень. Також, потенційним шляхом вирішення проблеми можна вважати проведення додаткового ранжування отриманих рекомендацій. Ціллю ранжування є виявлення одного рішення на базі порівняння альтернатив. Теоретично ранжування дозволить позбавитись можливих неоднозначностей, звужити вибірку і, таким чином, забезпечити максимально точний і якісний результат формування рекомендацій.

У ході дисертаційного дослідження для вирішення описаної проблеми було використано метод аналізу ієрархій (MAI). MAI - це метод організації та аналізу складних рішень із використанням математики та психології. Основне

застосування методу полягає в підтримці прийняття рішень за допомогою ієрархічної композиції завдання та рейтингування альтернативних рішень [51]. Даний метод складається з трьох частин: кінцевої мети або проблеми, яку необхідно вирішити, всіх можливих рішень, які називають альтернативами, і критеріїв, за якими відбувається оцінка альтернатив. МАІ забезпечує раціональну основу для необхідного рішення, визначаючи його критерії та альтернативні варіанти, а також пов'язуючи ці елементи із загальною метою [52].

Його використання зумовлено тим, що у більшості випадків користувач матиме справу з двома чи трьома альтернативами на вибір, та потребуватиме зовнішньої оцінки отриманих рекомендацій. Таку оцінку надаватимуть експерти, що і буде використано у МАІ.

Використання МАІ полягає у п'яти кроках:

- 1) Представлення задачі у вигляді ієрархічної структури;
- 2) Визначення пріоритетів елементів у ієрархічній структурі та побудова попарних порівнянь;
- 3) Визначення вагових коефіцієнтів з попарних порівнянь;
- 4) Перевірка узгодженості матриці порівнянь;
- 5) Ієрархічний синтез.

Першим кроком є побудова ієрархічної структури задачі, яка буде включати у собі мету, критерії, альтернативи та інші фактори, що потенційно впливатимуть на вибір. Зрозуміло, що кожен елемент такої ієрархії представляє різні аспекти задачі. Практичною реалізацією ієрархічної

структури може слугувати дерево ієрархій з критеріями та альтернативами. Таке дерево зображене на рис. 2.5.

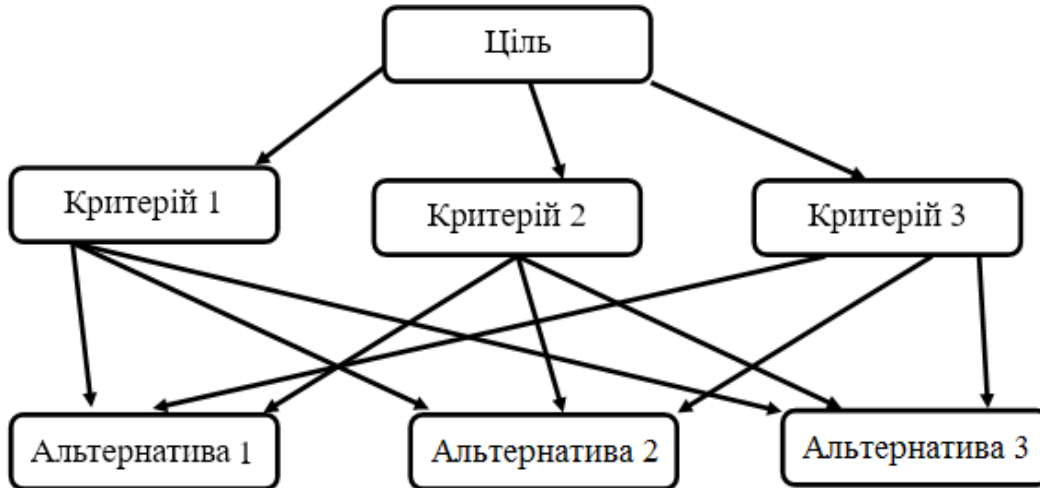


Рис. 2.5. Дерево ієрархій.

Наступним кроком є визначення пріоритетів, що визначають відносну важливість або перевагу тих чи інших елементів у побудованому дереві ієрархій. Для встановлення відносної важливості користуються шкалою відношень Сааті, що зображена у таблиці 2.1. Ця шкала дозволяє експерту ставити певне число, яке визначає відповідність ступеня переваги одного фактору перед іншим, для перетворення лінгвістичної оцінки у числову [53].

Таблиця 2.1

**Шкала відношень (ступеня значимості дій)**

Ступінь значимості	Визначення	Пояснення
1	Однакова значимість	Дві дії мають однаковий внесок у досягнення цілі
3	Слабка значимість	Існують не достатньо переконливі міркування на

		користь переваги однієї з дій
5	Істотна значимість	Наявні надійні дані для того, щоб продемонструвати перевагу однієї з дій
7	Очевидна значимість	Переконливе свідчення на користь однієї дії перед іншою
9	Абсолютна значимість	Незаперечні переконливі свідчення на користь переваги однієї дії перед іншою
2,4,6,8	Проміжні значення між сусідніми судженнями	Ситуація, коли необхідне компромісне рішення

Правомірність цієї шкали доведена практично при порівнянні з багатьма іншими шкалами. При використанні зазначеної шкали експерт, порівнюючи два об'єкти в змісті досягнення цілі, розташованої на вищому рівні ієрархії, повинен поставити у відповідність цьому порівнянню число в інтервалі від 1 до 9 або обернене до нього. У тих випадках, коли важко розрізнити скільки є проміжних градацій від абсолютної до слабкої переваги або цього не потрібно в конкретній задачі, може використовуватися шкала з меншим числом градацій. Гранично шкала має дві оцінки: 1 - об'єкти рівнозначні; 2 - перевага одного об'єкта над іншим. [53]

Побудувавши ієрархію, застосовується метод попарних порівнянь, що дозволяє сформувати відповідну матрицю попарних порівнянь. На рис. 2.6



зображена інструкція для побудови. Таку матрицю також необхідно побудувати для кожного критерію, визначеного у ієрархії.

	$E_1$	$E_2$	...	$E_n$
$E_1$	1	$a_{12}$	...	$a_{1n}$
$E_2$	$1/a_{12}$	1	...	$a_{2n}$
...	...	...	...	
$E_n$	$1/a_{1n}$	$1/a_{2n}$	...	1

Рис. 2.6. Інструкція для побудови матриці попарних порівнянь.

Матриця попарних порівнянь  $A$  у подальших обчисленнях дозволить оцінити узгодженість суджень експертів на основі трьох наступних величин:

- Максимальне власне значення
- Індекс узгодженості
- Індекс послідовності співвідношень.

Максимальне власне значення можна знайти за наступною формулою:

$$\lambda_{max} = e^T A W \quad (2.1)$$

де для обчислення індекса узгодженості використовується наступна рівність:

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (2.2)$$

Наступна формула дозволяє визначити індекс послідовності співвідношень:

$$CR = \frac{CI}{RI} \quad (2.3)$$

де  $e^T = (1,1, \dots, 1)^k$  – одиничний вектор,  $k$  – показник ступеня,  $W$  – власний вектор матриці.  $RI$  – випадковий показник, котрий залежить від  $n$  (тобто від кількості альтернатив) і визначається за таблицею 2.2, що наведена нижче:

**Таблиця 2.2**

**Значення табличного індексу узгодженості**

Порядок матриці (n)	1	2	3	4	5	6	7	8	9	10
Табличний індекс (RI)	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Табличне значення індексу може набувати і більшого значення, якщо альтернатив більше десяти. Тим не менш, зазвичай кількість альтернатив знаходиться в діапазоні від трьох до десяти.

Власний вектор матриці, який є необхідний для знаходження максимального значення також відповідає ваговим коефіцієнтам. Алгоритм знаходження власного вектора складається з наступних кроків:

1. Для кожного рядка матриці попарних порівнянь знайти середнє геометричне її елементів:

$$a_{ij} = (a_{ij}^1 * a_{ij}^2 * \dots * a_{ij}^S)^{\frac{1}{S}} \quad (2.4)$$

2. Знайти суму всіх середніх геометричних;
3. Поділити кожне середнє геометричне на їх суму (нормування).

Узгодженість визначається за наступною нерівністю:

$$CR < 10\% \quad (2.5)$$

Якщо нерівність виконується, то матрицю попарних порівнянь можна вважати узгодженою. Обчислення узгодженості необхідно так само провести і для усіх інших критеріїв, за якими визначаються альтернативи.

Після усіх обчислень узгодженості, наступним кроком буде визначення пріоритету критеріїв з огляду на їх важливість для досягнення мети та проведення ієрархічного синтезу. Цей процес полягає у визначення пріоритету альтернатив з огляду на їх важливість для досягнення цілі.

Для прийняття рішення необхідно обрати пріоритет альтернативи з найбільшим числовим значенням. Обрана альтернатива вважатиметься обраною експертами і буде включена у рекомендації користувача.

## **2.5 Методи оцінки розмірності даних**

Серед методів для визначення розмірності даних та інформації в цілому слід виділити наступні:

*t-SNE*. Т-розподілене вкладення стохастичної близькості (*t-SNE*) це неконтрольований нелінійний метод, який використовується в основному для дослідження даних і візуалізації багатовимірних даних. Іншими словами, метод *t-SNE* надає розуміння того, як розташовуються дані у багатовимірному просторі. Він був розроблений Лоренсом ван дер Маатенсом та Джеффри Хінтоном у 2008 році [54]. Алгоритм *t-SNE* обчислює міру подібності між парами екземплярів у просторі високої розмірності та у просторі низької розмірності [55]. Він моделює ймовірнісний розподіл сусідів навколо кожної

точки. Тут термін сусіди відноситься до набору точок, найближчих до кожної точки. У вихідному багатовимірному просторі це моделюється як розподіл Гаусса. У двовимірному вихідному просторі це моделюється як  $t$ -розподіл. Мета процедури – знайти таке відображення на двовимірний простір, яке мінімізує різницю між цими двома розподілами по всіх точках. Головний параметр, який контролює вибірку, називається *перплексією*. *Перплексія* приблизно еквівалентна кількості найближчих сусідів, які враховуються при зіставленні вихідного та вибіркового розподілів для кожної точки.

*UMAP*. Апроксимація та проєкція однорідного різноманіття (*UMAP*) – за своєю суттю працює дуже схоже на *t-SNE* – обидва використовують алгоритми компонування графа для розміщення даних у низькорозмірному просторі. Простіше кажучи, *UMAP* будує подання даних у вигляді багатовимірного графа, а потім оптимізує низькорозмірний граф, щоб він був якомога структурніше схожим. У той час як математика, що використовується у *UMAP* для побудови багатовимірного графа, є достатньо просунутою, ідеї які за нею стоять є напрочуд простими. Алгоритм заснований на трьох припущеннях щодо даних:

- Дані поступово розподілені на Рімановому многовиді [56];
- Ріманова метрика є локально сталою (або може бути апроксимована як така);
- Многовид є локально пов'язаним.

Виходячи з цих припущень, можна змоделювати многовид із нечіткою топологічною структурою. Вбудовування здійснюється шляхом пошуку низькорозмірної проєкції даних, що має максимально близьку еквівалентну нечітку топологічну структуру [57].

*PCA*. Аналіз головних компонентів (*PCA*) – популярний метод аналізу великих наборів даних, що містять велику кількість вимірів/ознак на одне

спостереження, що підвищує інтерпретацію даних при збереженні максимального обсягу інформації і дозволяє візуалізувати багатовимірні дані. Формально *PCA* – це статистичний метод зменшення розмірності набору даних. Це досягається шляхом лінійного перетворення даних у нову систему координат [58], в якій (більшу частину) варіації даних можна описати з меншою кількістю вимірювань, ніж вихідні дані. У багатьох дослідженнях використовуються перші два основні компоненти, щоб відобразити дані у двох вимірах та візуально ідентифікувати кластери тісно пов'язаних точок даних. Аналіз основних компонентів знаходить застосування у багатьох галузях, як-от популяційна генетика, дослідження мікробіома і атмосферні науки.[59]

Щодо *UMAP* та *t-SNE*, обидва методи є нелінійними та недетермінованими і засновані на упорядковуванні точок у сусідні графи. *UMAP* забезпечує відображення у скорочений простір, яке використовується для нових даних при передбаченні координат у скороченому просторі.

Найбільша різниця між вихідними даними *UMAP* порівняно з *t-SNE* полягає у балансі між локальною та глобальною структурою – *UMAP* часто значно краще зберігає глобальну структуру в остаточній проекції. Це означає, що міжкластерні відносини потенційно володіють більшим пріоритетом, ніж у *t-SNE*. Однак важливо відзначити, що оскільки *UMAP* і *t-SNE* обов'язково спотворюють багатовимірну форму даних при проектуванні на нижчі виміри, будь-яка задана вісь або відстань у нижчих вимірах, як і раніше, не піддаються прямої інтерпретації з точки зору таких методів як *PCA*.

Загалом, *UMAP* є чудовим ресурсом для зменшення розмірності. *t-SNE* в основному призначений виключно для побудови графіків, і не рекомендується до використання у модельному конвеєрі у зв'язку з тим, що неможливо використати однаковий *t-SNE* на тестовому наборі.

*t-SNE* відрізняється від *PCA* тим, що зберігає лише невеликі попарні відстані або локальну подібність, тоді як *PCA* займається збереженням великих попарних відстаней для максимізації дисперсії. На рисунку 2.7 досить добре зображується підхід методів *PCA* та *t-SNE*, використовуючи класичний набір даних “*Swiss Roll*” [60]. На рисунку можна побачити, що через нелінійність цього набору даних та завдяки збереження великих відстаней, *PCA* неправильно зберігає структуру даних.

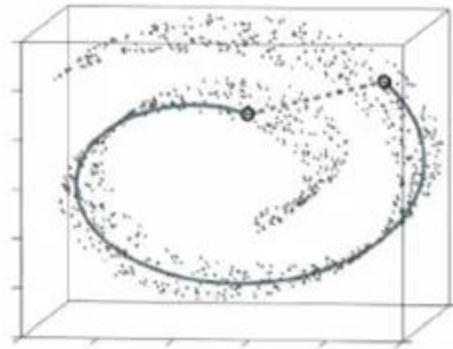


Рис. 2.7. Класичний набір даних “*Swiss Roll*”. Збереження невеликої відстані за допомогою *t-SNE* (суцільна лінія) порівняно з максимізацією дисперсії *PCA*.

*PCA* є закономірним лінійним відображенням і має тенденцію захоплювати глобальну структуру. Алгоритм використовує підхід на основі проєкцій, а саме, дозволяє визначити найближчу гіперплощину до даних, а також створити проєкцію усіх точок навчальної множини на визначену гіперплощину.

На фоні очевидно переваги *PCA* над іншими розглянутими методами (враховуючи, що набори даних, з’являється необхідність додатково розглянути лінійні методи, що працюють за схожим з *PCA* принципом, зокрема:

- *FA*. Основна мета факторного аналізу (*FA*) – не просто зменшити розмірність даних. Факторний аналіз - це корисний підхід для пошуку

прихованих змінних, які не вимірюються безпосередньо в одній змінній, а виводяться з інших змінних набору даних. Ці приховані змінні називають факторами.

- *LDA*. Лінійний дискримінантний аналіз (*LDA*) зазвичай використовують для багатокласової класифікації. Його також можна використовувати як метод зменшення розмірності. *LDA* найкраще поділяє або розрізняє навчальні екземпляри за їхніми класами. Основна відмінність між *LDA* та *PCA* полягає в тому, що *LDA* знаходить лінійну комбінацію вхідних ознак, яка оптимізує роздільність класів, у той час як *PCA* намагається знайти набір некорельованих компонентів максимальної дисперсії у наборі даних. Ще одна ключова відмінність між ними полягає в тому, що *PCA* - це неконтрольований алгоритм, тоді як *LDA* - це контрольований алгоритм, в якому враховуються позначки класів. Важливо також звернути увагу, що для методу *LDA* присутні деякі обмеження, зокрема, щоб застосувати *LDA*, дані повинні бути нормально розподілені. Набір даних також має містити позначки відомих класів. З іншого боку, *PCA* потребує масштабованих даних. Однак для *PCA* мітки класів не потрібні. Максимальна кількість компонентів, яку може знайти *PCA*, — це кількість вхідних функцій у вихідному наборі даних.
- *SVD*. Цей метод зменшує лінійну розмірність за допомогою сингулярного розкладу матриці (*SVD*). Він добре працює з розрідженими даними, в яких більшість значень рядків дорівнюють нулю. У той же час, *PCA* націлений на роботу із щільними даними. Тим не менш, *SVD* також можна використовувати із щільними даними, проте ефективність буде значно нижчою. Ще одна ключова

відмінність між *SVD* і *PCA* полягає в тому, що факторизація для *SVD* виконується на матриці даних, а факторизація для *PCA* виконується на коваріаційній матриці.

Враховуючи лінійність процесу формування індивідуальної освітньої траєкторії, що була описана у першому розділі, та відсутність необхідності у складній факторизації, у ході дослідження був здійснений вибір в сторону методу *PCA*. Вибір даного методу також зумовлений використанням онтології для визначення навчальних дисциплін і представлення індивідуальної освітньої траєкторії через функцію багатьох змінних, враховуючи, що змінні відповідатимуть навчальним курсам, а розмірність матриці для *PCA* - кількістю дисциплін.

## **Висновки до розділу 2**

1. Подано формальну модель рекомендаційної системи. Визначено множину суб'єктів (агентів) рекомендаційної системи та представлено їх взаємодію та логіку поведінки.
2. Розроблено концептуальну модель формування рекомендацій навчальних дисциплін для їх включення у ІОТ. Етапи формування рекомендацій зображені на діаграмі діяльності.
3. Описані та проаналізовані різноманітні методи для зменшення розмірності даних. Були розглянуті їх особливості та недоліки, що дозволило



вибрати метод РСА як такий, що задовольняє цілі дисертаційного дослідження.

4. Детально проаналізовані та описані усі етапні моделі у формуванні рекомендацій: тест на профорієнтацію, визначення семантичної близькості та метод аналізу ієрархій.

## **РОЗДІЛ ІІІ. МЕТОДИ КОМПЛЕКСНОГО АНАЛІЗУ**

Мета третього розділу – описати методи збору та аналізу даних. Зокрема, мова йде про парсинг даних з веб-сайтів ІТ-вакансій, визначення семантичної близькості, використання методу аналізу ієрархій, та застосування методу головних компонент. Визначення семантичної близькості забезпечується використанням методів мови програмування РНР; додатково розглядається метод Левенштейна та його недоліки при визначенні семантичної близькості. Експериментальні обчислення доводять ефективність та практичність використання методу аналізу ієрархій та методу головних компонент у рекомендаційній системі.

### **3.1 Метод аналізу даних з веб-сайтів ІТ-вакансій.**

У попередньому розділі був описаний процес визначення професійної схильності через профорієнтаційні тести. Такі тести є необхідними для збору даних про користувача з метою надання йому рекомендацій. Як було описано раніше, зібрані дані зберігаються у вигляді домінуючого типу особистості користувача. У користувача може бути декілька домінуючих типів, тож необхідно обрати професії, що відповідають усім визначеним. На основі цих професій буде здійснюватися пошук відповідних ІТ-вакансій.

Використання додаткового інструментів для збору даних є необхідним кроком у зв'язку з обмеженою інформацією про користувача. У розробленій в ході дослідження системі для аналізу користувача використовуються лише результати тестів на профорієнтацію, незважаючи потенційну можливість збору додаткової інформації. Така особливість зумовлена необхідністю, в

першу чергу, даних про професію та вимог, які представлені до неї на ринку праці, а не в детальній характеристиці користувача. Відповідно, для формування рекомендацій по ІТ-спеціальностям необхідно проаналізувати існуючі вакансії та зрозуміти, який попит наразі присутній на професійному ринку. Це дозволить визначити професійні компетентності та залучити стейкхолдерів у процес формування ІОТ.

Таким чином, в якості інструменту для збору даних про вакансії виступатиме парсер. Парсер (англ. *parser*; від *parse* – аналіз, розбір), він же синтаксичний аналізатор, – частина програми, що перетворює вхідні дані (як правило, текст) в певний структурований формат, необхідний для завдань подальшого їх (даних) аналізу та використання. Технічно, парсер виконує синтаксичний аналіз даних[61].

Загалом виділяють п'ять основних різновидів парсингу:

*Описовий.* Альтернативна назва цього – описовий аналіз даних. Метою є систематизація емпіричних фактів. Аналіз покликаний відповісти питанням: «Що сталося?». Наприклад, у такому дослідженні можуть бути дані про обсяг продажу за минулий місяць, або інформація про чисельність запитів на ту чи іншу послугу. За допомогою описового парсингу здійснюється пошук інформації, її узагальнення та угруповання [62].

При цьому така техніка не здатна виявити причини події, що відбулася, тому описовий аналіз, як правило, використовується в поєднанні з іншими способами.

*Дослідницький.* Призначенням цього парсингу є пошук взаємозв'язків між даними та складання гіпотез. Без виконання такої дослідницької дії інформація про ці взаємозв'язки та змінні буде неповною.

Класична сфера використання аналізу розвідувальних даних – видобуток інформації. Досліджуючи отримані відомості та виявляючи взаємозв'язки, можна зрозуміти причини процесів.

*Діагностичний.* Тут виявляється причина події, що сталася. За рахунок порівняння різних даних, виявлення закономірностей та зв'язків між подіями, діагностичний парсинг визначає фактори, що вплинули на їхнє виникнення. Аналізуючи цю інформацію, фірма має можливість нівелювати проблеми після визначення їх фундаментальних причин [62].

*Прогностичний.* Метою прогностичного парсингу є визначення подій, які можуть статися у майбутньому. Для цього застосовуються дані, отримані в результаті вищеописаних технік. Крім того, використовуються алгоритми та методи штучного інтелекту, а також машинного навчання. За рахунок знаходження взаємозв'язків і причин майбутні події стають з різним ступенем передбачуваними.

Точність прогнозу визначається кількістю інформації, знайденої на попередньому етапі, та якістю інтелекту алгоритмів. За допомогою такого парсингу можливо з достатнім ступенем точності передбачити кількість продажів у наступному місяці, уявити, яким чином поведитимуться покупці та припустити багато іншого.

*Попередньо-записуючий.* Такий парсинг вважається найскладнішою та найдорожчою аналітичною технікою. У цьому випадку застосовуються результати, що належать до категорій аналізаторів. Застосовуються права *ML* та *AI*, права та правила ведення бізнесу, нейронні права [62].

Об'єктом парсингу може бути будь-яка граматично структурована система: інформація, закодована природною мовою, мовою програмування, математичними виразами тощо. Наприклад, якщо вихідний масив даних є *HTML*-сторінкою, парсер може вичленувати з коду інформацію і перевести її в текст, зрозумілий для людини. Або конвертувати в *JSON* — формат для програм та скриптів [63]. На рис. 3.1 зображена загальна схема роботи парсера.

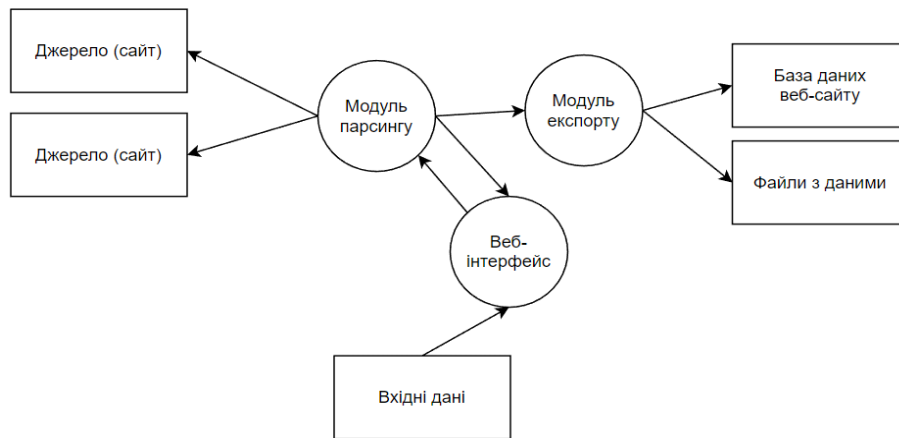


Рис. 3.1. Схема роботи парсера.

Доступ парсера до сайту можливий:

- через протоколи *HTTP*, *HTTPS* чи веб-браузер;
- з використанням бота, який має права адміністратора.

Отримання даних парсером – семантичний аналіз вихідного масиву інформації. Програма розбиває його на окремі частини (лексеми): слова,

словосполучення тощо. Парсер проводить їх граматичний аналіз, перетворюючи лінійну структуру тексту на деревоподібну (синтаксичне дерево). Приклад древа синтаксичного аналізу: вираз *expr*, термін *term*, факт *fact*, ідентифікатор *ID* [64]. Також приклад синтаксичного дерева наведено на рисунку 3.2.

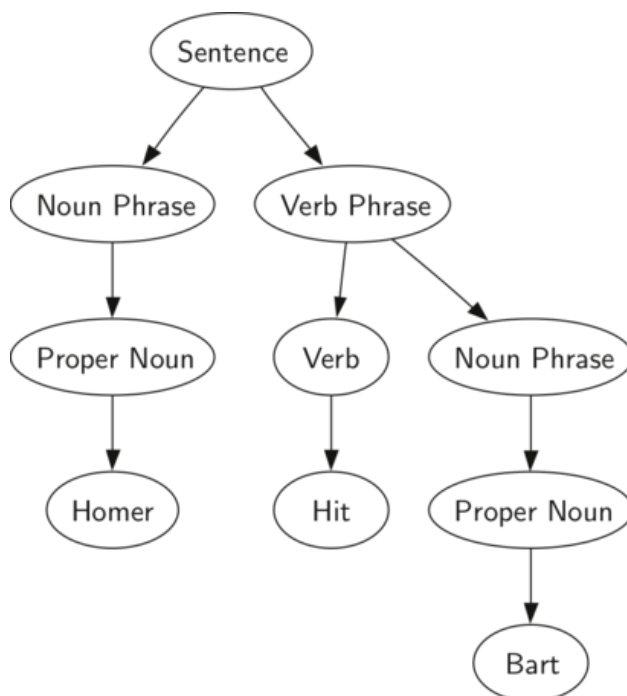


Рис. 3.2. Дерево синтаксичного аналізу простої пропозиції [65].

Така форма спрощує «розуміння» інформаційного масиву комп'ютерною програмою та буває двох типів:

- дерево залежностей - така структура складається з компонентів, що знаходяться в ієрархічних відносинах один до одного;
- дерево складових - у структурі цього типу компоненти перебувають у тісній залежності один з одним, але без ієрархічних відносин.

Також результат роботи парсера може бути поєднанням моделей. Програма діє за одним із двох алгоритмів:

- Спадний парсинг. Аналіз здійснюється від загального до приватного, а синтаксичне дерево розростається донизу, в напрямку більш детальної розбивки.

- Висхідний парсинг. Аналіз і побудова синтаксичного дерева здійснюються знизу вгору, з нижнього символу рядка, а потім встановлює все більші синтаксичні зв'язки. Це робиться доти, доки не буде досягнуто початкового символу коду.

Вибір конкретного методу парсингу залежить від кінцевої мети. У будь-якому випадку, парсер повинен вміти вичленяти із загального масиву тільки необхідні дані, а також перетворювати їх у зручний для вирішення завдання формат, наприклад, у формат *JSON*, *CSV* або навіть таблицю *Excel* [66].

При виборі парсера слід враховувати формати виводу, які підтримує сервіс, його здатність працювати з сучасними сайтами — наприклад, підтримку елементів керування *Ajax*, а також можливості автоматизації та форми звітності [61].

Застосування програм-парсерів дозволяє:

- автоматизувати процес аналізу та знижувати навантаження на співробітників, перенаправляти їх час та сили на вирішення інших завдань;
- прискорювати аналіз великого обсягу інформації, наприклад, кількох сотень сторінок інтернет-магазину або велику базу даних;
- виявляти помилки на сайті або будь-якому іншому інформаційному продукті, якщо в програмі задані налаштування на їх пошук.

- висока точність вирішення конкретного завдання, мінімальна ймовірність помилки, адже механізм не випускає нічого зі свого поля зору, на відміну від людини. Дефекти можуть виникнути лише при прорахунках у налаштуванні парсера;
- автоматична конвертація даних у потрібний формат; відсутність ідентифікації парсингу як DDoS-атаки.

До недоліків парсерів можна віднести не завжди релевантний аналіз даних. Однак у більшості випадків це залежить від можливостей програми, якості її налаштування користувачем. У більшості випадків інформація, що видається парсером, потребує незначної обробки для подальшого використання.

наявність захисту від даного способу індексування інформації у певних IP-адрес і неможливість збору всіх потрібних даних, що впливає з цього. Також можливі перерви у роботі через численність користувачів на сайті. Це відбувається через те, що парсер не може оновити веб-сторінку для оптимального завантаження.

Парсинг застосовується в будь-яких областях, де потрібно проаналізувати та систематизувати великі обсяги даних, в тому числі і отримання інформації про вакансії з відповідних веб-сайтів.

Принцип роботи парсерів для веб-сторінок однаковий, зазвичай він складається з трьох етапів.

1) Запит-відповідь. Перший крок – запитати у цільового сайту вміст певної *URL*-адреси. В результаті парсер отримує запитану інформацію у форматі *HTML*.



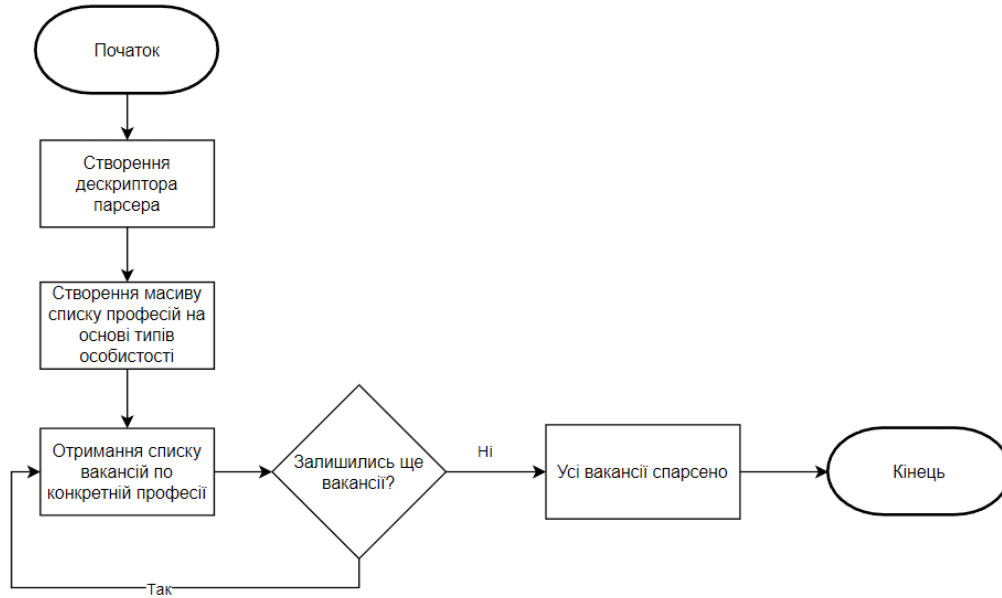
2) Розбір та вилучення. Синтаксичний аналіз зазвичай застосовується до будь-якої мови веб-програмування. Цей процес представляє собою інтерпретацію коду у вигляді тексту та створення відповідної структури в пам'яті. Це дозволить інтерпретатору “розуміти” дані з якими йому доведеться працювати.

Простіше кажучи, парсер бере HTML-код і витягує звідти відповідну інформацію — таку, як заголовок сторінки, абзаци, підзаголовки, посилання, виділення жирним, потрібні теми тощо, проводячи парсинг тексту.

3) Завантаження даних. Отримані дані завантажуються та зберігаються. Формат файлу визначається таким чином, щоб його можна було відкрити в іншій потрібній програмі. Для Google Таблиць це, наприклад, CSV, для парсингу бази даних - JSON і таке інше.

У веб-розробці та просуванні серед пошукових систем використовується велика кількість безкоштовних та платних програм для парсингу сайтів. Від простих онлайн-сервісів до браузерних плагінів і ПЗ для настільних ПК. Тим не менш, використання чужих напрацювань знижує цінність дисертаційного дослідження. Таким чином, було вирішено використовувати мову програмування РНР для написання програми, яка би задовольняла вимоги та цілі рекомендаційної системи.

В результаті, був розроблений алгоритм парсингу, який здійснює пошук по веб-сайтам (наприклад - <https://www.work.ua/>) на предмет співпадіння ІТ-вакансії і визначеної професії користувача. Алгоритм роботи програми парсингу зображений на рис. 3.3.



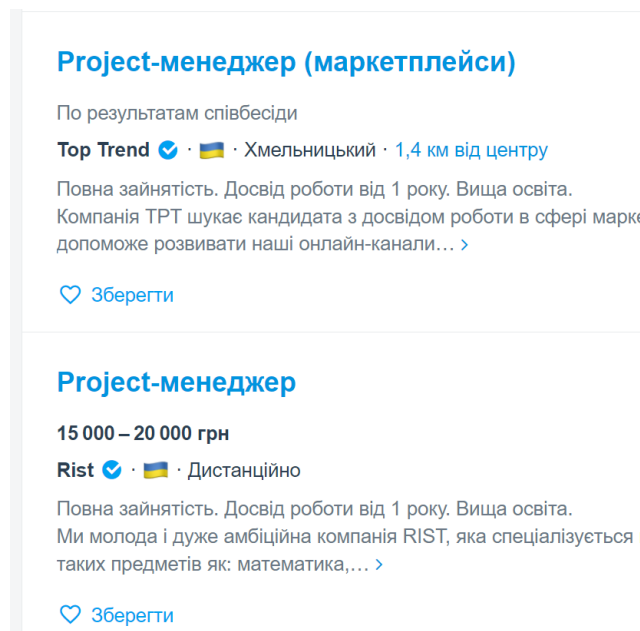
*Рис. 3.3. Алгоритм роботи парсеру.*

### **3.1.1 Аналіз даних ІТ-вакансій.**

Для парсингу, очевидно, не можуть бути використані попередньо отримані домінуючі типи особистості користувача, так як пошук по таким характеристикам навряд чи дасть необхідний результат. Це зумовлено тим, що більшість вакансій на ринку заповнені однаково, за шаблонами, без врахування попередньо пройдених профорієнтаційних тестів. Роботодавці часто вимагають у кандидатів володіння досвідом у конкретних сферах чи галузях. З попереднього аналізу ІТ-вакансій, дуже мала їх кількість згадує особистісні якості, хоча і трапляються винятки, як, наприклад, з вимогою «володіти інтелектуальним складом особистості». Тим не менш, така характеристика є загальною і може підійти будь-кому, тому на неї звертають дуже мало уваги.

Таким чином, пошук необхідно здійснювати по словам, що найкраще підходять під той чи інший домінуючий тип особистості – мова йде про ІТ-професії, які належать кожному типу. Так склалося, що у більшості профорієнтаційних тестів Холланда, після його проходження, одразу рекомендують цілий список професій, які підходять під визначений професійний тип. Якщо обмежити цей список лише ІТ-професіями, це дозволить отримати потужну базу пошуку, яка надзвичайно популярна на ринку праці. Відповідно, у ході дослідження було обрано список ІТ-професій згідно з наступним джерелом [67].

При аналізі ринку праці та популярних веб-сайтів для проведення експериментальних обчислень був обраний веб-сайт <https://www.work.ua/>. Додатково проведено короткий огляд вакансій на предмет схожості з джерелом [67]. Огляд вакансій зображений на рис. 3.4 – 3.7.



The image shows two job listings from the work.ua website. The first listing is for a 'Project manager (marketplace)' position at 'Top Trend' in Khmelnytskyi, with a salary of 15,000 - 20,000 UAH. The second listing is for a 'Project manager' position at 'Rist', which is remote, with the same salary range. Both listings mention 'Full-time position' and 'Higher education'.

**Project-менеджер (маркетплейси)**  
По результатам співбесіди  
**Top Trend** · Хмельницький · 1,4 км від центру  
Повна зайнятість. Досвід роботи від 1 року. Вища освіта.  
Компанія ТРТ шукає кандидата з досвідом роботи в сфері маркет допоможе розвивати наші онлайн-канали... >  
Зберегти

**Project-менеджер**  
15 000 – 20 000 грн  
**Rist** · Дистанційно  
Повна зайнятість. Досвід роботи від 1 року. Вища освіта.  
Ми молода і дуже амбіційна компанія RIST, яка спеціалізується на таких предметів як: математика,... >  
Зберегти

Рис. 3.4. Огляд вакансій «Проект-менеджер».

## Опис вакансії

Ми молода і дуже амбіційна компанія RIST, яка спеціалізується на викладанні таких предметів як: математика, українська мова, англійська мова, історія України та інших, для учнів 2 — 11 класів!

Наразі ми займаємось розробкою власної CRM-системи, яка значно оптимізує роботу нашої команди. Наш проект знаходить наstadії MVP і ми плануємо найближчим часом ввести його в реалізацію. Тож шукаємо класного Project-менеджера в нашу дружню родину ❤️


Ти наш ідеальний кандидат, якщо маєш:

- сильні комунікативні, організаторські та лідерські навички;
- знання методології та інструментів управління проектами;
- не менше 1 року успішного досвіду на посаді Project-менеджера;
- розуміння, що таке бізнес процеси і навички з покращення цих процесів у команді.



Обов'язки:

- ведення проекту від етапу планування до реалізації;
- написання user stories;
- робота в Jira;
- комунікація із клієнтами та їх сапорт;
- тестування системи на рівні користувача;
- менеджмент команди з 2+ осіб;
- проведення основних церемоній Скраму;
- People management: підтримка та мотивація команди, 1–1 зустрічі.



Рис. 3.5. Огляд вакансії «Проект менеджер».

**Full stack програміст** 


65 000 грн



etree Group   · Дистанційно

Повна зайнятість. Досвід роботи від 5 років. Вища освіта.  
etree Group — дистриб'ютор ІТ обладнання, програмного забезпечення та побутової електроніки, а також... >



 Зберегти  Гаряча

---


**Middle Backend Developer (PHP, Laravel)** 



ВіЯр   · VIP · Київ · 7,6 км від центру

Повна зайнятість. Досвід роботи від 2 років. Вища освіта.  
«ВіЯр» — провідна українська компанія з виробництва та продажу меблевих комплектуючих. Компанія, не зважаючи на... >

 Зберегти  Гаряча

---

**Програміст 1С у Городок** 

Т.В. Fruit   · Городок (Львівська обл.), шукаємо у Львові

Повна зайнятість. Досвід роботи від 1 року.  
Міжнародна Група компаній Т.В. Fruit — провідний переробник фруктів, ягід та овочів в світі. До складу... >



 Зберегти  Гаряча

Рис. 3.6. Огляд вакансій «Програміст».

### Опис вакансії

**etree Group** — дистриб'ютор ІТ обладнання, програмного забезпечення та побутової електроніки, а також інтернет-магазин, заснований у 2009 році в Кельні, Німеччина, і має в своєму портфоліо понад 40 000 товарів від 50 виробників. Ми продаємо та доставляємо по всьому світу монітори, проектори, принтери, сканери, ноутбуки, ПК, аксесуари та багато іншої електроніки. Завдяки високій якості роботи та гарному сервісу etree Group завоювала довіру партнерів, клієнтів та постачальників по всьому світу. Наразі ми маємо головний офіс у Кельні, Німеччина, та два офіси у Вроцлаві, Польща, та Одесі, Україна. Оскільки ІТ-індустрія стрімко розвивається, ми активно розширюємо нашу команду.

#### Ваші обов'язки

- Створення системи електронного документообігу для таких документів, як замовлення, рахунки-фактури та платежі
- Розробка та підтримка EDI-рішень для задоволення потреб клієнтів та постачальників
- Тісна співпраця з клієнтами та іншими внутрішніми командами для розуміння бізнес-вимог та перетворення їх у технічні рішення
- Співпрацювати з іншими розробниками для інтеграції рішень EDI з існуючими системами
- Виконувати тестування, налагодження та усунення несправностей для забезпечення цілісності системи EDI
- Бути в курсі останніх тенденцій та розробок в області технологій EDI

*Рис. 3.7. Огляд вакансії «Програміст».*

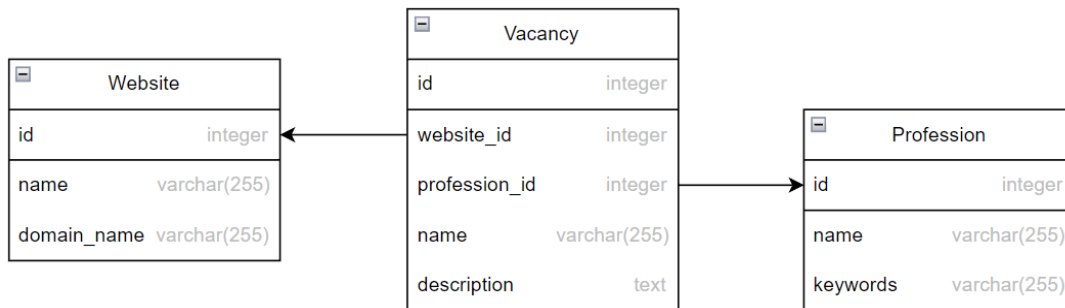
Відповідно, результатом роботи парсингу будуть вимоги до кандидатів, що прописані чи не у кожній ІТ-вакансії. Зазвичай саме ці текстові блоки містять ключові слова, які потенційно володіють семантичною близькістю до дисциплін з освітньо-професійних програм.

Результати парсингу записуються для зберігання та забезпечення доступності інструментами системи за допомогою додатково розробленої база даних, що містить наступну інформацію:

- Ключові слова професії;
- Назву та домен веб-сайту;
- Назву та опис вакансії.

Відповідні інформаційні таблиці, що містять інформацію про веб-сайти, вакансії та професії наведені на рисунку 3.8. Кожна професія, що знаходиться через пошук у вакансіях володіє певним набором ключових слів. Множина

таких даних є необхідною для подальших обчислень та визначення семантичної близькості.



*Рис. 3.8. Інформаційні таблиці, що містять дані, отримані в результаті парсингу.*

### **3.2 Знаходження семантичної близькості на основі методів роботи з даними.**

Як було зазначено у попередньому розділі, процес визначення навчальних дисциплін, які рекомендовані до включення у індивідуальну освітню траєкторію користувача полягає у поетапному порівнянні текстів, що отримані у результаті парсингу і представляють його професійну спрямованість та текстів, що отримані з силабусів і представляють собою ключові слова з онтології дисциплін. Якщо ключові слова дисциплін ідентичні, або ж семантично близькі до професійних нахилів студента, можемо вважати такі дисципліни рекомендованими до включення у ІОТ.

У статтях [7] і [68] описуються різні методи для знаходження семантичної близькості між двома словами. У першому випадку використовуються контекстні множини, щоб алгоритм мав базис для пошуку і порівняння. У другій статті описується метод Левенштейна. Використання

даного методу аргументується тим, що для проблеми формування індивідуальної освітньої траєкторії достатньо буде прямого порівняння відстані між словами. Якщо відстань буде достатньо малою, значить порівнювальні слова можна вважати семантично близькими. Проте даний алгоритм володіє рядом недоліків, серед яких низька точність та потреби у великих об'ємах пам'яті. Особливо це помітно при порівнянні великих текстових файлів або пошуку цілих фраз чи словосполучень, без врахування семантичних особливостей, що часто присутні у описах вакансій, таких як англіцизми або перефразовані вимоги.

У попередніх розділах було описано і обумовлене використання методів роботи з даними крім рекомендаційних алгоритмів. Враховуючи величезну популярність веб-технологій, вибір для дисертаційного дослідження був зроблений саме у їх користь.

Відповідно, методи роботи з парсингом даних представлені у вигляді методу *strpos*, що написаний на мові програмування *PHP* [69]. Даний метод приймає два рядки і числове значення з трьома відповідними параметрами:

*Haystack* - рядок, в якому проводиться пошук.

*Needle* - Залежно від задуманої поведінки, параметр *needle* має бути або явно приведений до рядка, або повинен бути виконаний явний виклик *chr()*.

*Offset* - Якщо цей параметр вказано, пошук буде розпочато із зазначеної кількості символів з початку рядка. Якщо задано негативне значення, відлік позиції початку пошуку буде здійснено з кінця рядка.

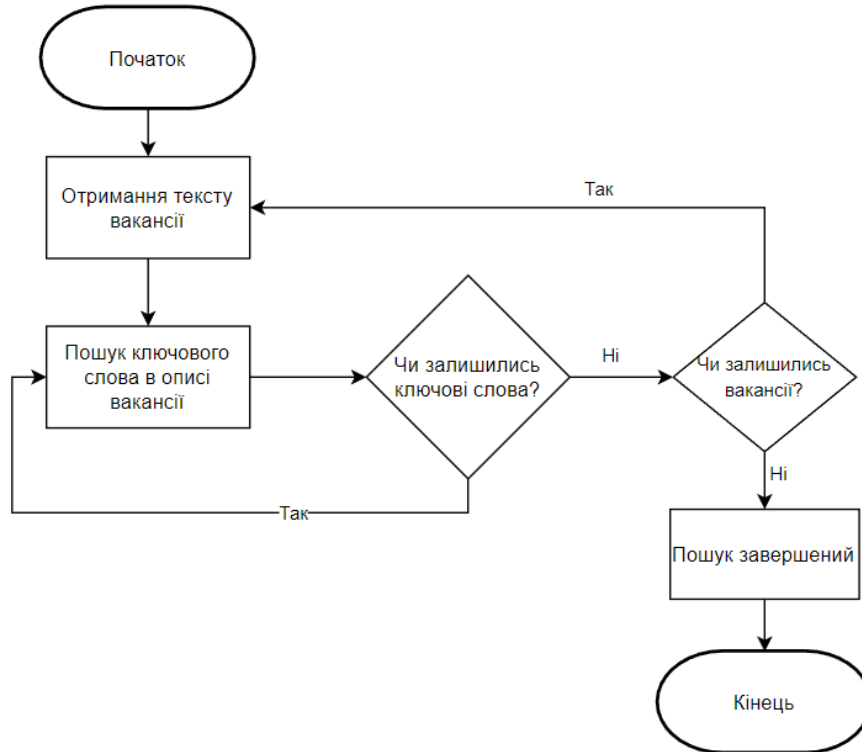
Для визначення семантичної близькості використовується метод *similartext*, він дозволяє знаходити семантичну близькість між двома текстовими рядками. Даний метод обчислює ступінь схожості двох рядків за алгоритмом, описаним у книзі [70]. Реалізація алгоритму у використаному методів не використовує стека, як у оригіналі, натомість, застосовуються

рекурсивні виклики, що у деяких випадках може прискорити процес. Складність алгоритму становить  $O(N^3)$ , де  $N$  - довжина рядка, який є довшим за інший з двох порівнюваних. Алгоритм повертає кількість символів, що збігаються, в обох рядках. Кількість символів, що збігаються, обчислюється шляхом знаходження найдовшого першого загального підрядка, а потім рекурсивним виконанням цього для префіксів і суфіксів. Довжина всіх знайдених загальних підрядків також додається. На виході отримуємо відсоткове значення того, наскільки порівнювальні рядки схожі між собою.

Описаний *RHP* метод при використанні у дисертаційному дослідженні може видавати значну похибку, так як порівняння здійснюється не просто з текстовими рядками, а величезними текстовими масивами, отриманими в результаті парсингу. Тому було вирішено доповнити даний метод використанням *substr\_count*. Проте цей метод на виході дає числове значення, а не процентне відношення, так як розмірність є різною. Тому з'явилася необхідність у введенні додаткової усередненої характеристики, що повертає кількість входжень кожної альтернативи.

Алгоритм роботи методу зображений на рисунку 3.9.





*Рис. 3.9. Блок-схема роботи алгоритму, який здійснює пошук семантичної близькості.*

### **3.2.1 Перевірка роботи алгоритму знаходження семантичної близькості**

Для перевірки алгоритму були взяті дані з освітньо-професійної програми «Системний аналіз» за перший (бакалаврський) рівень вищої освіти на офіційному веб-сайті Національного університету «Львівська політехніка» [71]. Зауважимо, що серед усіх вибіркового компонентів, доступні лише ті, що належать до трьох вибіркового блоків: Системи і методи прийняття рішень, Консолідована інформація та Аналіз даних. Додатково для вибору будуть також доступні і загальноуніверситетські дисципліни, проте вони значно збільшують об'єм вибірки. Відповідно, з'являється необхідність у

використанні методів зменшення розмірності даних. Використання такого методу описано у наступному підрозділі.

Суто для дисциплін вибіркового блоку користувачу буде запропоновано вибрати один вибірковий блок з трьох. Саме з дисциплінами з обраного блоку і буде здійснюватися перевірка алгоритму для визначення семантичної близькості.

У дистрибутивній семантиці слова зазвичай представляються як вектори у багатовимірному просторі їх контекстів. Семантична схожість обчислюється як косинусна близькість між векторами двох слів і може набувати значень у проміжку  $[-1 \dots 1]$  (на практиці часто використовуються тільки значення вище 0). Значення 0 приблизно означає, що ці слова не мають схожих контекстів і їх значення не пов'язані один з одним. Значення 1 навпаки свідчить про повну ідентичність їх контекстів і, отже, про близьке значення [72].

Відповідно, для перевірки результатів вважатимемо, що користувач обрав вибірковий блок 1: Системи і методи прийняття рішень. Список дисциплін у цьому блоці наступний:

- Технології бізнес-аналізу
- Моделювання бізнес-процесів
- Управління вимогами в ІТ-проектах
- Інженерія програмного забезпечення
- Ділові комунікації
- Планування ІТ-проектів
- Аналітичні сховища даних
- Інновації в бізнес-аналізі

Множина ключових слів, що отримана з онтології виглядає наступним чином:

$$C = \{c_1, c_2, \dots, c_n\},$$

Де  $n$  – кількість слів у множині.

Пошук відбуватиметься по рядкам у текстах ІТ-вакансій, що були попередньо отримані в результаті парсингу, використовуючи метод *strops*, що був описаний у попередньому підрозділі.

Відповідно до отриманих результатів, проведемо експериментальні обчислення використовуючи метод Левенштейна. Використання даного методу для дисциплін з вибіркового блоку освітніх програм було описано у даній роботі [7]. Метод (відстань) Левенштейна - це кількість операцій, необхідних для перетворення одного рядка в інший. Формула наведена нижче:

$$lev(a, b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ lev(\text{tail}(a), \text{tail}(b)) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} lev(\text{tail}(a), b) \\ lev(a, \text{tail}(b)) \\ lev(\text{tail}(a), \text{tail}(b)) \end{cases} & \text{otherwise} \end{cases}, \quad (3.1)$$

де *tail* певного рядка  $x$  є рядком для усіх символів, крім першого  $x$ , і  $x[n]$  є  $n$ -им символом рядка  $x$ , починаючи відлік від нуля.

Для наглядності, застосуємо алгоритм Левенштейна для спеціально відібраної множини слів  $C$ . Це дозволить зрозуміти, чи справді *PHP* методи *similartext* та *substr\_count* мають значну перевагу при здійсненні аналізу даних ніж використання відстані Левенштейна.

$C = \{\text{бізнес-аналіз, моніторинг, оптимізація, даними, бізнес-аналітик, організація, дані}\}$

Здійснюємо порівняння для наступної пари слів із множини  $C$ :

$c_1 = \{\text{бізнес-аналіз}\}$  та  $c_2 = \{\text{моніторинг}\}$ . Ці два слова розміщуються по обох осях матриці, котру можна візуалізувати за алгоритмом Вагнера - Фішера [73] і обчислити за формулою (3.1):

		м	о	н	і	т	о	р	и	н	г
	0	1	2	3	4	5	6	7	8	9	10
б	1	1	2	3	4	5	6	7	8	9	10
і	2	2	2	3	3	4	5	6	7	8	9
з	3	3	3	3	4	4	5	6	7	8	9
н	4	4	4	3	4	5	5	6	7	7	8
е	5	5	5	4	4	5	6	6	7	8	8
с	6	6	6	5	5	5	6	7	7	8	9
-	7	7	7	6	6	6	6	7	8	8	9
а	8	8	8	7	7	7	7	7	8	9	9
н	9	9	9	8	8	8	8	8	8	8	9
а	10	10	10	9	9	9	9	9	9	9	9
л	11	11	11	10	10	10	10	10	10	10	10
і	12	12	12	11	10	11	11	11	11	11	11
з	13	13	13	12	11	11	12	12	12	12	12

Рис. 3.10. Матриця Вагнера-Фішера для пари слів {бізнес-аналіз, моніторинг}

Остання позиція у матриці є шуканою відстанню Левенштейна і дорівнює числу 12. Схожим чином були обчислені і представлені відстані Левенштейна на рисунках 3.11 - 3.13

	б і з н е с - а н а л і т и к															
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
б	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
і	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
з	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12
н	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11
е	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10
с	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9
-	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8
а	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7
н	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6
а	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4	5
л	11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	4
і	12	11	10	9	8	7	6	5	4	3	2	1	0	1	2	3
з	13	12	11	10	9	8	7	6	5	4	3	2	1	1	2	3

Рис. 3.11. Матриця Вагнера-Фішера для пари слів {бізнес-аналіз, бізнес-аналітик}

	о п т и м і з а ц і я											
0	0	1	2	3	4	5	6	7	8	9	10	11
о	1	0	1	2	3	4	5	6	7	8	9	10
р	2	1	1	2	3	4	5	6	7	8	9	10
г	3	2	2	2	3	4	5	6	7	8	9	10
а	4	3	3	3	3	4	5	6	6	7	8	9
н	5	4	4	4	4	4	5	6	7	7	8	9
і	6	5	5	5	5	5	4	5	6	7	7	8
з	7	6	6	6	6	6	5	4	5	6	7	8
а	8	7	7	7	7	7	6	5	4	5	6	7
ц	9	8	8	8	8	8	7	6	5	4	5	6
і	10	9	9	9	9	9	8	7	6	5	4	5
я	11	10	10	10	10	10	9	8	7	6	5	4

Рис. 3.12. Матриця Вагнера-Фішера для пари слів {організація, оптимізація}

		д	а	н	и	м	и
	0	1	2	3	4	5	6
д	1	0	1	2	3	4	5
а	2	1	0	1	2	3	4
н	3	2	1	0	1	2	3
і	4	3	2	1	1	2	3

Рис. 3.13. Матриця Вагнера-Фішера для пари слів {дані, даними}

Очевидно, що для таких пар слів як {управління, управління} відстань Левенштейна буде дорівнюватиме нулю, так як слова є повністю однаковими. Для усіх інших розглянутих пар слів відстань Левенштейна визначається як  $x \in 1 \leq x \leq 3$ . Ми можемо вважати, що при таких значеннях порівнювальні слова можна вважати семантично близькими. Звісно, можуть траплятися і винятки чи похибки з семантично схожими словами, чи навіть з синонімами. Необхідність у таких відхиленнях значення  $x$  зумовлена потенційними неточностями у даних, які отримані в результаті парсингу, чи у описах дисциплін та їх ключових словах.

При виборі алгоритмів реалізації пошуку семантичної близькості у рекомендаційній системі були враховані як плюси так і мінуси використання відстані Левенштейна. Для їх вирішення необхідно підключати додаткові алгоритми, метрики та ресурси, що значно би збільшило складність практичної реалізації. Таким чином, для досягнення цілей дисертаційного дослідження, вибір був зроблений на користь методів, що уже реалізовані на мові програмування *PHP*.

Для перевірки з методами, що використані у рекомендаційній системі використаємо уже описану множину  $S$ .

Семантична близькість між бізнес-аналіз і моніторинг дорівнює 7  
Семантична близькість між бізнес-аналіз і оптимізація дорівнює 9  
Семантична близькість між бізнес-аналіз і даними дорівнює 8  
Семантична близькість між бізнес-аналіз і бізнес-аналітик дорівнює 24  
Семантична близькість між бізнес-аналіз і організація дорівнює 9  
Семантична близькість між бізнес-аналіз і дані дорівнює 7  
Семантична близькість між моніторинг і оптимізація дорівнює 6  
Семантична близькість між моніторинг і даними дорівнює 4  
Семантична близькість між моніторинг і бізнес-аналітик дорівнює 11  
Семантична близькість між моніторинг і організація дорівнює 8  
Семантична близькість між моніторинг і дані дорівнює 6  
Семантична близькість між оптимізація і даними дорівнює 7  
Семантична близькість між оптимізація і бізнес-аналітик дорівнює 7  
Семантична близькість між оптимізація і організація дорівнює 17  
Семантична близькість між оптимізація і дані дорівнює 5  
Семантична близькість між даними і бізнес-аналітик дорівнює 8  
Семантична близькість між даними і організація дорівнює 7  
Семантична близькість між даними і дані дорівнює 6  
Семантична близькість між бізнес-аналітик і організація дорівнює 10  
Семантична близькість між бізнес-аналітик і дані дорівнює 7  
Семантична близькість між організація і дані дорівнює 7

Рис. 3.14. Результати порівняння слів з множини  $S$  методами  $RHP$ .

Як помітно на рисунку 3.14, деякі пари слів повертають доволі високий числовий результат, що є підставою вважати їх семантично близькими. Звісно, трапляються і випадки, коли відстань Левенштейна може визначити відстань точніше, наприклад, пара слів {дані, даними} та дуже низьке числове значення схожості. Проте такі випадки можливі лише при одноразовому порівнянні слів, в той час як  $RHP$  методи оперують з великими масивами даних і додатково враховують кількість входжень рядків. При порівнянні з інформацією про IT-вакансії випадки неточності та й похибка в цілому зводиться до дуже низьких значень, що наближені до нуля. Таким чином, нею можна знехтувати і вважати  $RHP$  методи як такими, що краще задовольняють цілі дисертаційного дослідження.

Отже, експериментальні обчислення дозволили зрозуміти, що результати за різними методами є достатньо схожими, хоча і можуть відрізнятися у певних випадках. Не зважаючи на вищу потенційну точність методу з використанням відстані Левенштейна, у дисертаційному дослідженні були використані методи *similartext* та *substr\_count* завдяки значно вищій швидкості їх роботи. Потенційна похибка при використанні автоматичних методів є незначною та може бути проігнорована задля досягнення результату.

### **3.3 Застосування методу головних компонент.**

Попри отримання освітньо-професійних компонент та дисциплін з вибіркового блоку, необхідно також розглянути загальноуніверситетські дисципліни на предмет їх включення у індивідуальну освітню траєкторію студента. У попередньому розділі були досліджені методи аналізу даних та наведено коротку характеристику кожного методу. Додатково було обрано метод головних компонент для досягнення цілі у виборі дисциплін серед великих інформаційних масивів.

Зважаючи на використання методу головних компонент (РСА) необхідно визначити набір характеристик для кожної дисципліни з подальшим факторним аналізом і зменшенням їх розмірності. Ідея використання методу РСА для включення дисциплін у індивідуальну траєкторію полягає у знаходження кількох головних компонент, що відповідатимуть певним характеристикам дисципліни, та їх подальшим зменшенням хоча би до двох. Це дозволить зберегти приблизно дев'яносто відсотків з інформаційного масиву загальноуніверситетських дисциплін.

Парсинг загальноуніверситетських дисциплін здійснюється з веб-сайту Національного університету “Львівська Політехніка” [39]. Так як у



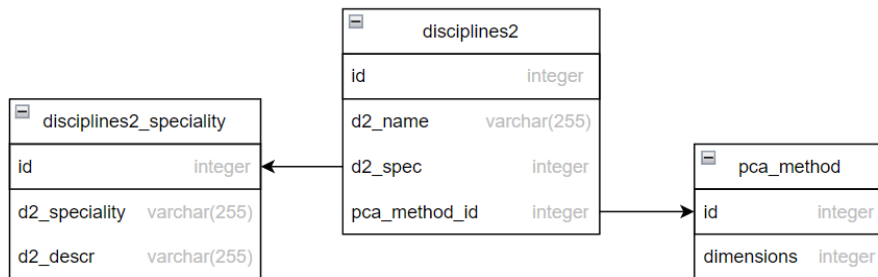
дисертаційному дослідженні ми розглядаємо лише ІТ-спеціальностей, то і парсинг даних буде логічно обмежити відповідними інститутами.

Таким чином, використовуючи попередньо описану технологію парсингу даних, для роботи методу головних компонент використаємо дисципліни з інституту комп'ютерних наук та інформаційних технологій [39].

Усі дисципліни отримані таким чином додатково записуються, щоб консолідувати інформацію та забезпечити доступність для модулів системи (модуль визначення загальноуніверситетських дисциплін, котрий описаний у наступному розділі). Дані записуються у базі даних, що містить наступну інформацію:

- Назви загальноуніверситетських дисциплін;
- Опис дисциплін та спеціальність;
- Числові значення характеристик дисциплін для методу головних компонент.

Відповідні інформаційні таблиці з даними по загальноуніверситетським дисциплінам наведені на рисунку 3.14.



*Рис. 3.15. Інформаційні таблиці, що містять дані, отримані в результаті парсингу.*

У кожної отриманої дисципліни є силабус, котрий детально її описує. На основі цієї інформації оберемо п'ять характеристик, завдяки яким можна здійснити вибір дисципліни для включення у ІОТ. Цими характеристиками є:

- Мета вивчення дисципліни;
- Завдання;
- Необхідність у обов'язкових попередніх та супутніх навчальних дисциплінах;
- Належність до спеціальності;
- К-сть рекомендованої літератури.

Відповідно, ціллю методу головних компонент буде зменшення характеристик хоча би до двох, що дозволить порівняти дисципліни і визначити найоптимальнішу для студента. При цьому важливо зберегти оригінальні дані, тобто врахувати усі характеристики.

Дисципліни отримані у вигляді наступної множини:

$$D = \{d_1, d_2, \dots, d_n\},$$

де  $n$  - кількість загальноуніверситетських дисциплін. Проведемо експериментальні обчислення з наступними заданими значеннями:

$D = \{\text{Математичні моделі представлення знань, Бізнес-планування та управління проектами, Основи підприємництва та бізнес-планування в ІТ, Інформаційні технології організації бізнесу, Об'єктно-орієнтоване програмування, Технології штучного інтелекту в інженерії даних, Україна-ЄС: цифрова трансформація, Етика бізнесу, Комп'ютерна схемотехніка та архітектура комп'ютерних систем, Технології веб розробки та дизайну}\}.$

В реальності система здатна опрацювати більше сотні, або навіть тисячі загальноуніверситетських дисциплін. Проте, задля наглядної демонстрації результатів ми використаємо невелику вибірку у десять одиниць. Результати від зменшеної вибірки не відрізнятимуться. Дані по характеристикам дисциплін були присвоєні на основі деяких спостережень згідно характеристикам і записані у вигляді наступним компонент:

$$X_1 = \{10, 8, 10, 10, 9, 7, 10, 8, 9, 8\}$$

$$X_2 = \{3, 1, 3, 3, 2, 0, 3, 2, 1, 2\}$$

$$X_3 = \{2, 2, 3, 2, 4, 6, 2, 5, 4, 5\}$$

$$X_4 = \{-7, -5, -7, -7, -6, -4, -7, -5, -6, -5\}$$

$$X_5 = \{-2, 1, -2, -2, -1, 2, -2, 0, -1, 0\}$$

Метою першого кроку є стандартизація діапазону безперервних вихідних змінних, щоб кожна з них вносила рівний внесок у аналіз. Ідея стандартизації полягає у перетворенні даних (масштабуванні) таким чином, щоб їх можна було порівняти. Вона не є обов'язковою при незначній дисперсії змінних, проте забезпечує більшу точність при отриманні результатів.

Математично стандартизація визначається як знаходження різниці середнього значення та поділу його на стандартне відхилення для кожного значення кожної змінної:

$$Z = \frac{\text{value} - \text{mean}}{\text{standard deviation}}$$

де *value* - значення до стандартизації (отримане в результаті спостережень), *mean* - середнє арифметичне змінної, *standard deviation* – стандартне відхилення.

Результати стандартизації наведені у таблиці 3.1, що наведена нижче.

Таблиця 3.1

## Результати стандартизації

Дисципліни	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
$d_1$	0.9995	0.9487	-0.9939	-0.9995	-0.9167
$d_2$	-0.8178	-0.9487	-0.9939	0.8178	1.1988
$d_3$	0.9995	0.9487	-0.3313	-0.9995	-0.9167
$d_4$	0.9995	0.9487	-0.9939	-0.9995	-0.9167
$d_5$	0.09087	0	0.3313	-0.09087	-0.2115
$d_6$	-1.7265	-1.8974	1.6565	1.7265	1.9039
$d_7$	0.9995	0.9487	-0.9939	-0.9995	-0.9167
$d_8$	-0.8178	0	0.9939	0.8178	0.4936
$d_9$	0.09087	-0.9487	0.3313	-0.09087	-0.2115
$d_{10}$	-0.8178	0	0.9939	0.8178	0.4936

Коли всі змінні знаходяться у одному масштабі, ми формуємо коваріаційну матрицю за наступною формулою:

$$Cov_{jk} = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k), \quad (3.2)$$

Де  $\bar{x}$  є середнім значенням вектора  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ . Обчислимо коваріацію  $(X_1, X_2)$  за формулою (3.2):

$$Cov_{12} = \frac{7.7584}{10-1} = 0.862$$

Коваріацію  $(X_3, X_1)$ :

$$Cov_{31} = \frac{-6.9237}{10 - 1} = -0.7693$$

і тд.

Результати наведені у матриці нижче:

$$\begin{pmatrix} 1 & 0.862 & -0.7693 & -1 & -0.9754 \\ 0.862 & 1 & -0.6286 & -0.862 & -0.892 \\ -0.7693 & -0.6286 & 1 & 0.7693 & 0.6489 \\ -1 & -0.862 & 0.7693 & 1 & 0.9754 \\ -0.9754 & -0.892 & 0.6489 & 0.9754 & 1 \end{pmatrix}$$

Володіючи матрицею, здійснюємо обчислення власних векторів і власних значень для знаходження головних компонент -  $PC$ . Власні значення наведені у таблиці 3.2, а власні вектори наведені у таблиці 3.3

**Таблиця 3.2**

**Власні значення головних компонент**

Параметри	$PC_1$	$PC_1$	$PC_1$	$PC_1$	$PC_1$
Власне значення	4.3754	0.4416	0.1729	0.01013	0
% дисперсії	87.5074	8.8323	3.4578	0.2025	0
Сукупно (%)	87.5074	96.3397	99.7975	100	100

**Таблиця 3.3**

**Власні вектори головних компонент**

Вектор( $PC_1$ )	Вектор( $PC_2$ )	Вектор( $PC_3$ )	Вектор( $PC_4$ )	Вектор( $PC_5$ )
0.473	0.04391	-0.3284	0.4081	0.7071
0.4357	0.3441	0.823	0.1203	0
-0.3857	0.8805	-0.1927	0.1972	0
-0.473	-0.04391	0.3284	-0.4081	0.7071
-0.4626	-0.3202	0.2643	0.7833	0

Отже, на даному етапі використання методу головних компонент, нам вдалося зменшити розмірність вибірки таким чином, що три з п'яти характеристик ( $X_3 - X_5$ ) можна відкинути, адже їх цінність для досягнення цілі зовсім незначна. Перші дві головні компоненти містять приблизно 96% усіх даних, що підтверджується даними з власних значень, а також діаграмою (рис 3.16) та проектуванням на осі координат (рис. 3.17).

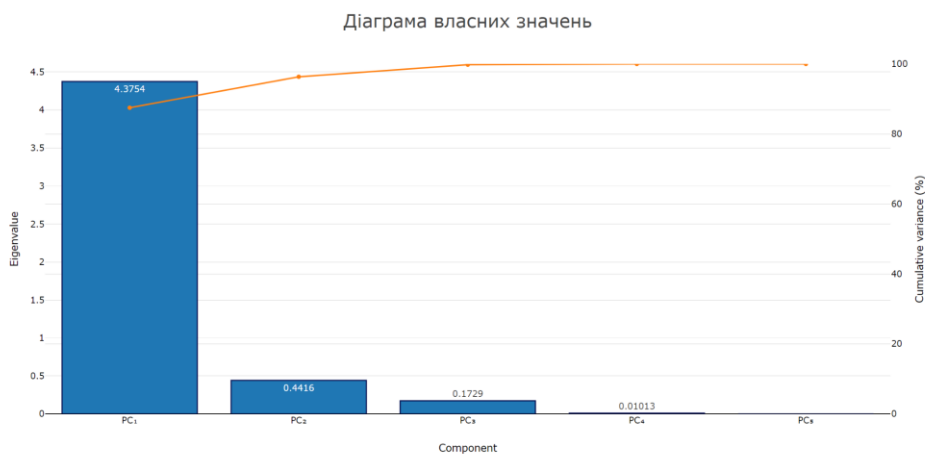


Рис. 3.16. Діаграма власних значень по кожній компоненті.

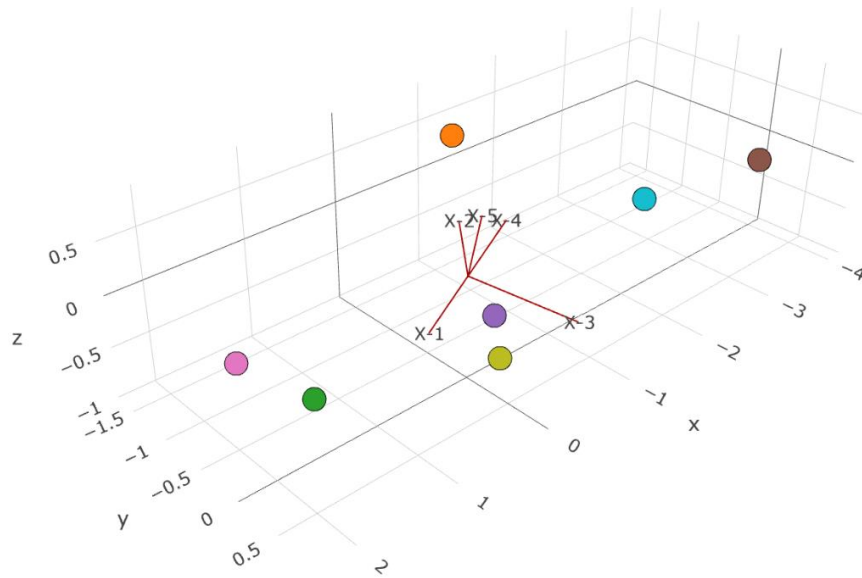


Рис. 3.17. Проектування даних на осі координат у трьох вимірах.

Після знаходження власних векторів, формально, метод головних компонент можна вважати завершеним. Проте отримані дані не є готовими для оперування рекомендаційною системою, на їх основі неможливо сформулювати будь-які рекомендації. Необхідно переорієнтувати дані з вихідних осей на ті, що представлені головними компонентами. Цей процес виконується через знаходження добутку транспонованого вектора та транспонованого набору вихідних даних (матриці) за наступною формулою:

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T,$$

де *FeatureVector* - новий вектор, який включає дані з перших двох власних векторів, а *StandardizedOriginalDataSet* – стандартизований набір даних, що отримані внаслідок спостережень.

$$FeatureVector^T = \begin{pmatrix} 0.473 & 0.4357 & -0.3857 & -0.473 & -0.4626 \\ 0.04391 & 0.3441 & 0.8805 & -0.04391 & -0.3202 \\ -0.3284 & 0.823 & -0.1927 & 0.3284 & 0.2643 \\ 0.4081 & 0.1203 & 0.1972 & -0.4081 & 0.7833 \\ 0.7071 & 0 & 0 & 0.7071 & 0 \end{pmatrix}$$

$$StandardizedOriginalDataSet^T = \begin{pmatrix} 10 & 8 & 10 & 10 & 9 & 7 & 10 & 8 & 9 & 8 \\ 3 & 1 & 3 & 3 & 2 & 0 & 3 & 2 & 1 & 2 \\ 2 & 2 & 3 & 2 & 4 & 6 & 2 & 5 & 4 & 5 \\ -7 & -5 & -7 & -7 & -6 & -4 & -7 & -5 & -6 & -5 \\ -2 & 1 & -2 & -2 & -1 & 2 & -2 & 0 & -1 & 0 \end{pmatrix}$$

$$FinalDataSet = \begin{pmatrix} 9.5019 & 5.3507 & 9.1162 & 9.5019 & 6.8862 & 1.9636 & 9.5019 & 5.0919 & 6.4505 & 5.0919 \\ 4.1802 & 2.3557 & 5.0607 & 4.1802 & 5.1890 & 5.1256 & 4.1802 & 5.6615 & 4.8449 & 5.6615 \\ -4.0278 & -3.5673 & -4.2205 & -4.0278 & -4.3151 & -4.24 & -4.0278 & -3.5867 & -5.1381 & -3.5867 \\ 6.1264 & 6.6033 & 6.3236 & 6.1264 & 6.3676 & 7.2389 & 6.1264 & 6.5319 & 6.2473 & 6.5319 \\ 2.1213 & 2.1213 & 2.1213 & 2.1213 & 2.1213 & 2.1213 & 2.1213 & 2.1213 & 2.1213 & 2.1213 \end{pmatrix}$$

Отриманий набір даних є більш зручним та зрозумілішим як для користувача, так і для системи. До того ж, як тепер помітно і неозброєним оком, у компонентах  $X_3 - X_5$  практично відсутня кореляція даних, і усі значення зведені до одного (найбільше це помітно у компоненті  $X_5$ ). Це демонструє важливість лише перших двох компонент і незначну кількість даних, якими володіють інші. Тобто на наступному етапі ми можемо ігнорувати значення по головним компонентам  $X_3 - X_5$  та обчислювати значення лише перших двох головних компонент ( $X_1$  та  $X_2$ ). Відповідно, ми сумуємо значення компонент і обираємо максимальне для досягнення результату.

$$9.5019 + 4.1802 = 13,6821$$

$$5.3507 + 2.3557 = 7,7064$$

$$9.1162 + 5.0607 = 14,1769$$

$$9.5019 + 4.1802 = 13,6821$$

$$6.8862 + 5.1890 = 12,0752$$

$$1.9636 + 5.1256 = 7,0892$$

$$9.5019 + 4.1802 = 13,6821$$

$$5.0919 + 5.6615 = 10,7534$$

$$6.4505 + 4.8449 = 11,2954$$

$$5.0919 + 5.6615 = 10,7534$$

Відповідно, для вище описаного набору даних рекомендованою дисципліною буде  $d_3 = \{ \text{Основи підприємництва та бізнес-планування в ІТ} \}$ . Також варто звернути увагу на однакове значення дисциплін  $d_1 = \{ \text{Математичні моделі представлення знань} \}$ ,  $d_4 = \{ \text{Інформаційні технології організації бізнесу} \}$  та  $d_7 = \{ \text{Україна-ЄС: цифрова трансформація} \}$ . При інакших вхідних даних спостережень результат може і буде відрізнятися, так як при великій вибірці у сотню і більше дисциплін, шанс на однакові числові



значення є мізерним. Підтвердження експерименту полягає у відповідності дисципліни ІТ-спеціальностям та її високий коефіцієнт при виборі студентами.

Програмна реалізація описаного методу по формуванні рекомендацій на основі загальноуніверситетських дисциплін представлена у останньому розділі за допомогою веб-технологій.

### **3.4 Експериментальні обчислення за методом аналізу ієрархій для ранжування дисциплін.**

Метод аналізу ієрархій був теоретично описаний у попередньому розділі. Проте, для підтвердження його актуальності та практичності для використання у дисертаційному дослідженні, необхідно додатково провести експериментальні обчислення. Ціллю експерименту буде зімітувати вибір дисциплін експертами для їх рекомендації студенту ІТ-спеціальності.

Нехай, для ранжування були отримані наступні альтернативи (дисципліни):

- Комп'ютерна лінгвістика;
- Проблемно-орієнтоване програмування;
- Планування ІТ-проектів;

Ці дисципліни належать до вибіркового компонент освітньо-професійної програми “Системний аналіз” для першого рівня (бакалавр) вищої освіти за 2022 навчальний рік [71].

Додатково необхідно визначити критерії, по яким буде визначитися пріоритет альтернатив. Критерії не повинні залежати від навчальних

дисциплін і є фіксованими незалежно від вхідних даних. У ході дослідження було вирішено обрати наступні критерії:

- Особисті хобі та інтереси;
- Потенційна заробітна плата;
- Професійні перспективи;
- Можливість працювати у суміжних галузях.

Такі критерії вибрані на основі аналізу ринку праці та зацікавленості абітурієнтів при виборі дисциплін.

Відповідно до отриманих даних будуюмо дерево ієрархій (рис. 3.12). На верхньому рівні ієрархії знаходиться ціль, яку необхідно досягнути. В нашому випадку це «обрати дисципліну». Нижні рівні (другий і третій) представляють критерії та альтернативи (дисципліни), відповідно.

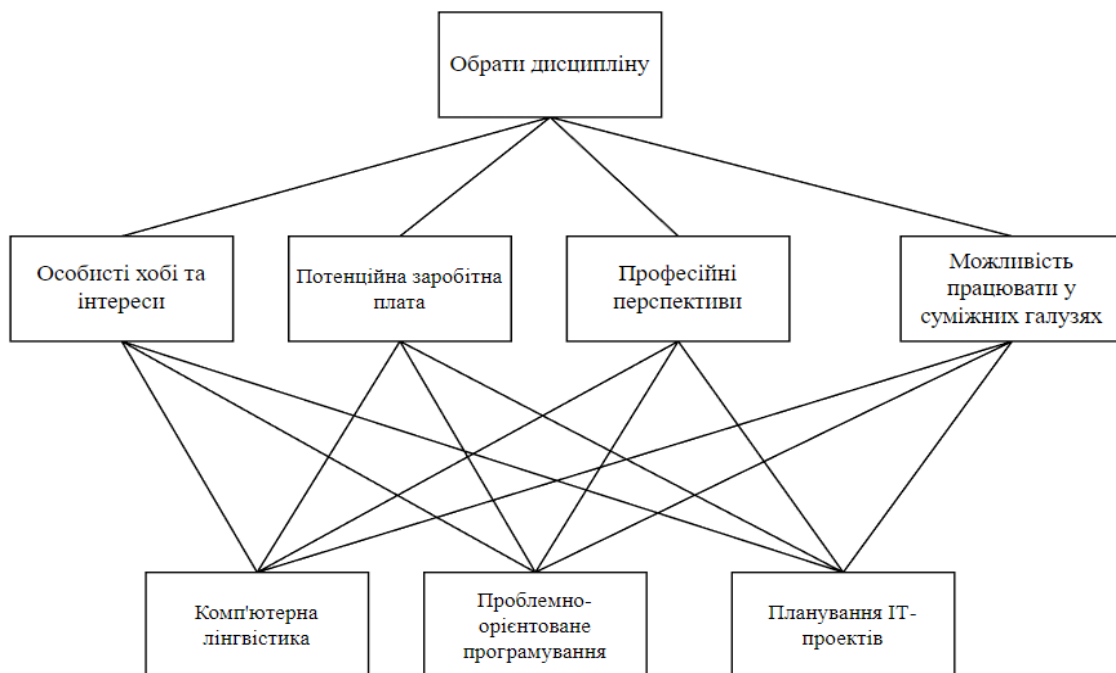


Рис. 3.18. Дерево ієрархій, побудоване за МАІ

Наступним кроком є визначення пріоритетів, що представляють відносну важливість або перевагу елементів у побудованому дереві ієрархій. Для навчальних дисциплін буде визначений пріоритет щодо кожного критерію прийняття рішень, у той час як для критеріїв пріоритет буде визначений на основі їх важливості для досягнення мети. Потім пріоритети будуть об'єднані в усій ієрархії, щоб визначити загальний пріоритет для кожної дисципліни. Дисципліна із найвищим пріоритетом буде найкращою альтернативою, а співвідношення пріоритетів дисциплін вказуватиме на їхню відносну потужність щодо поставленої цілі. Пріоритети будуть виведені шляхом проведення ряду обчислень: попарні порівняння за участю всіх елементів ієрархії. Елементи на кожному рівні порівнюватимуться по два з огляду на їхній внесок. Результати цих порівнянь введені в матрицю, яка обробляється математично для отримання пріоритетів усіх елементів на рівні.

Для побудови матриці спершу необхідно визначити вагу пари альтернатив за кожним критерієм. Для цього використаємо шкалу відношень, що була наведена у таблиці 2.1. В результаті у таблиці 3.4 була визначена вага альтернатив за першим критерієм – «Особисті хобі та інтереси».

**Таблиця 3.4**

**Порівняння альтернатив за критерієм «Особисті хобі та інтереси»**

<b>Альтернатива</b>	<b>Вага альтернативи</b>	<b>Альтернатива</b>	<b>Вага альтернативи</b>
<i>Комп'ютерна лінгвістика</i>	5	<i>Проблемно-орієнтоване програмування</i>	1

<i>Комп'ютерна лінгвістика</i>	1	<i>Планування IT-проектів</i>	3
<i>Проблемно-орієнтоване програмування</i>	1	<i>Планування IT-проектів</i>	8

Використовуючи вище наведену таблицю, будемо відповідну матрицю попарних порівнянь:

$$(1 \ 5 \ 1/3 \ 1/5 \ 1 \ 1/8 \ 3 \ 8 \ 1)$$

Використовуючи формулу (2.4) знаходимо власний вектор:

$$W = (0,271 \ 0,066 \ 0,661)$$

Далі здійснюємо оцінку узгодженості суджень експертів за даними, з яких була побудована матриця попарних порівнянь. Нижче наведено обчислення, що були проведені за формулами (2.1), (2.2) і (2.3):

$$\lambda_{max} = (1 \ 1 \ 1) * \left(1 \ 5 \ \frac{1}{3} \ \frac{1}{5} \ 1 \ \frac{1}{8} \ 3 \ 8 \ 1\right) * (0,271 \ 0,066 \ 0,661) = 3,02$$

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{3,02 - 3}{3 - 1} = 0,445$$

$$CR = \frac{CI}{RI} = \frac{0,445}{0,58} = 0,017$$

Здійснюємо переведення отриманої величини у відсотки:

$$CR = 0,017 * 100\% = 1,7\%$$

Нерівність (2.5) виконується, так як отримане значення  $CR$  є меншим ніж 10%. Відповідно, можна вважати дану матрицю узгодженою.

Схожим чином проводимо розрахунки за іншими критеріями. Для критерію «Потенційна заробітна плата» порівняння альтернатив наведено у таблиці 3.5.

Таблиця 3.5

## Порівняння альтернатив за критерієм «потенційна заробітна плата»

Альтернатива	Вага альтернативи	Альтернатива	Вага альтернативи
<i>Комп'ютерна лінгвістика</i>	1	<i>Проблемно-орієнтоване програмування</i>	4
<i>Комп'ютерна лінгвістика</i>	3	<i>Планування ІТ-проектів</i>	1
<i>Проблемно-орієнтоване програмування</i>	7	<i>Планування ІТ-проектів</i>	1

Матриця попарних порівнянь:

$$(1 \ 1/3 \ 4 \ 3 \ 1 \ 7 \ 1/3 \ 1/7 \ 1)$$

Власний вектор:

$$W = (0,260 \ 0,653 \ 0,085)$$

$$\lambda_{max} = (1 \ 1 \ 1) * (1 \ 1/3 \ 4 \ 3 \ 1 \ 7 \ 1/3 \ 1/7 \ 1) * (0,260 \ 0,653 \ 0,085) = 3,10$$

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{3,1 - 3}{3 - 1} = 0,05$$

$$CR = \frac{CI}{RI} = \frac{0,05}{0,58} = 0,086 * 100\% = 8,6\%$$

Нерівність (2.5) виконується. Матриця за критерієм «потенційна заробітна плата» є узгодженою. Для критерію «професійні перспективи» побудована таблиця 3.6 альтернатив.

**Таблиця 3.6**

**Порівняння альтернатив за критерієм «Професійні перспективи»**

<b>Альтернатива</b>	<b>Вага альтернативи</b>	<b>Альтернатива</b>	<b>Вага альтернативи</b>
<i>Комп'ютерна лінгвістика</i>	1	<i>Проблемно-орієнтоване програмування</i>	4
<i>Комп'ютерна лінгвістика</i>	4	<i>Планування IT-проектів</i>	1
<i>Проблемно-орієнтоване програмування</i>	8	<i>Планування IT-проектів</i>	1

Матриця попарних порівнянь:

$$(1 \ 4 \ 1/4 \ 1/4 \ 1 \ 1/8 \ 4 \ 8 \ 1)$$

Власний вектор:

$$W = (0,222 \ 0,069 \ 0,707)$$

$$\lambda_{max} = (1 \ 1 \ 1) * (1 \ 4 \ 1/6 \ 1/4 \ 1 \ 1/7 \ 6 \ 7 \ 1) * (0,222 \ 0,069 \ 0,707) = 3,03$$

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{3,03 - 3}{3 - 1} = 0,015$$

$$CR = \frac{CI}{RI} = \frac{0,015}{0,58} = 0,025 = 2,5\%$$

За критерієм «Можливість працювати в суміжних галузях» порівняння альтернатив можна побачити у таблиці 3.7.

**Таблиця 3.7**

**Порівняння альтернатив за критерієм «Можливість працювати в суміжних галузях»**

<b>Альтернатива</b>	<b>Вага альтернативи</b>	<b>Альтернатива</b>	<b>Вага альтернативи</b>
<i>Комп'ютерна лінгвістика</i>	5	<i>Проблемно-орієнтоване програмування</i>	1
<i>Комп'ютерна лінгвістика</i>	9	<i>Планування ІТ-проектів</i>	1
<i>Проблемно-орієнтоване програмування</i>	3	<i>Планування ІТ-проектів</i>	1

Матриця попарних порівнянь:

$$(1 \ 5 \ 9 \ 1/5 \ 1 \ 3 \ 1/9 \ 1/3 \ 1)$$

Власний вектор:

$$W = (0,751 \ 0,178 \ 0,070)$$

$$\lambda_{max} = (1 \ 1 \ 1) * (1 \ 3 \ 4 \ 1/3 \ 1 \ 6 \ 1/4 \ 1/6 \ 1) * (0,751 \ 0,178 \ 0,070) = 3,02$$

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{3,02 - 3}{3 - 1} = 0,01$$

$$CR = \frac{CI}{RI} = \frac{0,01}{0,58} = 0,017 = 1,7\%$$

Після оцінки альтернатив по всіх критеріях, необхідно також здійснити оцінку критеріїв з огляду на їх важливість для досягнення мети. Порівняння критеріїв за метою зображене в таблиці 3.8.

**Таблиця 3.8**

**Критерії порівняння щодо досягнення цілі**

<b>Критерій</b>	<b>Вага критерію</b>	<b>Критерій</b>	<b>Вага критерію</b>
<i>Особисті хобі та інтереси</i>	5	<i>Потенційна заробітна плата</i>	1
<i>Особисті хобі та інтереси</i>	4	<i>Професійні перспективи</i>	1
<i>Особисті хобі та інтереси</i>	8	<i>Можливість працювати у суміжних галузях</i>	1
<i>Потенційна заробітна плата</i>	1	<i>Професійні перспективи</i>	3
<i>Потенційна заробітна плата</i>	4	<i>Можливість працювати у суміжних галузях</i>	1
<i>Можливість працювати у суміжних галузях</i>	1	<i>Професійні перспективи</i>	6

Відповідна матриця попарних порівнянь для критеріїв матиме наступний вигляд:



$$(1 \ 5 \ 4 \ 8 \ 1/5 \ 1 \ 1/3 \ 4 \ 1/4 \ 3 \ 1 \ 6 \ 1/8 \ 1/4 \ 1/6 \ 1)$$

Власний вектор матриці:

$$W = (0,592 \ 0,119 \ 0,242 \ 0,044)$$

$$\lambda_{max} = (1 \ 1 \ 1 \ 1) * \left( 1 \ 5 \ 4 \ 8 \ \frac{1}{5} \ 1 \ \frac{1}{3} \ 4 \ \frac{1}{4} \ 3 \ 1 \ 6 \ 1/8 \ 1/4 \ 1/6 \ 1 \right) \\ * (0,592 \ 0,119 \ 0,242 \ 0,044) = 4,19$$

$$CI = \frac{\lambda_{max} - n}{n - 1} = \frac{4,19 - 4}{4 - 1} = 0,063$$

$$CR = \frac{CI}{RI} = \frac{0,063}{0,9} = 0,07 = 7\%$$

Так як матриця узгоджена, можна використати знайдений власний вектор для синтезу пріоритету альтернатив до поставленої мети. Для цього необхідно перемножити пріоритети:

$$(0,271 \ 0,066 \ 0,661) * 0,592 = (0,160 \ 0,039 \ 0,391)$$

$$(0,260 \ 0,653 \ 0,085) * 0,119 = (0,030 \ 0,077 \ 0,010)$$

$$(0,222 \ 0,069 \ 0,707) * 0,242 = (0,053 \ 0,016 \ 0,171)$$

$$(0,751 \ 0,178 \ 0,070) * 0,044 = (0,033 \ 0,007 \ 0,003)$$

Для наглядної демонстрації результати представлені у таблиці 3.9. У цій же таблиці і проведені фінальні операції сумування елементів ієрархії. Згідно з таблицею, найвищий пріоритет до мети має третя альтернатива – «Планування ІТ-проектів» зі значення 0,575. Відповідно, саме цю дисципліну і буде рекомендовано користувачу.

**Таблиця 3.9**

**Загальні пріоритети для всіх альтернатив**

<b>Альтернативи</b>	<i>Особисті хобі та інтереси</i>	<i>Потенційна заробітна плата</i>	<i>Професійні перспективи</i>	<i>Можливість працювати у суміжних галузях</i>	<b>Мета</b>
<i>Проблемно-орієнтоване програмування»</i>	0,160	0,030	0,053	0,033	0,276
<i>Комп'ютерна лінгвістика</i>	0,039	0,077	0,016	0,007	0,139
<i>Планування IT-проектів</i>	0,391	0,010	0,171	0,003	0,575
<b>Загалом</b>	0,59	0,117	0,24	0,043	1

### **Висновки до розділу 3**

1. Описано метод аналізу даних з веб-сайтів IT-вакансій, що дозволяє отримати професійні характеристики користувача. В основі даного методу використовується метод парсингу, який дозволяє здійснювати пошук ключових слів у описах IT-вакансій на веб-сайтах пошуку роботи.

2. Розроблено алгоритм пошуку семантичної близькості на основі мови програмування PHP з використанням методів . Додатково проведені експериментальні обчислення з використанням відстані Левенштейна для порівняння точності з розробленим алгоритмом.

3. Через необхідність у включенні загальноуніверситетських дисциплін та відповідною роботою з величезними масивами даних, було застосовано метод головних компонент. Його застосування дозволило

зменшити розмірність даних та обрати одну конкретну дисципліну для включення у рекомендації.

4. Проведено експериментальні обчислення для ранжування дисциплін з використанням методу аналізу ієрархій. Усі тестові обчислення були здійсненні з випадковими вхідними даними. Для вхідних даних були обрані випадкові дисципліни з вибіркового компонентів освітньо-професійної програми «Системний аналіз».

## РОЗДІЛ IV. ПРОГРАМНІ ОСОБЛИВОСТІ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

Метою четвертого розділу було розроблення структури рекомендаційної системи на основі веб-застосунку. Веб-застосунок реалізований з можливостями користувацького інтерфейсу та представленою програмною реалізацією. Додатково представлена практична реалізація методів, моделей та алгоритмів, що були описані у попередніх розділах з використанням веб-технологій.

### 4.1 Функціональне призначення

Описані у попередніх розділах моделі, методи та алгоритми складають основу розробленого веб-застосунку разом з базою даних. Кожен етап формування рекомендацій є практично реалізований та взаємопов'язаний з іншими модулями системи.

Головним завданням створеного веб-застосунку є надання рекомендацій користувачу при формуванні його індивідуальної освітньої траєкторії. Рекомендації надаються у вигляді списку навчальних дисциплін для включення у ІОТ. Для веб-застосунку описаний процес можна представити у вигляді наступних етапів:

- 1) Старт системи;
- 2) Створення профілю користувача або, у разі наявності існуючого профілю, авторизація користувача;
- 3) Проходження користувачем тесту на профорієнтацію. Запис результатів тесту у БД;

4) Парсинг даних з веб-сайтів ІТ-вакансій та веб-сайту Національного університету «Львівська політехніка»;

5) Аналіз даних: визначення семантично близьких дисциплін та застосування методу головних компонент для зменшення розмірності загальноуніверситетських дисциплін;

6) Запис результатів аналізу даних у БД;

7) Застосування МАІ для зменшення об'єму рекомендованих дисциплін та збільшення точності рекомендацій;

8) Представлення результату роботи системи користувачу..

9) Кінець роботи системи.

Описані етапи роботи веб-застосунку також зображені на рисунку 4.1 за допомогою блок-схем та на рис. 4.2 у вигляді UML діаграми. Додатково було визначено акторів, що працюють з системою (користувач та адміністратор).

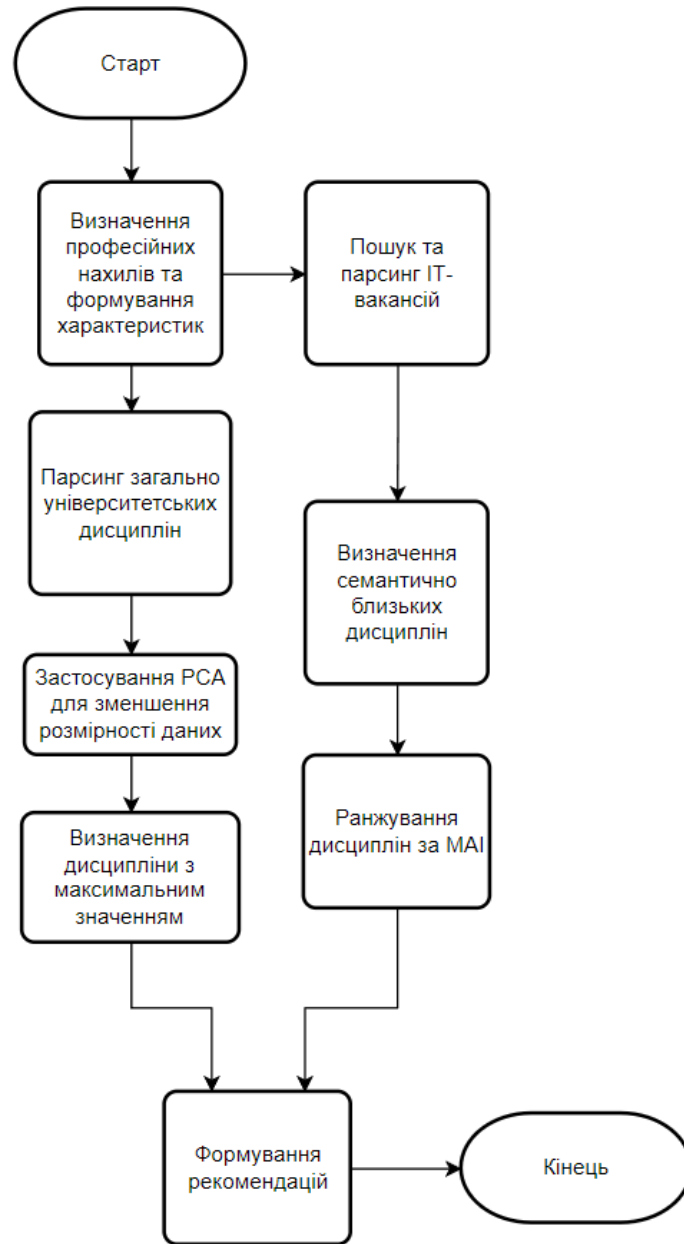


Рис. 4.1. Блок-схема етапів роботи веб-застосунку.

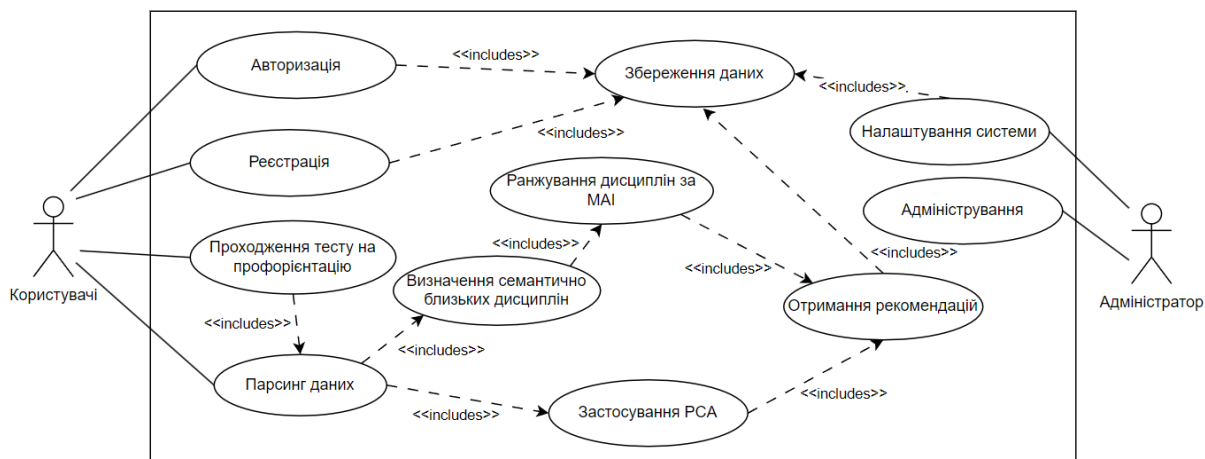


Рис. 4.2. Діаграма варіантів використання веб-застосунку.

Для веб-застосунку існує дві сутності, які з нею взаємодіють: *Користувач* та *Адміністратор*. При користуванні системою актор *Користувач* може здійснювати наступні дії:

- Зареєструватись/Авторизуватись у системі;
- Ввести інформацію про себе та бажані навчальні дисципліни;
- Пройти тест на профорієнтацію;
- Виключити небажані дисципліни зі списку рекомендацій;
- Отримати фіналізований список рекомендованих дисциплін з

можливістю їх подальшого включення у ІОТ.

Актор *Адміністратор* забезпечує стабільну роботу системи, проводить моніторинг її стану та виправляє помилки, якщо такі виникають.

## 4.2 Структурно-функціональна модель веб-застосунку

У основі системи використовується трирівнева архітектура. Для розробленого веб-застосунку вона зображена на рис. 4.3. Під цим типом архітектури розуміється наявність трьох типів компонентів:

- Клієнтський шар – найвищий рівень системи з інтерфейсом користувача. Основна функція інтерфейсу – представлення задач і результатів зрозумілих користувачу;

- Логічний шар – цей шар координує систему, опрацьовує команди, виконує логічні завдання та обчислення. Опрацьовує дані між двома навколишніми шарами.

- Шар даних – тут зберігається інформація і витягується з бази даних і файлової системи. Інформація відправляється в логічний шар для опрацювання і в кінці повертається користувачу.

**Підсистема визначення профорієнтації** дозволяє користувачу пройти тест, що визначить домінуючі типи особистості, які необхідні для подальшого визначення профорієнтації. Підсистема включає у собі базу даних тесту на профорієнтацію, модуль проходження тесту та базу даних результатів тестування.

*БД тесту* дозволяє зберігати дані тесту Холланда на профорієнтацію з варіантами відповідей та результатами.

*Модуль тесту на визначення профорієнтації* – даний модуль є першочерговим і основним у зборі даних про користувача, адже він надає можливість проходження тесту і запису результатів у *базу даних результатів тесту*.

**Підсистема аналізу даних** забезпечує роботу парсера у зборі даних про ІТ-вакансій та навчальні дисципліни.

*Модуль парсингу вакансій* – цей модуль забезпечує парсинг інформації на веб-сайтах ІТ-вакансій та освітніх веб-сайтах.

Паралельно з парсингом інформації про ІТ-вакансії необхідно отримати дані про дисципліни, зокрема, загальноуніверситетські, які також можуть бути



включені у ІОТ. За цей процес відповідає *модуль парсингу загальноуніверситетських дисциплін*

Отримані дані зберігаються у *БД вакансій* та *БД дисциплін*, відповідно.

***Підсистема вибору дисциплін*** виконує дві функції: зменшує вибірку дисциплін та визначає саме ті дисципліни, які найкраще підходять користувачу.

*Модуль визначення семантичної близькості* – модуль забезпечує пошук ключових слів онтології у текстових описах ІТ-вакансій. Знайдені співпадиння вважаються семантично близькими.

*Модуль методу головних компонент* – модуль використовує метод головних компонент для зменшення кількості загальноуніверситетських дисциплін. Додатково обирається та дисципліна, як найкраще підходить користувачу.

***Підсистема експертної оцінки*** дозволяє проводити додаткове ранжування альтернатив (дисциплін) для визначення найкращої на основі критеріїв та експертних оцінок.

*Модуль ранжування дисциплін за МАІ* безпосередньо здійснює порівняння альтернатив використовуючи МАІ на основі *БД експертних оцінок*.

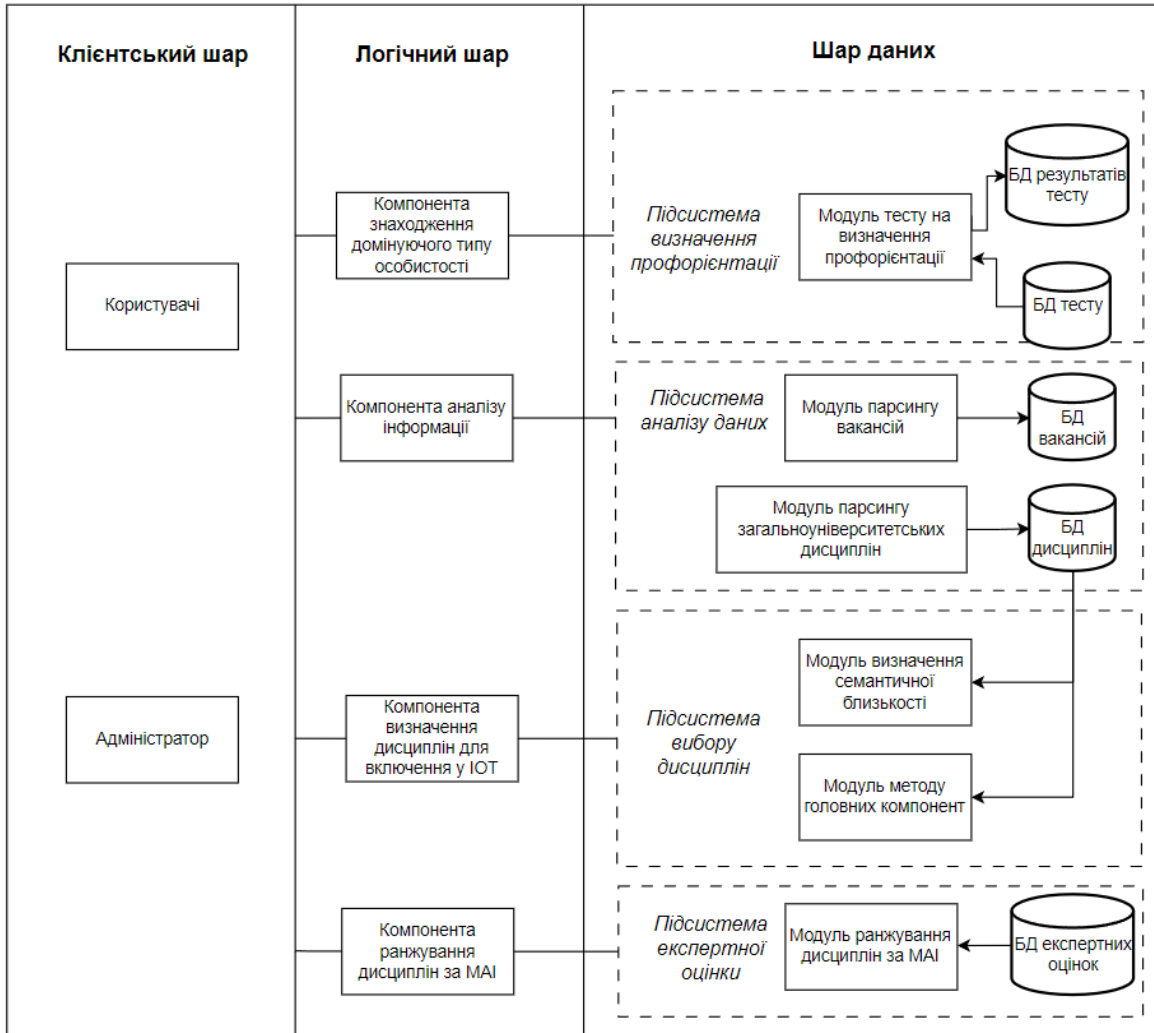


Рис. 4.3. Архітектура веб-застосунку.

На рисунку 4.4 представлений алгоритм роботи веб-застосунку, в процесі її експлуатації користувачем.

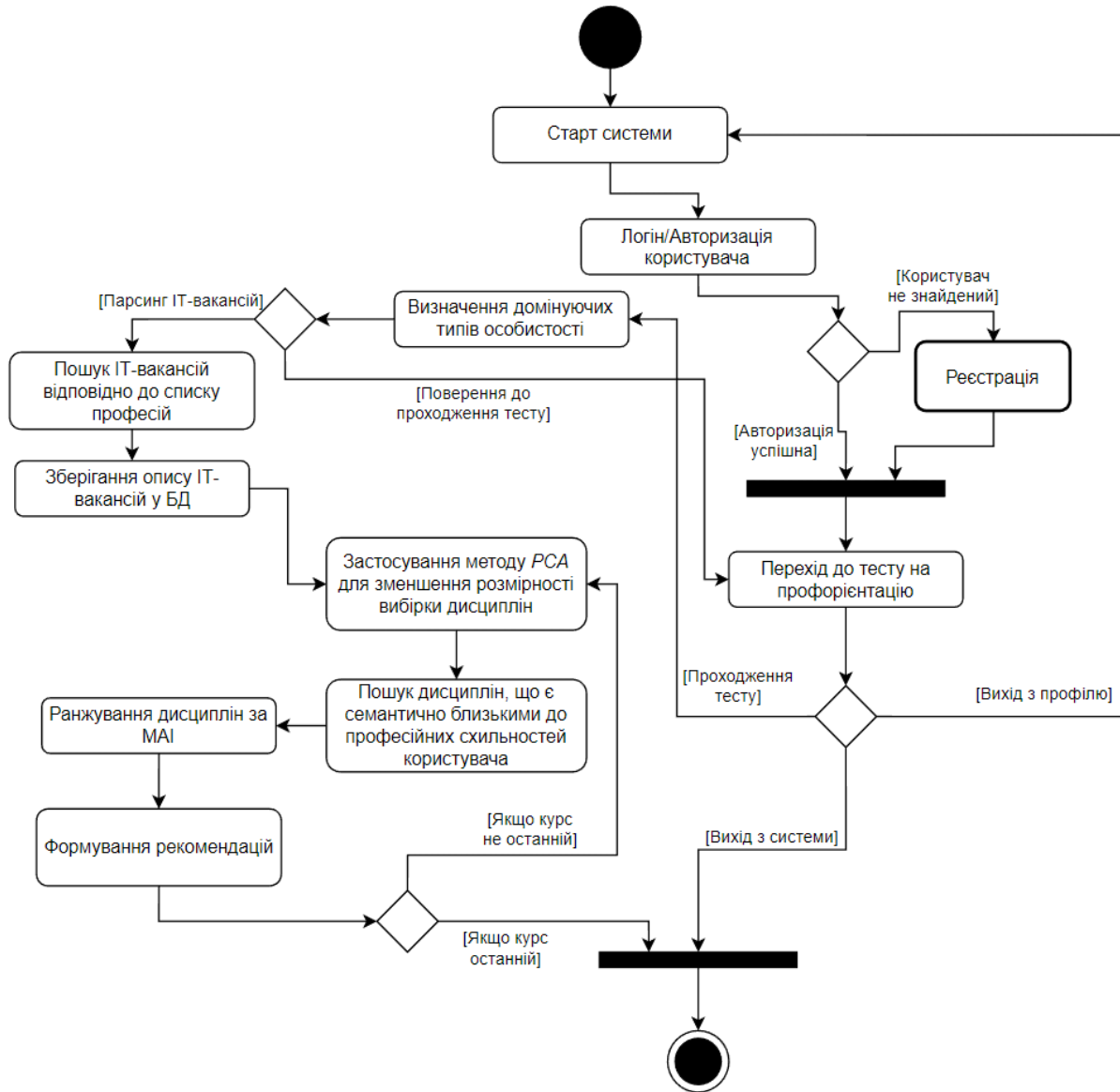


Рис. 4.4. Алгоритм функціонування веб-застосунку.

### 4.3 Особливості програмної реалізації.

Основний метод доступу до рекомендаційної системи – веб-застосунок, що розташований за доменним адресом. Використання веб-технологій для реалізації системи зумовлене широкою та зручною доступністю, об’ємною

базою даних на сервері та можливістю різноманітних модифікацій у майбутньому.

Архітектура веб-застосунку побудована на базі клієнт-серверної технології, що дозволяє працювати як в локальному, так і в мережевому режимі. Схема такої технології зображена на рисунку 4.5

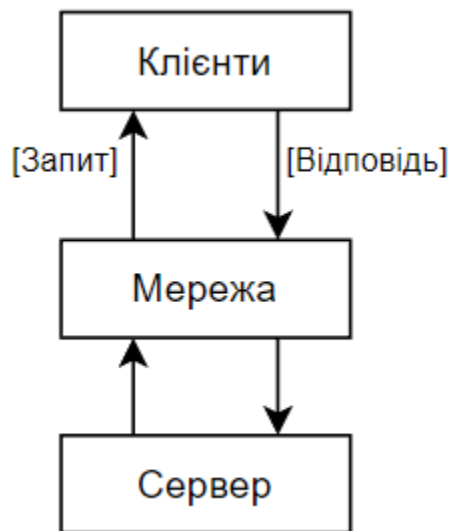


Рис. 4.5. Модель клієнт-серверної технології.

Клієнтська частина, написаний з використанням мови гіпертекстової розмітки *HTML* та спеціальною мовою розмітки даних (стилів) *CSS*. Серверна частина веб-застосунку використовує вільну систему керування реляційними базами даних - *MySQL* та скриптову мову програмування, що створена для генерації *HTML*-сторінок на стороні вебсервера – *PHP*. Використання *PHP* дозволяє сховати усі програмні процеси від користувача, на відміну від скриптові мови *JavaScript*, так як браузер отримує уже готовий *HTML* код для роботи.

*PHP* (гіпертекстовий препроцесор) відомий як мова сценаріїв загального призначення, яку можна використовувати для розробки динамічних та інтерактивних веб-сайтів. Це була одна з перших серверних мов, яку можна було вбудувати в *HTML*, що полегшувало додавання функцій до веб-сторінок без необхідності звертатися до зовнішніх файлів для отримання даних. Його використання змінювалося з роками, завдяки регулярним оновленням (версія 8.0 була випущена в листопаді 2020 року) [74], додаванням функцій і розблокуванню нових можливостей.

Мова *PHP* не є домінуючою при використанні у веб-середовищах, хоча і володіє певними особливостями та перевагами перед іншими. Розглянемо деякі ключові переваги, що доводять важливість *PHP* у веб-розробці [75]:

- *Скриптова мова*: скриптові мови інтерпретуються іншою програмою під час їх запуску, звідси витікає відсутність необхідності у компіляції. Скриптові мови можуть також інтерпретуватися на стороні сервера або на стороні клієнта (в браузері).
- *На стороні сервера*: *PHP* це скриптова мова на стороні сервера, що обробляється інтерпретатором *PHP* на веб-сервері; результат обробки відправляється у веб-браузер у вигляді простого *HTML*.
- *Відкритий вихідний код*: *PHP* доступний для безкоштовного завантаження та використання.
- *Об'єктно-орієнтований*: об'єктно-орієнтоване програмування (ООП) використовує концепцію «об'єктів» для зберігання даних та функцій, що допомагають створювати більш складні веб-програми для багаторазового використання. ООП було додано до *PHP5*.
- *Швидкість*: *PHP* використовує свою пам'ять, зводячи до мінімуму навантаження на сервер та підвищуючи продуктивність. *PHP* може бути на 382% швидшим, ніж *Python*, і на 195% швидше, ніж *Ruby*.

- *Простота*: синтаксис *PHP* легко зрозумілий для вивчення, незалежно від того, чи програма створюється з нуля, чи використовуються існуючі фреймворки або надбудови.
- *Хороша підтримка*: *PHP* підтримує всі провідні бази даних (*MySQL*, *SQLite*, *ODBC*) та сумісний з більшістю серверів (*Apache*, *IIS* тощо). Він також підтримується усіма популярними платформами (*Windows*, *Mac OS*, *Linux* тощо) і може додатково підтримуватися фреймворками *PHP* (*Laravel*, *CodeIgniter*, *Symfony*) та багатьма добре укомплектованими та перевіреними бібліотеками. На рисунку 4.6 представлена актуальність фреймворків, що використовують *PHP* відповідно до опитувань, що проведені JetBrains [76].

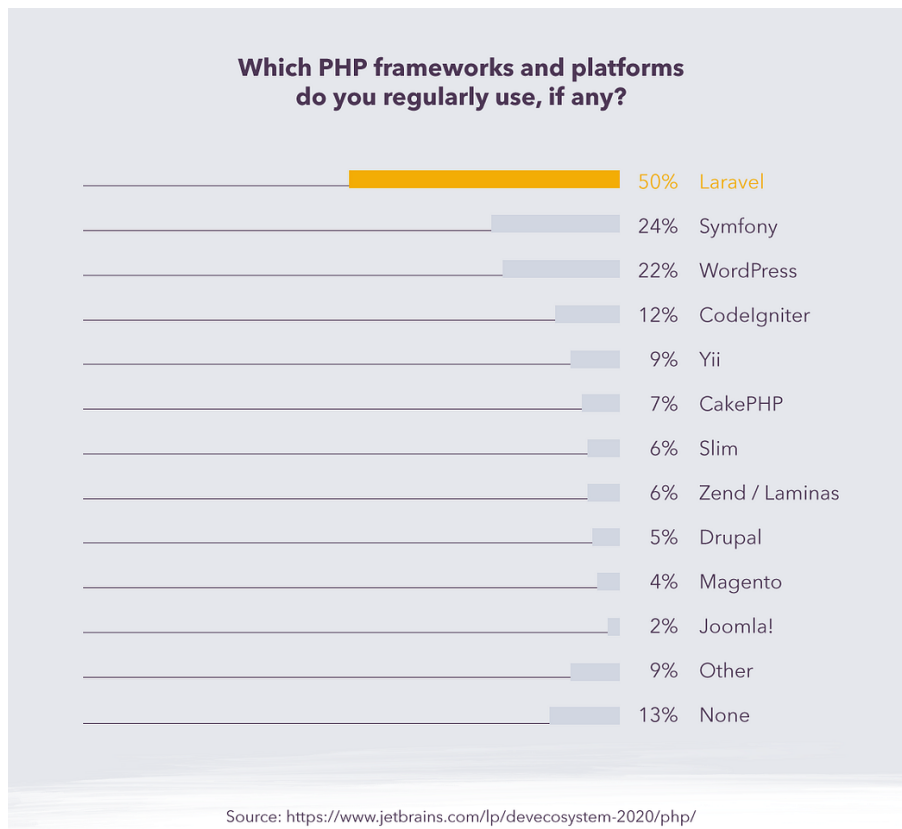


Рис. 4.6. Актуальність фреймворків, що використовують *PHP* [76].

Використання саме *PHP* у дисертаційному дослідженні зумовлене відсутністю необхідності у активній взаємодії користувача з інтерфейсом. Це є прерогативою мови програмування *JavaScript*

*JavaScript* — це мова скриптів (або програмування), яка дозволяє реалізувати складні функції на веб-сторінках і «оживити веб-сторінки». В основному будь-який веб-ресурс складаються з трьох частин: *HTML*, *CSS* і *Javascript* [77].

У той час, як *HTML* вважається скелетом веб-сторінки, а *CSS* «шкірою», *Javascript* задає зв'язки та взаємодію між структурою та стилем цієї структури. До основних функцій *Javascript* належить:

- Оновлення вмісту всередині структури.
- Зміна кольорів та розміру шрифту.
- Анімація зображень.

*Javascript* також іноді називають скриптовою мовою, хоча це все ще мова програмування. Проте головна відмінність полягає у тому, що програма, наприклад, написана мовою програмування *C*, обов'язково повинна бути скомпільована перед тим, як її можна буде запустити. В той же час *Javascript* не потрібно компілювати. Код уже містить серію команд, які інтерпретуються одна за одною під час виконання (на відміну від запуску при процесі компіляції).

*JavaScript* зазвичай або вбудовано у веб-сторінку, або включено у відповідний *.js* файл. Зазвичай *Javascript* запускається уже після завантаження *HTML* і *CSS* на сторінці, тому він розміщується у кінці тегу *body* в *HTML* коді.

Хоча *Javascript* в оригіналі було створено, щоб зробити веб-сайти та онлайн-ресурси інтерактивними, він часто використовується і для інших цілей, зокрема:

- Для розробки мобільних додатків (*React Native*, *Native Script*, *Ionic*);
- Для розробки браузерних ігор;
- Внутрішня веб-розробка (*Nodejs*).

PHP	JavaScript
<ul style="list-style-type: none"><li>• Server-side scripting language</li><li>• Used for back-end development</li><li>• More secure (as is not visible in browser)</li><li>• Helps to build high-level interactive web pages</li><li>• Quite slow performance</li><li>• More features available</li><li>• Combined with HTML</li><li>• MariaDB, MySQL, and PostgreSQL;</li><li>• WordPress, Drupal, Joomla</li><li>• Best for e-commerce and other websites using CMS</li></ul>	<ul style="list-style-type: none"><li>• Client-side scripting language</li><li>• Mainly used for front-end development</li><li>• Has tools for enhancing security but needs more effort to do so</li><li>• Helps to build user-friendly creative web pages</li><li>• Fast performance</li><li>• Less load on a server and less server traffic</li><li>• Combined with HTML, XML, Ajax</li><li>• AngularJS and ReactJS:</li><li>• MongoDB, CouchDB, and NoSQL</li><li>• Best for dynamic SPAs</li></ul>

Рис. 4.7. Основні відмінності між *Javascript* і *PHP* [78].

Розглянувши особливості обидвох мов програмування, а також провівши порівняння, що наведене на рис. 4.7, вибір в сторону мови *PHP* стає очевидним, адже більша частина усіх обчислень на веб-сайті відбувається на стороні сервера, а не зі сторони користувача – звідси і відсутність необхідності у активній взаємодії та додатковій інтерактивності.

#### 4.3.1 Особливості реалізації бази даних.



Основою будь-якої інформаційної системи є її база даних. Така ж ситуація і для рекомендаційних систем, які, очевидно, побудовані на зборі та аналізі даних. У дисертаційному дослідженні база даних використовує реляційну систему управління *MySQL*. Фрагмент бази даних системи зображений на рисунку 4.8.

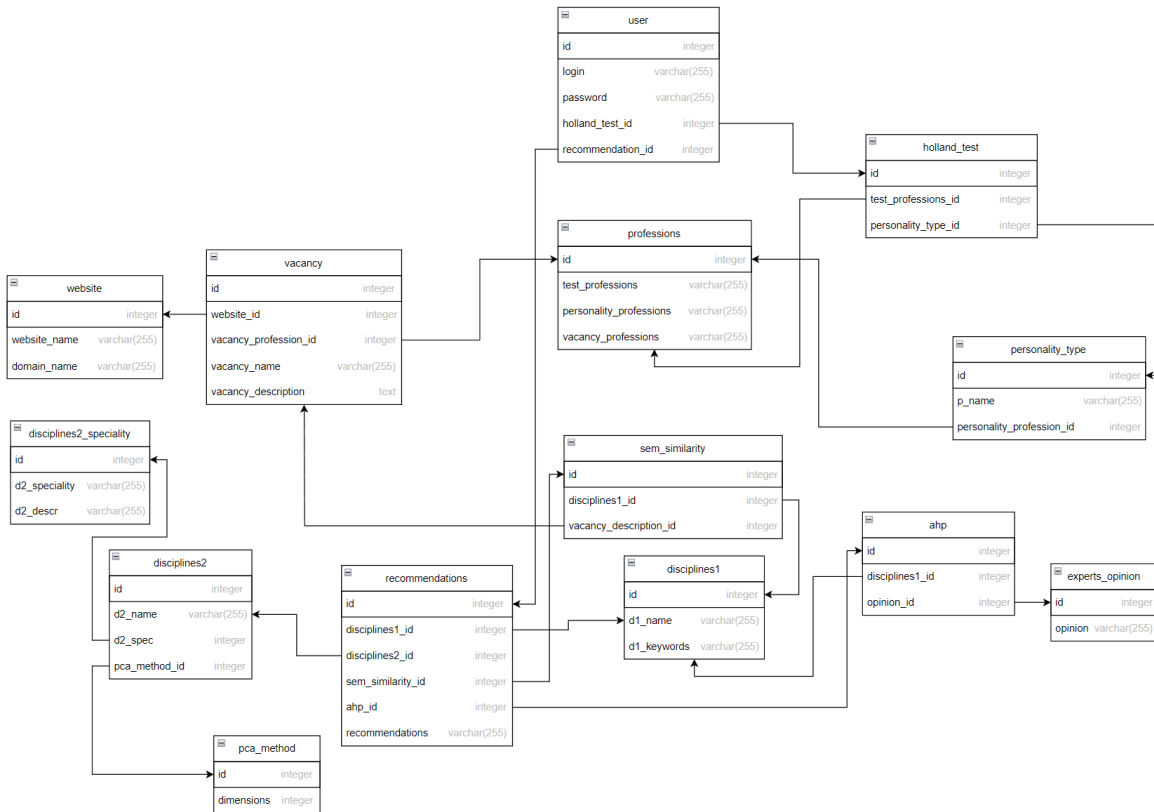


Рис. 4.8. Фрагмент бази даних.

Логічна складова бази даних складається з п'ятих частин:

1) Сутності БД, які містять інформацію з профілю користувача та списки усіх професій:

а) “Профіль користувача (*user*)” з атрибутами логіну (*login*), пароля (*password*), ідентифікатору тесту на профорієнтацію

(*holland\_test\_id*) та ідентифікатора рекомендацій (*recommendation\_id*).

b) “*Списки професій (professions)*” з атрибутами професій, що використовуються у тесті на профорієнтацію (*test\_professions*) та для здійснення парсингу на основі результатів тест (*personality\_professions*). Додатково присутній атрибут професій, що отримані в результаті парсингу ІТ-вакансій (*vacancy\_professions*)

2) Сутності БД, що містять інформацію про тест на профорієнтацію (тест Голланда):

a) “*Тест Голланда (holland\_test)*” з атрибутами ідентифікатора списку професій для тесту (*test\_professions\_id*) та ідентифікатора типів особистості (*personality\_type\_id*).

b) “*Тип особистості (personality\_type)*” з атрибутом назви типу особистості (*p\_name*) та ідентифікатором професій, що відповідають типам особистості (*personality\_profession\_id*).

3) Сутності БД, які зберігають інформацію про ІТ-вакансії та результати парсингу:

a) “*ІТ-вакансії (vacancy)*” з атрибутом ідентифікатора веб-сайту (*website\_id*), ідентифікатора професій ІТ-вакансій (*vacancy\_profession\_id*), назвою ІТ-вакансії (*vacancy\_name*) та відповідним описом вакансії (*vacancy\_description*).

b) “*Веб-сайт, де здійснювався парсинг (website)*” з атрибутами інформації по веб-сайту (*website\_name*) та його доменного імені (*domain\_name*).

4) Сутності БД, які зберігають інформацію про всі дисципліни з якими працює система, результати знаходження семантичної близькості та рекомендації для користувача:

- a) “Дисципліни (*disciplines1*)” з наступними атрибутами: назва дисципліни (*d1\_name*), ключові слова дисциплін (*d1\_keywords*). Дана сутність зберігає інформацію лише про компоненти вибіркового блоку освітніх програм (тобто дисципліни, що отримані з онтології).
- b) “Дисципліни (*disciplines2*)” з наступними атрибутами: назва дисципліни (*d2\_name*), ключові слова дисциплін (*d2\_spec*), ідентифікатор методу PCA (*pca\_method\_id*). Дана сутність зберігає інформацію про загальноуніверситетські дисципліни.
- c) “Метод визначення семантичної близькості (*sem\_similarity*)” з ідентифікатором дисциплін (*disciplines\_id*) та описом вакансії (*vacancy\_description\_id*).
- d) “Рекомендації (*recommendations*)” з атрибутами ідентифікатора дисциплін (*disciplines1\_id*) та (*disciplines2\_id*), семантичної близькості (*sem\_similarity\_id*) і методу аналізу ієрархій (*ahp\_id*). Також присутній атрибут рекомендацій (*recommendations*), що містить інформацію, яка пропонується користувачу на виході системи.

5) Сутності БД, які містять інформацію про математичні методи - PCA та MAI:

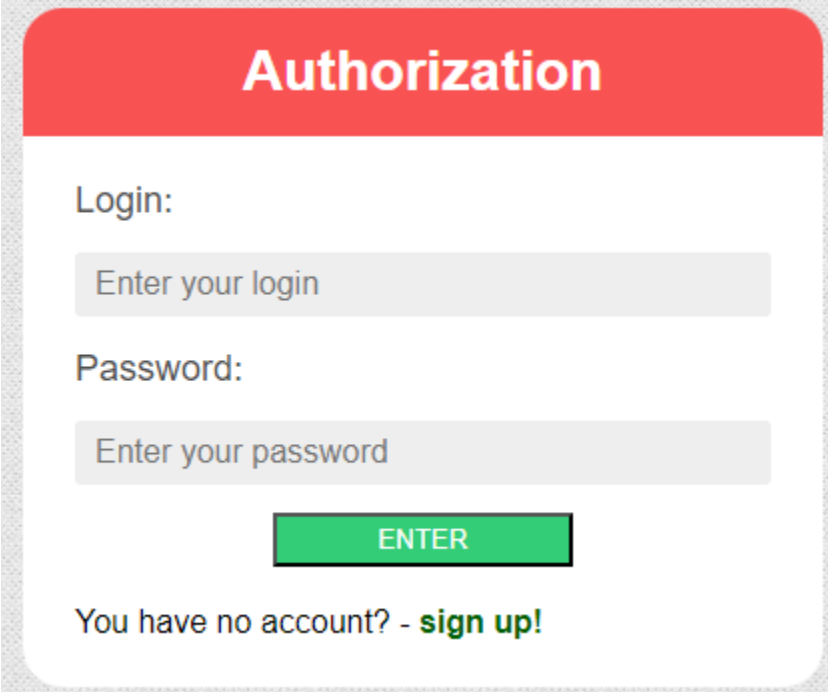
- a) “Метод аналізу ієрархій (*ahp*)” з ідентифікатором дисциплін (*disciplines\_id*) та експертними оцінками (*opinion*).
- b) “Експертні оцінки (для MAI) (*experts\_opinion*)” з ідентифікатором та атрибутом оцінок експертів (*opinion\_id*).

- с) “Метод PCA (*pca\_method*)” з ідентифікатором та атрибутом розмірності (*dimensions*).

#### 4.3.2 Опис реалізації користувацького інтерфейсу.

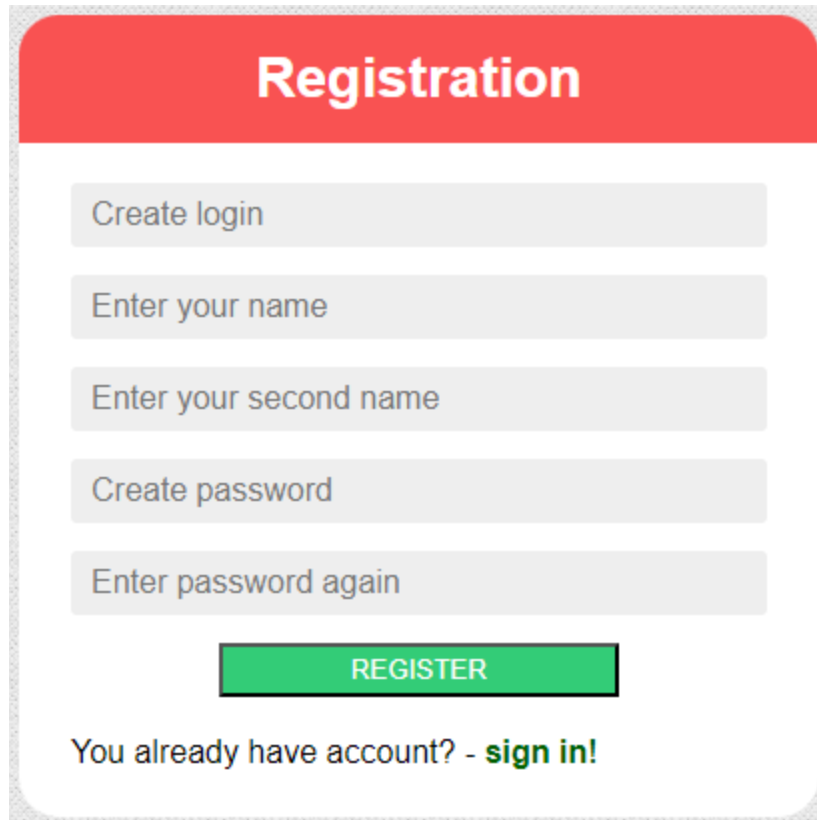
Візитною картою будь-якого веб-сайту є його користувацький інтерфейс. Як було згадано раніше, для побудови були використані технології *HTML* та *CSS*, а також *PHP*. Через відсутність скриптової мови *Javascript*. інтерфейс є дещо обмежений у функціоналі, проте це не створює проблем для користувача.

При вході на веб-сайт користувачу одразу пропонується здійснити логін у систему або ж зареєструватися. Авторизація представлена на рисунку 4.9. Реєстрація зображена на рисунку 4.10.



The image shows a login form with a red header containing the word "Authorization" in white. Below the header, there are two input fields: "Login:" followed by a light gray box with the placeholder text "Enter your login", and "Password:" followed by a light gray box with the placeholder text "Enter your password". Below these fields is a green button with the text "ENTER" in white. At the bottom of the form, there is a link that says "You have no account? - sign up!" where "sign up!" is in green.

Рис. 4.9. Сторінка логіну.

The image shows a registration form with a red header containing the word "Registration" in white. Below the header are five light gray input fields with the following labels: "Create login", "Enter your name", "Enter your second name", "Create password", and "Enter password again". Below these fields is a green button with the text "REGISTER" in white. At the bottom of the form, there is a link that says "You already have account? - sign in!".

**Registration**

Create login

Enter your name

Enter your second name

Create password

Enter password again

**REGISTER**

You already have account? - **sign in!**

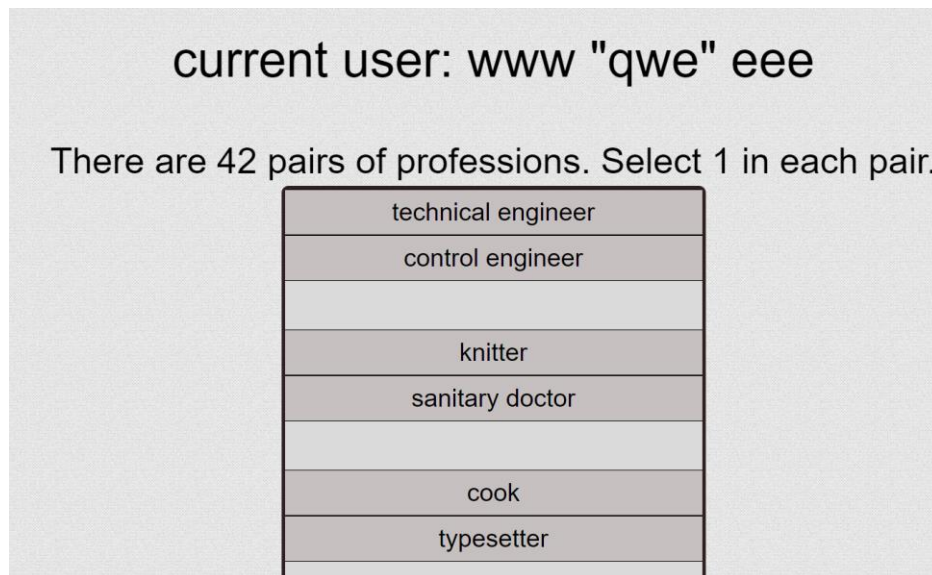
*Рис. 4.10. Сторінка реєстрації.*

Профіль користувача водночас послуговує і головною сторінкою. Це зроблено з ціллю обмежити доступи незареєстрованим користувачам, так як система повинна зберігати і обробляти персональні дані, що стає неможливим при відкритому доступі незареєстрованим користувачам.

Проходження тесту на профорієнтацію виглядає таким чином, що користувачу пропонується один варіант на вибір серед сорока двох пар професій. На рис. 4.11 та рис. 4.12 зображені фрагменти з інтерфейсу при проходженні тесту. Відповідно, на рис. 4.13 представлені результати тесту: усі шість типів особистості з набраними балами, а також підсумковий результат у вигляді коду Холланда.

На рис. 4.14 зображене практичне застосування методу головних компонент для зменшення розмірності даних. Використовуються функції *changeDimension()* та *applayingPca()* скриптової мови програмування *PHP*.

На рис. 4.15 і 4.16 зображені результати обчислень за методом аналізу ієрархій. Користувач отримує лише кінцеву інформацію без проміжних обчислень чи кроків. В кінці йому пропонують одну альтернативу за МАІ, котра отримала найбільшу вагу в результаті усіх обчислень. Ця альтернатива (навчальна дисципліна) рекомендується для включення у індивідуальну освітню траєкторію.



*Рис. 4.11. Перший фрагмент сторінки тесту на профорієнтацію.*

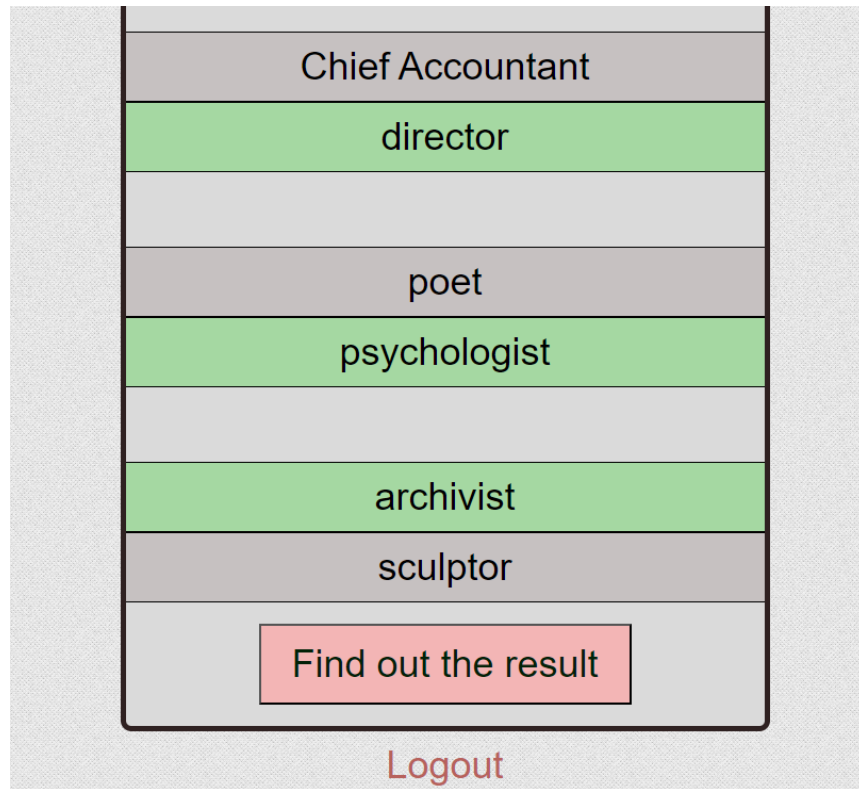


Рис. 4.12. Другий фрагмент сторінки тесту на профорієнтацію.

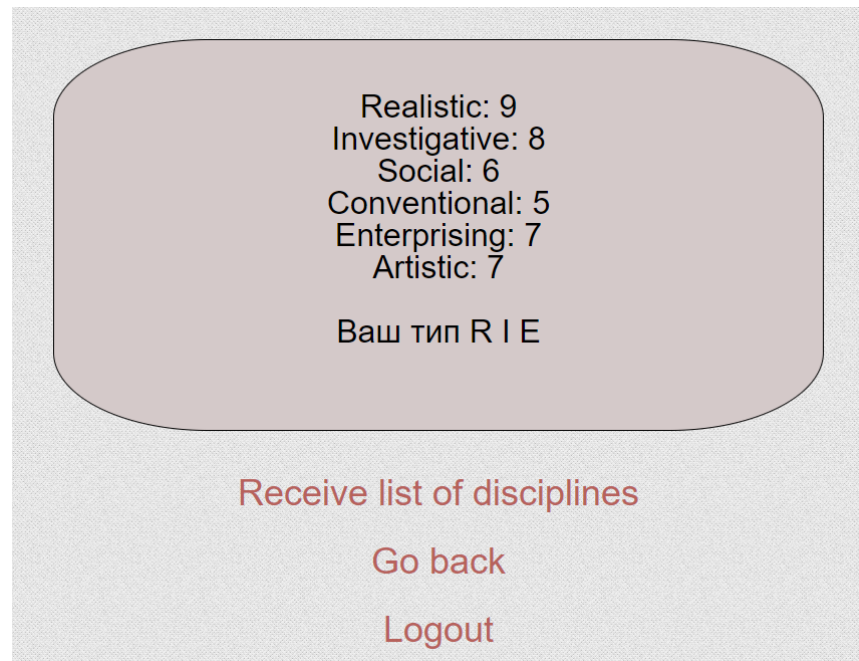


Рис. 4.13. Результати тесту.

```

19 $p = new PCA\PCA($points);
20 $p->changeDimension(1);
21 $p->applayingPca();
22 $result = $p->getArray();
23 foreach ($subjects as $key => $value)
24 {
25     $subjects[$key][4] = [$result[0][$key], $result[1][$key]];
26 }
27 foreach ($subjects as $key => $value)
28 {
29     $subjects[$key][5] = $subjects[$key][4][0] + $subjects[$key][4][1];
30 }
31 $max = 0;
32 $max_key = 0;
33 foreach ($subjects as $key => $value)
34 {
35     if ($subjects[$key][5] > $max)
36     {
37         $max = $subjects[$key][5];
38         $max_key = $key;
39     }
40 }
41 echo "<pre>";
42 $sub = $subjects[$max_key][0];
43 var_dump($subjects);
44 echo "Найбільша кількість балів {$max} у дисципліни {$sub}";
45 echo "</pre>";
46

```

Рис. 4.14. Практичне застосування методу головних компонент на мові

*RHP.*

Analytic hierarchy process showed the following results:

Планування IT-проектів:	31.07%
Інженерія програмного забезпечення:	34.21%
Моделювання бізнес-процесів:	34.71%

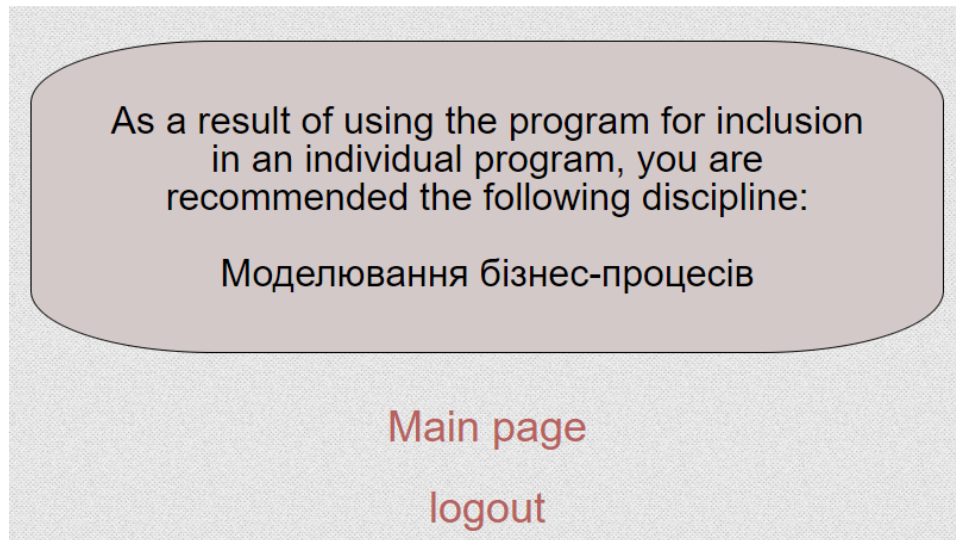
Receive recommendations

Go back

Logout

Рис. 4.15. Ранжування за MAI.





*Рис. 4.16. Сторінка з рекомендаціями.*

### 4.3.3. Технічні характеристики

Так як методом доступу до системи є веб-сайт, користувачам не потрібно нічого завантажувати чи встановлювати. Конкретні вимоги до операційної системи користувача відсутні. Необхідно лише забезпечити доступ до інтернету, так як веб-сайт працює у режимі онлайн. У таблиці 4.1 наведені детальні характеристики веб-застосунку

**Таблиця 4.1**

#### **Параметри веб-застосунку**

<b>Параметри</b>	<b>Одиниці виміру</b>	<b>Значення</b>
Швидкодія мови програмування	Оп/мс	1000
Об'єм пам'яті для збереження даних	Мб	10

Час на обробку запиту користувача	мс	150
Об'єм програмного коду	К-сть рядків коду	1500

#### **Висновки до розділу 4**

1. Розроблено структуру веб-застосунку який використовує напрацювання дослідження для формування рекомендацій навчальних дисциплін з можливістю їх подальшого включення у індивідуальну освітню траєкторію.

2. Наведено алгоритми функціонування веб-застосунку, описано функціональне призначення та представлена функціонально-структурна модель. Додатково проаналізовані технології веб-програмування та обґрунтовано вибір мови веб-програмування PHP для застосування у застосунку.

3. Продемонстровано користувацький інтерфейс у вигляді послідовних зображень процесу експлуатації веб-застосунку користувачем.

## ВИСНОВКИ

У дисертаційній роботі розв'язано актуальне наукове завдання розроблення системи рекомендацій навчальних дисциплін з їх подальшим включення у індивідуальну освітню траєкторію для студентів ІТ-спеціальностей.

При цьому отримано такі результати:

1. На основі аналізу існуючих інформаційних технологій, систем та моделей що використовуються для супроводу процесу формування індивідуальної освітньої траєкторії, запропоновано концептуальну схему етапів формування рекомендацій навчальних дисциплін для студентів ІТ-спеціальностей.

2. Розроблено модель процесу аналізу даних щодо визначення професійних схильностей студентів на основі профорієнтаційного тесту Джона Холланда. Додатково визначені відповідні професії до кожного типу особистості для полегшення процесу парсингу даних у рекомендаційній системі.

3. З використанням веб-технологій створено програму парсингу даних з веб-сайтів пошуку роботи. Процес парсингу використовує попередньо отримані дані про профорієнтацію студента і знаходиться відповідні для нього вакансії.

4. На основі даних, що попередньо отримані в результаті парсингу, проводять пошук та аналіз на предмет семантичної близькості до елементів онтології навчальних дисциплін. Онтологія отримана на основі освітньо-професійної програми з Національного університету «Львівська Політехніка».

5. З використанням сервісу хостингу створено веб-сайт, який реалізує рекомендаційну систему, що формує рекомендації користувача для включення навчальних дисциплін у індивідуальну освітню траєкторію. Розроблений веб-ресурс доступний усім користувачам мережі інтернет через доменний адрес.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Закон України «Про освіту» від 05.09.2017 № 2145-VIII. URL: <http://zakon2.rada.gov.ua/laws/show/2145-19>.
2. Каньковський, І. Є. (2013). Індивідуальні освітні траєкторії як необхідність сучасного процесу професійної підготовки фахівця. *Професійна освіта: проблеми і перспективи*, 4, 62-65.
3. The World Economic Forum. [Електронний ресурс] // Global Information Technology Report. – 1 April 2015. – Режим доступу до ресурсу: <https://www.weforum.org/reports/global-information-technology-report-2015/#indexId=NRI&economy=UKR>
4. Дубас Ю. В., Кунанець Н. Е. Застосування методу багатовимірної середньої при формуванні індивідуальних навчальних траєкторій студентів ІТ-спеціальностей. *Вісник Національного університету “Львівська політехніка”*. Серія: Інформаційні системи та мережі. 2020. Вип. 7. С. 70–77.
5. Dubas Y., Kunanets N. Definition of professional vector in educational systems of training future IT specialists *Вісник Національного університету “Львівська політехніка”*. Серія: Інформаційні системи та мережі. 2021. Вип. 9. С. 142–150.
6. Dubas Y., Mudrokha V., Pasichnyk V. Information system for the formation of individual educational trajectories of students of IT specialties. *CEUR Workshop Proceedings*. 2021. Vol. 2851: Proc. of the 2nd intern. workshop IT project management ITPM 2021, Slavsko, Ukraine, Febr. 16-18, 2021. Vol. 1. P. 184–194.
7. Dubas Y., Kunanets N. Use of ontology in information system to assist in the formation of an individual educational trajectory of students. *Вісник*

Хмельницького національного університету. Серія: Технічні науки. 2023. № 1 (317). С. 9–17.

8. Dubas Y., Kunanets N. The role of the analytic hierarchy process in the formation of a student's individual educational trajectory. *Manažérska Informatika*. 2022. R. 1, č. 2.

9. Karlgren Jussi An algebra for recommendations: Using reader data as a basis for measuring document proximity // Department of Computer and Systems Sciences, Stockholm University , 1990.

10. Вікіпедія. Netflix Prize [Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/Netflix\\_Prize](https://ru.wikipedia.org/wiki/Netflix_Prize)

11. Ricci, Francesco; Rokach, Lior; Shapira, Bracha (2022). "Recommender Systems: Techniques, Applications, and Challenges". In Ricci, Francesco; Rokach, Lior; Shapira, Bracha (eds.). *Recommender Systems Handbook* (3 ed.). New York: Springer. pp. 1–35.

12. "playboy Lead Rise of Recommendation Engines - TIME". *TIME.com*. 27 May 2010. Archived from the original on May 30, 2010. Retrieved 1 June 2015.

13. Resnick, Paul, and Hal R. Varian. "Recommender systems." *Communications of the ACM* 40, no. 3 (1997): 56-58.

14. Wavemakers. How Do Search Engines Work? [Електронний ресурс]. – Режим доступу: <https://wavemakers.co/how-search-engines-work/>

15. Medium. Brief on Recommender Systems [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd>

16. Papers with Code. Collaborative Filtering [Електронний ресурс]. – Режим доступу: <https://paperswithcode.com/task/collaborative-filtering>

17. Nitin M., Saumya C., Aanchal V., Sunita T. "Research Problems in Recommender systems". *Journal of Physics Conference* January 2021

18. Strands @ Recked: Recommendations 101 [Електронний ресурс]. – Режим доступу: <https://dokumen.tips/amp/technology/strands-recked-recommendations-101.html>

19. 5 Problems of Recommender Systems. ReadWrite [Електронний ресурс]. – Режим доступу: [https://readwrite.com/5\\_problems\\_of\\_recommender\\_systems/](https://readwrite.com/5_problems_of_recommender_systems/)

20. N. Lathia, S. Hailes, L. Capra, and X. Amatriain, “Temporal diversity in recommender systems,” in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 210–217, ACM, 2010.

21. W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, “Transfer learning in collaborative filtering for sparsity reduction,” in AAAI, vol. 10, pp. 230–235, 2010.

22. Положення про організацію освітнього процесу в Національному університеті «Львівська політехніка» [Електронний ресурс]. – Режим доступу: <https://lpnu.ua/sites/default/files/2020/pages/2005/polozhennya-pro-organizaciyu-osvitnogo-procesu.pdf>

23. Положення про формування та реалізацію індивідуальних навчальних планів студентів в Національному університеті «Львівська політехніка» [Електронний ресурс]. – Режим доступу: <https://lpnu.ua/sites/default/files/2020/pages/2001/svo-lp-0102-polozhennya-pro-formuvannya-ta-realizaciyu-individualnikh-navchalnikh-planiv-studentiv.PDF>

24. Порядок вибору навчальних дисциплін студентами Національного університету “Львівська політехніка” [Електронний ресурс]. – Режим доступу: [https://lpnu.ua/sites/default/files/attach/2017/6816/polozhennyaosvitniy\\_proces\\_vi\\_dnovleno.pdf](https://lpnu.ua/sites/default/files/attach/2017/6816/polozhennyaosvitniy_proces_vi_dnovleno.pdf)

25. Відомості про самооцінювання освітньої програми. Національний університет "Львівська політехніка" [Електронний ресурс]. – Режим доступу:

<https://lpnu.ua/sites/default/files/2022/program/18743/vso291mizhnarodna-informaciya.pdf>

26. Положення про формування та реалізацію індивідуальних навчальних планів студентів. Національний університет "Львівська політехніка" [Електронний ресурс]. – Режим доступу: [https://old.lpnu.ua/sites/default/files/attach/2018/9927/cvo\\_01.02\\_polozhennya\\_pro\\_formuvannya\\_ta\\_realizaciyu\\_indyvidualnyh\\_navchalnyh\\_planiv\\_studentiv.pdf](https://old.lpnu.ua/sites/default/files/attach/2018/9927/cvo_01.02_polozhennya_pro_formuvannya_ta_realizaciyu_indyvidualnyh_navchalnyh_planiv_studentiv.pdf)

27. Порядок організації навчання студентів за індивідуальними графіками. Національний університет "Львівська політехніка" [Електронний ресурс]. – Режим доступу: <https://lpnu.ua/sites/default/files/2020/pages/2011/svo0207poryadkorganizaciyinavchannyazaindyvidualnymygrafikamy.pdf>

28. Національна стратегія розвитку освіти в Україні на 2012-2021 роки [Електронний ресурс] – Режим доступу до ресурсу: <http://zakon3.rada.gov.ua/laws/show/344/2013>.

29. Наказ № 368 від 04.04.2016 р. [Електронний ресурс] // Міністерство освіти і науки України. – 2016. – Режим доступу до ресурсу: <http://old.mon.gov.ua/files/normative/2016-04-19/5432/nmo-368.pdf>.

30. ICTs in Education for People with Special Needs : specialized training course. Moscow : UNESCO Institute for Information Technologies in Education, 2006. 160 p.

31. Сучасні засоби ІКТ підтримки інклюзивного навчання : навчальний посібник / [А. В. Гета, В. М. Заїка, В. В. Коваленко та ін.] ; за заг. ред. Ю. Г. Носенко. – Полтава : ПУЕТ, 2018. – 261 с

32. E. de Corte, "Improving Higher Education Students' Learning Proficiency by Fostering their Self-regulation Skills", European Review. 2016. Vol. 24 (2), pp. 264-276.



33. Концептуальні засади професійного розвитку особистості в умовах євроінтеграційних процесів: зб. наук. статей / за ред. В.Г. Кременя, М.Ф. Дмитриченко, Н.Г. Ничкало – Уклад.: М.В. Артюшина, В.П. Тищенко та ін. – К.: НТУ, 2015. – 768 с.

34. Інформаційні ресурси бібліотек закладів освіти: формування та використання: практ. посіб. / НАПН України, ДНПБ України ім. В. О. Сухомлинського ; [Григоревська О. В., Матвійчук О. Є., Вараксіна Н. В., Кропачева Н. М., Бойко С. Т. ; наук. ред.: Пономаренко Л. О.]. – Вінниця : Нілан-ЛТД, 2022. – 181 с. – (Серія «На допомогу професійній самоосвіті працівників освітянських бібліотек» ; вип. 28 ). ISBN 978-966-924-984-5

35. Берко А. Ю. Інформаційне забезпечення інтелектуальних систем професійної орієнтації / А. Ю. Берко, У. Я. Коляса // Вісник Національного університету "Львівська політехніка". Серія : Електроніка : [зб. наук. пр.] / М-во освіти і науки України ; відп. ред. Д. Заячук. – Львів : Вид-во Нац. ун-ту "Львів. політехніка", 2006. – С. 41-49.

36. (PDF) Інформаційна система формування індивідуальної освітньої траєкторії студентів. Доступно з: [https://www.researchgate.net/publication/354577447\\_Information\\_System\\_for\\_the\\_Formation\\_of\\_Students'\\_Individual\\_Educational\\_Trajectory](https://www.researchgate.net/publication/354577447_Information_System_for_the_Formation_of_Students'_Individual_Educational_Trajectory).

37. (PDF) Система інформаційного забезпечення процесу формування індивідуальної освітньої траєкторії студентів [Електронний ресурс] – Режим доступу до ресурсу: <https://journal.iitta.gov.ua/index.php/itlt/article/view/1390/1062>.

38. (PDF) Система рекомендації навчальних матеріалів для персоналізації навчального шляху [Електронний ресурс] – Режим доступу до ресурсу: <https://core.ac.uk/download/pdf/323532858.pdf>.

39. Каталог навчальних дисциплін. Національний університет «Львівська Політехніка» [Електронний ресурс] – Режим доступу до ресурсу: <https://directory.lpnu.ua/subjects>

40. Джон Філіп Голланд. Wikiwand [Електронний ресурс]. – Режим доступу:

[https://www.wikiwand.com/uk/%D0%94%D0%B6%D0%BE%D0%BD\\_%D0%A4%D1%96%D0%BB%D1%96%D0%BF\\_%D0%93%D0%BE%D0%BB%D0%B%D0%B0%D0%BD%D0%B4](https://www.wikiwand.com/uk/%D0%94%D0%B6%D0%BE%D0%BD_%D0%A4%D1%96%D0%BB%D1%96%D0%BF_%D0%93%D0%BE%D0%BB%D0%B%D0%B0%D0%BD%D0%B4)

41. The History Of John Holland, PhD. Self-directed Search [Електронний ресурс]. – Режим доступу: <https://self-directed-search.com/about-john-holland/>

42. Holland code assessment and RIASEC. Career Key [Електронний ресурс]. – Режим доступу: [https://www.careerkey.org/fit/personality/holland-code-assessment-riasec#:~:text=The%20term%20Holland%20Code%2C%20Holland,\)%20and%20Conventional%20\(C\).](https://www.careerkey.org/fit/personality/holland-code-assessment-riasec#:~:text=The%20term%20Holland%20Code%2C%20Holland,)%20and%20Conventional%20(C).)

43. Gruber T. A translation approach to portable ontologies // Knowledge Acquisition. – 1993. – № 5 (2). – P. 199–220.

44. Онтології у контексті інтеграції інформації: представлення, методи та інструменти побудови / О.М. Овдій, Г.Ю. Проскудіна // Проблеми програмування. — 2004. — N 2,3. — С. 353-365. — Бібліогр.: 56 назв. — укр.

45. Литвин В. В. Бази знань інтелектуальних систем підтримки прийняття рішень. – Львів: Вид-во Львівської політехніки, 2011. – 240 с.

46. Литвин В. В. Мультиагентні системи підтримки прийняття рішень, що базуються на прецедентах та використовують адаптивні онтології // Радіоелектроніка, інформатика, управління. – 2009. – № 2 (21). – С. 120–126

47. Литвин В. В., Оборська О. В., Вовнянка Р. В. Метод моделювання процесу підтримки прийняття рішень у конкурентному середовищі // Математичні машини й системи. – 2014. – № 1. – С. 50–57.

48. Lee WN, Shah N, Sundlass K, Musen M. Comparison of ontology-based semantic-similarity measures. *AMIA Annu Symp Proc.* 2008 Nov 6;2008:384-8. PMID: 18999312; PMCID: PMC2655943.

49. Development and Application of a Metric on Semantic Net / Rada R., et al. // *IEEE Trans. on Systems, Man and Cybernetics.* — 1989. — 19(1). — P. 17—30.

50. Leacock C., Chodorow M. Combining local context and WordNet similarity for word sense identification // *WordNet: An electronic lexical database / Fellbaum C. (ed.). — Cambridge, MA: MIT press, 1998. — P. 265—283.*

51. Saaty, Thomas L.; Peniwati, Kirti (2008). *Group Decision Making: Drawing out and Reconciling Differences.* Pittsburgh, Pennsylvania: RWS Publications. ISBN 978-1-888603-08-8.

52. Saracoglu, B.O. (2013). Selecting industrial investment locations in master plans of countries. *European Journal of Industrial Engineering* 7 (4): 416–441.

53. Saati, T. (1993). *Priniatie reshenii. Metod analiza ierarkhii.* Moskva : Radio i sviaz (in Russian).

54. van der Maaten, L.J.P.; Hinton, G.E. (Nov 2008). Visualizing Data Using t-SNE. *Journal of Machine Learning Research* 9: 2579–2605.

55. T-розподілене вкладення стохастичної близькості. Вікіпедія [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/T-%D1%80%D0%BE%D0%B7%D0%BF%D0%BE%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B5\\_%D0%B2%D0%BA%D0%BB%D0%B0%D0%B4%D0%B5%D0%BD%D0%BD%D1%8F\\_%D1%81%D1%82%D0%BE%D1%85](https://uk.wikipedia.org/wiki/T-%D1%80%D0%BE%D0%B7%D0%BF%D0%BE%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B5_%D0%B2%D0%BA%D0%BB%D0%B0%D0%B4%D0%B5%D0%BD%D0%BD%D1%8F_%D1%81%D1%82%D0%BE%D1%85)

[%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%BD%D0%BE%D1%97%D0%B1%D0%BB%D0%B8%D0%B7%D1%8C%D0%BA%D0%BE%D1%81%D1%82%D1%96#cite\\_ref-MaatenHinton\\_1-0](#)

56. Ріманів многовид. Вікіпедія [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%A0%D1%96%D0%BC%D0%B0%D0%BD%D1%96%D0%B2\\_%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D0%B2%D0%B8%D0%B4](https://uk.wikipedia.org/wiki/%D0%A0%D1%96%D0%BC%D0%B0%D0%BD%D1%96%D0%B2_%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D0%B2%D0%B8%D0%B4)

57. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Umap-learn [Електронний ресурс]. – Режим доступу: <https://umap-learn.readthedocs.io/en/latest/>

58. Coordinate system. Wikipedia [Електронний ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Coordinate\\_system](https://en.wikipedia.org/wiki/Coordinate_system)

59. Jolliffe, Ian T.; Cadima, Jorge (2016-04-13). "Principal component analysis: a review and recent developments". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 374 (2065): 20150202.

60. Research Swiss Rolls Dataset. GitHub [Електронний ресурс]. – Режим доступу: [https://github.com/ajschumacher/gadsc1/blob/master/dataset\\_research/john\\_k\\_swissrolls.md](https://github.com/ajschumacher/gadsc1/blob/master/dataset_research/john_k_swissrolls.md)

61. Parser. Javatpoint [Електронний ресурс]. – Режим доступу: <https://www.javatpoint.com/parser>

62. Parser. TechTarget [Електронний ресурс]. – Режим доступу: <https://www.techtarget.com/searcharchitecture/definition/parser>

63. Types of Parsers in Compiler Design. GeeksforGeeks [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/types-of-parsers-in-compiler-design/>

64. Вакуленко Ю. О. Впровадження та перспективи використання програм-парсерів в закладах вищої освіти. – Вінниця – 2021.
65. Bradley N. Miller and David L. Ranum. 2011. Problem Solving with Algorithms and Data Structures Using Python SECOND EDITION (2nd. ed.). Franklin, Beedle & Associates Inc., USA.
66. Погорелов О. В. Створення інструменту для автоматизованого парсингу сайтів на основі технології Node.js. – Миколаїв – 2021.
67. IT професії. Старосалтівська територіальна громада [Електронний ресурс]. – Режим доступу: <https://stsaltiv.gov.ua/useful-info/it-profesii>
68. Анісімов А. В., Глибовець М. М., Марченко О. О, Кисенко В. К. Метод обчислення семантичної близькості для слів природної мови // National University of Kyiv-Mohyla Academy – 1 January 2011
69. Function strpos. Php.net [Електронний ресурс]. – Режим доступу: <https://www.php.net/manual/en/function.strpos.php>
70. Ian Oliver. 1993. Decision Graphs - An Extension of Decision Trees. Pearson Education. ISBN 978-0131004139.
71. Освітньо-професійна програма “Системний аналіз”. Національний університет “Львівська політехніка” [Електронний ресурс]. – Режим доступу: <https://lpnu.ua/sites/default/files/2021/program/12813/sa-opp-2022.pdf>
72. A.O. Nykonenko Semantic relatedness calculation method // Taras Shevchenko National University of Kyiv, Ukraine – 2016 – № 3
73. R. A. Wagner, M. J. Fischer. The string-to-string correction problem. J. ACM 21 1 (1974). P. 168—173
74. PHP 8.0 released. Php.net [Електронний ресурс]. – Режим доступу: <https://www.php.net/releases/8.0/en.php>

75. Why is PHP Still So Widely Used? Medium.com [Электронный ресурс].  
– Режим доступа: <https://medium.com/@radhwanrouihm/why-is-php-still-so-widely-used-e61984089903>

76. The State of Developer Ecosystem 2020 / PHP / JetBrains.com [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/lp/devecosystem-2020/php/>

77. What is Javascript? A Really Brief Explanation. Medium.com [Электронный ресурс]. – Режим доступа: <https://medium.com/analytics-vidhya/what-is-javascript-a-really-brief-explanation-3cd111dd87d9>

78. PHP vs Javascript: a Short Comparison. Scand.com [Электронный ресурс]. – Режим доступа: <https://scand.com/company/blog/php-vs-javascript-difference-between/>

**ДОДАТОК А**  
**ПРОГРАМНИЙ КОД РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ**

**ДОДАТОК Б**  
**АКТИ ВПРОВАДЖЕННЯ**