

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів автоматизації планування роботи команди виконавців ІТ-проекту
(тема)

Виконав:
студент 2 курсу, групи УПГІТм-22-1
Максим ДУДКА
(власне ім'я, прізвище)

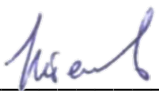
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва освітньої програми)

Керівник проф. каф. ІУС, Дмитро МІХНОВ
(посада, власне ім'я, прізвище)

Допускається до захисту
Зав. кафедри


(підпис)

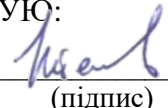
Костянтин ПЕТРОВ
(власне ім'я, прізвище)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Інформаційних управляючих систем
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 

(підпис)

« 01 » квітня 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Дудці Максиму Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів автоматизації планування роботи команди виконавців ІТ-проєкту
затверджена наказом університету від 01 квітня 2024 р. № 258Ст
2. Термін подання студентом роботи до екзаменаційної комісії 8 червня 2024р.
3. Вихідні дані до роботи: науково-технічні публікації; джерела інтернету; науково-технічна література, що стосуються теми кваліфікаційної роботи.
4. Перелік питань, що потрібно опрацювати в роботі: аналіз існуючих методів підбору персоналу, огляд сучасних технологій, оцінка ефективності методу.
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри): загальна характеристика роботи, існуюча модель підбору виконавців ІТ-проєкту, покращена модель підбору персоналу ІТ-проєкту.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Визначення типів ІТ-підприємств, для яких виконується дослідження	01.04.2024	виконано
2	Огляд існуючих методів для атоматизації планування роботи команди виконавців ІТ-проекту	13.04.2024	виконано
3	Аналіз переваг та недоліків застосування існуючих методів для визначеного типу ІТ-підприємств	15.04.2024	виконано
4	Огляд критеріїв оцінювання методів рекомендацій	20.04.2024	виконано
5	Вибір критеріїв для оцінки виконавця ІТ-проекту	25.04.2024	виконано
6	Огляд технологій для прототипу системи що реалізовує запропонований метод	30.04.2024	виконано
7	Перевірка запропонованого методу	15.05.2024	виконано
8	Подання студентом роботи для перевірки на плагіат	01.06.2024	виконано
9	Надання роботи на підпис науковому керівнику	03.06.2024	виконано
10	Попередній захист роботи	04.06.2024	виконано
11	Надання роботи на рецензію	05.06.2024	виконано
12	Надання роботи на підпис завідувачу кафедри	08.06.2024	виконано
13	Захист кваліфікаційної роботи	10.06.2024	виконано

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Дмитро МІХНОВ
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 83 стор., 4 рис., 30 табл., 35 джерел, 2 додатки.

МЕТОД КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ, РЕКОМЕНАЦІЙНА СИСТЕМА, ПІДБІР ВИКОНАВЦІВ ІТ-ПРОЄКТІВ, ПЛАНУВАННЯ ІТ-ПРОЄКТУ.

Метою даної роботи є дослідження моделей і методів автоматизації підбору виконавців для ІТ-проектів з використанням колаборативної фільтрації та прототип системи, яка використовує запропонований метод.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є процес підбору виконавців для ІТ-проектів.

Предметом дослідження являються методи аналізу і оптимізації процесу підбору виконавців.

Теоретичними результатами дослідження є опис методу колаборативної фільтрації та його основних етапів, а також розробка методики оцінки кандидатів на основі їхніх попередніх проектів.

Практичним результатом є апробація запропонованого методу автоматизації підбору виконавців для ІТ-проектів та розробка прототипу системи, що включає інтерфейс користувача для фільтрації кандидатів і застосування колаборативної фільтрації для рекомендацій.

Новизна дослідження полягає в дослідженні та розробці методу автоматизації підбору виконавців з використанням колаборативної фільтрації, описі сценаріїв його використання, описі взаємодії користувачів із системою та опис архітектури рішення.

ABSTRACT

The explanatory note to the qualification work: 83 p., 4 fig., 30 tabl., 35 sources, 2 appendixes.

METHOD OF COLLABORATIVE FILTERING, RECOMMENDATION SYSTEM, SELECTION OF IT PROJECT EXECUTORS, IT PROJECT PLANNING.

The purpose of this work is to research models and methods of automating the selection of performers for IT projects using collaborative filtering and a prototype system that uses the proposed method.

The object of research within the framework of the master's qualification work is the process of selecting executors for IT projects.

The subject of the study is methods of analysis and optimization of the process of selecting performers.

The theoretical results of the research are a description of the method of collaborative filtering and its main stages, as well as the development of a methodology for evaluating candidates based on their previous projects.

The practical result is the approbation of the proposed method of automating the selection of performers for IT projects and the development of a system prototype that includes a user interface for candidate filtering and the application of collaborative filtering for recommendations.

The novelty of the study consists in the research and development of the method of automating the selection of performers using collaborative filtering, the description of its use scenarios, the description of user interaction with the system, and the description of the solution architecture.

ЗМІСТ

Вступ	9
1 Проблема підбору виконавців для ІТ-проєкту	11
1.1 Огляд проблеми при підборі фахівців для виконання ІТ проєкту	11
1.2 Огляд особливостей ІТ-компаній	12
1.3 Стислий огляд існуючих методів підбору персоналу для ІТ проєкту	15
1.4 Постановка задачі дослідження	16
2 Огляд та аналіз методів підбору персоналу в ІТ-проєктах	18
2.1 Огляд існуючих методів	18
2.2 Аналіз критеріїв для оцінки якості рекомендаційних систем	23
2.3 Огляд критеріїв оцінювання методів рекомендацій	27
2.4 Вибір критеріїв для оцінки виконавця ІТ-проєкту	29
3 Розробка методу для додатку для внутрішньої системи ІТ-компанії	30
3.1 Пропонування методу для підбору виконавців ІТ-проєкту з покращеннями	30
3.2 Необхідні дані для роботи метода та способи їх перенесення в систему підбору виконавців ІТ-проєкту	35
4 Практичне застосування методу для покращення рекомендацій виконавців ІТ-проєкту	37
4.1 Загальний опис запропонованої системи для реалізації запропонованого метода	37
4.2 Детальний опис екранів додатку що реалізує запропонований метод	38
4.3 Загальний опис запропонованої системи для реалізації запропонованого метода	42
4.4 Порівняння результатів підбору кандидатів з використанням колаборативної фільтрації та без неї	58

Висновки	60
Перелік джерел посилання	62
Додаток А Код	66
Додаток Б Графічний матеріал роботи	71

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AWS –Amazon Web Services;
GCP –Google Cloud Platform;
OCI –Oracle Cloud Infrastructure.

ВСТУП

Сучасні інформаційні технології дозволяють вирішувати складні завдання, пов'язані з оптимізацією управління ІТ-проектами. Одним з ключових аспектів цього процесу є підбір виконавців, який може значно вплинути на успішність проекту. Традиційні методи відбору часто є трудомісткими, суб'єктивними і не враховують усіх важливих аспектів продуктивності кандидатів.

Автоматизація процесу підбору виконавців для ІТ-проектів підвищує його точність та об'єктивність. Колаборативна фільтрація, як один з найефективніших методів надання рекомендацій, базується на аналізі історичних даних та взаємодій користувачів. Цей підхід дозволяє обробляти великі обсяги інформації та надавати персоналізовані рекомендації, що значно покращує процес відбору кандидатів.

На даний момент існує безліч систем управління ІТ-проектами, проте більшість з них не враховують специфіку підбору виконавців з використанням колаборативної фільтрації. Відбір виконавців зазвичай здійснюється на основі стандартних критеріїв, таких як досвід, навички та рекомендації, що не завжди дозволяє знайти найкращих кандидатів для конкретного проекту.

Колаборативна фільтрація дозволяє враховувати додаткові фактори, такі як попередні проекти, оцінки продуктивності та взаємодії з іншими виконавцями. Це сприяє точнішому прогнозуванню потенційної продуктивності кандидатів у нових проектах та підвищує загальну ефективність управління проектами.

Таким чином, автоматизація процесу підбору виконавців з використанням колаборативної фільтрації забезпечує високу якість та продуктивність ІТ-проектів. Це дозволяє здійснювати ефективний підбір

персоналу, що відповідає сучасним вимогам ринку та сприяє успішному завершенню проєктів.

Об'єктом дослідження в рамках даної магістерської роботи є процес підбору виконавців для IT-проєктів.

Предметом дослідження є методи оптимізації процесу підбору виконавців.

Метою даної роботи є дослідження методів автоматизації підбору виконавців для IT-проєктів з використанням колаборативної фільтрації та розробка системи, яка об'єднує переваги існуючих методів, мінімізуючи їхні недоліки.

Ця магістерська робота виконана згідно з методичними вказівками 2024 року щодо розробки та оформлення магістерської роботи за спеціальністю 122 Комп'ютерні науки та освітньою програмою «Управління проєктами в галузі інформаційних технологій» [1].

1 ПРОБЛЕМА ПІДБОРУ ВИКОНАВЦІВ ДЛЯ ІТ ПРОЄКТУ

1.1 Огляд проблеми при підборі фахівців для виконання ІТ проєкту

Підбір правильної команди є одним із вирішальних факторів успіху будь-якого ІТ-проєкту. Цей процес включає визначення і розподіл ролей, які максимально відповідають індивідуальним компетенціям і досвіду учасників. Важливість правильного вибору виконавців особливо підкреслюється у випадках, коли висока складність проєкту вимагає різноманітних спеціалізацій і технічних навичок [2].

Традиційні методи підбору, такі як аналіз резюме, співбесіди та рекомендації систем з фільтрацією, часто не дозволяють повною мірою оцінити реальний внесок потенційних учасників в їх попередні проєкти та їхній практичний досвід. Такі методи можуть призвести до підбору кандидатів, які теоретично відповідають вимогам, але на практиці не можуть ефективно впоратися з конкретними завданнями через брак необхідного досвіду або спеціалізованих навичок [2].

Досвід роботи у конкретних технологічних стеках або проєктах схожої спрямованості може значно підвищити шанси на успіх проєкту. Програмісти та інженери, які раніше працювали з певними мовами програмування або платформами, мають краще розуміння потенційних проблем і нюансів, що зустрічаються під час розробки, що дозволяє їм швидше адаптуватися до проєктних вимог.

Однією з основних проблем традиційних підходів до підбору є їх висока трудомісткість та суб'єктивність. Рішення про найм здебільшого базуються на суб'єктивному оцінюванні інтерв'юерами, що може не завжди відображати реальні навички та потенціал кандидата. Це створює ризики неоптимального розподілу ресурсів та може вплинути на загальний прогрес і якість проєкту [2].

Автоматизація процесів підбору може допомогти уникнути багатьох із цих проблем та забезпечити більш об'єктивний та точний підбір. Системи можуть аналізувати великі обсяги інформації, включаючи історію завершених проєктів, роль в команді, використані технології, нотис період і виходячи з цього пропонувати кандидатів, які найкраще підходять для конкретних ІТ-проєктів.

1.2 Огляд особливостей ІТ-компаній

У середовищі інформаційних технологій (ІТ), що швидко розвивається, компанії відіграють ключову роль у стимулюванні інновацій, постачанні передових рішень і формуванні майбутнього цифрової трансформації. Ці організації суттєво відрізняються за розміром, структурою та характером проєктів, які вони здійснюють, кожна з яких вносить унікальний внесок у технологічну екосистему. Розуміння цих відмінностей має вирішальне значення для адаптації підходів до управління ІТ-проєктами, зокрема для автоматизації планування роботи для команд, яким доручено реалізувати ІТ-проєкти. Класифікація ІТ-компаній за розміром [3]:

- стартапи та малі ІТ-компанії. Стартапи та малі компанії, які характеризуються економною діяльністю, часто керуються інноваціями та гнучкістю. Зазвичай у них є невелика команда, зосереджена на розробці одного продукту або набору тісно пов'язаних послуг. Гнучкість і згуртованість таких команд дозволяють здійснювати швидкі цикли розробки, але часто з обмеженими ресурсами;

- ІТ-компанії середнього розміру: ці компанії встановили стабільну присутність на ринку з більш широким асортиментом продуктів і послуг. Маючи в своєму розпорядженні більше ресурсів, компанії середнього розміру можуть виконувати більші проєкти та мати більше спеціалізованих

команд. Однак вони зберігають баланс між гнучкістю невеликих компаній і орієнтацією на процеси більших корпорацій;

- великі ІТ-компанії: Великі ІТ-компанії чи підприємства працюють у глобальному масштабі, пропонуючи широкий спектр продуктів і послуг у різних секторах. Ці компанії мають значні ресурси, включаючи великі різноманітні команди ІТ-фахівців, а їх діяльність характеризується складними процесами управління проєктами та сильним акцентом на масштабованості та безпеці.

Розподілення за типом і напрямком:

- компанії з розробки програмного забезпечення: спеціалізуються на створенні програмних продуктів, від мобільних і веб-додатків до корпоративних програмних рішень. У цих компаніях часто працює широкий спектр спеціалістів, включаючи розробників, тестувальників і менеджерів проєктів;

- ІТ-консалтингові фірми: ці фірми пропонують експертні поради та послуги, щоб допомогти підприємствам узгодити свої ІТ-стратегії з бізнес-цілями. Їхня робота може варіюватися від консультацій щодо найкращого використання технологій до впровадження складних систем або заходів кібербезпеки;

- постачальники хмарних послуг та інфраструктури. З розвитком хмарних обчислень компанії, що спеціалізуються на наданні хмарної інфраструктури, послуг платформи та програмного забезпечення як послуги (SaaS), відіграють вирішальну роль в ІТ-екосистемі. Вони дозволяють компаніям використовувати масштабовані обчислювальні ресурси на вимогу;

- компанії з виробництва обладнання та мереж: ці компанії зосереджуються на фізичних компонентах ІТ-інфраструктури, включаючи сервери, пристрої зберігання та мережеве обладнання. Вони мають вирішальне значення для розгортання та обслуговування ІТ-систем.

Робота зосереджена на оптимізації підбору ІТ-виконавців ІТ-проєкту, визнання різноманітної екосистеми ролей у ІТ-проєктах має вирішальне значення для контекстуалізації даного підходу та його застосовності. Класифікація робітників-інженерів ІТ-компанії [4]:

- інженери програмного забезпечення: ці професіонали відповідають за проєктування, розробку та підтримку програмного забезпечення. Серед іншого, вони можуть спеціалізуватися на інтерфейсній, бек-енд, повноцінній і мобільній розробці. Запропонований метод дуже актуальний для інженерів програмного забезпечення, оскільки він спрямований на розподіл завдань кодування на основі їхніх конкретних навичок та історичних моделей кодування;

- інженери із забезпечення якості (QA): інженери із забезпечення якості зосереджуються на забезпеченні якості програмного забезпечення за допомогою ручного та автоматизованого тестування. Вони виявляють помилки, впроваджують протоколи тестування та тісно співпрацюють з розробниками, щоб забезпечити відповідність програмного забезпечення всім вимогам і стандартам. Наш метод може допомогти у призначенні завдань із забезпечення якості, які відповідають досвіду інженера з конкретними рамками або методологіями тестування;

- інженери DevOps: спеціалізуючись на інтеграції розробки та операцій, інженери DevOps працюють над автоматизацією та оптимізацією життєвого циклу розробки програмного забезпечення. Вони керують конвеєрами CI/CD, інфраструктурою та системами моніторингу. Запропонований підхід може допомогти визначити завдання DevOps, узгоджені з досвідом інженера в конкретних інструментах або середовищах.

Класифікація менеджерів ІТ-компанії:

- менеджери проєктів: ці люди відповідають за планування, виконання та закриття проєктів. Вони гарантують виконання проєктів вчасно, в межах бюджету та відповідність стандартам якості;

- менеджери продуктів: менеджери продуктів контролюють розробку продуктів від концепції до запуску. Вони тісно співпрацюють з інженерними командами та зацікавленими сторонами, щоб визначити бачення продукту та roadmap;
- СТО (Chief Technology Officer): СТО відповідає за технологічний напрямок компанії. Вони приймають стратегічні рішення щодо впровадження нових технологій і методологій;
- генеральний директор (головний виконавчий директор): генеральний директор контролює загальну діяльність і результативність компанії.

1.3 Стислий огляд існуючих методів підбору персоналу для ІТ проєкту

Оглядаючи методи зрозуміло що використання передових обчислювальних методів та їх інтеграція з системами управління проєктами є багатообіцяючим підходом. Отож буде розглянути можливі методи оптимізацій для системи що буде рекомендувати кандидатів на ІТ-проєкт.

- метод оптимізації метод заснований на вмісті (Content-Based Filtering): Цей підхід може використовувати детальні профілі кандидатів, що включають технічні навички, попередній досвід, освітній рівень та інші релевантні характеристики. Система аналізує ці дані для створення рекомендацій, порівнюючи вимоги проєкту з профілями кандидатів [5];
- нейронні мережі та глибоке навчання: Моделі глибокого навчання можуть виявляти складні шаблони та залежності в даних, що може допомогти у виявленні оптимальних кандидатів для конкретних задач. Наприклад, використання рекурентних нейронних мереж для аналізу історії проєктів та внесків кандидатів [6];

– кластеризація: Алгоритми кластеризації можуть допомогти у групуванні схожих кандидатів або проєктів на основі різних атрибутів. Це дозволяє системі швидше визначати потенційно підходящі групи для нових проєктів;

– колаборативна фільтрація [5]. Один з основних методів, використовуваних у системах рекомендацій. Його суть полягає у використанні даних про попередні взаємодії користувачів для прогнозування та рекомендації елементів, які можуть їх зацікавити. Цей метод можна застосовувати не тільки для рекомендації товарів або фільмів, але й для визначення оптимальних кандидатів для роботи над ІТ-проєктами.

Додавання методів для оптимізації рекомендації кандидатів має великий потенціал. Такий підхід не тільки оптимізує процес планування, але й гарантує, що проєкт підходить програмісту, враховуючи його індивідуальний досвід і знання. Таким чином, він прагне максимізувати показники успішності проєктів, підвищити продуктивність команди та створити середовище, де програмісти залучені та мотивовані роботою, яка відповідає їхнім сильним сторонам.

1.4 Постановка задачі дослідження

Враховуючи різноманіття ІТ-компаній, очевидно, що підхід до управління ІТ-проєктами, включаючи підбір виконавців ІТ-проєкту, має бути пристосований до конкретного контексту кожної компанії. Ця робота зосереджена саме на великих ІТ-компаніях, де складність і масштаб проєктів вимагають складних методів для планування виконання проєктів. У таких середовищах проблема підбору виконавців стає особливо гострою. Великі компанії часто мають великі репозиторії попередніх проєктів, що робить

можливим використовувати цю велику кількість даних для більш точного підбору виконавців ІТ-проєкту, які мають найбільш відповідний досвід.

У цьому контексті ціллю даної роботи є створення методу та прототипу що його використовує для автоматизації підбору виконавців ІТ проєктів. Для цього треба вирішити наступні задачі:

- порівняти існуючі методи, їх плюси та мінуси в залежності від ситуації;
- сформулювати критерії оцінки рекомендації;
- визначити метод для цілей дослідження;
- модернізувати існуючий метод.

2 ОГЛЯД ТА АНАЛІЗ МЕТОДІВ ПІДБОРУ ПЕРСОНАЛУ В ІТ-ПРОЄКТАХ

2.1 Огляд існуючих методів

Існує декілька основних методів до розробки рекомендаційних систем, що можуть бути застосовані для підбору персоналу, кожен з яких має свої унікальні характеристики, переваги та недоліки. Вибір відповідного методу залежить від специфіки конкретного завдання, доступних даних та ресурсів.

Метод, заснований на вмісті (Content-Based Filtering), нейронні мережі та глибоке навчання, кластеризація та колаборативна фільтрація є серед найбільш широко застосовуваних підходів у цій галузі [7]. Кожен з них відрізняється підходом до обробки даних і забезпеченням рекомендацій, що впливає на їхню ефективність у різних сценаріях.

Щоб краще зрозуміти, який метод найкраще підходить для конкретної ситуації, важливо розглянути їхні переваги та недоліки, враховуючи особливості та цілі проєкту.

Таблиця 2.1 – Порівняння переваг та недоліків Content-Based Filtering (сформовано на основі джерела [8])

Переваги	Недоліки
Метод заснований на вмісті (Content-Based Filtering)	
Індивідуалізація: Цей метод дозволяє створювати чітко персоналізовані рекомендації, оскільки кожен користувач отримує рекомендації, які базуються на його власних інтересах і поведінці. Це дає можливість відповідати унікальним вподобанням кожного користувача, що є великою перевагою для систем, орієнтованих на індивідуальний підхід.	Обмежена диверсифікація: Рекомендації, як правило, базуються на тих самих характеристиках і типах об'єктів, які вже подобаються користувачеві, що може призвести до одноманітності рекомендацій та обмеженого охоплення інших можливих інтересів.

Кінець таблиці 2.1

Переваги	Недоліки
<p>Незалежність від інших користувачів: Метод заснований на аналізі профілю користувача, тому не потребує значної кількості даних про інших користувачів, що може бути важливо в умовах конфіденційності або при обмеженому обсязі даних.</p>	<p>Відсутність здатності виявляти нові інтереси: Оскільки рекомендації базуються на попередніх взаємодіях, система може не рекомендувати нові типи об'єктів, з якими користувач ще не взаємодіяв, що обмежує можливість виявляти нові інтереси.</p>
<p>Прозорість: Рекомендації можуть легко пояснюватися, оскільки вони базуються на чітко визначених атрибутах та характеристиках, відомих користувачу, що робить систему більш зрозумілою для кінцевих користувачів.</p>	<p>Обмеженість атрибутів: Якість рекомендацій залежить від повноти та деталізації описових атрибутів об'єктів. Якщо атрибути об'єктів недостатньо чітко визначені або неповні, це може призвести до менш точних рекомендацій.</p>
<p>Легкість оновлення: Цей метод дозволяє легко додавати нові атрибути та властивості, що забезпечує адаптивність системи до змін у вподобаннях користувачів або додаткових характеристик об'єктів, які система повинна враховувати.</p>	

Застосування: Оптимально підходить для середовищ, де кожен користувач має чітко визначені інтереси, а також у випадках, коли база даних про інших користувачів обмежена або існують проблеми конфіденційності даних. Зазвичай використовується в системах рекомендацій книг, статей, музики, а також для профілювання кандидатів для різних ролей.

Таблиця 2.2 – Порівняння переваг та недоліків нейронних мереж та глибокого навчання (сформовано на основі джерела [9])

Переваги	Недоліки
Нейронні мережі та глибоке навчання	
<p>Виявлення складних залежностей: Нейронні мережі та глибоке навчання дозволяють виявляти складні закономірності та залежності в даних, які не виявляються традиційними методами. Це дозволяє створювати більш точні та комплексні рекомендації.</p>	<p>Вимоги до обчислювальних ресурсів: Навчання та використання моделей глибокого навчання вимагає значних обчислювальних ресурсів, що може бути дорого для промислового використання. Це вимагає спеціального обладнання та інфраструктури.</p>
<p>Адаптивність: Моделі глибокого навчання можуть бути переналаштовані та масштабовані для різних типів задач та обсягів даних. Вони можуть адаптуватися до змін у даних та потребах користувачів, що забезпечує їх високу гнучкість.</p>	<p>Складність налаштування: Для налаштування та оптимізації моделей глибокого навчання потрібні досвідчені спеціалісти, що робить процес тривалим та ресурсомістким. Це може бути складно для організацій, які тільки починають працювати з такими технологіями.</p>
<p>Масштабованість: Підходить для обробки великих обсягів даних, завдяки можливості використовувати розподілені обчислення та оптимізації. Це робить цей підхід ефективним навіть у дуже великих системах.</p>	<p>Невизначеність результатів: Моделі глибокого навчання можуть бути важко інтерпретувати, що ускладнює пояснення рекомендацій та прийняття рішень на їх основі. Це може знизити довіру до рекомендацій, які не підкріплені прозорими аргументами.</p>
<p>Адаптація до нових даних: Моделі глибокого навчання можуть навчатися в режимі реального часу, постійно адаптуючись до нових даних. Це дозволяє системі з часом підвищувати свою точність та ефективність.</p>	<p>Чутливість до навчальних даних: Моделі можуть навчатися на упереджених або нерелевантних даних, що впливає на якість рекомендацій. Це вимагає ретельної перевірки даних та відфільтрування шуму.</p>

Застосування: Ефективно в ситуаціях, де доступні великі обсяги даних, є можливість використовувати потужні обчислювальні ресурси, а також є потреба у виявленні складних закономірностей. Зазвичай використовується у великих рекомендаційних системах для потокових сервісів, електронної комерції та для аналізу великих обсягів даних.

Таблиця 2.3 – Порівняння переваг та недоліків методу кластеризації (сформовано на основі джерела [10])

Переваги	Недоліки
Кластеризація	
Групування подібних елементів: Кластеризація дозволяє виявляти природні групи користувачів або об'єктів, що дозволяє визначати загальні тенденції та рекомендації для кожної групи. Це забезпечує сегментацію користувачів на основі їх подібних вподобань.	Відсутність індивідуалізації: Усі користувачі в одному кластері отримують однакові рекомендації, що обмежує рівень персоналізації та може призвести до менш точної відповідності вподобанням.
Гнучкість: Існує багато різних алгоритмів кластеризації з різними підходами до групування, що дозволяє вибрати найбільш відповідний алгоритм для конкретного набору даних та цілей.	Проблеми з масштабуванням: Кластеризація може бути складною при роботі з дуже великими наборами даних, які можуть не поміщатися в пам'ять або оброблятися за прийнятний час.
Швидкість: У великих наборах даних кластеризація може бути швидшим методом, ніж інші підходи, особливо при використанні оптимізованих алгоритмів.	Залежність від алгоритму: Результати кластеризації сильно залежать від обраного алгоритму та його гіперпараметрів, що може впливати на якість та точність результатів.
Виявлення аномалій: Кластеризація допомагає виявляти аномальні групи або об'єкти, що не відповідають жодному з кластерів, що дозволяє визначати нетипові або рідкісні елементи.	Труднощі з визначенням кількості кластерів: Підбір оптимальної кількості кластерів часто потребує додаткового аналізу та експериментів, що може бути трудомістким та ресурсомістким процесом.

Застосування: Добре підходить для визначення загальних тенденцій у великих групах користувачів або об'єктів, особливо в умовах, де персоналізація менш важлива. Використовується для аналізу клієнтів у маркетингу, визначення тенденцій та сегментів ринку, а також для виявлення аномалій.

Таблиця 2.4 – Порівняння переваг та недоліків методу колаборативної фільтрації (сформовано на основі джерела [8])

Переваги	Недоліки
Колаборативна фільтрація	
<p>Індивідуалізація: Метод колаборативної фільтрації враховує індивідуальні уподобання користувача, аналізуючи його історію взаємодії з об'єктами та порівнюючи з іншими користувачами. Це дозволяє створювати персоналізовані рекомендації на основі схожості вподобань.</p>	<p>Проблема холодного старту: Важко надати рекомендації новим користувачам або новим елементам, оскільки про них ще немає даних, що обмежує можливість надання точних рекомендацій з самого початку.</p>
<p>Диверсифікація: Рекомендації можуть бути різноманітними завдяки аналізу вподобань різних користувачів та об'єктів, що розширює можливості рекомендаційної системи та відкриває нові інтереси для користувачів.</p>	<p>Залежність від великих даних: Ефективність системи значно знижується при відсутності великої кількості даних про взаємодії користувачів та об'єктів, що може бути викликом для невеликих компаній або нових проєктів.</p>
<p>Адаптивність: Система постійно навчається на нових даних і може швидко адаптуватися до змін у вподобаннях користувачів, що дозволяє забезпечити актуальні рекомендації.</p>	<p>Складність реалізації: Колаборативна фільтрація вимагає спеціальних алгоритмів для ефективної роботи з великими обсягами даних та забезпечення точності рекомендацій,</p>

	що може бути складним для реалізації.
--	---------------------------------------

Кінець таблиці 2.4

Переваги	Недоліки
Покращення з об'ємом даних: Ефективність системи покращується з часом, оскільки обсяги даних взаємодії користувачів та об'єктів збільшуються, що дозволяє системі враховувати більше вподобань та варіантів поведінки.	Сприйнятливості до шуму: Помилки у взаємодіях або упередженість можуть призвести до неточних або неправильних рекомендацій, що вимагає ретельного фільтрування даних.

Застосування: Найкраще працює в середовищах, де є великі обсяги даних про взаємодію користувачів та об'єктів, наприклад у рекомендаційних системах для електронної комерції, потокових сервісів або соціальних мереж.

2.2 Аналіз критеріїв для оцінки якості рекомендаційних систем

Оцінка якості рекомендаційних систем є важливою складовою процесу розробки, оскільки вона дозволяє визначити, наскільки ефективно система задовольняє потреби користувачів. Існує декілька популярних методів оцінки, кожен з яких має свої плюси та мінуси.

Таблиця 2.5 – Порівняння переваг та недоліків Precision (сформовано на основі джерела [11])

Переваги	Недоліки
Точність прогнозування (Precision)	
Специфічність: Цей показник допомагає оцінити, наскільки точно система вибирає релевантні елементи для рекомендацій.	Ігнорування відхилення: Не враховує випадки, коли релевантні елементи не були рекомендовані.

Precision вимірює частку релевантних рекомендацій серед усіх наданих рекомендацій. Це часто використовується для вимірювання ефективності систем, що рекомендують обмежену кількість об'єктів.

Таблиця 2.6 – Порівняння переваг та недоліків Recall (сформовано на основі джерела [11])

Переваги	Недоліки
Виклик (Recall)	
Повнота: Вимірює, наскільки повно система здатна виявляти всі релевантні елементи.	Можливе зниження точності: Вищий виклик може призвести до зниження точності, оскільки система може включати більше нецільових рекомендацій.
Корисний для критичних застосувань: Де важливо не пропустити релевантні елементи.	

Recall вимірює частку релевантних елементів, що були рекомендовані системою, серед усіх релевантних елементів у датасеті.

Таблиця 2.7 – Порівняння переваг та недоліків F1 Score (сформовано на основі джерела [11])

Переваги	Недоліки
F1 Score	
Баланс: Збалансовану оцінку між релевантних елементів і нерелевантних. Забезпечує втратою включенням	Не враховує розподіл класів: Може не відображати реальну ефективність у випадках, коли класи мають дуже різний розподіл.

F1-бал є гармонійним середнім між Precision і Recall, що дозволяє оцінити баланс між цими двома показниками.

Таблиця 2.8 – Порівняння переваг та недоліків Mean Absolute Error (MAE) та Root Mean Squared Error (RMSE) (сформовано на основі джерела [11])

Переваги	Недоліки
Mean Absolute Error (MAE) та Root Mean Squared Error (RMSE)	
Кількісні оцінки: Надають чітку кількісну міру розбіжності між прогнозами та фактичними значеннями.	Середнє значення може бути вводити в оману: Одне велике відхилення може значно вплинути на середнє значення, особливо в RMSE.
Нечутливість до розподілу помилок: RMSE надає більшу вагу більшим помилкам, що може бути як перевагою, так і недоліком залежно від контексту.	

Mean Absolute Error (MAE) та Root Mean Squared Error (RMSE) вимірюють середню помилку між прогнозованими та фактичними оцінками, які система надає.

Для оцінки якості рекомендаційної системи, що буде розроблена в рамках дипломної роботи, було обрано використання Mean Absolute Error (MAE) та Root Mean Squared Error (RMSE). Ці метрики забезпечують чіткі кількісні оцінки розбіжностей між прогнозами та фактичними значеннями, що дозволяє детально аналізувати ефективність системи.

Mean Absolute Error (MAE): Ця метрика обчислює середню абсолютну помилку між прогнозованими та фактичними оцінками. Вона є простою для розуміння та інтерпретації, надаючи чітку міру точності системи [11].

Root Mean Squared Error (RMSE): Ця метрика обчислює квадратний корінь з середньої квадратичної помилки. Вона надає більшу вагу більшим помилкам, що може бути корисним для виявлення значних відхилень у прогнозах [11].

Використання цих метрик дозволить отримати точні та надійні оцінки ефективності рекомендаційної системи, що є критично важливим для успішної реалізації нашого проєкту. У наступних розділах детально буде описано реалізацію системи, що використовує ці методи, з детальним аналізом їх ефективності.

2.3 Огляд критеріїв оцінювання методів рекомендацій

Для вибору оптимального методу рекомендацій необхідно встановити відповідні критерії оцінювання для систем рекомендацій. Визначені наступні ключові критерії [12]:

- швидкість: Критично важлива для систем, які функціонують у режимі реального часу, оскільки висока швидкість обробки даних забезпечує своєчасне надання рекомендацій;
- точність: Основна характеристика, яка визначає здатність системи точно визначати інтереси та потреби користувачів. Висока точність рекомендацій є критичною для ефективності системи;
- вартість обчислень: Тісно пов'язана зі швидкістю, цей критерій особливо важливий у контексті використання хмарних обчислень, де вартість зазвичай залежить від використання дискового простору, трафіку даних та часу процесора.

Для оцінки критеріїв Швидкість, Точність та Вартість обчислень використовуватиметься спрощений діапазон оцінок: Високий, Середній, Низький [12]. Ці оцінки допоможуть визначити, наскільки добре потенційні методи рекомендацій відповідають встановленим потребам та вимогам. Результати порівняння методів наведено у вигляді таблиці 2.9.

Таблиця 2.9 – Порівняння існуючих методів підвищення точності рекомендацій (сформовано на основі джерела [12])

Методи	Швидкість	Точність	Вартість обчислень
Content-Based Filtering	Середня	Висока	Висока
Нейронні мережі та глибоке навчання	Середня	Середня	Висока
Кластеризація	Висока	Низька	Середня
Колаборативна фільтрація	Середня	Висока	Середня

На основі встановлених критеріїв оцінювання можна зробити висновок, що колаборативна фільтрація є оптимальним методом для реалізації нашої рекомендаційної системи. Вона забезпечує високу точність при середній швидкості та вартості обчислень, що робить її найбільш підходящою для задоволення вимог нашої системи.

2.4 Вибір критеріїв для оцінки виконавця ІТ-проєкту

Для оцінки того, наскільки добре кандидат підходить для нового ІТ-проєкту, було вирішено використовувати складнішу методичку, яка включає оцінки з попередніх проєктів за кількома складовими. Цей підхід дозволяє об'єктивно оцінити компетентність та надійність виконавців на основі їхньої минулої продуктивності.

Для розрахунку оцінок попередніх проєктів використовуються наступні кроки:

- збір даних: Збираються оцінки кандидатів за всі виконані проєкти. Оцінки включають такі складові: Technical Skill, Productivity, Product Contribution, Soft skills [13]. Кожен проєкт має свої коефіцієнти для цих складових (у сумі 1);

- розрахунок оцінок за складовими: Для кожного кандидата обчислюються середні оцінки за всіма попередніми проєктами для кожної складової;

- розрахунок загального балу: Загальний бал для кандидата на новий проєкт розраховується як сума оцінок за складовими, помножених на відповідні коефіцієнти для цього проєкту;

Формула 1.1 розрахунку загального балу виглядає наступним чином:

$$\text{Загальний бал} = \sum(\text{Оцінка складової} \times \text{Коефіцієнт складової}), \quad (1.1)$$

Ми можемо використати оцінки попередніх проєктів як критерій для порівняння ефективності рекомендацій, отриманих за допомогою колаборативної фільтрації та без неї. Це дозволяє оцінити наскільки точними є рекомендації і як добре вони відображають історичні дані про продуктивність кандидатів.

Використання оцінок попередніх проєктів із складовими та коефіцієнтами як критерію оцінки кандидатів є ефективним інструментом для підбору виконавців у нові ІТ-проєкти. Цей підхід забезпечує об'єктивність, прозорість та врахування історичних даних, що сприяє більш точному та надійному відбору.

3 РОЗРОБКА МЕТОДУ ДЛЯ ДОДАТКУ ДЛЯ ВНУТРІШНЬОЇ СИСТЕМИ ІТ-КОМПАНІЇ

3.1 Пропонування методу для підбору виконавців ІТ-проєкту з покращеннями

Для розробки прототипу рекомендаційної системи було обрано метод колаборативної фільтрації як основний підхід до аналізу та рекомендацій. Колаборативна фільтрація ефективно використовує дані про фахівців та проєктів що вони реалізовували для прогнозування та рекомендації. Однак, щоб підвищити точність та релевантність рекомендацій, було вирішено доповнити базовий метод методом колаборативної фільтрації (рис 2).

В сучасних ІТ-компаніях підбір персоналу для проєктів це складний та багатоетапний процес, що включає декілька важливих етапів: збір даних про кандидатів, аналіз цих даних, оцінка відповідності кандидатів вимогам проєкту та прийняття рішення про найм. Сучасний процес підбору має певні недоліки, зокрема високу трудомісткість та суб'єктивність, що можуть призводити до неоптимального вибору кандидатів. У цьому підрозділі буде розглянуто існуючий процес підбору персоналу та запропоновано вдосконалений метод, заснований на колаборативній фільтрації.

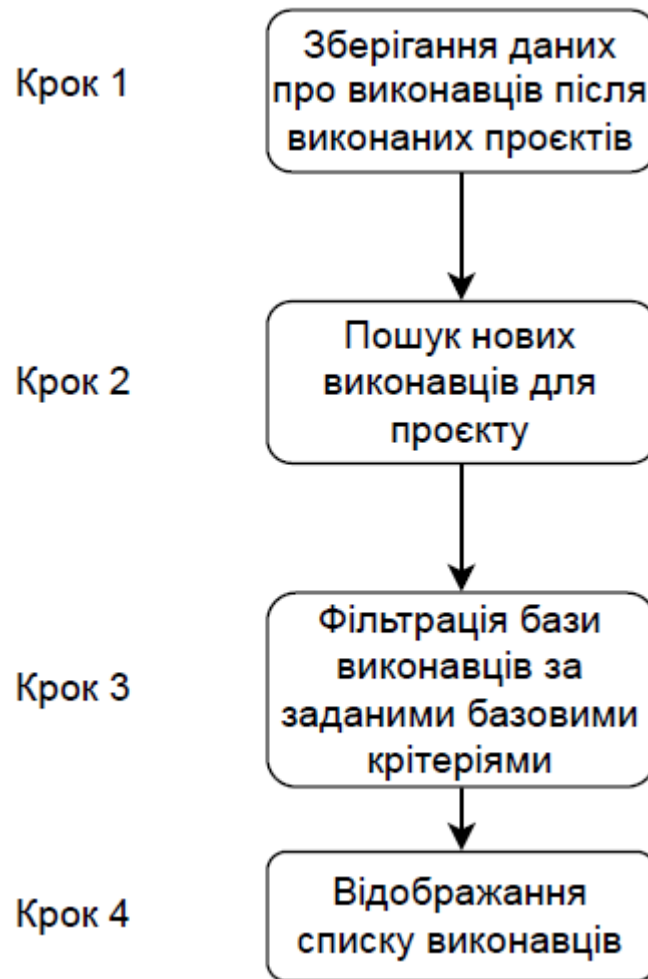


Рисунок 1 – Існуючий процес підбору виконавців

На рисунку 1 зображено існуючий процес підбору виконавців для ІТ-проєкту, який складається з наступних кроків:

- крок 1. Зберігання даних про виконавців після виконаних проєктів: всі дані про завершені проєкти зберігаються у базі даних. Це включає інформацію про учасників, їхні ролі, навички та результати проєкту;
- крок 2. Пошук нових виконавців для проєкту: коли з'являється новий проєкт, починається пошук відповідних кандидатів;
- крок 3. Фільтрація бази виконавців за базовими критеріями: кандидати фільтруються за основними критеріями, такими як наявність певних навичок, досвід роботи, місце розташування тощо;

– крок 4. Відображення списку виконавців: після фільтрації користувачам показується список кандидатів, які відповідають заданим критеріям;

Основні недоліки сучасного підходу включають:

– суб'єктивність. Рішення про найм часто базуються на суб'єктивному оцінюванні інтерв'юерів;

– обмежена точність. Традиційні методи підбору не завжди дозволяють повною мірою оцінити реальний внесок потенційних учасників та їхній практичний досвід.

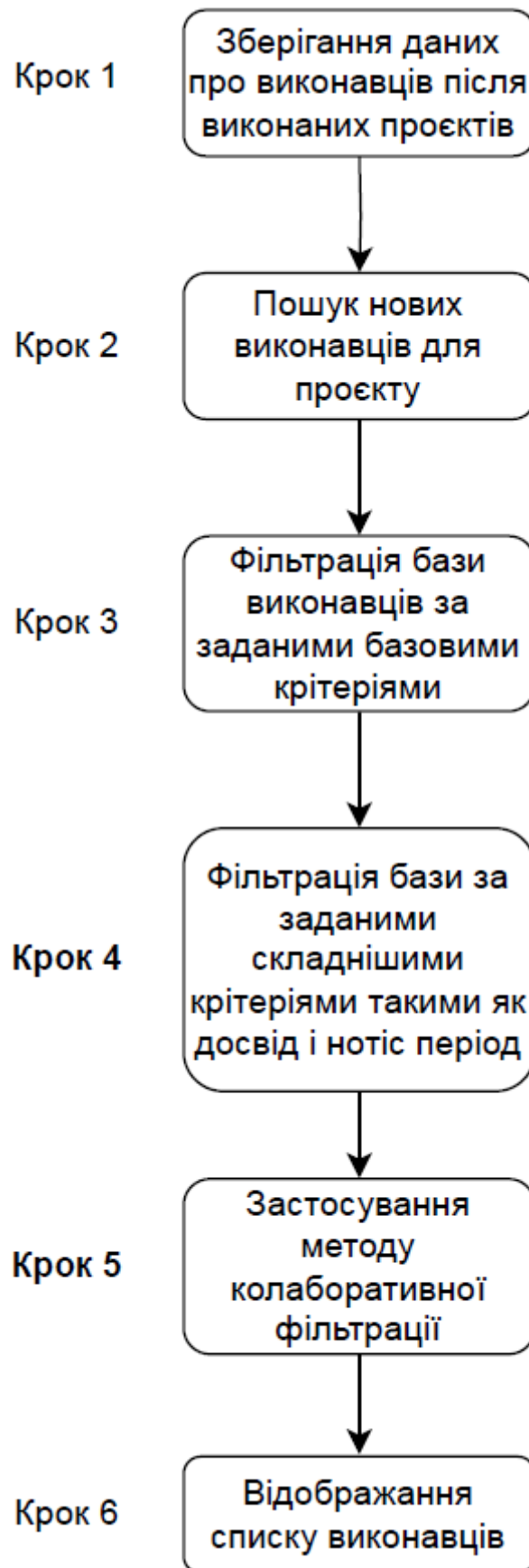


Рисунок 2 - Покращений процес підбору виконавців

На рисунку 2 зображено вдосконалений метод підбору виконавців для ІТ-проектів з використанням колаборативної фільтрації, який включає наступні кроки:

- крок 1. Зберігання даних після виконаних проектів як і в сучасному процесі, всі дані про завершені проекти зберігаються у базі даних;
- крок 2. Пошук нових виконавців для проекту починається пошук відповідних кандидатів після визначення нових потреб проекту;
- крок 3. Фільтрація бази виконавців за базовими критеріями перший етап фільтрації виконується за основними критеріями, такими як наявність певних навичок, досвід роботи, місце розташування тощо;
- крок 4. Фільтрація бази виконавців за складнішими критеріями додатковий етап фільтрації виконується за складнішими критеріями, такими як досвід у конкретних проектах, результати попередніх проектів, технічні навички, продуктивність, внесок у продукт та soft skills;
- крок 5. Застосування методу колаборативної фільтрації на основі зібраних даних та складних критеріїв застосовується метод колаборативної фільтрації, який аналізує попередні взаємодії та переваги користувачів, щоб надати більш релевантні рекомендації;
- крок 6. Відображення результату після всіх етапів фільтрації та застосування колаборативної фільтрації користувачам показується список кандидатів, які найбільш відповідають потребам проекту.

Запропонований метод має наступні переваги:

- вища точність. Колаборативна фільтрація дозволяє аналізувати історичні дані та взаємодії для покращення точності рекомендацій;
- ефективність. Автоматизація процесу знижує трудомісткість та прискорює підбір кандидатів.

Вдосконалений процес підбору персоналу, заснований на колаборативній фільтрації, сприяє створенню більш ефективної та адаптивної системи, здатної давати більш точні рекомендації. Це забезпечує

високоякісний підбір персоналу, що значно підвищує шанси на успіх ІТ-проектів.

3.2 Необхідні дані для роботи метода та способи їх перенесення в систему підбору виконавців ІТ-проекту

Для успішної роботи системи рекомендацій виконавців ІТ-проектів, що використовує метод колаборативної фільтрації, необхідно мати доступ до значного обсягу даних. Ці дані забезпечують можливість точної та релевантної фільтрації кандидатів на основі їх попереднього досвіду, навичок та результатів.

Основні категорії даних, які повинні бути доступні системі, включають:

- репозиторій проектів компанії: опис проектів, тривалість, технології, що використовувалися, та результати. Конкретні ролі, які виконували учасники проектів, та задачі, які вони вирішували;
- дані про працівників компанії: інформація про технічні навички, попередній досвід, освіту, сертифікації та спеціалізації. Історія участі в проєктах, виконувані ролі, успішність виконання завдань;
- фідбеки та оцінки: оцінки роботи в проєктах, зворотний зв'язок про професійні та міжособистісні якості. Формальні та неформальні оцінки, що відображають загальну ефективність та надійність працівників;
- інші дані: географічне розташування працівників та інші релевантні дані, що можуть впливати на процес підбору, такі як доступність, мовні навички тощо.

Перенесення даних з існуючих баз даних компанії до нової системи рекомендацій може бути здійснено кількома способами. Нижче пропонуються основні методи, які можуть бути використані:

- міграції баз даних: налаштування підключення до існуючих баз даних компанії та міграція даних безпосередньо через SQL-запити або спеціальні інструменти для міграції баз даних або ж використання скриптів на мовах програмування (наприклад, Python, Java) для автоматизації процесу міграції даних з однієї бази даних в іншу;
- перенесення даних через CSV файли: експорт даних з існуючих баз даних у формат CSV, що є універсальним та легко читаємим. Багато сучасних баз даних та фреймворків підтримують імпорт даних з CSV файлів, що спрощує процес міграції;
- веб-сервіси та API: використання API для отримання даних в режимі реального часу з існуючих систем компанії. Це дозволяє оновлювати дані безпосередньо та забезпечує актуальність інформації.

Вибір конкретного методу перенесення даних залежить від обсягу даних, технічних можливостей компанії та вимог до швидкості та точності міграції. У багатьох випадках може бути доцільним комбінувати декілька методів для забезпечення максимально ефективного перенесення даних.

У цьому розділі було розглянуто вдосконалений метод підбору виконавців для IT-проектів, який включає етапи фільтрації кандидатів за критеріями, а також застосування колаборативної фільтрації для надання більш релевантних рекомендацій. Запропонований метод дозволяє підвищити об'єктивність та точність процесу відбору кандидатів, що сприяє більш ефективному управлінню проектами та досягненню високих результатів.

Наступним кроком є детальний опис практичного застосування запропонованого методу. Буде розглянуто, як саме реалізувати цю

методологію в реальних умовах, описано архітектуру системи та наведено приклад її використання на практиці.

4 ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДУ ДЛЯ ПОКРАЩЕННЯ РЕКОМЕНДАЦІЙ ВИКОНАВЦІВ ІТ-ПРОЄКТУ

4.1 Загальний опис пропонованої системи для реалізації запропонованого метода

Система рекомендацій, яку планується розробити, спрямована на оптимізацію процесу підбору персоналу для ІТ-проєктів з використанням методу колаборативної фільтрації. Система буде включати два основних інтерфейсних екрани, які забезпечать ефективно та інтуїтивно зрозуміле користування:

Екран логіну:

- мета: екран логіну призначений для безпечного доступу до системи. Він вимагатиме від користувачів ввести свої облікові дані, такі як ім'я користувача та пароль;
- особливості: будуть включені засоби безпеки, такі як хешування паролів та можливість відновлення доступу для забезпечення захисту персональних і фірмових даних.

Екран фільтрації виконавців:

- мета: цей екран буде ядром системи, де користувачі можуть застосовувати різноманітні фільтри для відбору потенційних виконавців для ІТ-проєктів. Функціональність буде базуватися на колаборативній фільтрації, дозволяючи користувачам вибирати кандидатів на основі аналізу подібності між поточними проєктними вимогами та історичними даними про виконані проєкти і персонал;

– особливості: користувачі зможуть задавати складні критерії фільтрації, такі як технічні навички, досвід роботи у певних проєктах, технологічні стеки, а також часові періоди досвіду. Система надаватиме динамічне ранжування кандидатів згідно з релевантністю до заданих критеріїв.

Ця система буде спроектована так, щоб бути швидкою, точною і ефективною у виборі найкращих кандидатів для конкретних ІТ-проєктів, мінімізуючи ручну роботу та покращуючи якість підбору персоналу.

4.2 Детальний опис екранів додатку що реалізує запропонований метод

Екран логіну (рис. 3) є першим кроком користувача в інтеракції з системою рекомендацій персоналу для ІТ-проєктів. Цей інтерфейс призначений для забезпечення безпеки доступу до системи та її функціональностей.

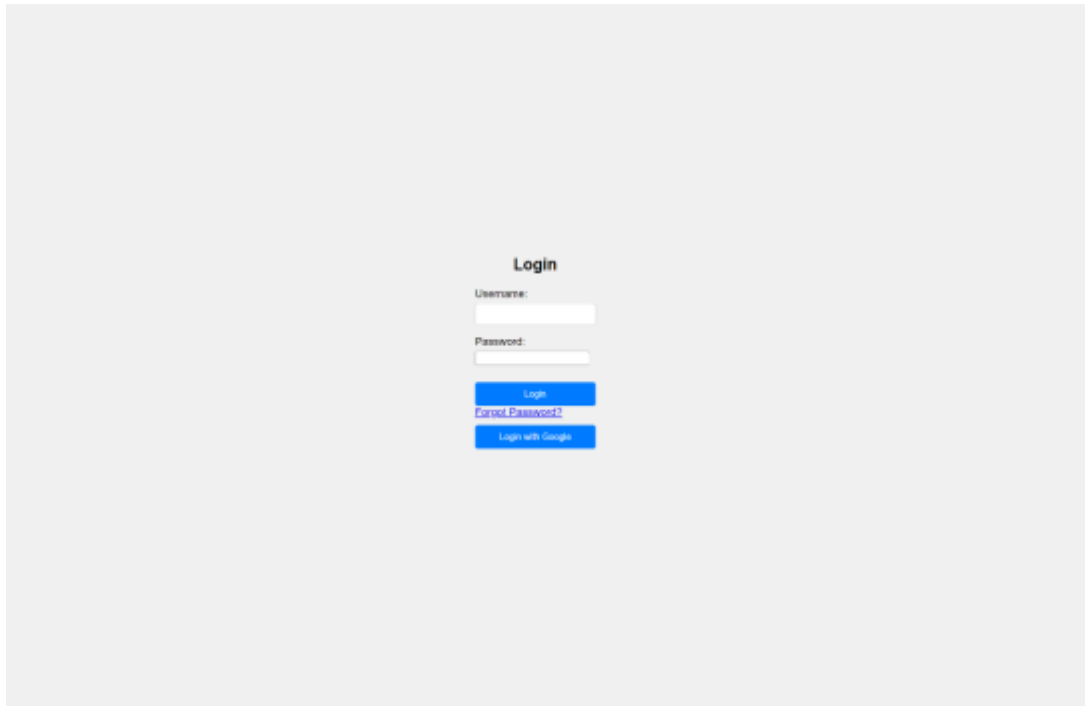


Рисунок 3 – Екран логіну

Ось основні компоненти та функції, які будуть реалізовані на екрані логіну:

- поля введення: користувачі повинні ввести своє ім'я користувача та пароль. Поля будуть захищені з використанням сучасних методів шифрування, щоб забезпечити конфіденційність даних;
- кнопка входу: після введення облікових даних, користувачі натискатимуть кнопку входу для перевірки та авторизації в системі;
- відновлення паролю: у випадку, якщо користувач забуде свій пароль, буде надана можливість його відновлення через електронну пошту. Користувачам буде запропоновано ввести свою електронну адресу, на яку надійде лист з інструкціями для відновлення доступу;
- безпека: система використовуватиме безпечні протоколи аунтифікації та авторизації, забезпечуючи захист інформації від несанкціонованого доступу;

– вхід через сторонні сервіси: користувачі матимуть змогу використовувати свої облікові записи Google для швидкого входу в систему. Це не тільки спрощує процес логіну, але й забезпечує додатковий рівень безпеки, оскільки ці платформи мають потужні засоби аутентифікації та безпеки.

Цей екран логіну розроблено з міркувань зручності користувачів та високого рівня безпеки, щоб забезпечити надійний та ефективний доступ до системи рекомендацій. Використання передових технологій та інтеграція з популярними платформами дозволяє зробити процес логіну не тільки безпечним, але й максимально зручним для кінцевого користувача.

Екран фільтрації (рис. 4) виконавців є ключовим компонентом системи, де користувачі мають змогу визначити і відібрати потенційних кандидатів для участі в ІТ-проектах.

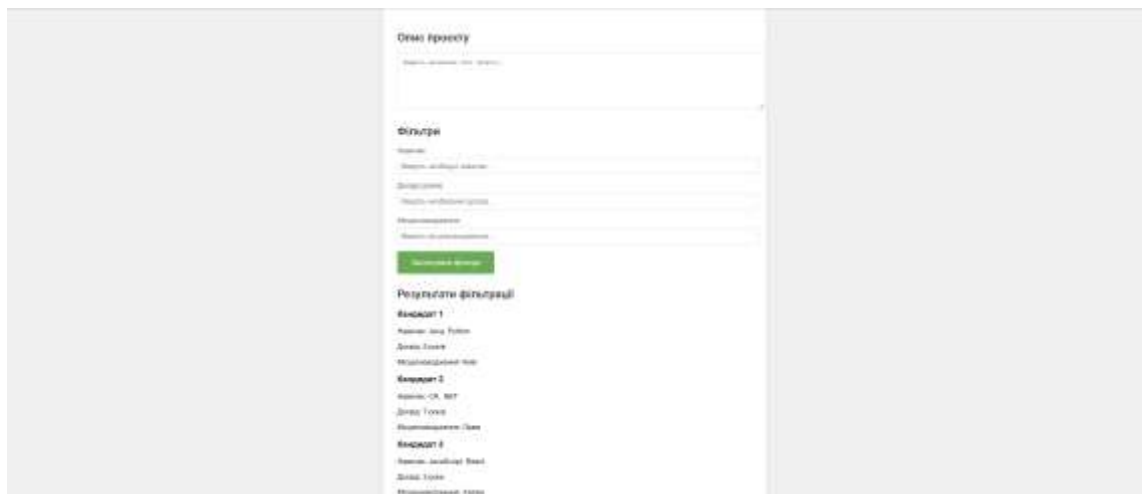


Рисунок 4 – Екран фільтрації

Цей інтерфейс включає наступні елементи та функціональні можливості:

- поле вводу: користувачі можуть ввести детальний опис проєкту, для якого шукають виконавців. Це може включати інформацію про технологічні вимоги, тривалість проєкту, основні задачі та інші ключові деталі, які допомагають у відборі кандидатів;
- збереження інформації: введена інформація зберігається в системі, щоб можна було використовувати її для автоматичного підбору кандидатів згідно з вимогами проєкту;
- налаштування фільтрів: користувачі можуть застосувати різні фільтри, такі як технічні навички, досвід роботи в певних проєктах, знання технологічних стеків, і так далі. Це дозволяє точніше відповідати вимогам проєкту до кваліфікації виконавців;
- інтерактивний вибір: фільтри можуть бути легко змінені або налаштовані згідно з поточними потребами, що дозволяє користувачам динамічно адаптувати процес пошуку;
- під капотом: для обробки введеної інформації та фільтрів система використовує метод колаборативної фільтрації, який аналізує існуючі дані про виконавців та їхній внесок у минулі проєкти;
- результати фільтрації: на основі обраного методу, система пропонує список кандидатів, які найбільш точно відповідають встановленим критеріям. Цей список може бути переглянутий користувачами, які можуть подальше редагувати фільтри або вибирати кандидатів для співбесіди.

Інтерфейс екрану фільтрації розроблено таким чином, щоб максимально спростити і оптимізувати процес підбору персоналу для ІТ-проєктів.

4.3 Загальний опис запропонованої системи для реалізації запропонованого метода

У процесі розробки рекомендаційної системи, яка базується на колаборативній фільтрації, правильний вибір технологічного стеку є не просто важливим, але вирішальним фактором, що впливає на цілий ряд критичних аспектів проєкту. Від вибору технологій залежать не тільки базові можливості та загальна ефективність системи, але й швидкість її розробки, зручність подальшої підтримки, можливості масштабування та адаптивність до змінюваних вимог бізнесу та користувачів.

У виборі компонентів для такої системи необхідно врахувати багато факторів, включаючи сумісність технологій, їхню взаємодію, продуктивність, надійність, а також доступність кваліфікованих розробників. Технологічний стек повинен бути добре адаптованим до великих обсягів даних та складних алгоритмів обробки, які є типовими для систем колаборативної фільтрації. Водночас, важливо враховувати можливість інтеграції з іншими системами та даними, а також забезпечення високої доступності та відмовостійкості.

Крім технічних аспектів, важливо врахувати економічну ефективність обраних рішень: вартість ліцензій, обладнання, підтримки та оновлень. Оптимальний технологічний стек дозволить не тільки ефективно розв'язати поставлені задачі, але й оптимізувати витрати та забезпечити легкість внесення змін і масштабування системи у відповідь на зростаючі обсяги даних чи зміну бізнес-процесів.

В цьому розділі буде детально розглянуто конкретні технології та інструменти, які були обрані для реалізації нашої рекомендаційної системи. Основну увагу буде приділено мовам програмування, базам даних, фреймворкам для обробки великих даних, а також іншим інструментам, що забезпечують потрібні можливості для ефективної роботи системи. Також

буде розглянуто особливості кожної технології, її переваги та потенційні виклики в контексті специфічних задач.

При розгляді мов програмування для розробки рекомендаційної системи, особливу увагу слід приділити вибору мов для різних компонентів системи: бекенду, фронтенду та десктопної версії. Вибір мови програмування впливає на продуктивність, масштабованість, доступність розробників та інтеграцію з іншими системами та інструментами.

Порівняльні таблиці технологій для бекенду:

Таблиця 4.1 – Порівняння переваг та недоліків Java для бекенду (сформовано на основі джерела [14])

Переваги	Недоліки
Java	
Масштабованість: Java ефективно використовується в великих розподілених системах і підтримує високу продуктивність з можливістю обробки великих обсягів даних.	Висока вимогливість до пам'яті: програми на Java можуть вимагати більше оперативної пам'яті порівняно з іншими мовами.
Багата екосистема: велика кількість бібліотек і фреймворків, таких як Spring і Hibernate, полегшує розробку складних додатків.	Швидкість запуску: через те, що Java-додатки вимагають для виконання запуску JVM вони можуть мати повільний старт.
Багатопоточність: вроджена підтримка багатопоточності дозволяє ефективно використовувати апаратні ресурси.	

Таблиця 4.2 – Порівняння переваг та недоліків Rust для бекенду (сформовано на основі джерела [15])

Переваги	Недоліки
Rust	
Безпека пам'яті: Rust запобігає помилкам, пов'язаним із управлінням пам'яттю, без потреби в сміттєзбиранні.	Крута крива навчання: управління власністю і borrowing може бути складним для освоєння новим розробникам.
Багатопоточність: Rust розроблено з можливостями безпечної багатопоточності, що знижує ризик помилок у роботі з даними.	Обмежена екосистема: хоча екосистема Rust розвивається швидко, вона все ще значно менш насичена порівняно з Java або C#.
Висока продуктивність: забезпечує продуктивність, порівнянну з C/C++, що робить його відмінним вибором для вимогливих систем.	

Таблиця 4.3 – Порівняння переваг та недоліків C# для бекенду (сформовано на основі джерела [16])

Переваги	Недоліки
C#	
Інтегрованість з .NET Framework: можливість створення надійних та швидких веб-додатків з використанням ASP.NET.	Залежність від Microsoft: найкраще працює на Windows, хоча .NET Core розширює можливості на Linux та macOS.

Кінець таблиці 4.3

Переваги	Недоліки
Багатий набір інструментів: інтеграція з Visual Studio надає потужні інструменти для розробки та налагодження.	Пам'ять та продуктивність: може бути менш продуктивним порівняно з мовами, які мають безпосередній доступ до апаратних ресурсів.

Таблиця 4.4 – Порівняння переваг та недоліків Go (Golang) для бекенду (сформовано на основі джерела [17])

Переваги	Недоліки
Go (Golang)	
Простота та швидкість розробки: Мова має простий синтаксис і забезпечує швидку компіляцію.	Менш потужні бібліотеки: На деякі задачі можуть не існувати готові рішення, порівняно з більш зрілими мовами, як Java або C#.
Вбудована підтримка конкурентності: Go має вбудовані можливості для легкої реалізації конкурентних операцій за допомогою горути.	
Хороша підтримка мікросервісів: Чудово підходить для будівництва масштабованих мікросервісних архітектур завдяки своїй ефективності та легкості розгортання.	

Вибір мови для бекенду вимагає зваження всіх цих факторів, щоб забезпечити баланс між продуктивністю, розробкою, масштабованістю та підтримкою в майбутньому.

Порівняльні таблиці технологій для фронтенду:

Таблиця 4.5 – Порівняння переваг та недоліків JavaScript для фронтенду (сформовано на основі джерела [18])

Переваги	Недоліки
JavaScript	
Універсальність: JavaScript є стандартною мовою для розробки веб-додатків, підтримуваною всіма основними браузерами.	Проблеми з безпекою: JavaScript схильний до різних видів атак, таких як cross-site scripting (XSS).
Багата екосистема: Велика кількість бібліотек та фреймворків, таких як React, Angular та Vue.js, що дозволяють створювати багаті і динамічні користувацькі інтерфейси.	Керування станом: Управління станом у великих додатках може бути складним без використання додаткових бібліотек або фреймворків.

Кінець таблиці 4.5

Таблиця 4.6 – Порівняння переваг та недоліків TypeScript для фронтенду (сформовано на основі джерела [19])

Переваги	Недоліки
TypeScript	
Строга типізація: Додає строгу типізацію до JavaScript, що покращує розробку великих додатків і сприяє виявленню помилок на ранніх стадіях.	Крива навчання: Вимагає від розробників знань про типи та інтерфейси, що може бути бар'єром для новачків.

Кінець таблиці 4.6

Переваги	Недоліки
Сумісність з JavaScript: TypeScript повністю сумісний з JavaScript, що дозволяє легко інтегрувати його в існуючі проєкти.	Час компіляції: Програми на TypeScript можуть вимагати більше часу для компіляції порівняно з звичайним JavaScript.
Підтримка з боку розробників: Широка підтримка спільнотою і інструментами розробки.	

Flutter – це фреймворк від Google для створення кросплатформних додатків. Він дозволяє розробляти додатки одночасно для мобільних, веб- та десктопних платформ, використовуючи одну і ту ж базу коду.

Таблиця 4.7 – Порівняння переваг та недоліків Flutter для фронтенду (сформовано на основі джерела [20])

Переваги	Недоліки
Flutter	
Кросплатформеність: Можливість створення додатків для Android, iOS, Windows, Mac, Linux та браузеру з однієї кодової бази.	Розмір додатку: Додатки на Flutter можуть мати більший розмір файлу порівняно з нативними додатками.
Продуктивність: Висока продуктивність завдяки компіляції в машинний код, що забезпечує швидку та плавну роботу інтерфейсів.	Залежність від платформи: Хоча Flutter дозволяє створювати кросплатформені додатки, існування певних платформоспецифічних можливостей може потребувати додаткової інтеграції або налаштування.

Кінець таблиці 4.7

Переваги	Недоліки
Багатий набір віджетів: Велика кількість предвстановлених віджетів, що полегшує створення складних і візуально привабливих користувацьких інтерфейсів.	Спільнота: Хоча спільнота Flutter швидко зростає, вона все ще менша порівняно з спільнотами інших великих фреймворків, таких як React.

Кожна з цих мов і фреймворків має свої сильні та слабкі сторони, і вибір між ними залежить від специфічних вимог проєкту, досвіду команди та інших технічних критеріїв.

Розглянемо популярні бази даних, такі як PostgreSQL, MySQL, Oracle, MSSQL та MariaDB, щоб зрозуміти, як кожна з них може відповідати потребам рекомендаційної системи. Оцінка буде включати плюси та мінуси кожної системи управління базами даних.

Таблиця 4.8 – Порівняння переваг та недоліків PostgreSQL (сформовано на основі джерела [21])

Переваги	Недоліки
PostgreSQL	
Підтримка складних запитів: PostgreSQL відома своєю підтримкою складних SQL запитів і транзакцій, що робить її ідеальною для складних аналітичних додатків, де необхідне глибоке даних маніпулювання.	Високі вимоги до адміністрування: PostgreSQL може вимагати більше зусиль з налаштування та оптимізації для досягнення максимальної продуктивності порівняно з іншими СУБД.

Кінець таблиці 4.8

<p>Високий рівень дотримання стандартів SQL: PostgreSQL часто вважають однією з найбільш стандартно відповідних систем управління базами даних, що спрощує міграцію та інтеграцію з іншими системами.</p>	
<p>Підтримка розширень: Ця СУБД може бути розширена з додатковими функціями, які можуть бути інтегровані в базу даних як модулі.</p>	

Таблиця 4.9 – Порівняння переваг та недоліків MySQL (сформовано на основі джерела [22])

Переваги	Недоліки
MySQL	
<p>Широка підтримка: MySQL підтримується більшістю хостинг-провайдерів, що робить її однією з найпростіших опцій для розгортання веб-додатків.</p>	<p>Обмеження у підтримці транзакцій: Версії MySQL до введення InnoDB мали обмежену підтримку транзакцій, що могло бути критичним для додатків, які потребують сильної цілісності даних.</p>
<p>Велика спільнота: Завдяки своїй популярності, MySQL має велику спільноту розробників, що забезпечує гарну документацію і</p>	<p>Менш гнучка у порівнянні з PostgreSQL: MySQL може бути менш гнучкою в певних аспектах, таких як підтримка складних запитів</p>

підтримку.	або нетрадиційних типів даних.
------------	--------------------------------

Таблиця 4.10 – Порівняння переваг та недоліків Oracle (сформовано на основі джерела [23])

Переваги	Недоліки
Oracle	
Підтримка великих обсягів даних: Oracle оптимізована для роботи з величезними обсягами даних та високонавантаженими додатками, що робить її вибором для великих корпорацій.	Вартість: Одна з найбільш дорогих СУБД на ринку.
Розширені можливості безпеки: Oracle надає комплексні можливості для забезпечення безпеки даних, включаючи шифрування, аудит та контроль доступу.	Складність: Oracle має високу криву навчання через складність своїх функцій та інструментів.
	Залежність від вендора: Використання Oracle може призвести до залежності від одного постачальника, що може обмежувати гнучкість і збільшувати вартість.

Кінець таблиці 4.10

Таблиця 4.11 – Порівняння переваг та недоліків MSSQL (сформовано на основі джерела [24])

Переваги	Недоліки
MSSQL	
Інтегровані бізнес-рішення: Підтримка SQL Server Reporting Services та SQL Server Analysis Services забезпечує потужні можливості для бізнес-аналітики.	Висока вартість володіння: Ліцензування MSSQL може бути значно дорожчим, особливо для великих обсягів даних або великої кількості користувачів.
Оптимізація під Windows: SQL Server оптимізовано для роботи з Windows, що може забезпечити підвищену продуктивність на цій платформі.	Обмежена крос-платформеність: Хоча останні версії SQL Server покращили підтримку Linux, їхня функціональність все ще може бути обмеженою порівняно з версіями для Windows.

Кінець таблиці 4.11

Таблиця 4.12 – Порівняння переваг та недоліків MariaDB (сформовано на основі джерела [25])

Переваги	Недоліки
MariaDB	
Висока сумісність з MySQL: MariaDB забезпечує високу сумісність з MySQL, що дозволяє легко мігрувати системи без переписування застосунків.	Менша підтримка у порівнянні з MySQL: Хоча MariaDB має активну спільноту, вона все ще не може зрівнятися з рівнем підтримки, який надається корпорацією Oracle для MySQL.
Покращення продуктивності: MariaDB вносить численні покращення в	Ризики в майбутніх розробках: Оскільки MariaDB відокремилася від

продуктивність та доступність функцій, яких немає в стандартній MySQL.	MySQL, існує потенційний ризик розбіжностей у майбутніх функціях і сумісності.
--	--

Ретельний вибір бази даних має вирішальне значення для забезпечення стабільності, продуктивності та масштабованості рекомендаційної системи, а також для оптимізації витрат і ресурсів.

У контексті рекомендаційних систем, кешування відіграє критичну роль у забезпеченні високої продуктивності та миттєвої відповіді користувачам, особливо при великих обсягах даних та великій кількості запитів. Нереляційні бази даних, такі як MongoDB, Redis та Memcached, є популярними рішеннями для кешування завдяки їхнім унікальним характеристикам та оптимізації для швидкого доступу до даних. Переваги та недоліки кожної в наступних таблицях:

Таблиця 4.13 – Порівняння переваг та недоліків MongoDB (сформовано на основі джерела [26])

Переваги	Недоліки
MongoDB	
Гнучка схема даних: MongoDB використовує формат BSON, що дозволяє зберігати документи без строгої схеми. Це робить MongoDB ідеальним для динамічних даних і швидкої ітерації розробки.	Використання пам'яті: MongoDB може використовувати значний обсяг оперативної пам'яті для зберігання робочих наборів даних, що може стати проблемою в обмежених ресурсах.

Таблиця 4.14 – Порівняння переваг та недоліків Redis (сформовано на основі джерела [27])

Переваги	Недоліки
Redis	
Швидкість: Redis є базою даних в оперативній пам'яті, що забезпечує швидкий доступ до даних. Він ідеально підходить для реалізації високопродуктивних кешів.	Обмеження пам'яті: Оскільки дані зберігаються в оперативній пам'яті, обсяг даних, який може бути ефективно оброблений, обмежений доступною пам'яттю.
Структури даних: Redis підтримує різноманітні структури даних, такі як рядки, списки, множини, хеші та черги, що робить його дуже гнучким у використанні.	
Підтримка транзакцій: Redis підтримує операції, які можуть бути виконані атомарно, що є корисним для забезпечення цілісності даних.	

Кінець таблиці 4.14

Таблиця 4.15 – Порівняння переваг та недоліків Memcached (сформовано на основі джерела [28])

Переваги	Недоліки
Memcached	
Простота: Memcached легко налаштовується і використовується,	Обмежені функціональні можливості: Не підтримує стільки

забезпечуючи швидке зберігання ключ-значення в пам'яті.	структур даних, як Redis, і має менші можливості управління даними.
---	---

Кінець таблиці 4.15

Переваги	Недоліки
Зниження навантаження на базу даних: Часто використовується для зниження навантаження на реляційні бази даних шляхом кешування результатів запитів.	Відсутність постійності: Memcached не підтримує постійність, тому втрата пам'яті сервера може призвести до втрати даних.

Вибір відповідної бази даних для кешування в рекомендаційних системах залежить від специфічних вимог до швидкості, гнучкості управління даними та обсягу пам'яті. Кожна з наведених технологій має свої переваги та обмеження, які слід розглядати при їх виборі.

Хмарні платформи пропонують різні сервіси та ресурси для розгортання, управління та масштабування додатків. Вибір правильної хмарної платформи може суттєво вплинути на продуктивність, вартість та успіх проєкту що буде використовувати запропонований метод. Буде розглянуто п'ять популярних хмарних сервісів.

Таблиця 4.16 – Порівняння переваг та недоліків AWS (Amazon Web Services) (сформовано на основі джерела [29])

Переваги	Недоліки
AWS (Amazon Web Services)	
Різноманітність сервісів: AWS	Складність ціноутворення:

пропонує один з найширших асортиментів обчислювальних ресурсів та сервісів, включаючи EC2 для віртуальних серверів, S3 для зберігання даних, Lambda для виконання коду без серверів.	Модель ціноутворення AWS може бути складною через велику кількість опцій і можливих доплат.
--	---

Кінець таблиці 4.16

Переваги	Недоліки
Масштабованість: AWS підтримує масштабування від невеликих до найбільших корпоративних вимог, дозволяючи компаніям адаптувати ресурси відповідно до своїх потреб.	Комплексність інтерфейсу: Багатофункціональний інтерфейс може бути перевантаженим для новачків або малого бізнесу.
Надійність: Використання глобальної мережі дата-центрів забезпечує високу доступність і надійність сервісів.	

Таблиця 4.17 – Порівняння переваг та недоліків Oracle Cloud (сформовано на основі джерела [30])

Переваги	Недоліки
Oracle Cloud	
Спеціалізація на базах даних: Oracle пропонує рішення, які оптимізовані для високопродуктивних баз даних, забезпечуючи передові можливості для управління даними.	Відносно обмежений вибір сервісів: Не пропонує такий широкий спектр хмарних сервісів, як деякі конкуренти.
Підтримка Enterprise рівня:	Фокус на корпоративних

Забезпечує рішення, які є ідеальними для великих корпорацій з складними вимогами до даних і застосунків.	користувачах: Може бути менш привабливим для стартапів або малих компаній через високу вартість та складність продуктів.
--	--

Таблиця 4.18 – Порівняння переваг та недоліків Azure (сформовано на основі джерела [31])

Переваги	Недоліки
Azure	
Інтеграція з Windows: Azure легко інтегрується з Windows Server, SQL Server, і .NET, що робить її ідеальною для компаній, що використовують Microsoft продукти.	Складність налаштувань: Нові користувачі можуть зіткнутися з викликами при налаштуванні і використанні Azure через складність платформи.
Гібридні можливості: Azure підтримує гібридні розгортання, що дозволяє компаніям поєднувати хмарні та локальні ресурси.	Залежність від інших продуктів Microsoft: Найбільшу вигоду отримують користувачі, що вже вкладені в екосистему Microsoft.

Таблиця 4.19 – Порівняння переваг та недоліків Google Cloud Platform (GCP) (сформовано на основі джерела [32])

Переваги	Недоліки
Google Cloud Platform (GCP)	
Інновації в області машинного навчання: GCP є лідером в наданні сервісів машинного навчання і штучного інтелекту, з потужними API для розробки.	Крута крива навчання: Інноваційні технології можуть вимагати спеціальних знань для ефективного використання.
Аналітика даних: Забезпечує	Менший вибір географічних

передові рішення для аналізу великих даних, включаючи BigQuery.	регіонів: В порівнянні з AWS і Azure, GCP має меншу кількість дата-центрів.
---	---

Таблиця 4.20 – Порівняння переваг та недоліків DigitalOcean (сформовано на основі джерела [33])

Переваги	Недоліки
Google Cloud Platform (GCP)	
Простота та чистота: DigitalOcean пропонує інтуїтивно зрозумілий інтерфейс та просту модель ціноутворення, що є ідеальним для розробників та малих команд.	Функціональні обмеження: Не пропонує такої широти сервісів, як більші провайдери.
Спільнота: Має активну спільноту з багатьма навчальними матеріалами та підтримкою.	Ресурсні обмеження: Може не вистачати опцій для великих або дуже навантажених додатків.

На основі проведеного аналізу було обрано технологічний стек для реалізації рекомендаційної системи, що включає наступні компоненти:

- мова програмування: Java для бекенду, яка забезпечує високу масштабованість, продуктивність та багату екосистему бібліотек і фреймворків;
- фронтенд: Flutter для створення кросплатформних додатків, що дозволяє розробляти єдиний додаток для десктопу, вебу та мобільних пристроїв;
- бази даних: PostgreSQL для зберігання основних даних про виконавців, проекти та оцінки;

- Redis для кешування даних і забезпечення швидкого доступу до них;
- Oracle Cloud Infrastructure для забезпечення високої надійності, безпеки та масштабованості системи.

Цей вибір технологій забезпечить високу продуктивність, надійність та зручність у розробці та підтримці системи. Використання Java для бекенду та Flutter для фронтенду дозволить створити ефективну та гнучку систему, яка відповідає сучасним вимогам до IT-проєктів. PostgreSQL та Redis забезпечать надійне зберігання та швидкий доступ до даних, а Oracle Cloud Infrastructure надасть потужну хмарну платформу для розгортання та масштабування системи.

4.4 Порівняння результатів підбору кандидатів з використанням колаборативної фільтрації та без неї

Для оцінки ефективності використання колаборативної фільтрації ми порівняли результати підбору кандидатів для нового проєкту з використанням колаборативної фільтрації та без неї.

Таблиця 4.21 – Порівняння результатів підбору виконавців IT-проєкту

	Результати без використання колаборативної фільтрації	Результати з використанням колаборативної фільтрації
Кандидат	Executor1	Executor12
Навички	Java, Python	Java, Rust
Досвід	5 років	5 років

Середній бал попередніх проєктів	4.25	4.71
Оцінка для нового проєкту		4.31

Колаборативна фільтрація показала кращі результати у підборі кандидата для нового проєкту, оскільки дозволяє врахувати більш широкий спектр факторів, що впливають на продуктивність кандидата. Хоча середні оцінки попередніх проєктів можуть надати об'єктивну інформацію про продуктивність кандидата, колаборативна фільтрація може врахувати додаткові аспекти, такі як взаємодія з іншими виконавцями, специфічні навички та досвід, релевантні для нового проєкту.

Використання колаборативної фільтрації для підбору кандидатів у IT-проєкти демонструє більшу ефективність порівняно з підходом, заснованим лише на середніх оцінках попередніх проєктів. Колаборативна фільтрація дозволяє врахувати більш широкий спектр факторів і забезпечує точніші прогнози щодо продуктивності кандидатів у нових проєктах.

ВИСНОВКИ

У процесі виконання даної дипломної роботи було досліджено та розроблено метод автоматизації підбору виконавців для IT-проектів з використанням колаборативної фільтрації. Основні досягнення та результати роботи можна підсумувати наступним чином:

Аналіз існуючих методів підбору персоналу: Проведено детальний аналіз традиційних методів підбору виконавців для IT-проектів, включаючи їх переваги та недоліки. Виявлено, що традиційні методи є трудомісткими та суб'єктивними, що може призводити до неоптимального вибору кандидатів.

Огляд сучасних технологій: Розглянуто різні методи оптимізації, включаючи методи на основі вмісту, нейронні мережі, кластеризацію та колаборативну фільтрацію. Визначено, що колаборативна фільтрація є найбільш підходящим методом для розробки рекомендаційної системи, оскільки вона забезпечує високу точність та адаптивність рекомендацій.

Розробка методу оцінки кандидатів: Запропоновано методiku оцінки кандидатів на основі їхніх оцінок за попередні проекти з використанням складових, таких як технічні навички, продуктивність, внесок у проект та м'які навички. Це дозволяє отримати об'єктивну оцінку компетентності виконавців.

Реалізація системи: Розроблено прототип системи рекомендацій, що включає два основних інтерфейсних екрани: екран логіну та екран фільтрації виконавців. Система використовує метод колаборативної фільтрації для надання рекомендацій та дозволяє користувачам задавати складні критерії фільтрації.

Оцінка ефективності методу: Проведено порівняння результатів підбору виконавців з використанням та без використання колаборативної фільтрації. Виявлено, що застосування колаборативної фільтрації значно покращує точність рекомендацій та підвищує об'єктивність процесу відбору кандидатів.

Результати роботи демонструють ефективність запропонованого методу автоматизації підбору виконавців для ІТ-проектів. Використання колаборативної фільтрації дозволяє забезпечити високу точність рекомендацій, знизити суб'єктивність процесу та підвищити загальну ефективність управління проектами. Це сприяє підвищенню продуктивності команд та успішності ІТ-проектів.

Дана кваліфікаційна робота та список літератури були оформлені відповідно до ДСТУ [34] [35].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проектами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2024. – 28 с.
2. Дудка М.В. Дослідження методів автоматизації планування роботи команди виконавців ІТ-проєкту. //28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь ХХІ столітті». Зб. матеріалів форуму. Т.6. Конференція «Інформаційні інтелектуальні системи» – Харків: ХНУРЕ. 2024.
3. Small, Medium, or Large: Does Business Size Matter? Digital Resource. URL: <https://www.yourdigitalresource.com/post/does-business-size-matter> (дата звернення: 29.05.2024).
4. How to Structure and Build an Effective Engineering Team. LinearB. URL: <https://linearb.io/blog/how-to-structure-and-build-an-effective-engineering-team> (дата звернення: 29.05.2024).
5. Моделі роботи рекомендаційних систем. Драга Я.Ю. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2017/11/23.pdf> (дата звернення: 29.05.2024).
6. Using Neural Networks for Your Recommender System. Nvidia Blog URL: <https://developer.nvidia.com/blog/using-neural-networks-for-your-recommender-system> (дата звернення: 29.05.2024).
7. Рекомендаційна система. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D1%96%D0%B9%D0%BD>

D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0
(дата звернення: 29.05.2024).

8. Recommendation System – Understanding The Basic Concepts. Analytics Vidhya. URL: <https://www.analyticsvidhya.com/blog/2021/07/recommendation-system-understanding-the-basic-concepts/> (дата звернення: 29.05.2024).

9. Understanding Deep Learning: exploring its Advantages and Disadvantages. Yantra. URL: <https://www.yantrallp.com/blog/understanding-deep-learning-exploring-its-advantages-and-disadvantages/> (дата звернення: 29.05.2024).

10. The latest in Data and Analytics. Data Rundown. URL: <https://datarundown.com/#topics> (дата звернення: 29.05.2024).

11. Recommender Systems: Machine Learning Metrics and Business Metrics. Neptune.ai. URL: <https://neptune.ai/blog/recommender-systems-metrics> (дата звернення: 29.05.2024).

12. Дослідження методів формування рекомендацій в проєктах побудови рекомендаційних систем в соціальних мережах. Бондар Д.В.. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/f34b559b-8725-4fb7-8453-7dfbd23b4409/content> (дата звернення: 29.05.2024).

13. Engineering Performance Review. Railsware. URL: <https://railsware.com/blog/engineering-performance-review/> (дата звернення: 29.05.2024).

14. 12 pros and cons of Java for your project. EPAM Anywhere. URL: <https://anywhere.epam.com/en/blog/pros-and-cons-java> (дата звернення: 29.05.2024).

15. Rust for Backend Development. MasteringBackend. URL: <https://masteringbackend.com/posts/rust-for-backend-development> (дата звернення: 29.05.2024).

16. Backend Development: A Comprehensive Guide to Server-Side Programming Languages. Stackify. URL: <https://stackify.com/backend->

development-a-comprehensive-guide-to-server-side-programming-languages/
(дата звернення: 29.05.2024).

17. The Pros And Cons Of Programming In Go. Mobcoder. URL: <https://blog.mobcoder.com/golang-pros-and-cons/> (дата звернення: 29.05.2024).

18. The Advantages and Disadvantages of JavaScript. freeCodeCamp. URL: <https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/> (дата звернення: 29.05.2024).

19. JavaScript vs Typescript. svitla. URL: <https://svitla.com/blog/javascript-vs-typescript> (дата звернення: 29.05.2024).

20. Is Flutter Good for Web Development? Here's What You Need to Know. FE MAG. URL: <https://www.frontendmag.com/tips/is-flutter-good-for-web-development/> (дата звернення: 29.05.2024).

21. Techieclues. Techieclues. URL: <https://www.techieclues.com> (дата звернення: 29.05.2024).

22. MS Access Database, Why Choose It? Blue Claw Database Developer Resource. URL: <https://blueclawdb.com> (дата звернення: 29.05.2024).

23. Oracle Database: what it is and how it works. Ionos. URL: <https://www.ionos.com/digitalguide/hosting/technical-matters/oracle-database/> (дата звернення: 29.05.2024).

24. The Advantages and Disadvantages of Microsoft SQL Server. Rothrobot. URL: <https://rothrobot.com/the-advantages-and-disadvantages-of-microsoft-sql-server/> (дата звернення: 29.05.2024).

25. Why MariaDB Is “Better” Than MySQL. Percona. URL: <https://www.percona.com/blog/why-mariadb-is-better-than-mysql/> (дата звернення: 29.05.2024).

26. MongoDB NoSQL Solution: Advantages And Disadvantages. Developer.com. URL: <https://www.developer.com/database/mongodb-advantages-disadvantages/> (дата звернення: 29.05.2024).

27. 13 Advantages And Disadvantages of Redis. ProsConsLab. URL: <https://prosconslab.com/articles/32/13-advantages-and-disadvantages-of-redis> (дата звернення: 29.05.2024).
28. Memcached vs Redis: Choose Your In-Memory Cache. Kinsta. URL: <https://kinsta.com/blog/memcached-vs-redis/> (дата звернення: 29.05.2024).
29. AWS advantages and Disadvantages. upGrad. URL: <https://www.knowledgehut.com/blog/cloud-computing/aws-advantages-and-disadvantages> (дата звернення: 29.05.2024).
30. Oracle Cloud Infrastructure. TrustRadius. URL: <https://www.trustradius.com/products/oracle-cloud-infrastructure/reviews?q=pros-and-cons#product-demos> (дата звернення: 29.05.2024).
31. 15 Microsoft Azure Advantages and Disadvantages. BrandonGaille. URL: <https://brandongaille.com/15-microsoft-azure-advantages-and-disadvantages/> (дата звернення: 29.05.2024).
32. Advantages And Disadvantages of Google Cloud Platform. Ca. URL: <https://cloudtecharena.com/advantages-and-disadvantages-of-google-cloud-platform/> (дата звернення: 29.05.2024).
33. DigitalOcean Hosting Review PROS & CONS (2023) – Excellent Choice For Hosting. KasaReviews. URL: <https://www.kasareviews.com/digitalocean-hosting-review-pros-cons/> (дата звернення: 29.05.2024).
34. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.
35. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.