

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр _____ ННЦЗФН
(повна назва)

Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Розробка вебзастосунку: «Віртуальний помічник для запису на прийом до
лікаря» із реалізацію рекомендаційної системи на основі колаборативної
фільтрації
(тема)

Виконав:
здобувач _____ четвертого року навчання,
групи _____ ІТШЗ-21-1

_____ Антон Коншин
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна

Освітня програма _____ Штучний інтелект

(повна назва освітньої програми)

Керівник _____ ст. викл. Ігор Бібічков
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____
(підпис)

_____ Олег ЗОЛОТУХІН
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Центр _____ ННЦЗФН _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

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

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Коншину Антону Андрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка вебзастосунку: "Віртуальний помічник для запису на прийом до лікаря" із реалізацію рекомендаційної системи на основі колаборативної фільтрації _____

затверджена наказом університету від 7 травня 2025 р. № 80Стз

2. Термін подання студентом роботи до екзаменаційної комісії 25 червня 2025 р.

3. Вихідні дані до роботи анонімізований датасет RateMDs із текстовими відгуками й рейтингами пацієнтів, початкова схема БД та каркас веб-застосунку ASP.NET Core MVC + EF Core, технічні вимоги (Python \geq 3.11, ASP.NET Core \geq 8.0, PostgreSQL \geq 16, Docker 23), нормативна база (методичні вказівки 2024 р., ДСТУ 3008:2015), базові Python-скрипти для ETL-процесу й тренування моделі SVD-based CF, а також граничні вимоги до продуктивності (RMSE \leq 0,9; t-відгуку \leq 500 мс). _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі _____

2) Формування системи даних _____

3) Розробка веб-застосунку та системи колаборативної фільтрації _____

РЕФЕРАТ

Пояснювальна записка: 52 с., 7 рис., 2 табл., 1 дод., 11 джерел.

КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ASP.NET CORE MVC, DOCKER, ENTITY FRAMEWORK, POSTGRESQL, PYTHON, WEB.

Об'єкт дослідження – інтелектуальна система рекомендацій лікарів для пацієнтів, інтегрована у веб-застосунок запису на прийом.

Предмет дослідження – побудова та реалізація системи рекомендацій на основі колаборативної фільтрації, а також створення повнофункціонального веб-застосунку на основі сучасних архітектурних рішень.

Мета роботи – розробити веб-застосунок «Віртуальний помічник для запису на прийом до лікаря» з вбудованим модулем рекомендацій, який використовує історичні дані взаємодій пацієнтів для прогнозування найбільш релевантних спеціалістів.

Методи дослідження – огляд галузі охорони здоров'я, аналіз існуючих систем електронного запису, математичне моделювання алгоритмів колаборативної фільтрації, розробка веб-застосунку та рекомендаційного модуля з використанням сучасних технологій Python, ASP.NET Core MVC, PostgreSQL та Docker.

Анотація – У роботі проаналізовано, спроектовано та реалізовано інформаційну систему, яка поєднує функціональність медичного веб-сервісу із рекомендаційною підсистемою на основі методів колаборативної фільтрації.

ABSTRACT

Bachelor's thesis contains: 52 pp., 7 fig., 2 tabl., 1 ann., 11 references.

ASP.NET CORE MVC, COLLABORATIVE FILTERING, DOCKER, ENTITY FRAMEWORK, POSTGRESQL, PYTHON, RECOMMENDER SYSTEMS, WEB.

Object of research – an intelligent doctor recommendation system for patients integrated into an appointment booking web application.

Subject of research – the design and implementation of a recommendation system based on collaborative filtering, as well as the development of a full-featured web application using modern architectural solutions.

Purpose of the work – to develop a web application «Virtual Assistant for Doctor Appointment Booking» with a built-in recommendation module that utilizes historical patient interaction data to predict the most relevant specialists.

Research methods – review of the healthcare domain, analysis of existing e-booking systems, mathematical modeling of collaborative filtering algorithms, development of the web application and recommendation module using modern technologies including Python, ASP.NET Core MVC, PostgreSQL, and Docker.

Abstract – The project includes the analysis, design, and implementation of an information system that combines the functionality of a medical web service with a recommendation subsystem based on collaborative filtering methods.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз предметної галузі та постановка задачі.....	11
1.1 Стан розвитку веб-застосунків у медичній сфері.....	11
1.2 Еволюція та типологія рекомендаційних систем у медицині	12
1.3 Актуальні проблеми при онлайн-виборі лікаря.....	15
1.4 Передумови побудови інтелектуальної рекомендаційної системи ...	17
1.5 Постановка задачі.....	21
2 Формування системи даних	24
2.1 Збір необроблених даних	24
2.2 Проектування структури даних і бази даних	25
2.1.1 Схема genders	26
2.1.2 Схема міст.....	27
2.1.3 Схема статусів запису.....	28
2.1.4 Схема користувачів.....	28
2.1.5 Схема записів на прийом	29
2.1.6 Схема оцінок.....	30
2.1.7 Схема відгуків	31
2.1.8 Схема рекомендацій	32
2.1.9 Схема подібності користувачів і лікарів	32
2.1.10 Схема зв'язків лікар-клініка	33
2.1.11 Схема specialties	34
2.1.12 Схема clinics	34
2.1.13 Схема doctors	35
2.1.14 Схема appointment_statuses	36
2.3 Обробка текстів	37
3 Розробка веб-застосунку та системи колаборативної фільтрації.....	39
3.1 Реалізація колаборативної фільтрації	39

3.1.1 Теоретичні основи колаборативної фільтрації	39
3.1.2 Принципи колаборативної фільтрації.....	39
3.1.3 Алгоритмічна реалізація	40
3.1.4 Приклад роботи системи	44
3.2 Розробка візуального інтерфейсу	45
Висновки	49
Перелік джерел посилання	50
Додаток А Відомість кваліфікаційної роботи	52

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект;

ASP.NET Core MVC – веб-фреймворк компанії Microsoft для розробки веб-застосунків;

Baseline – базова модель прогнозування оцінок користувачів;

CF – Collaborative Filtering – колаборативна фільтрація;

Docker – програмне забезпечення для розгортання застосунків у контейнерах;

EF Core – Entity Framework Core – легковагий ORM-фреймворк від Microsoft;

NLP – Natural Language Processing – обробка природної мови, галузь штучного інтелекту;

ORM – Object-Relational Mapping – технологія програмування, що дозволяє конвертувати дані між несумісними типами систем.

ВСТУП

В умовах стрімкого розвитку цифрових технологій медична сфера все активніше інтегрує інноваційні рішення з метою підвищення якості обслуговування пацієнтів та ефективності роботи закладів охорони здоров'я. Однією з найперспективніших технологій є впровадження інтелектуальних рекомендаційних систем, здатних надавати персоналізовані поради щодо вибору лікаря. У зв'язку зі зростанням обсягів відкритих даних про медичні послуги, пацієнтів та відгуки, з'являється можливість використовувати алгоритми машинного навчання, зокрема колаборативної фільтрації, для створення високоточних систем рекомендацій. Це дозволяє значно зменшити суб'єктивність у виборі лікаря, скоротити час на пошук відповідного фахівця та підвищити рівень задоволеності користувача.

Сучасні веб-застосунки для запису до лікаря зазвичай надають базову інформацію про фахівців, фільтрацію за спеціалізацією та рейтингом, але рідко включають глибоку персоналізацію. Застосування колаборативної фільтрації дозволяє подолати це обмеження шляхом аналізу схожих вподобань інших пацієнтів. Така система не тільки полегшує вибір лікаря на основі колективного досвіду, але й має здатність адаптуватися до змін у поведінці користувачів та поповнення бази даних новими записами.

Метою цієї кваліфікаційної роботи є розробка інтелектуального веб-застосунку «Віртуальний помічник для запису на прийом до лікаря», основною функцією якого є рекомендаційна система на основі колаборативної фільтрації. Завданнями роботи є: аналіз сучасних підходів до рекомендацій у медичній сфері, формування якісного набору даних для навчання моделі, реалізація алгоритму рекомендацій на основі схожості вподобань користувачів, проектування повноцінної структури бази даних та побудова веб-інтерфейсу для взаємодії з кінцевими користувачами.

Об'єктом дослідження є система онлайн-запису до лікаря з підтримкою персоналізованих рекомендацій. Предметом дослідження є побудова та реалізація алгоритму рекомендаційної системи, що враховує взаємодію користувача з іншими учасниками системи, історію оцінок та контентні ознаки лікарів. У процесі дослідження використано методи колаборативної фільтрації, обробки текстових відгуків (NLP [1]), а також засоби розробки веб-застосунків та проєктування реляційних баз даних.

Система ґрунтується на гібридному підході: базові прогнози рейтингів (Baseline) комбінуються з підрахунком схожості між користувачами або лікарями за допомогою метрик (наприклад, косинусної міри), а також враховуються текстові скарги пацієнтів, які отримані із відкритих відгуків. Це дозволяє забезпечити високий рівень релевантності рекомендацій навіть у випадках з обмеженою інформацією або відсутністю історії у нових користувачів.

У даній кваліфікаційній роботі здійснено повний цикл реалізації системи: створено структуру бази даних у PostgreSQL з урахуванням реальних сценаріїв взаємодії користувача, реалізовано механізм генерації рекомендацій мовою Python з подальшою інтеграцією у середовище ASP.NET Core MVC, розгорнуто базу даних у контейнері Docker, побудовано веб-інтерфейс, що підтримує повну функціональність онлайн-запису до лікаря та відображення персоналізованих рекомендацій. Результати тестування показали стабільність, масштабованість та зручність системи для кінцевого користувача.

Таким чином, результати цієї роботи можуть мати значну практичну цінність як у рамках цифровізації охорони здоров'я, так і як базова основа для розширення функціональності рекомендаційних сервісів у сфері електронної медицини.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Стан розвитку веб-застосунків у медичній сфері

Медична галузь традиційно характеризується складною структурою організації сервісів для пацієнтів. Така складність зумовлена не лише багатопрофільністю спеціалістів, а й варіативністю клінічних сценаріїв, широким спектром медичних послуг та високими вимогами до точності й конфіденційності даних. Крім того, пацієнти звертаються за допомогою з унікальними скаргами, мають різні рівні обізнаності про доступні варіанти лікування і часто покладаються на особистий досвід чи думки знайомих. Це створює значне навантаження на систему первинного доступу до медичних послуг і ускладнює пошук відповідного фахівця, особливо у великих містах.

Протягом останнього десятиліття індустрія охорони здоров'я демонструє активну диджиталізацію, зумовлену як глобальними технічними трендами, так і внутрішніми запитами на ефективні, прозорі та зручні сервіси для населення. У країнах з розвинутою цифровою інфраструктурою електронні медичні сервіси стали невід'ємною частиною системи охорони здоров'я: пацієнти можуть онлайн переглядати історію хвороб, замовляти рецепти, отримувати консультації та записуватись на прийом до лікаря. Успішна реалізація таких сервісів базується на інтеграції даних, стандартизації процесів і впровадженні інтелектуальних технологій для аналізу запитів користувача.

На практиці ж значна частина існуючих веб-застосунків для запису до лікаря залишається на рівні базового агрегування інформації. Такі сервіси, як правило, реалізовані у вигляді каталогів медичних установ або списків спеціалістів із фільтрами за спеціалізацією, регіоном або рейтингом. Вони забезпечують мінімально необхідний функціонал для вибору фахівця, проте не враховують поведінкових моделей користувача, не аналізують історичні патерни звернень і не пропонують жодних персоналізованих рішень.

Іншими словами, взаємодія з подібними платформами вимагає від користувача повного контролю за вибором – від формування запиту до аналізу результатів. У такій ситуації пацієнт змушений самотійно переглядати десятки або сотні профілів, ознайомлюватись із відгуками, оцінювати релевантність досвіду лікаря до власних потреб і при цьому не мати жодної гарантії, що зроблений вибір є оптимальним. Це особливо критично для вразливих груп користувачів – людей похилого віку, пацієнтів із хронічними захворюваннями або тих, хто звертається по медичну допомогу вперше.

Брак механізмів персоналізованого супроводу значно знижує ефективність цифрових сервісів. Вибір лікаря часто зводиться до інтуїтивного рішення або механічного сортування за кількістю зірок, що не завжди відображає реальний рівень компетенції або сумісність пацієнта з обраним фахівцем. Це, у свою чергу, може призводити до повторних звернень, невдоволення сервісом або навіть погіршення стану здоров'я через несвоєчасне або неадекватне лікування.

У таких умовах актуальним стає впровадження інтелектуальних систем, здатних адаптуватися до потреб конкретного користувача, використовувати його історію взаємодії, аналізувати подібні випадки й формувати обґрунтовані рекомендації. Це відкриває шлях до нової генерації медичних веб-застосунків, які не просто агрегують інформацію, а виступають активним посередником у прийнятті рішень, зменшуючи когнітивне навантаження на пацієнта та підвищуючи якість обслуговування загалом.

1.2 Еволюція та типологія рекомендаційних систем у медицині

Рекомендаційні системи в медичній сфері мають давню історію, яка бере початок із задач підтримки прийняття клінічних рішень. У перших реалізаціях вони відігравали роль експертних систем, що аналізували

симптоми, анамнез пацієнта та клінічні стандарти для формулювання діагностичних припущень або пропозицій щодо подальшого обстеження. Такі системи були спрямовані виключно на допомогу медичному персоналу, і в основі їх роботи лежали чітко визначені правила, побудовані на основі доказової медицини або досвіду фахівців.

Проте зі стрімким зростанням обчислювальних потужностей, появою великих обсягів цифрових даних та розвитком машинного навчання відбулося зрушення в парадигмі: рекомендаційні системи почали дедалі більше орієнтуватися не лише на клініцистів, а й на самих пацієнтів. Особливо помітним це стало в контексті пошуку лікаря, вибору медичної установи, оцінки якості обслуговування та персоналізованого планування маршруту лікування. Це розширення функцій відбулося не лише завдяки технічному прогресу, а й через соціальну потребу – пацієнти стали активнішими учасниками медичного процесу, що вимагало нових форм підтримки прийняття рішень уже на рівні користувача.

Сучасні рекомендаційні системи у медицині базуються на різних підходах, які можна класифікувати за характером джерел даних та логікою формування рекомендацій. Одним із базових напрямів є контентно-орієнтовані системи, які враховують властивості об'єктів – зокрема, характеристики лікарів (спеціалізація, стаж, місце роботи, відгуки) та намагаються підібрати спеціаліста, подібного до того, що був позитивно оцінений раніше. Такі системи добре працюють у випадках, коли користувач має сформований досвід взаємодії, однак мають обмежену здатність до узагальнення у нових ситуаціях.

Інший підхід представлений демографічними системами, які будують припущення на основі соціальних, вікових, географічних або статевих характеристик пацієнта. Вони часто використовуються як допоміжні фільтри, але в ізоляції не здатні забезпечити високий рівень персоналізації, оскільки не враховують фактичну поведінку користувача в системі.

Найбільш перспективним і технічно ефективним є підхід колаборативної фільтрації, особливо в умовах наявності значного обсягу історичних даних. Основна ідея полягає в тому, що вибір користувача може бути спрогнозований на основі схожості з іншими користувачами. У цьому випадку алгоритм не аналізує безпосередньо характеристики об'єкта (лікаря), а досліджує структуру оцінок, взаємодій або текстових відгуків, щоб знайти «сусідів» із подібною поведінкою. Якщо кілька користувачів з подібним профілем схвалили певного лікаря, є висока ймовірність, що новий користувач також буде ним задоволений.

Перевага колаборативної фільтрації полягає в її здатності виявляти неочевидні патерни вибору, які не пов'язані з явними атрибутами. Наприклад, двоє пацієнтів можуть мати різний вік і проживати в різних містах, але подібність у структурі їх звернень, оцінок і скарг дозволить системі зробити релевантне припущення щодо наступного вибору. Це дозволяє системі навчатися з колективного досвіду, що особливо цінно в медичному контексті, де традиційні маркери не завжди є показовими.

Нарешті, гібридні системи об'єднують переваги згаданих підходів. Вони комбінують сигнали з різних джерел: контентну інформацію про лікарів, демографічні дані користувачів, історію взаємодій та результати обробки текстів відгуків. Такий підхід дозволяє зменшити слабкі сторони окремих моделей (наприклад, проблему cold start або ризик упереджених рекомендацій) і забезпечити більш стабільні й точні результати у широкому спектрі сценаріїв.

У сфері медицини колаборативна фільтрація, особливо у поєднанні з NLP-модулями, демонструє високу релевантність. Кожна оцінка, залишена пацієнтом, кожне текстове враження після прийому чи взаємодія із системою стає сигналом, який підсилює точність моделі. Таким чином, алгоритм поступово будує уявлення про переваги користувача і трансформує загальний досвід усієї системи в індивідуальні поради.

Отже, еволюція рекомендаційних систем у медицині – це шлях від жорстко закодованих правил до адаптивних моделей, здатних гнучко реагувати на індивідуальні потреби пацієнтів. Сучасна практика демонструє, що саме колективний досвід, узятий у поєднанні з персональними характеристиками, забезпечує найкращий результат у виборі медичного фахівця.

1.3 Актуальні проблеми при онлайн-виборі лікаря

Попри стрімкий розвиток цифрових сервісів, процес вибору лікаря через інтернет залишається доволі складним і недостатньо структурованим для більшості користувачів. Пацієнт, який шукає спеціаліста онлайн, стикається з низкою проблем, що суттєво ускладнюють прийняття обґрунтованого рішення, знижують рівень довіри до системи та збільшують ризик хибного вибору. Це стосується як недосконалості самих платформ, так і обмеженості представленої інформації.

Одним із головних бар'єрів є непрозорість системи оцінювання лікарів. У багатьох сервісах рейтинги формуються без зрозумілого обґрунтування, не деталізуються за критеріями (наприклад, професіоналізм, сервіс, емпатія), а середня оцінка подається без урахування кількості відгуків або часової динаміки. Це спотворює сприйняття та створює хибне враження про якість послуг. Крім того, часто відсутня можливість перевірити автентичність самих відгуків або їх прив'язку до реальних прийомів, що знижує достовірність отриманої інформації.

Ще однією проблемою є обмежений доступ до історичних даних пацієнта. Якщо користувач не має персонального кабінету з історією звернень, система не може враховувати попередній досвід або контекст – наприклад, які лікарі вже були відвідані, з яким результатом, чи були повторні звернення. Через це втрачається важлива інформація, яка могла би суттєво покращити персоналізацію рекомендацій.

Не менш суттєвою є розбіжність між реальними очікуваннями пацієнта та тим, що відображено у профілі лікаря. Стандартні параметри на зразок спеціалізації, стажу, статі чи місця роботи хоч і важливі, але самі по собі не гарантують сумісності або емоційного комфорту. Для багатьох користувачів значення мають неформальні характеристики: стиль комунікації, уважність, відкритість до діалогу, досвід лікування певного типу пацієнтів тощо. У структурованій анкеті лікаря ці аспекти зазвичай не представлені, і саме вони стають предметом пошуку у відгуках.

У таких умовах відсутність механізмів персоналізації стає ключовою перешкодою. Новий користувач змушений орієнтуватися лише на загальну інформацію, не отримуючи жодних підказок щодо того, який лікар найбільш відповідає його індивідуальним потребам. Це особливо ускладнює вибір для тих, хто вперше звертається до медичної установи або не має медичного бекграунду, щоб самостійно проаналізувати профіль спеціаліста.

Часто пацієнти змушені самостійно переглядати десятки сторінок з профілями, шукати співзвучні відгуки, порівнювати досвід інших користувачів, тобто виконувати роль аналітичної системи вручну. У цьому контексті стає очевидною причина, чому багато хто продовжує покладатися на особисті рекомендації знайомих або колег. Така поведінка імітує основи колаборативної фільтрації: людина довіряє вибору тих, хто перебував у подібній ситуації та мав позитивний досвід.

Ці припущення підтверджуються й статистично. Згідно з даними опитування, проведеного компанією Software Advice у 2020 році, переважна більшість користувачів звертається до сайтів із відгуками саме на етапі первинного пошуку лікаря: 71% респондентів використовують такі ресурси як перший крок у виборі нового медичного постачальника. Ще 23% перевіряють онлайн-профілі навіть після особистих рекомендацій, щоб переконатися у правильності вибору. Нарешті, 6% використовують ці сервіси для переоцінки поточного лікаря. Сукупно це означає, що 94%

пацієнтів так чи інакше покладаються на онлайн-джерела під час ухвалення рішення, що перетворює відгуки в один із ключових факторів у сучасному медичному середовищі (рисунок 1.1).

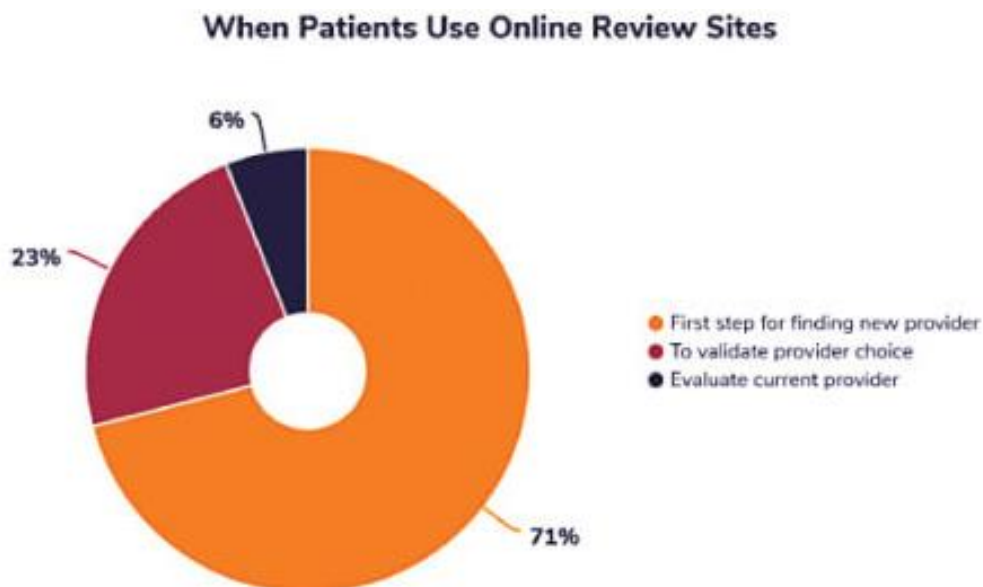


Рисунок 1.1 – Використання пацієнтами онлайн-сайтів із відгуками під час вибору медичного постачальника (за даними Software Advice, 2020)

У цьому контексті створення інтелектуальних систем, які не лише агрегують ці дані, а й здатні виявляти схожі шаблони поведінки серед пацієнтів, аналізувати відгуки з урахуванням тону та змісту, адаптувати рекомендації під конкретну ситуацію користувача – виглядає не лише доречним, а й неминучим етапом еволюції цифрової медицини. Такий підхід дозволяє перейти від пасивного інтерфейсу до активної допомоги, що значно підвищує якість прийнятого рішення та загальну задоволеність користувача сервісом.

1.4 Передумови побудови інтелектуальної рекомендаційної системи

Сучасні сервіси медичного онлайн-запису дедалі частіше позиціонуються не лише як інструменти бронювання часу прийому, а як

повноцінні інформаційні платформи, здатні супроводжувати користувача в процесі прийняття рішень щодо вибору фахівця. Однак реальний функціонал більшості наявних систем не відповідає цим очікуванням. Як було показано у попередньому підрозділі, наявні сервіси залишають користувача наодинці з великим масивом малоструктурованої інформації – профілів лікарів, відгуків, узагальнених рейтингів – не надаючи жодної адаптивної підтримки у виборі. Така ситуація створює як суб’єктивні труднощі (когнітивне перевантаження, втома від вибору), так і об’єктивні втрати якості (хибний вибір, низький рівень задоволеності, пропущені візити тощо).

У цих умовах виникає природна потреба в інтелектуальному посереднику – системі, яка б не просто акумулювала дані, а й брала участь в аналізі потреб користувача, враховувала історію попередніх рішень, орієнтувалася на успішний досвід інших пацієнтів і формувала персоналізовані рекомендації щодо лікаря. Ідеальною технологічною основою для побудови такого посередника є алгоритми колаборативної фільтрації, які здатні адаптуватися до контексту та вчитися на сукупному досвіді користувачів системи.

Колаборативна фільтрація дозволяє поєднати масовість цифрової поведінки пацієнтів з індивідуальними потребами конкретної особи, формуючи на цій основі точкові рекомендації, які з часом лише покращуються. Алгоритм не потребує жорсткої ручної настройки або експертних правил – замість цього він автоматично виявляє приховані подібності між користувачами та об’єктами рекомендацій (лікарями), ґрунтуючись на оцінках, відгуках або інших формах взаємодії. Це особливо важливо в медичному середовищі, де класичні фільтри за спеціалізацією або регіоном не дозволяють врахувати емоційні, психологічні чи поведінкові аспекти вибору.

Ще одним ключовим фактором, що обґрунтовує необхідність побудови рекомендаційної системи, є природа самих даних у медичних

застосунках. На відміну від інших сфер (наприклад, інтернет-магазинів), тут кожен акт взаємодії – оцінка, відгук, запис на прийом – є не лише транзакцією, а й сигналом про якість медичного контакту. Це надає високий семантичний рівень кожному фрагменту даних і дозволяє використовувати навіть обмежені масиви для побудови значущих моделей прогнозування.

Крім того, важливо враховувати, що медичний запис – це контекстуальна дія, яка відбувається в умовах обмеженого часу, можливого стресу або невизначеності. У таких ситуаціях користувачеві потрібна не повна свобода вибору, а довірена система супроводу, здатна підказати варіанти на основі зваженого аналізу. Інтелектуальна рекомендаційна система може виступати саме в такій ролі: не диктувати рішення, а пропонувати обґрунтовані альтернативи, які базуються на історичних закономірностях і узгоджуються з особистим профілем користувача.

Окремої уваги заслуговує можливість інтеграції текстових відгуків у модель. За допомогою методів обробки природної мови (NLP) можна аналізувати не лише оцінки, а й змістовні повідомлення користувачів, виявляючи контекст скарг, стиль комунікації лікаря, повторювані теми (наприклад, дитячі захворювання, болі в спині, хронічні випадки). Це дозволяє підвищити якість рекомендацій для нових користувачів (cold start), які ще не встигли залишити оцінки, але вже мають певні очікування, які можна зіставити з існуючими патернами.

Важливим є й той факт, що сучасні веб-платформи зазвичай мають технічну інфраструктуру, готову до розширення. Використання реляційних баз даних, REST-архітектури та контейнеризованого середовища (Docker) створює передумови для ефективної інтеграції рекомендаційного модуля без суттєвих змін у бізнес-логіці застосунку. Це означає, що впровадження інтелектуального компонента не потребує повного редизайну системи – достатньо модульного розширення, яке доповнює, а не замінює існуючий функціонал.

Таким чином, створення веб-застосунку з інтегрованим інтелектуальним модулем рекомендацій постає не як експеримент чи додатковий функціонал, а як логічне продовження еволюції цифрових сервісів у медичній сфері. Різке зростання популярності електронних платформ серед пацієнтів, які прагнуть швидкого та зручного доступу до медичних послуг, вимагає від розробників нових рішень, що здатні забезпечити не лише інформативність, а й релевантність пропозицій. Водночас зростає усвідомлення того, що ефективна медична взаємодія не може бути уніфікованою – вона повинна адаптуватися до потреб конкретної людини, враховувати її контекст, попередній досвід і навіть стиль комунікації.

Наявність цифрових слідів поведінки користувачів – історії оцінок, текстів відгуків, повторних звернень – формує багатий інформаційний простір, у якому можна виявляти закономірності та будувати прогнози. Цей інформаційний потенціал залишається практично невикористаним у більшості сучасних систем, які працюють за принципом агрегатора. Запровадження алгоритмів колаборативної фільтрації дозволяє змінити цю ситуацію, перетворивши сирі дані на дієвий інструмент персоналізованої підтримки. Крім того, сучасні технологічні підходи, зокрема використання контейнеризації, REST-архітектури, реляційних БД та мов програмування, орієнтованих на обробку даних, створюють сприятливе середовище для впровадження таких систем без необхідності повного перевизначення інфраструктури.

Практичний досвід реалізації подібних підходів у суміжних галузях – електронній комерції, стримінгових сервісах, освітніх платформах – свідчить про значне зростання ефективності, коли користувач отримує рекомендації, сформовані не випадково, а на основі аналізу мільйонів подібних рішень. Перенесення цього принципу у медичну сферу, з її особливим статусом, делікатністю даних та високою відповідальністю, відкриває новий рівень якості цифрових сервісів. Інтелектуальна система

рекомендацій не лише підвищує зручність користувача, а й виконує роль адаптивного навігатора в складному середовищі медичних рішень, де правильний вибір має безпосереднє відображення на здоров'ї та добробуті людини.

У результаті впровадження такої системи змінюється сама логіка взаємодії з цифровим сервісом: користувач більше не є пасивним споживачем інформації, а перетворюється на активного учасника персоналізованої траєкторії вибору. Це підвищує довіру до платформи, знижує рівень фрустрації при пошуку, скорочує час на прийняття рішення і сприяє формуванню довгострокової лояльності до сервісу. Таким чином, впровадження рекомендаційної підсистеми є не лише доцільним з погляду технологій, а й необхідним із позиції користувацького досвіду та загальної трансформації медичних цифрових екосистем.

1.5 Постановка задачі

Аналіз сучасного стану медичних веб-сервісів, практик вибору лікарів та можливостей персоналізації в цифровому середовищі засвідчив наявність чітко вираженого розриву між інформаційною насиченістю платформ і їхньою здатністю надавати користувачам релевантні рекомендації. Незважаючи на стрімкий розвиток електронних систем запису на прийом, більшість з них залишаються лише інструментами для фільтрації фахівців за базовими ознаками – такими як спеціалізація чи географічна доступність – без урахування реальних потреб користувача, історії його звернень або досвіду інших пацієнтів із подібними запитами. У результаті, ключовий момент взаємодії – вибір лікаря – часто виявляється непрозорим, трудомістким і неефективним.

У таких умовах доцільним виглядає створення інтелектуальної системи нового покоління, яка поєднує класичну функціональність веб-застосунку з можливостями персоналізованих рекомендацій. Йдеться про

платформу, здатну не лише забезпечити стандартний функціонал (реєстрація, вибір лікаря, перегляд профілів, запис на прийом), а й формувати релевантні підказки на основі аналізу колективного досвіду користувачів. Центром такого рішення виступає алгоритмічна підсистема колаборативної фільтрації, яка навчається на історичних даних і прогнозує найдоцільніші варіанти для кожного конкретного користувача.

З огляду на вищеописане, основною метою є створення системи, яка поєднає функціонал веб-застосунку для запису до лікаря та інтелектуальної рекомендаційної підсистеми на основі колаборативної фільтрації. Така система повинна мати змогу:

- отримувати та зберігати дані про взаємодії користувачів із лікарями (оцінки, записи, повторні звернення). Передбачається, що всі релевантні дії користувача – як-от залишення оцінки після прийому, написання текстового відгуку або повторна взаємодія з тим самим лікарем – будуть зафіксовані у системі та включені до профілю для подальшого аналізу;

- обчислювати коефіцієнти подібності між користувачами та/або лікарями. Розрахунок схожості має базуватися на історії оцінок, зокрема з використанням косинусної міри або іншої метрики, що дозволить ідентифікувати найбільш близькі за вподобаннями пари користувачів або об'єктів;

- будувати прогнози рейтингів за допомогою baseline-моделі, а також item-to-item підходу з використанням косинусної подібності. Це дозволить формувати рекомендації навіть у ситуаціях обмеженого контексту, поєднуючи глобальну середню оцінку з індивідуальними зсувами (bias) і структурою локальних подібностей між лікарями;

- адаптуватися до нових користувачів і лікарів, вирішуючи cold start-проблему. Система повинна забезпечувати прийнятну якість рекомендацій навіть за відсутності значної історії – зокрема, через використання

контентної інформації або загальних шаблонів взаємодії на основі демографічних і текстових ознак;

– реалізувати інтерфейс, у якому рекомендації відображаються динамічно в контексті інших функцій застосунку: фільтрації, реєстрації, профілю лікаря. Важливо забезпечити, щоб персоналізовані поради були невід’ємною частиною основного користувацького досвіду: інтегровані у список лікарів, автоматично з’являлися у відповідних розділах, змінювалися при оновленні фільтрів або появи нових даних.

Запропонована система має не лише продемонструвати технічну реалізованість описаного підходу, а й відповідати вимогам стабільної, масштабованої та зручної цифрової платформи. Передбачається, що її архітектура буде сумісна з сучасними веб-фреймворками та базами даних, а розгортання – можливим у контейнерному середовищі без складної конфігурації. При цьому особлива увага приділяється користувацькому досвіду: попри складність внутрішньої логіки, інтерфейс має залишатися інтуїтивним і доступним навіть для людей без спеціальної підготовки.

Таким чином, сформульована задача передбачає не лише розробку окремих компонентів системи, а комплексну реалізацію повноцінного веб-застосунку, у якому алгоритмічні інструменти персоналізації тісно поєднані з традиційними функціями електронного медичного запису.

2 ФОРМУВАННЯ СИСТЕМИ ДАНИХ

2.1 Збір необроблених даних

Якість і успішність будь-якої сучасної рекомендаційної системи напряду залежить від глибини та релевантності вихідних даних. Для задачі персоналізованих рекомендацій лікарів базовим джерелом інформації є масиви реальних відгуків і оцінок, залишених пацієнтами після візиту до медичних спеціалістів. Однак, на практиці навіть найбільші відкриті медичні датасети – такі як RateMDs – мають свої особливості і обмеження, які необхідно враховувати з перших кроків побудови системи.

Вихідний датасет RateMDs містить понад два мільйони анонімізованих відгуків, кожен з яких включає текстову частину, числові оцінки за кількома категоріями (helpfulness, knowledgeability, punctuality, staff) та атрибутивну інформацію про лікаря (ID, спеціалізація, місце роботи тощо). Відсутність постійного ідентифікатора користувача є як обмеженням (неможливість класичного user-based CF), так і запорукою захисту приватності. Важливо, що дані мають нерівномірний розподіл: переважна більшість лікарів отримує лише декілька відгуків, а невелика група – сотні й тисячі, що створює спарсеність (розрідженість) даних і підвищує вимоги до алгоритмів.

Збір даних відбувається шляхом автоматизованого завантаження CSV-файлів, попередньої перевірки їхньої структурної цілісності (наявність усіх ключових полів, коректність кодування, відсутність дубльованих записів). У подальшому, на цьому етапі можливе збагачення інформації: наприклад, отримання додаткових довідників спеціалізацій або локацій для коректнішої фільтрації та валідації записів.

Суттєве значення має етичний аспект: навіть за відсутності персональних даних важливо дотримуватися принципів відповідального

використання медичної інформації, уникати спроб “ідентифікації” за непрямими ознаками.

Таким чином, уже на етапі збору даних формується основа для подальшого етапу – проектування структури системи, оскільки якість і формат вихідних записів впливають як на вибір алгоритмів, так і на майбутню продуктивність системи.

2.2 Проектування структури даних і бази даних

Проектування бази даних є одним із найважливіших етапів у створенні веб-застосунку, що використовує рекомендаційну систему. Від правильної структури таблиць залежить не лише швидкість роботи сервісу, а й можливість для подальшого розширення функціоналу, а також точність алгоритмів рекомендацій. У цьому проєкті використано реляційну базу даних, яка дозволяє ефективно зберігати, обробляти та аналізувати великі масиви взаємопов'язаних даних.

Центральними сутностями виступають користувачі, лікарі, клініки, записи на прийом, рейтинги й відгуки. Додатково введено низку допоміжних довідників, які сприяють нормалізації даних та забезпечують ефективність і гнучкість запитів. Всі таблиці спроектовано з урахуванням принципів нормалізації (до третьої нормальної форми), що мінімізує дублювання інформації та підвищує швидкість роботи системи.

Довідкові таблиці виконують роль стандартизованих наборів даних, що використовуються багаторазово. Наприклад, таблиця статей визначає можливі значення для користувачів (чоловіча, жіноча, інше), а таблиця ролей користувачів чітко відокремлює пацієнтів, лікарів та адміністраторів системи, що важливо для розмежування доступу. Довідник міст дозволяє уніфікувати інформацію про місцезнаходження клінік та користувачів, що є суттєвим фактором при пошуку лікаря пацієнтом.

Ключовою є таблиця користувачів, що містить базову інформацію та автентифікаційні дані. Вона пов'язана з таблицею лікарів, яка додає специфічні атрибути, такі як спеціалізація, досвід роботи й особиста біографія. Це рішення дозволяє гнучко керувати ролями користувачів без зайвого дублювання інформації.

Записи на прийом об'єднують пацієнтів, лікарів та клініки, і мають свій статус (створено, скасовано, завершено). Важливим моментом є чітке розмежування таблиць рейтингу і відгуків, що дозволяє системі окремо аналізувати кількісні та якісні характеристики взаємодій пацієнтів і лікарів.

Для реалізації рекомендаційної системи введено спеціальні таблиці для кешування рекомендацій, які формуються за допомогою колаборативної фільтрації, а також окремі таблиці для зберігання матриць подібності між користувачами й лікарями. Це істотно прискорює обчислення рекомендацій у реальному часі.

Особливою є також таблиця зв'язків між лікарями та клініками, яка підтримує many-to-many взаємозв'язок. Це рішення дозволяє одному лікарю працювати у декількох клініках, а клінікам – наймати кількох лікарів, що відповідає реальним сценаріям використання сервісу.

Таким чином, представлена структура бази даних є комплексною, але водночас чітко організованою і продуманою до деталей. Це забезпечує гнучкість, швидкість роботи та простоту розширення системи в майбутньому, створюючи надійний фундамент для ефективної роботи алгоритмів рекомендацій.

2.1.1 Схема genders

Таблиця genders використовується для зберігання довідника статей. Це дозволяє стандартизувати запис інформації про стать користувачів і уникати помилок при введенні даних у пов'язані таблиці системи (лістинг 2.1).

Лістинг 2.1 – SQL-реалізація таблиці genders

```
CREATE TABLE genders (  
  gender_id SERIAL PRIMARY KEY,  
  gender VARCHAR(10) UNIQUE  
);
```

Поля таблиці genders:

– gender_id (primary key): унікальний ідентифікатор статі.

Використовується як зовнішній ключ у таблиці users;

– gender: назва статі (наприклад, «чоловіча», «жіноча», «інше»).

Значення є унікальним, що запобігає дублюванню.

2.1.2 Схеми міст

Таблиця cities призначена для зберігання списку міст. Це дає можливість уніфіковано вказувати місце проживання користувачів чи розташування клінік (лістинг 2.2).

Лістинг 2.2 – SQL-реалізація таблиці cities

```
CREATE TABLE cities (  
  city_id SERIAL PRIMARY KEY,  
  city_name VARCHAR(100)  
  UNIQUE  
);
```

Поля таблиці cities:

– city_id (primary key): унікальний ідентифікатор міста, що використовується у зовнішніх ключах інших таблиць;

– city_name: унікальна назва міста.

2.1.3 Схеми статусів запису

Таблиця `roles` визначає ролі користувачів у системі: пацієнт, лікар або адміністратор. Це дозволяє чітко розмежувати функціональні можливості різних категорій користувачів (лістинг 2.3).

Лістинг 2.3 – SQL-реалізація таблиці `roles`

```
CREATE TABLE roles (  
    role_id SERIAL PRIMARY KEY,  
    role_name VARCHAR(20) UNIQUE NOT NULL -- 'patient',  
'doctor', 'admin' );
```

Поля таблиці `roles`:

- `role_id` (primary key): унікальний ідентифікатор ролі;
- `role_name`: назва ролі, що має бути унікальною (наприклад, «patient», «doctor», «admin»).

2.1.4 Схеми користувачів

Таблиця `users` містить основну інформацію про всіх користувачів – пацієнтів, лікарів та адміністраторів (лістинг 2.4).

Лістинг 2.4 – SQL-реалізація таблиці `users`

```
CREATE TABLE users (  
    user_id SERIAL PRIMARY KEY,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    password_hash VARCHAR(255) NOT NULL,  
    full_name VARCHAR(100),  
    gender_id INT REFERENCES genders(gender_id),  
    city_id INT REFERENCES cities(city_id),  
    role_id INT REFERENCES roles(role_id),  
    date_registered TIMESTAMP DEFAULT CURRENT_TIMESTAMP );
```

Поля таблиці users:

- user_id (primary key): унікальний ідентифікатор користувача;
- email: унікальна адреса електронної пошти, що використовується для входу в систему;
- password_hash: хеш пароля для забезпечення безпеки;
- full_name: повне ім'я користувача;
- gender_id: зовнішній ключ на таблицю genders;
- city_id: зовнішній ключ на таблицю cities;
- role_id: зовнішній ключ на таблицю roles;
- date_registered: дата та час реєстрації користувача у системі.

2.1.5 Схеми записів на прийом

Таблиця appointments використовується для зберігання записів на прийом. Вона є центральною для обліку взаємодій між пацієнтами, лікарями та клініками (лістинг 2.5).

Лістинг 2.5 – SQL-реалізація таблиці appointments

```
CREATE TABLE appointments (
    appointment_id SERIAL PRIMARY KEY,
    patient_id INT REFERENCES users(user_id) ON DELETE
CASCADE,
    doctor_id INT REFERENCES doctors(doctor_id) ON DELETE
CASCADE,
    clinic_id INT REFERENCES clinics(clinic_id),
    appointment_time TIMESTAMP NOT NULL,
    status_id INT REFERENCES appointment_statuses(status_id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Поля таблиці appointments:

- appointment_id (primary key): унікальний ідентифікатор запису;

- patient_id: зовнішній ключ на таблицю users (пацієнт);
- doctor_id: зовнішній ключ на таблицю doctors;
- clinic_id: зовнішній ключ на таблицю clinics;
- appointment_time: дата й час запланованого прийому;
- status_id: зовнішній ключ на таблицю appointment_statuses (статус запису);
- created_at: дата та час створення запису.

2.1.6 Схема оцінок

Таблиця ratings містить числові оцінки лікарів пацієнтами за різними критеріями (лістинг 2.6).

Лістинг 2.6 – SQL-реалізація таблиці ratings

```
CREATE TABLE ratings ( rating_id SERIAL PRIMARY KEY,
  doctor_id INT REFERENCES doctors(doctor_id),
  patient_id INT REFERENCES users(user_id),
  appointment_id INT REFERENCES
appointments(appointment_id),

  service INT CHECK (service BETWEEN 1 AND 5),
  professionalism INT CHECK (professionalism BETWEEN 1 AND
5), helpfulness INT CHECK (helpfulness BETWEEN 1 AND 5),
  knowledge INT CHECK (knowledge BETWEEN 1 AND 5),
  rating_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  UNIQUE (doctor_id, patient_id, appointment_id) );
```

Поля таблиці ratings:

- rating_id (primary key): унікальний ідентифікатор оцінки;
- doctor_id: зовнішній ключ на таблицю doctors;
- patient_id: зовнішній ключ на таблицю users;
- appointment_id: зовнішній ключ на таблицю appointments;

- service: оцінка рівня сервісу (від 1 до 5);
- professionalism: оцінка професіоналізму (від 1 до 5);
- helpfulness: оцінка готовності допомогти (від 1 до 5);
- knowledge: оцінка знань (від 1 до 5);
- rating_date: дата й час залишення оцінки;
- UNIQUE (doctor_id, patient_id, appointment_id): унікальність оцінки за конкретним поєднанням лікар–пацієнт–запис.

2.1.7 Схема відгуків

Таблиця reviews зберігає текстові відгуки пацієнтів про лікарів (лістинг 2.7).

Лістинг 2.7 – SQL-реалізація таблиці reviews

```
CREATE TABLE reviews ( review_id SERIAL PRIMARY KEY,
  doctor_id INT REFERENCES doctors(doctor_id),
  patient_id INT REFERENCES users(user_id),

  appointment_id INT REFERENCES
appointments(appointment_id),
  review_text TEXT, review_date TIMESTAMP DEFAULT
CURRENT_TIMESTAMP );
```

Поля таблиці reviews:

- review_id (primary key): унікальний ідентифікатор відгуку;
- doctor_id: зовнішній ключ на таблицю doctors;
- patient_id: зовнішній ключ на таблицю users;
- appointment_id: зовнішній ключ на таблицю appointments;
- review_text: текст відгуку;
- review_date: дата й час залишення відгуку.

2.1.8 Схема рекомендацій

Таблиця `recommendations` використовується для кешування результатів роботи рекомендаційної системи (лістинг 2.8).

Лістинг 2.8 – SQL-реалізація таблиці `recommendations`

```
CREATE TABLE recommendations ( rec_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(user_id),
    doctor_id INT REFERENCES doctors(doctor_id),
    predicted_rating FLOAT, approach VARCHAR(30),
    -- 'user-based', 'item-based' generated_at TIMESTAMP
DEFAULT CURRENT_TIMESTAMP
);
```

Таполя таблиці `recommendations`:

- `rec_id` (primary key): унікальний ідентифікатор запису;
- `user_id`: зовнішній ключ на таблицю `users`;
- `doctor_id`: зовнішній ключ на таблицю `doctors`;
- `predicted_rating`: прогнозований рейтинг лікаря для даного користувача;
- `approach`: тип підходу для рекомендації («user-based», «item-based»);
- `generated_at`: дата та час генерації рекомендації.

2.1.9 Схема подібності користувачів і лікарів

Таблиця `user_similarities` містить попередньо розраховані коефіцієнти схожості між користувачами (лістинг 2.9).

Лістинг 2.9 – SQL-реалізація таблиці `user_similarities`

```
CREATE TABLE user_similarities ( id SERIAL PRIMARY KEY,
    user_id1 INT REFERENCES users(user_id),
    user_id2 INT REFERENCES users(user_id),
```

Продовження лістингу 2.9

```

    similarity FLOAT,
    UNIQUE (user_id1, user_id2)
);

```

Поля таблиці user_similarities:

- id (primary key): унікальний ідентифікатор запису;
- user_id1: ідентифікатор першого користувача (пацієнта);
- user_id2: ідентифікатор другого користувача;
- similarity: коефіцієнт схожості (діапазон від –1 до 1).

2.1.10 Схема зв'язків лікар-клініка

Таблиця doctor_clinics фіксує зв'язки між лікарями та клініками, підтримуючи відношення багато-до-багатьох (лістинг 2.10).

Лістинг 2.10 – SQL-реалізація таблиці doctor_clinics

```

CREATE TABLE doctor_clinics (
    id SERIAL PRIMARY KEY,
    doctor_id INT REFERENCES doctors(doctor_id) ON DELETE
CASCADE, clinic_id INT REFERENCES clinics(clinic_id) ON DELETE
CASCADE, UNIQUE (doctor_id, clinic_id)
);

```

Поля таблиці doctor_clinics:

- id (primary key): унікальний ідентифікатор зв'язку;
- doctor_id: зовнішній ключ на таблицю doctors;
- clinic_id: зовнішній ключ на таблицю clinics;
- UNIQUE (doctor_id, clinic_id): запобігає дублюванню зв'язків для пари лікар–клініка.

2.1.11 Схема specialties

Таблиця specialties використовується як довідник спеціалізацій лікарів. Це дозволяє підтримувати уніфікований перелік медичних спеціальностей у системі та спрощує пошук за фахом (лістинг 2.11).

Лістинг 2.11 – SQL-реалізація таблиці specialties

```
CREATE TABLE specialties (  
    specialty_id SERIAL PRIMARY KEY,  
    name VARCHAR(100) UNIQUE NOT NULL  
);
```

Поля таблиці specialties:

- specialty_id (primary key): унікальний ідентифікатор спеціалізації;
- name: назва спеціалізації, наприклад «кардіолог», «невролог», «педіатр». Поле є унікальним.

2.1.12 Схема clinics

Таблиця clinics містить інформацію про медичні заклади, у яких працюють лікарі та ведеться прийом пацієнтів (лістинг 2.12).

Лістинг 2.12 – SQL-реалізація таблиці clinics

```
CREATE TABLE clinics (  
    clinic_id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    city_id INT REFERENCES cities(city_id),  
    address VARCHAR(255)  
);
```

Поля таблиці clinics:

- clinic_id (primary key): унікальний ідентифікатор клініки;

- name: назва клініки;
- city_id: зовнішній ключ на таблицю cities (місто, у якому знаходиться клініка);
- address: адреса клініки.

2.1.13 Схеми doctors

Таблиця doctors містить додаткові атрибути лікаря, що розширюють основний профіль користувача (лістинг 2.13).

Лістинг 2.13 – SQL-реалізація таблиці doctors

```
CREATE TABLE doctors (
    doctor_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(user_id) ON DELETE
CASCADE,
    specialty_id INT REFERENCES specialties(specialty_id),
    bio TEXT,
    experience_years INT,
    photo_url VARCHAR(255)
);
```

Поля таблиці doctors:

- doctor_id (primary key): унікальний ідентифікатор лікаря;
- user_id: зовнішній ключ на таблицю users (ідентифікатор користувача);
- specialty_id: зовнішній ключ на таблицю specialties (спеціалізація лікаря);
- bio: коротка біографія або опис профілю лікаря;
- experience_years: кількість років професійного досвіду;
- photo_url: посилання на фото лікаря.

2.1.14 Схеми appointment_statuses

Таблиця appointment_statuses є довідником статусів запису на прийом, які використовуються для управління життєвим циклом прийомів у системі (лістинг 2.14).

Лістинг 2.14 – SQL-реалізація таблиці appointment_statuses

```
CREATE TABLE appointment_statuses (  
    status_id SERIAL PRIMARY KEY,  
    status_name VARCHAR(30) UNIQUE  
);
```

Поля таблиці appointment_statuses:

- status_id (primary key): унікальний ідентифікатор статусу;
- status_name: назва статусу запису (наприклад, «created», «cancelled», «done»).

Сформована реляційна база даних для «Віртуального помічника запису на прийом до лікаря» складається з п'ятнадцяти основних таблиць, зображених на схемі (рисунок 2.1).

Цей набір охоплює всі дані, необхідні для повноцінної роботи сервісу: від довідників статі й ролей до таблиць рекомендацій і матриць схожості. Центральними у структурі є користувачі, лікарі та записи на прийом – саме вони утворюють ядро взаємодій, довкола якого вибудовується логіка рекомендаційної підсистеми.

Сукупний обсяг даних у реальному середовищі оцінюється у десятки гігабайтів і зростає пропорційно кількості пацієнтів, лікарів та залишених відгуків. Щоб забезпечити стабільну продуктивність при такому навантаженні, у схемі закладено низку оптимізацій: індексування ключових полів, кешування прогнозованих рейтингів, періодичне перерахування матриць схожості та можливість партиціонування найбільших таблиць (зокрема ratings і reviews).

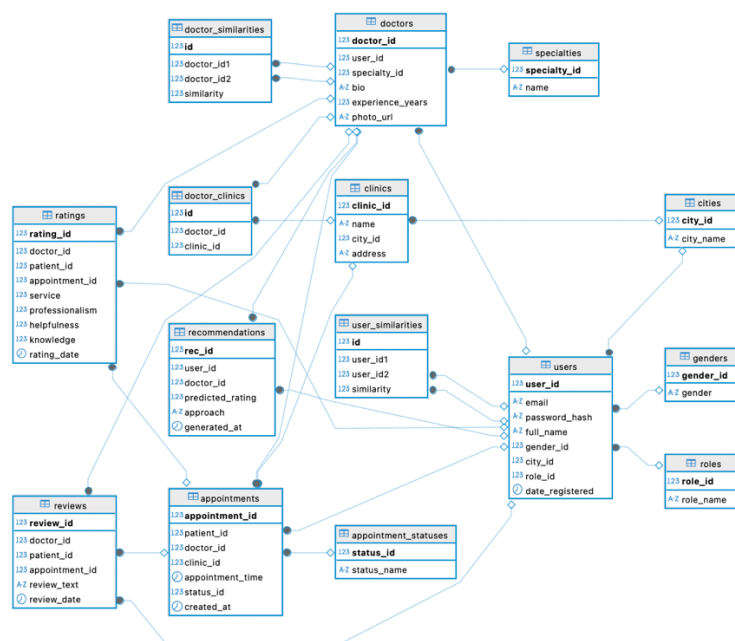


Рисунок 2.1 – Загальна схеми бази даних із запису на прийом до лікаря

Ці рішення дозволяють системі працювати швидко навіть при великому обсязі транзакцій і запитів, підтримуючи користувачів у режимі реального часу.

2.3 Обробка текстів

Відгуки пацієнтів у системах медичних рекомендацій – це цінний, але складний для обробки ресурс. Текстові дані містять не лише оцінки, а й опис скарг, побажань, емоційних вражень. Основна задача NLP-модуля полягає у перетворенні «сирого» тексту на структуровану інформацію, придатну для подальшого аналізу й побудови персоналізованих рекомендацій.

Перший крок – очищення тексту від зайвих символів, стандартизація регістру, видалення стоп-слів (беззмістовних слів типу «це», «той», «і» тощо). Далі застосовується лематизація: зведення слів до базової форми («головний», «головного», «головним» – усе зводиться до «головний»), що підвищує якість подальшої класифікації.

Наступний етап – векторизація. Найпоширеніший підхід – TF-IDF (Term Frequency–Inverse Document Frequency), який дозволяє виділити ключові слова, характерні саме для цього відгуку порівняно з усією базою. Для медичних систем особливо цінним є виявлення «медичних тем» – наприклад, автоматичне розпізнавання скарг: «біль у спині», «проблеми із зором», «дитяча алергія».

Далі відбувається класифікація текстів за проблемами або типом звернення (наприклад, за допомогою машинного навчання [6] або простіших евристик). Це дає змогу використовувати додаткову «ознаку» у системі рекомендацій: рекомендувати лікаря, що успішно вирішував подібні проблеми.

Окремий аспект – виявлення «тону» (sentiment analysis): чи був досвід пацієнта позитивним чи негативним, що додатково підвищує точність рекомендацій і дозволяє уникати лікарів із сумнівною репутацією. Наукові дослідження [2], [3], [4] підтверджують, що використання навіть базових NLP-підходів (лематизація, TF-IDF) дозволяє підвищити якість персоналізованих рекомендацій на 10–20% у порівнянні зі “сліпою” обробкою текстів.

Таким чином, NLP – це міст між «сирим» текстом і структурованими ознаками, які може використати як класична, так і сучасна рекомендаційна система [3], [7].

3 РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ТА СИСТЕМИ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ

3.1 Реалізація колаборативної фільтрації

3.1.1 Теоретичні основи колаборативної фільтрації

Колаборативна фільтрація (CF) у сучасних рекомендаційних системах є одним із найефективніших підходів до прогнозування індивідуальних вподобань користувачів[4]. У контексті задачі підбору лікаря CF дозволяє пропонувати пацієнтам таких спеціалістів, яких із високою ймовірністю схвалили б користувачі зі схожими оцінками та медичними ситуаціями. В цій роботі впроваджено як user-based, так і item-based модифікації алгоритму, що дозволяє підлаштовувати систему під різні сценарії використання та специфіку даних.

3.1.2 Принципи колаборативної фільтрації

Принцип CF полягає у створенні так званої “матриці взаємодій” – таблиці, у якій рядки відповідають користувачам (пацієнтам), стовпці – об’єктам (лікарям), а комірки – рейтингам або відгукам. Проте у реальному світі така матриця завжди розріджена: більшість пацієнтів взаємодіють лише з невеликою кількістю лікарів, і навпаки. Це породжує класичну проблему спарсеності, яка суттєво ускладнює як класичне user-based прогнозування, так і item-based підхід.

Враховуючи цю специфіку, у даному рішенні застосовано декілька ключових кроків.

Розрахунок «базових» оцінок: Оскільки у користувачів і лікарів можуть бути власні «упередження» до виставлення оцінок, система враховує глобальне середнє, а також індивідуальний «зсув» (bias) кожного

користувача й кожного лікаря. Це допомагає скоригувати «порожні» місця у даних і точніше спрогнозувати рейтинг навіть у випадках, коли схожість не може бути обрахована через відсутність даних.

Використання косинусної подібності: Для обрахунку схожості між користувачами (user-user) або лікарями (item-item) використовується косинусна метрика, що дозволяє зважувати внесок кожного учасника навіть у розріджених масивах, та ефективно масштабується на великі обсяги даних.

3.1.3 Алгоритмічна реалізація

В основі сучасних рекомендаційних систем лежить ідея виявлення латентних закономірностей у даних про взаємодію користувачів з об'єктами. У випадку веб-застосунку для вибору лікаря найочевиднішою і водночас найпрозорішою формалізацією є побудова матриці взаємодій користувач-контент. В нашому випадку це матриця, де кожен рядок відповідає окремому користувачу, а кожен стовпець – лікарю, якого бодай раз оцінювали пацієнти системи.

Формування матриці `user_item` – ключовий підготовчий крок, без якого неможливо реалізувати жоден класичний підхід до колаборативної фільтрації. Вибір саме `pivot`-таблиці, де пропущені значення (відсутні оцінки) заповнюються нулями, повністю виправданий: це дозволяє застосовувати алгебраїчні операції, зокрема швидке множення векторів і обрахунок косинусної схожості. Насправді, кожна строка такої матриці – це вектор, який описує переваги користувача у високовимірному просторі лікарів, що й відкриває можливість для використання класичних методів із теорії обробки сигналів та машинного навчання.

Вже на цьому етапі варто згадати, що навіть така проста структура створює низку теоретичних викликів. Передусім, матриця зазвичай сильно розріджена: більшість пацієнтів оцінює дуже малу частину лікарів, а велика

кількість лікарів отримує мало відгуків. Відповідно, заповнення пропущених значень нулями – це не лише технічний прийом, а й вибір гіпотези: відсутність оцінки інтерпретується як нульовий рівень довіри чи відсутність інформації, що не впливає на косинусну відстань між векторами.

Далі обчислюється глобальний середній рейтинг (грецька μ) – просте, але надзвичайно важливе статистичне узагальнення, що фіксує загальний «настрій» чи «схильність» до оцінювання в усій системі:

$$\mu = \frac{1}{|K|} \sum_{i,j \in K} R_{ij}. \quad (3.1)$$

де K – множина пар користувач-лікар із наявними оцінками.

Додавання персоналізованого зсуву (bias) для користувача та лікаря – це класичний підхід, відомий ще з робіт Yehuda Koren (Netflix Prize) [3], що дозволяє враховувати індивідуальні тенденції до заниження чи завищення оцінок. Наприклад, деякі користувачі дають виключно високі оцінки, а деякі лікарі, навпаки, отримують лише сувору критику.

Кодова реалізація цього етапу надзвичайно лаконічна (лістинг 3.1).

Лістинг 3.1 – Обчислення глобального середнього та індивідуальних зсувів оцінювання для користувачів і лікарів

```
global_mean = user_item.values[user_item.values > 0].mean()
user_bias = user_item.replace(0, np.nan).mean(axis=1) -
global_mean
doctor_bias = user_item.replace(0, np.nan).mean(axis=0) -
global_mean
```

Така компактність не лише спрощує підтримку системи, а й дозволяє масштабувати підхід на великі дані без втрати прозорості та відтворюваності результатів.

На наступному етапі будується матриця схожості між користувачами (user-user similarity) та лікарями (item-item similarity). Тут принципово важливою є косинусна міра схожості – класична метрика зі статистики та теорії інформації, яка ідеально підходить для роботи з розрідженими даними. Реалізація цього розрахунку у бібліотеці sklearn дозволяє уникнути ручної роботи з циклами та забезпечити коректність навіть при великій кількості пропущених даних:

$$S(u, v) = \frac{\sum_j R_{uj}R_{vj}}{\sqrt{\sum_j R_{uj}^2}\sqrt{\sum_j R_{vj}^2}}. \quad (3.2)$$

Аналогічно формується і матриця схожості між лікарями (лістинг 3.2).

Лістинг 3.2 – Матриця схожості між лікарями

```
user_similarity =
pd.DataFrame(cosine_similarity(user_item),
              index=user_item.index,
              columns=user_item.index)
```

Далі ключовим стає власне прогнозування рейтингу. Тут ми повертаємося до класичної user-based колаборативної фільтрації, де прогноз для лікаря, якого користувач ще не оцінював, базується на зваженому середньому оцінок тих користувачів, які є максимально подібними (згідно зі знайденою раніше матрицею схожості). Варто зауважити, що, згідно з теорією інформаційних фільтрів, оптимальність такого прогнозу досягається саме при використанні ваги, пропорційної коефіцієнту подібності. У вигляді коду це виглядає так (лістинг 3.3).

Лістинг 3.3 – Прогнозування рейтингу

```
for doctor in unrated_doctors:
    sim_scores = []
```

Продовження лістингу 3.3

```

ratings = []
for other_user, sim in similar_users.iteritems():
    r = user_item.at[other_user, doctor]
    if r > 0:
        sim_scores.append(sim)
        ratings.append(r)
if sim_scores:
    pred = np.average(ratings, weights=sim_scores)
else:
    pred = baseline_predict(user_id, doctor)
predictions[doctor] = pred

```

Цей фрагмент коду не просто розв'язує задачу прогнозу – він реалізує математичну формулу зваженого середнього, що класично записується як:

$$\hat{R}_{uj} = \frac{\sum_{v \in N_u(j)} S_{uv} R_{vj}}{\sum_{v \in N_u(j)} |S_{uv}|}, \quad (3.3)$$

де $N_u(J)$ – множина найбільш схожих до користувача u чи користувачів, які вже оцінювали лікаря J .

У випадку, коли не знайдено жодного релевантного користувача (наприклад, для нових лікарів чи при cold start), система повертається до baseline-прогнозу, що поєднує глобальне середнє та індивідуальні bias:

$$\hat{R}_{uj} = \mu + b_u + b_j. \quad (3.4)$$

Така гнучкість дозволяє системі працювати навіть у найгірших сценаріях розрідженості даних.

Кінцевий результат роботи коду – формування списку рекомендацій із прогнозованим рейтингом для кожного лікаря, відсортованого за спаданням (лістинг 3.4).

Лістинг 3.4. – Формування списку рекомендацій

```
recs = pd.DataFrame({'doctor_id':  
list(predictions.keys()),  
                    'predicted_rating':  
list(predictions.values())})  
return recs.sort_values('predicted_rating',  
ascending=False).head(top_k)
```

Ця структура даних ідеально підходить для інтеграції у сучасний веб-додаток: вона легко серіалізується у формат JSON, може бути відображена на інтерфейсі користувача і, головне, гарантує інтерпретованість кожного прогнозу.

Усе описане вище – приклад реалізації фундаментальних ідей рекомендаційних систем, де кожна строка коду відповідає певному теоретичному принципу, а кожна формула підкріплена багаторічною практикою використання у галузі інформаційного пошуку та машинного навчання.

3.1.4 Приклад роботи системи

Для конкретного користувача (наприклад, `patient_id=241`) система формує топ-рекомендації у вигляді списку лікарів із прогнозованим рейтингом задоволення (рисунок 3.1).

Результат є зрозумілим: перші у списку – ті лікарі, щодо яких прогнозована ймовірність позитивної оцінки є найвищою.

```

➤ Recommendations for user 241 (User-based):
  doctor_id predicted_rating
      102         4.72
      108         4.36
      106         4.27
      109         3.97
      104         3.90

Recommendations for user 241 (Item-based):
  doctor_id predicted_rating
      106         4.78
      108         4.45
      102         4.21
      109         3.85
      104         3.80

```

Рисунок 3.1 – Приклад роботи рекомендаційної системи

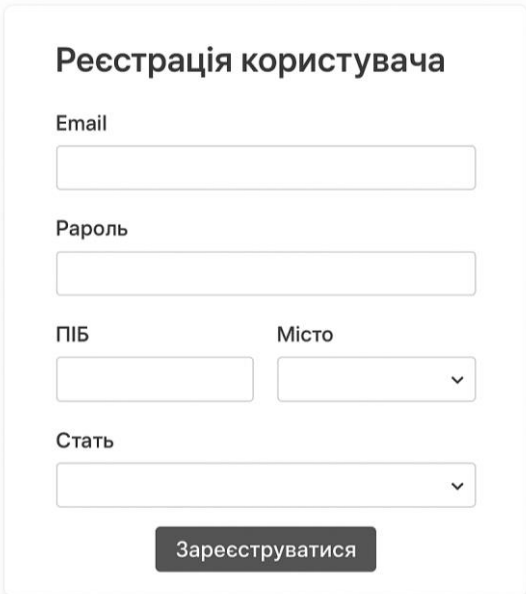
3.2 Розробка візуального інтерфейсу

Візуальний інтерфейс є ключовим компонентом системи рекомендацій, оскільки саме через нього користувачі взаємодіють із функціями веб-застосунку.

При проектуванні інтерфейсу були враховані сучасні принципи юзабіліті, що передбачають простоту використання, інтуїтивність навігації та зрозумілість представлення даних.

Основні компоненти інтерфейсу реалізовані з використанням фреймворку ASP.NET Core [8] MVC та EF Core[9], що забезпечує швидку та ефективну інтеграцію з розробленою базою даних на PostgreSQL через технологію Entity Framework Core. Це дозволило значно спростити процес створення форм і забезпечити безпечну взаємодію між клієнтською і серверною частинами застосунку.

Реєстрація користувачів реалізована через просту форму, яка включає такі поля як email, пароль, повне ім'я, стать та місто проживання. Після реєстрації роль користувача і дата створення профілю встановлюються автоматично. Це дозволяє мінімізувати кількість кроків для реєстрації, знижуючи бар'єр для нових користувачів (рисунок 3.2).



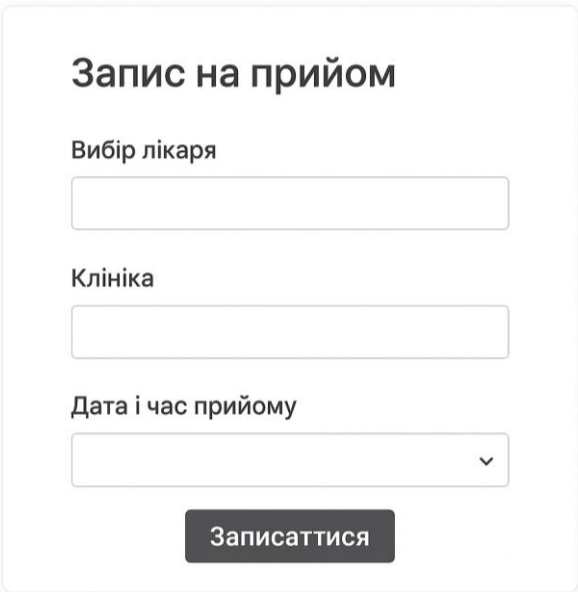
The screenshot shows a registration form with the following fields and controls:

- Реєстрація користувача** (User Registration)
- Email**: A text input field.
- Рароль** (Role): A text input field.
- ПІБ** (Full Name): A text input field.
- Місто** (City): A dropdown menu.
- Стать** (Gender): A dropdown menu.
- Зареєструватися** (Register): A dark button at the bottom.

Рисунок 3.2 – Інтерфейс реєстрації користувача

Форма запису на прийом забезпечує вибір лікаря та клініки, автоматичне заповнення доступних слотів часу прийому, що істотно підвищує зручність для пацієнта.

Статус запису автоматично оновлюється залежно від дій користувача або адміністратора (створено, скасовано, завершено) (рисунок 3.3).

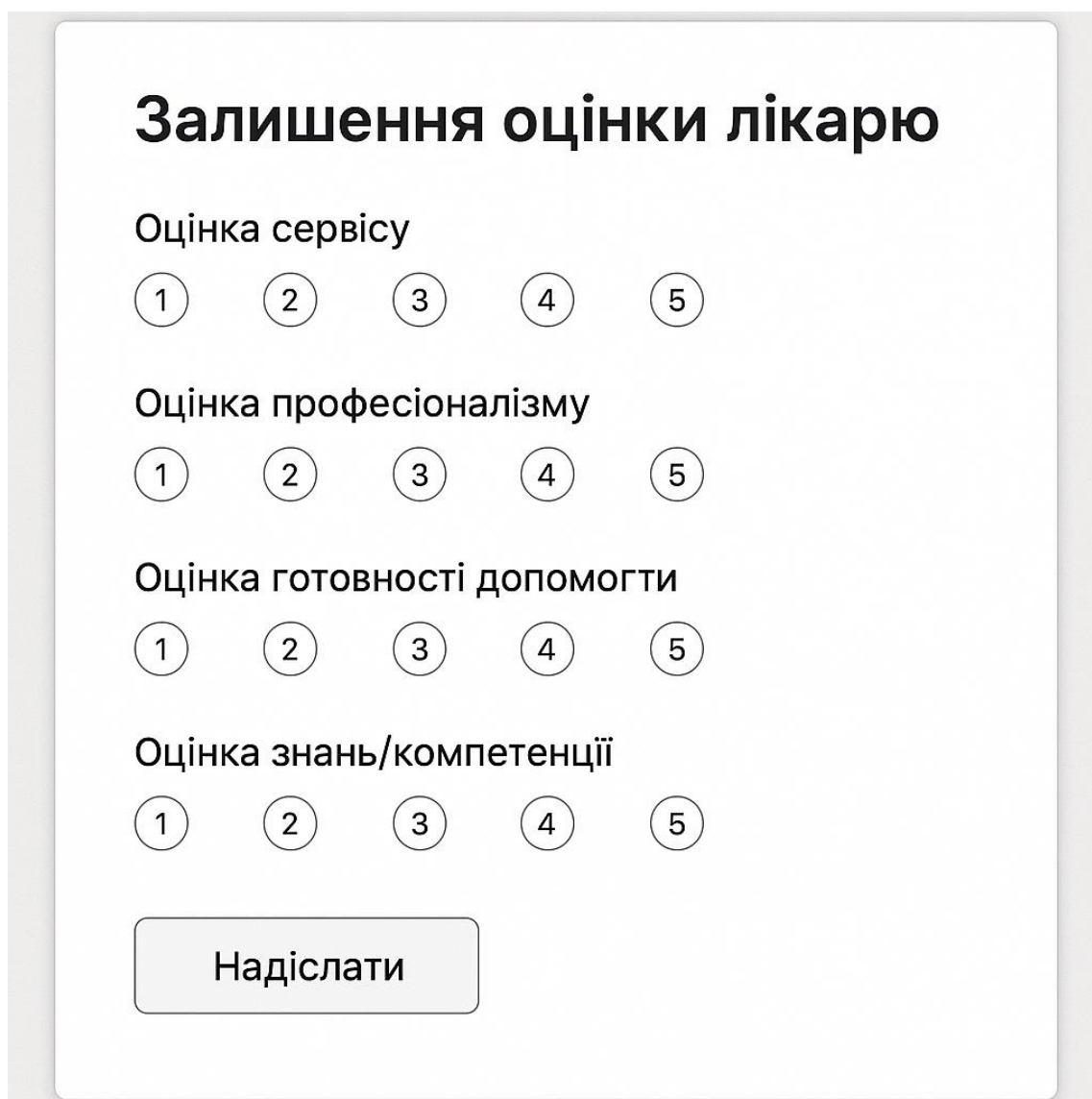


The screenshot shows an appointment form with the following fields and controls:

- Запис на прийом** (Appointment)
- Вибір лікаря** (Select Doctor): A text input field.
- Клініка** (Clinic): A text input field.
- Дата і час прийому** (Date and Time of Appointment): A dropdown menu.
- Записатися** (Book): A dark button at the bottom.

Рисунок 3.3 – Інтерфейс запису на прийом

Оцінювання лікарів після візиту реалізоване через інтуїтивний інтерфейс, що дозволяє виставляти рейтинги за чотирма ключовими критеріями: сервіс, професіоналізм, готовність допомогти, знання. Відгуки та оцінки автоматично додаються до профілю лікаря і впливають на майбутні рекомендації іншим користувачам (рисунок 3.4).



Залишення оцінки лікарю

Оцінка сервісу

① ② ③ ④ ⑤

Оцінка професіоналізму

① ② ③ ④ ⑤

Оцінка готовності допомогти

① ② ③ ④ ⑤

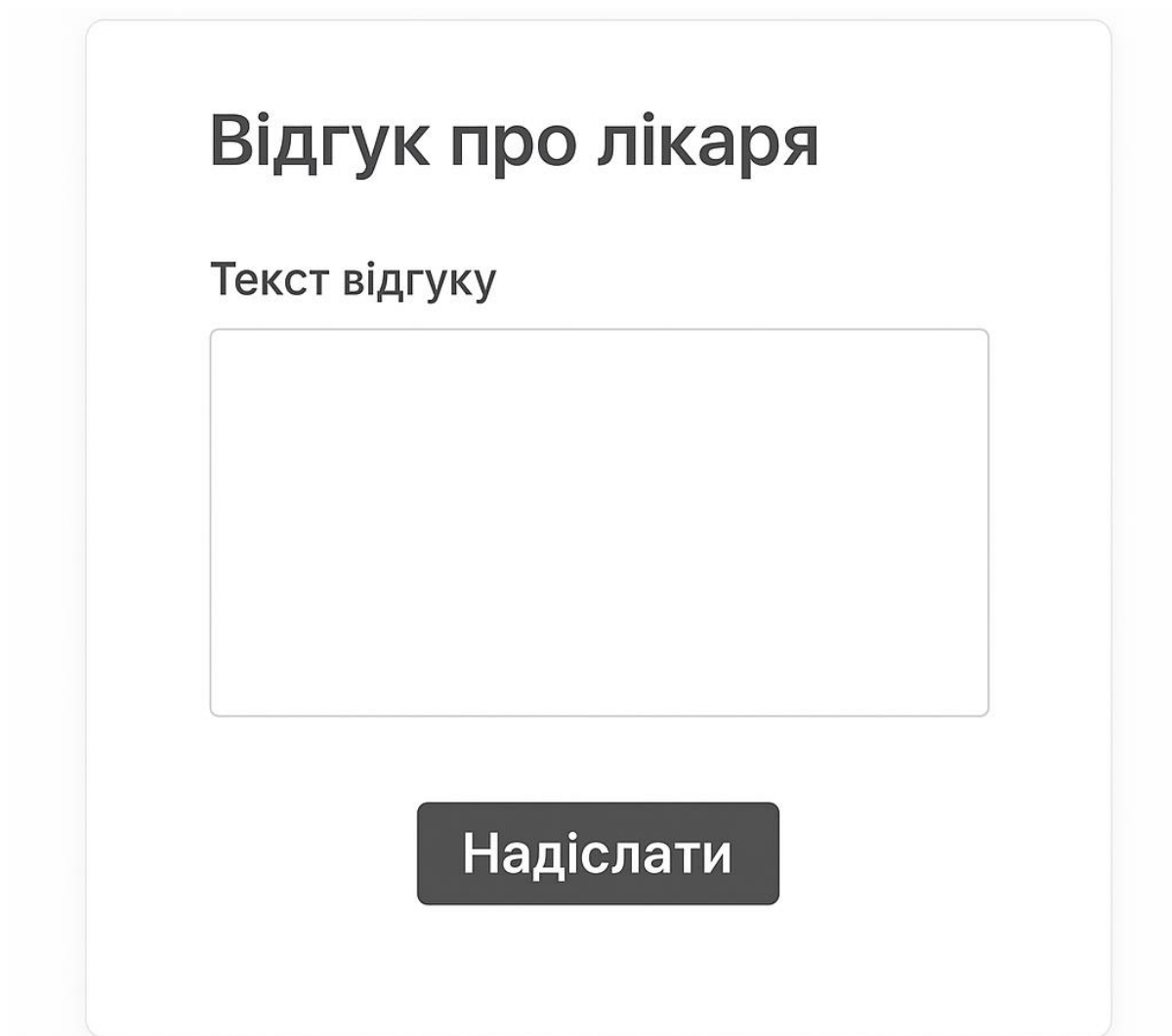
Оцінка знань/компетенції

① ② ③ ④ ⑤

Надіслати

Рисунок 3.4 – Інтерфейс залишення оцінки лікарю

Окремо реалізовано форму для написання текстових відгуків, яка дозволяє користувачам додати детальний опис свого візиту та вражень (рисунок 3.5).



The image shows a user interface for submitting a doctor review. It features a large heading, a label for the text input, a text area, and a submit button.

Відгук про лікаря

Текст відгуку

Надіслати

Рисунок 3.5 – Інтерфейс відгуку про лікаря

Таким чином, розроблений інтерфейс відповідає сучасним стандартам веб-дизайну, забезпечує високу швидкість взаємодії з користувачами, інтеграцію з рекомендаційними алгоритмами, а також дозволяє легко масштабувати систему в майбутньому.

ВИСНОВКИ

У межах даної кваліфікаційної роботи було розроблено веб-застосунок «Віртуальний помічник для запису на прийом до лікаря» з інтегрованою системою рекомендацій на основі колаборативної фільтрації. Здійснено повний цикл створення системи: від збору та підготовки вихідних даних, проектування та впровадження бази даних, розробки алгоритму рекомендацій до створення повноцінного веб-інтерфейсу для кінцевих користувачів. Розроблена система має потенціал для забезпечення високої ефективності та стабільності роботи в умовах реального використання. Використання колаборативної фільтрації дозволяє значно підвищити персоналізацію рекомендацій, скоротити час на пошук відповідного спеціаліста та покращити загальний досвід користувачів. Таким чином, результати цієї роботи мають важливе значення для цифровізації медичних послуг та можуть бути використані як основа для створення інших персоналізованих сервісів у сфері охорони здоров'я.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 De Croon R., Van Houdt L., Htun N. N., Štiglic G., Vanden Abeele V., Verbert K. Health Recommender Systems: Systematic Review. *Journal of Medical Internet Research*. 2021. 23(6): e18035. URL: <https://www.jmir.org/2021/6/e18035/> (date of access: 24.06.2025).

2 Sae-Ang A., Chairat S., Tansuebchueasai N., Fumaneeshoat O., Ingviya T., Chaichulee S. Drug Recommendation from Diagnosis Codes: Classification vs. Collaborative Filtering Approaches. *International Journal of Environmental Research and Public Health*. 2023. 20(1): 309. URL: <https://www.mdpi.com/1660-4601/20/1/309> (date of access: 24.06.2025).

3 Ерошенко О. Рекомендаційні системи в дистанційному навчанні. *Наука онлайн: міжнародний електронний науковий журнал*. 2018. № 6. URL: <https://nauka-online.com/publications/information-technology/2018/6/rekomendatelnnye-sistemy-v-distantsionnom-obuchenii/> (дата звернення: 24.06.2025).

4 Ajitsaria A. Build a Recommendation Engine With Collaborative Filtering – Real Python. URL: <https://realpython.com/build-recommendation-engine-collaborative-filtering/> (date of access: 24.06.2025).

5 Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: MIT Press, 2016. 775 p.

6 James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. 2nd ed. New York: Springer, 2021. 606 p.

7 Bird S., Klein E., Loper E. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Sebastopol, CA: O'Reilly Media, 2009. 504 p.

8 ASP.NET Core documentation – Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0> (date of access: 24.06.2025).

- 9 Entity Framework Core documentation – Microsoft Learn. URL: <https://learn.microsoft.com/en-us/ef/core/> (date of access: 24.06.2025).
- 10 PostgreSQL 17.5 Documentation. URL: <https://www.postgresql.org/docs/current/index.html> (date of access: 24.06.2025).
- 11 Docker Documentation. URL: <https://docs.docker.com/> (date of access: 24.06.2025)
- 12 Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*. 2009. Vol. 42, No. 8. P. 30–37.