

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ПАРАСОЧКІН ВІТАЛІЙ ВОЛОДИМИРОВИЧ

Допускається до захисту:

в.о. завідувача кафедри  
інформаційних технологій  
канд. техн. наук, доцент

\_\_\_\_\_ О. В. Зелінська

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

**РЕКОМЕНДАЦІЙНА СИСТЕМИ ДЛЯ ПОБУДОВИ  
МАРШРУТУ ТУРИСТИЧНИХ ПОДОРОЖЕЙ НА ОСНОВІ  
ВПОДОБАНЬ КОРИСТУВАЧА**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (магістерська) робота

Науковий керівник:

Антонов Ю.С., кан. фіз.-мат. наук, доцент,  
доцент кафедри інформаційних  
технологій

\_\_\_\_\_  
(Підпис)

Оцінка \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: \_\_\_\_\_  
(Підпис)

Вінниця 2024

## ЗМІСТ

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| ВСТУП .....                                                                   | 4  |
| РОЗДІЛ 1 .....                                                                | 6  |
| АНАЛІЗ ТА КЛАСИФІКАЦІЯ ІСНУЮЧИХ СИСТЕМ.....                                   | 6  |
| 1.1 Поняття експертних систем та їх класифікація. ....                        | 6  |
| 1.2 Рекомендаційні системи та особливості їх використання .....               | 15 |
| 1.3 Огляд аналогів та існуючих технологій .....                               | 23 |
| Висновок до першого розділу.....                                              | 25 |
| РОЗДІЛ 2 .....                                                                | 27 |
| ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМИ ТА ПОБУДОВА МОДЕЛІ<br>ПРОГРАМНОГО ПРОДУКТУ ..... | 27 |
| 2.1 Визначення ключового функціоналу веб-додатку .....                        | 27 |
| 2.2 Збір даних для відображення можливих варіантів.....                       | 35 |
| 2.3 Взаємодія користувача з сайтом.....                                       | 38 |
| Висновок до другого розділу .....                                             | 45 |
| РОЗДІЛ 3 .....                                                                | 47 |
| РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ .....                                                  | 47 |
| 3.1 Реалізація front-end частини.....                                         | 47 |
| 3.2 Реалізація back-end частини.....                                          | 58 |
| 3.3 Результати роботи рекомендаційної системи .....                           | 62 |
| Висновок до третього розділу.....                                             | 65 |
| ВИСНОВОК.....                                                                 | 67 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....                                               | 69 |

## АНОТАЦІЯ

**Парасочкін В.В.** Рекомендаційна система для побудови маршруту туристичних подорожей на основі вподобань користувача. Спеціальність 122 “Комп’ютерні науки”, Освітня програма “Комп’ютерні технології обробки даних (Data Science)”. Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній роботі представлено розробку рекомендаційної системи та веб-додатку для підбору подорожей по Україні. Досліджено основні аспекти рекомендаційної та експертної системи, методи обробки даних та підходи до розробки рекомендаційних систем. Запропонована система базується на аналізі взаємодії користувача з маршрутами і застосовує персоналізовані алгоритми для генерації рекомендацій.

Ключові слова: рекомендаційна система, експертна система, аналіз даних, веб-додаток, подорож, туризм.

73 с., 15 рис., 62 джерела.

**Parasochkin V.V.** Recommendation System for Building Tourist Travel Routes Based on User Preferences. Specialty 122 “Computer Science”, Educational Program “Computer Data Processing Technologies (Data Science)”. Vasyl Stus Donetsk National University, Vinnytsia, 2024.

The qualification paper presents the development of a recommendation system and a web application for selecting travel routes across Ukraine. The study examines key aspects of recommendation and expert systems, data processing methods, and approaches to the development of recommendation systems. The proposed system is based on the analysis of user interaction with routes and employs personalized algorithms to generate recommendations.

Keywords: recommendation system, expert system, data analysis, web application, travel, tourism.

73 p., 15 fig., 62 sources.

## ВСТУП

У сучасному світі використання експертних та рекомендаційних систем стало однією з важливих складових багатьох галузей, починаючи від науки і закінчуючи бізнесом. Ці системи сприяють значному покращенню різних процесів та рішень. У науковій сфері вони допомагають вченим аналізувати складні дані та робити прогнози, що сприяє новим відкриттям. У бізнесі такі системи автоматизують процеси, підтримують прийняття стратегічних рішень та покращують обслуговування клієнтів, тим самим підвищуючи ефективність та продуктивність.

Загалом, експертні та рекомендаційні системи стають невід'ємною частиною сучасного світу, відкриваючи нові перспективи для інновацій та покращень у різних галузях. Внаслідок цього зростає рівень ефективності та продуктивності в різних сферах життя. Хоча пересічній людині може бути незрозуміло, що таке експертна чи рекомендаційна система, такі системи вже протягом майже півстоліття активно використовуються у повсякденному житті, залишаючись непомітними для більшості людей.

У рамках цієї роботи ми зосередимося на створенні веб-сайту у вигляді рекомендаційної системи, яка буде пропонувати місця для відвідування та формувати маршрути туристичних подорожей на основі вподобань користувача. Перший етап включатиме огляд існуючих аналогічних технологій, збір даних та дослідження переваг і недоліків наявних сайтів. Це дозволить використовувати ці знання для створення власного продукту.

Другий етап передбачатиме безпосереднє створення продукту за допомогою сучасних веб-технологій, його впровадження та розміщення в мережі Інтернет для відкритого користування іншими користувачами. У цій роботі ми поєднаємо знання про рекомендаційні та експертні системи з практичним створенням повноцінного веб-сайту, що дозволить отримати високоякісний продукт.

Таким чином, дана робота сприятиме покращенню розуміння того, як рекомендаційні системи можуть бути інтегровані у веб-технології для створення

корисних та інноваційних продуктів. Це не лише підвищить рівень обслуговування користувачів, але й сприятиме розвитку нових підходів у побудові маршрутів туристичних подорожей, враховуючи індивідуальні вподобання кожного користувача.



## **РОЗДІЛ 1**

### **АНАЛІЗ ТА КЛАСИФІКАЦІЯ ІСНУЮЧИХ СИСТЕМ**

#### **1.1 Поняття експертних систем та їх класифікація.**

Варто почати з того, що єдиного правильного визначення, для такого широкого поняття, як експертна система не існує. Але слід зазначити, що натомість люди з певних галузь дають визначення цьому терміну виходячи від специфіки, структури та сфер застосування експертних систем. Однак можна сміливо дати узагальнене твердження.

Експертна система – це один з різновидів штучного інтелекту, який може імітувати знання та приймати в певній мірі успішні рішення на основі цих знань, в одній з визначених галузей. Цю систему для простоти і зрозумілості можна уявити як певний тип комп'ютерної програми, яка може містити в собі глибокі знання або певним досвідом в сфері спеціалізації такій як фінанси, медицина тощо. [1]

Тобто можна сказати, що це система, яка містить в собі знання і здібності одного або як буває в більшості випадків декількох експертів в певній галузі застосування і ця система може здійснювати логічні висновки на основі тих знань які має, що допомагає кінцевому користувачу вирішувати певний і специфічний тип проблем або завдань.

Також такі системи можуть проводити консультування, тестування, проектування, навчання, діагностування тощо. Також її можна охарактеризувати як систему, яка включає в себе базу знань, на основі яких вже може вирішувати завдання в тій чи іншій галузі. Також її ще можна охарактеризувати як методологія адаптації алгоритму пошуку успішних рішень. Простими словами, експертна система – це система, що імітує здатність людини-експерта в вузько направлених галузі приймати рішення.

Продовжуючи, варто навести хронологію того, як виникли експертні системи, та в яких сферах вони застосовувались. Отож, звідки почалась історія експертних систем?

- У 1960–1970 роках експертні системи були першою і доволі успішною спробою застосування штучного інтелекту [2], [3]. Перші кроки в цьому напрямку були спрямовані на те, щоб створити програми, які могли б вирішувати конкретні задачі в вузько направленій області, використовуючи знання, вбудовані в них за допомогою експертів в даній предметній області. Для прикладу, можна навести програму Dendral [4], розроблену в 1965 році, яка допомагала хіміка і людям пов'язаній з даною професією в ідентифікація органічних сполук.
- У наступне десятиріччя, 1970-1980 роках, експертні системи набули ширшого кола поширення, і почали використовуватись у різних галузях, таких як медицина, фінанси, інженерія, та управління. Можна навести як приклад, систему MYCIN [5], яку створили в 1976 році, і яку використовували для діагностики інфекційних хвороб шкіри та м'яких тканин. Щодо фінансової сфери, то там експертні системи застосовувались для прогнозу росту ринку акцій та управління портфелем для купівлі або продажу акцій.
- В 1980-1990 роках експертні системи отримали більше потужностей, і стали більш розповсюдженими. Наприклад, систему XCON [6], інженери використовували уже для автоматизації процесу проектування комп'ютерів.
- В 1990-2000 роках, підвищилась мультимедійність і інтерактивність в різноманітних сферах життя. Не обійшов цей прогрес стороною і сферу експертних систем. Так, до прикладу з'явилися інтерактивні системи як INTERNIST/CAD [7], яка допомагала лікарям діагностувати хвороби. В деяких з таких систем був перелік питань і відштовхуючись від відповіді пацієнта, можна було з відсотковим відношенням визначити який діагноз у пацієнта. В той же час, в фінансах покращили можливості попередніх років. Так, в той період часу експертні системи використовувалися для розпізнавання шахраїв та управління ризиками на ринку.

- З 2000 року по 2010 , експертні системи отримують можливість інтегруватись з іншими технологіями. Наприклад, IBM Watson Health [8] використовувався для того, щоб проводити аналіз певних медичних даних та допомагати в діагностиці. У фінансах, фінтех-стартапи використовують експертні системи для аналізу та рекомендацій як правильно інвестувати.
- Починаючи з 2010 і до сьогодні, експертні системи отримують все більше уваги, та стають більш ефективними. Наприклад, Ada Health [9], може допомогти пацієнтам у самодіагностиці, та радить як правильно в подальшому лікуватись. Фінансові радники використовують експертні системи для персоналізованих порад, щодо управління фінансами.

Експертна система на початку свого творення, розроблявся людьми, які досліджували штучний інтелект в 60-ті та 70-ті, але отримала комерційне застосування вже починаючи з 80-х.

- Експертна система моделює механізм мислення людини при вирішенні проблем в заданій предметній області, однак варто зазначити, що експертні системи ні в якому разі не повинні повністю показувати психологічну модель експерта в певній галузі, а тільки відтворювати за допомогою комп'ютера деякі прийоми розв'язання проблем, які експерт може використовувати;
- Система не тільки виконує обчислення, але й робить висновки на основі своїх знань. Ці знання, які зазвичай описані спеціальною мовою називають базою знань. [10] Вона потрібна для зберігання даних, що описують розглянуту область і також можуть містити правила, що описують перетворення, власне кажучи, даних цієї області.
- Завдяки своїй практичній орієнтованості, експертні системи стають незамінними інструментами в наукових дослідженнях та комерційній діяльності, дозволяючи вирішувати складні задачі, які раніше були доступні лише для досвідчених експертів.

- Важливою характеристикою експертної системи є її продуктивність, яка визначається часом, необхідним для отримання результату, та його достовірністю.
- Експертна система може не лише генерувати рішення, але й вміти чітко та аргументовано пояснювати, чому саме це рішення було запропоновано. Користувач повинен мати доступ до всієї інформації, що підтверджує обґрунтованість прийнятого рішення, адже це показує наглядний звіт роботи системи та сприяє кращому її розумінню.

Експертні системи зазвичай складаються з чотирьох ключових компонентів:

1. База знань - Цей сховище інформації містить правила, факти та приклади, які експертна система використовує для прийняття рішень. Прийнято описувати базу знань на фізичному та логічному рівнях. Фізичний рівень – рівень фізичного представлення даних описує адресування та організацію даних у зовнішній пам'яті комп'ютера. Логічний рівень даних представляє собою абстрактну модель предметної області, що описує її структуру, зв'язки між елементами та кількісні характеристики у вигляді даних.
2. Інтерфейс [11] користувача - завдяки цьому інтерфейсу користувач може взаємодіяти з системою, надаючи їй інформацію та отримуючи рекомендації. Користувач спілкується з експертною системою за допомогою інтерфейсу, надаючи їй дані та команди, які містять інструкції щодо обробки знань. Система ж надає результати у вигляді значень, присвоєних змінним. Користувачі експертних систем мають можливість отримати не лише результат, але й розгорнуте пояснення того, як він був отриманий
3. Машина логічного пояснення [12] - ця підсистема пояснює користувачеві, як експертна система прийшла до того чи іншого рішення. Розрізняють два види пояснень:

- Пояснення, які надаються користувачеві за вимогою, тобто користувач у будь-який момент може зажадати від експертної системи пояснення своїх дій.
- Пояснення отриманого рішення проблеми. Користувач може захотіти отримати пояснення після одержання рішення як воно було отримано. Експертна система повинна пояснити кожний етап своїх роздумів, що ведуть до знаходження розв'язку задачі. Незважаючи на те, що технологія роботи з експертною системою не є простою, інтерфейс користувача в цьому плані є дружнім і зрозумілим, тому і не виникає ніяких труднощів при веденні так званого «діалогу».

4. Машина виведення [13] - цей компонент використовує правила бази знань, щоб вивести нові знання з інформації, наданої користувачем.

На рисунку 1.1 можна побачити приблизну схему складу експертної системи:

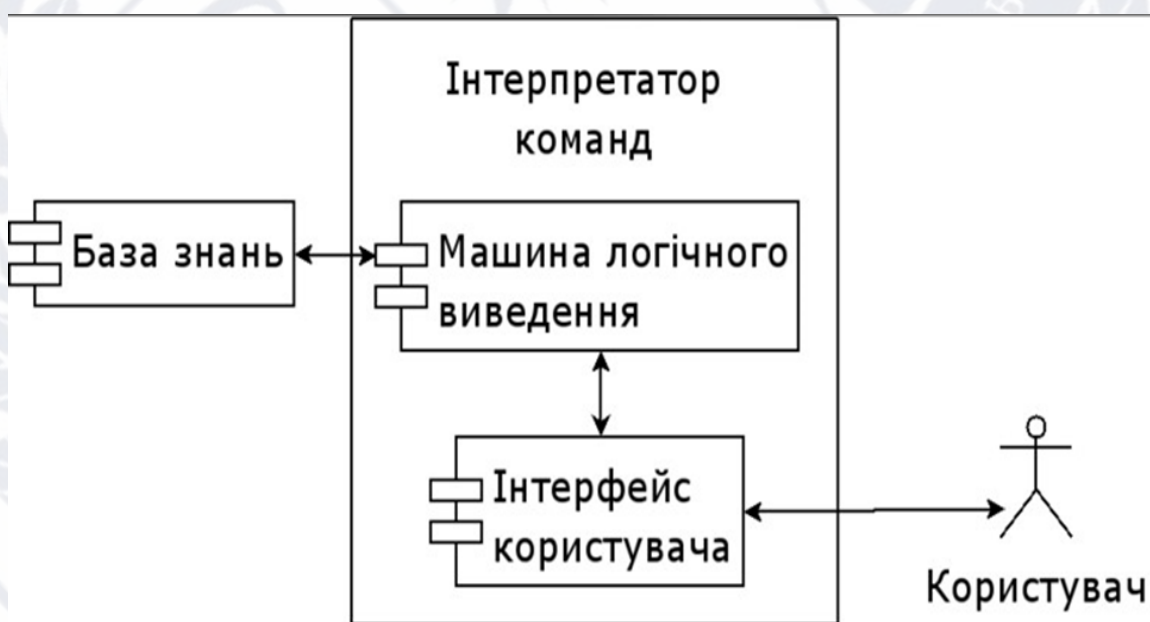


Рисунок 1.1 – Структура експертної системи

Основою ЕС виступають база знань (БЗ) та процедура висновку, які неможливо розділити і необхідно звертати увагу одночасно.

Класифікація експертних систем відбувається за певними ознаками, в залежності від контексту, проблемою, яку треба вирішити, області застосування тощо.

Розглянемо основні принципи класифікації експертних систем по черзі:

За метою створення:

1. Навчальні експертні системи [14]:

Мета таких експертних систем полягає в навчанні фахівців певної предметної області. Така експертна система надає наступні функціональності:

- Надання знання та пояснення з певної теми.
- Проводження тестування та оцінювання знань користувачів.
- Моделювання ситуації та дозволяє кінцевому користувачеві практикувати прийняття рішень.

До прикладів таких систем можна навести тренажери для навчання пілотів літаків і системи для навчання медичних працівників діагностиці захворювань.

2. Системи автоматизованого прийняття рішень [15]:

Метою таких експертних систем є автоматичне прийняття рішень в певних ситуаціях без втручання людини. До функціональності цього типу експертних систем можна віднести:

- Аналіз дані з датчиків та інших джерел.
- Використання знання та правила, щоб вибрати оптимальний варіант дій.
- Керування роботою технічних систем або процесів.

До прикладів таких систем можна віднести системи автоматичного виробництва та системи діагностики та прогнозування несправностей.

3. Експертні системи тиражування знань експертів [16]:

Метою цього типу експертних систем може бути збереження та власне тиражування знань та досвіду експертів в певній предметній області.

До функціональних особливостей можна сміливо віднести:

- Збирання та процес формалізації вигляду знань експертів.
- Надання доступу до таких знань користувачам.

- Дозвіл оновлювати та доповнювати базу знань.

До прикладів можна віднести системи проектування технічних систем та системи підтримки прийняття рішень.

4. Експертні системи для вирішення задач, також відомі як консультативні експертні системи [17]:

Мета цих систем полягає в наданні користувачам знань та рекомендацій необхідних для вирішення конкретних задач.

Функціональності наступні:

- Аналіз даних наданих користувачем.
- Використання знань та правил, для можливого запропонування можливих вирішень проблеми.
- Пояснення ходу міркувань.
- Інколи змога надавати інструменти та ресурси, які можуть допомогти користувачеві виконати задачу.

До прикладів можна віднести системи фінансового консультування та юридичні консультаційні системи.

Експертні системи також класифікують за типом задачі, яку вони вирішують:

1. Діагностичні системи [18] – системи, які використовуються для визначення причин проблеми або стану на основі опису, який представили. Як приклад, можна звернути увагу на систему діагностики автомобільних поломок.
2. Системи прогнозування [19] – системи, які можуть давати прогноз для майбутнього розвитку подій, на основі поточної ситуації або історичних даних. Наприклад, системи прогнозу погоди, системи оцінки майбутнього врожаю чи аналітичні ринкові системи для прогнозу в економічній сфері діяльності.
3. Навчальні системи – системи, які використовуються власне для навчання користувачів цією системою, або інших систем на основі певного накопиченого експертного знання. До прикладу можна сміливо віднести

системи вивчення мов програмування або система тренування медичних студентів.

4. Системи керування [20] – системи, які приймають рішення та видають певні інструкції і команди відповідно для керування процесами або об'єктами в реальному чи навіть віртуальному середовищі. Прикладом може бути система управління рухом транспорту.
5. Системи інтерпретації даних [21] – системи, яка вирішує одне з найпоширеніших і найпопулярніших завдань для експертних систем. В контексті інтерпретації даних, мається на увазі дослідження змісту даних, в результаті має бути коректна відповідь. Зазвичай, в таких системах передбачається багатоваріантний аналіз даних. До прикладу можна віднести систему, для ідентифікації та виявлення різноманітних видів океанських плавучих засобів.
6. Системи планування – системи, які знаходять план дій, які належать до об'єктів, здатних здійснювати деякі функції. Ці експертні системи використовують моделі поведінки реальних об'єктів для логічного виведення наслідків запланованої діяльності. З прикладів можна виділити системи для планування поведінки робота, або планування промислових замовлень.
7. Системи моніторингу [22] – системи, які здатні безперервно слідкувати за даними в реальному часі і сповіщати про вихід показників за певні встановлені норми. Однак, з головних проблем таких систем можна відзначити не спрацьовування в ситуаціях коли показники вийшли за норми, і помилкове сповіщення тоді, коли показники в нормі. Суть проблеми полягає у тому, що симптоми ситуацій коли система помиляється є неоднозначними, а також потребують урахування тимчасового контексту. До прикладу таких систем можна віднести систему FALCON, яка допомагає відслідковувати датчики на хімічному заводі.
8. Системи, що можуть приймати рішення замість людини [23] – експертні системи які приймають рішення на основі знань, є ключовими в

автоматизації аналізу відповідей та оцінювання. Вони працюють за алгоритмом штучного інтелекту і використовують бази знань, що зберігають досвід фахівців у конкретній галузі.

Експертні системи можуть аналізувати відповіді студентів у тестових середовищах, використовуючи алгоритми, які об'єктивно оцінюють різноманітні типи запитань. Це суттєво зменшує суб'єктивність оцінки, дозволяє відстежувати прогрес у навчанні та надавати студентам зворотний зв'язок, який би зазвичай надавав викладач.

Такі системи особливо корисні в автоматизованих системах контролю знань. Вони здатні проводити аналіз відповідей за різними критеріями, зокрема за точністю, повнотою відповіді, використанням правильних термінів, логічною побудовою відповіді тощо. Також вони можуть допомогти у виявленні прогалин у знаннях студента та забезпечувати гнучкість у навчанні за рахунок індивідуальних підходів до оцінювання.

Рішення таких систем ґрунтується на фактах і правилах, закладених у базі знань, тому вони імітують процес мислення людини-експерта, але при цьому їхні висновки об'єктивні та швидкі. Це дозволяє значно підвищити ефективність освітніх процесів, зокрема в дистанційному навчанні та у великих групах студентів, де ручне оцінювання займає багато часу.

Сучасні експертні системи використовують машинне навчання, що дозволяє їм самостійно покращувати свою базу знань на основі нових даних та аналізів. Це означає, що з часом такі системи стають більш точними та здатними до прийняття все складніших рішень.

В останній класифікації розглянемо можливість класифікувати експертні системи за зв'язком з реальним часом. Отож зв'язок експертних систем з реальним часом можна розділити на три основні групи:

1. Статичні експертні системи – системи, які працюють без прив'язки до часу, тобто можна сказати, що такі системи проводять аналіз вхідних даних

та роблять вихідні результати на основі сталих правил та бази знань. Виходячи з цього, можна зробити висновок що в них немає потреби в негайному і швидкому відгуку, тобто вони використовуються там, де час не є критичним, або коли важлива лише інформація, яку проаналізувала експертна система, без необхідності швидкого втручання. До прикладу можна віднести системи для діагностики дефектів в автомобілі.

2. Квazідинамічні (псевдо реальні) експертні системи – системи, які можуть забезпечувати швидку реакцію на зміни, але вони не обробляють інформацію в реальному часі. Замість цього вони використовують попередньо збережені дані для швидкого аналізу та вироблення рішень. Такі системи можуть надавати відповіді з невеликою затримкою, але вони не можуть гарантувати миттєвої реакції на події. До прикладу можна віднести складу мікробіологічну експертну систему, в якій змінюються лабораторні вимірювання.
3. Динамічні експертні системи – системи, які здатні обробляти вхідні дані та генерувати вихідні результати в реальному часі. Експертні системи співпрацюють з об'єктними датчиками у реальному часі, неперервно обробляють та аналізують дані, що надходять. Вони надають можливість негайно реагувати на зміни в навколишньому середовищі та приймати рішення у реальному часі. Такі системи дуже необхідні за умов, де потрібно швидко реагувати, тобто де час є критичним. Наприклад, у системах моніторингу в реанімаційних палатах.

### **1.2 Рекомендаційні системи та особливості їх використання**

Рекомендаційна система [24,25] – різновид системи, яка фільтрує інформацію і створює перелік об'єктів за рейтингом, яким користувач може надати перевагу. Для того, щоб надати найбільш рентабельну і корисну інформацію, при відтворенні рейтингу об'єктів, рекомендаційна система використовує інформацію з, так званого, профілю користувача.

Рекомендаційні системи з'явилися відносно недавно, хоча і історія появи систем має місце в далекому минулому, справжній виток їхнього розвитку і використання пов'язаний з появою на світ комп'ютерів та інтернету:

- 1990-2000 роки це перші згадки про рекомендаційні системи, які можна знайти в психології та маркетингу, де ще оцінювали питання про вплив рекомендацій на поведінку споживача продукту. Але як тільки з'явилися комп'ютери і збільшився обсяг даних, то тоді і збільшилися можливості для створення більш складних алгоритмів для аналізу рекомендацій
- В 2000-2010 роках відбувся значний розвиток у вивченні алгоритмів рекомендацій, саме в цьому періоді вченим з Массачусетського технологічного інституту (MIT) вдалось розробити алгоритм колаборативної фільтрації, який пізніше стане основою і базою для багатьох сучасних рекомендаційних систем.
- 2010 – дотепер. Разом з тим як з'явилися інтернет платформи, такі як Amazon, Netflix, Spotify, рекомендаційні системи отримали широке використання для збільшення рівня і кількості продажів. Також вони використовувались для того щоб користувач покращив користувацький досвід сайтом. За останні роки, завдяки стрімкому прогресу штучного інтелекту, сучасні технології, у тандемі з машинним навчанням, забезпечують можливість створення систем рекомендацій, які є точнішими та більш персоналізованими.

Загалом, можна сказати що, історія рекомендаційних систем є навіть історією злиття наукових досліджень, технологічних новацій та їх практичного застосування з метою збільшення зручності для користувачів та покращення їхнього онлайн-досвіду.

Рекомендаційна система також може бути в вигляді програмного забезпечення або певного алгоритму, який аналізує дані про користувачів, які користуються цими системами, і також аналізує взаємодію з певними об'єктами, як вже було сказано раніше. До прикладу користувач може взаємодіяти з

товарами на сайті магазину, з фільмами, музикою тощо. Після цього рекомендаційна система надає персоналізовані рекомендації після таких взаємодій. Ці системи застосовуються щоб допомогти користувачам знайти необхідний товар на сайті, або контент, який, скоріше за все, вони знайдуть корисним або цікавим, на основі їхній попередній дій або вподобань.

Рекомендаційні системи всередині себе можуть послуговуватись різноманітними методами і техніками, такими як фільтрація заснована на вмісті (аналіз властивостей об'єктів та те як ці об'єкти поєднуються з вподобаннями користувача), колаборативна фільтрація (аналіз взаємодій між користувачем та об'єктами) або гібридні підходи, що поєднують обидва способи.

Такі системи використовуються в різних галузях, від торгівлі в онлайн магазинах до вибору музики або відео на стрімінгових платформах, при цьому спрощуючи процес прийняття рішень для користувача, щоб задовільнити його в певному запиті.

Існує кілька типів рекомендаційних систем, кожен з яких використовує різні методи та підходи для надання персоналізованих рекомендацій користувачу:

1. Фільтрація на основі контенту [26] – цей метод для фільтрації рекомендаційної системи, заснований на тому, що персоналізує пропозиції на основі дій користувача. Вона аналізує різноманітні атрибути та ключові слова, пов'язані з об'єктами в базі даних, наприклад товари в онлайн-магазині, та співставляє їх з тим що використовував в пошуках або переглядав користувач. Тобто, можна сказати що створює певний профіль на основі взаємодій користувача, таких як покупки, оцінки, пошукових запитів і натискань.

Процес роботи фільтрації на основі контенту можна розбити на кілька ключових етапів:

- Збір контенту та визначення характеристик. На цьому етапі відбувається збір даних про об'єкти, які включені до рекомендаційної системи. Наприклад, якщо мова йде про фільми, то

система збиратиме інформацію про жанри, акторів, режисерів, тривалість фільму та інші характеристики. Для інших типів об'єктів (книги, музика, товари) цей підхід буде аналогічним: збираються релевантні атрибути для аналізу.

- Створення профілю користувача. На основі активності користувача (перегляди, оцінки, покупки, пошукові запити) система формує профіль, що відображає його вподобання та інтереси. Цей профіль може включати різноманітні параметри, що описують уподобання користувача, такі як жанри, категорії товарів, теми тощо.
- Співставлення контенту з профілем користувача. Після збору інформації про контент та створення профілю система починає аналізувати схожість між характеристиками об'єктів і вподобаннями користувача. Чим більше атрибутів об'єкта відповідають параметрам профілю, тим вища ймовірність того, що цей об'єкт буде запропонований у рекомендаціях.
- Ранжування рекомендацій. Система розташовує об'єкти за ступенем відповідності профілю користувача. Ті об'єкти, які найбільше відповідають його інтересам, будуть показані на перших позиціях. Це забезпечує користувачеві найбільш релевантні рекомендації.
- Оновлення рекомендацій. Рекомендаційна система постійно оновлює свої пропозиції відповідно до нових дій користувача. Наприклад, якщо користувач змінює свої уподобання або взаємодіє з новим контентом, його профіль оновлюється, і система адаптує рекомендації, щоб відображати ці зміни.

Таким чином, фільтрація на основі контенту дозволяє рекомендаційній системі гнучко та динамічно реагувати на зміни в поведінці користувача, забезпечуючи релевантність та точність рекомендацій на кожному етапі взаємодії з системою. Це особливо корисно тоді, коли даних про користувачів ще недостатньо для аналізу та рекомендацій.

2. Колаборативна фільтрація [27] – це поширений метод персоналізованої системи рекомендацій, який має здатність фільтрувати інформацію, наприклад дані про взаємодію з іншими подібними користувачами. Тобто цей підхід рекомендує об'єкти, які інші користувачі з подібними інтересами чи вподобаннями оцінили і звернули увагу. Оскільки він працює на основі прогнозування оцінок користувачів, вважається, що він виконує завдання регресії. Колаборативна фільтрація додатково поділяється на два типи:

- Колаборативна фільтрація користувач-користувач (user-user collaborative filtering) [28] – це метод рекомендаційної системи, який в своїй основі базується на спільному споживанні контенту між користувачами. Він слідкує за вподобаннями користувача з схожими інтересами на основі його взаємодії з іншими об'єктами, і на основі цього пропонує ці об'єкти користувачам з схожими інтересами відповідно. Використовує в своїй основі матрицю рейтингів, де рядки представлені безпосередньо користувачами, а стовпці – об'єкти споживання (фільми, продукти). Сама матриця заповнена рейтингами або оцінками, які вказують на взаємодію користувачів з об'єктами. Цей метод орієнтується на споживанні користувача, і ставить акцент саме на інтересах та вподобаннях користувачів, а не на характеристиках контенту.

### Колаборативна фільтрація користувач-користувач

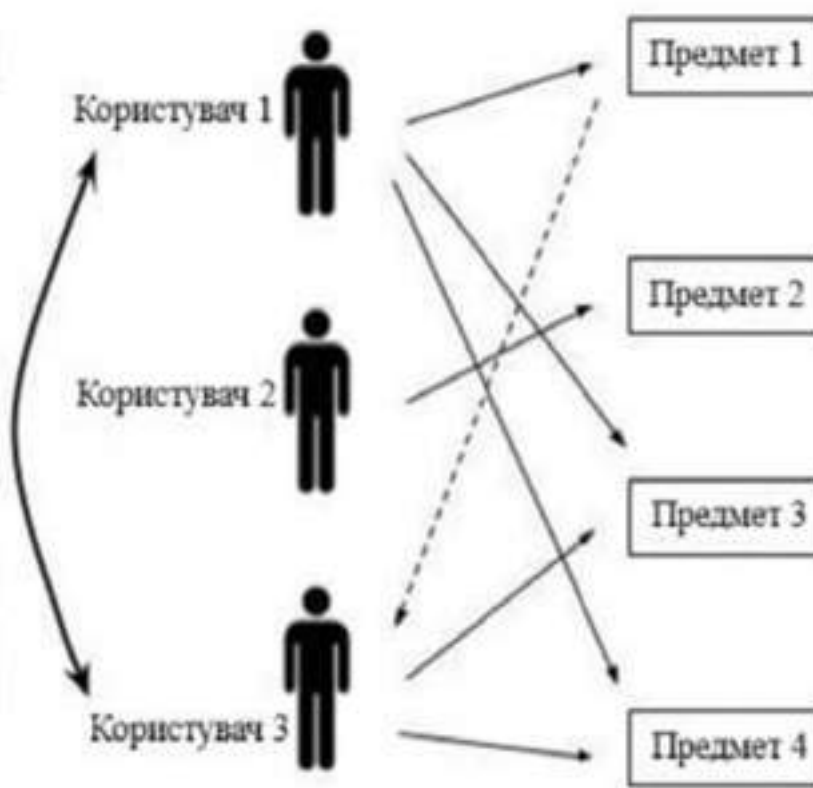


Рисунок 1.2 - Приклад колаборативної фільтрації на основі користувачів.

Звертаючи увагу на рисунок 1.2, найбільш підходящим для користувача 1, буде рекомендувати предмет, які розглядає користувач 3. Тобто в нашому випадку це предмет, який є спільний для обох користувачів, тобто предмет під номером 1.

Використовується в електронній комерції, відео та аудіо стрімінгу, безпосередньо для рекомендацій користувачу новинок, або схожих товарів.

З переваг можна виокремити можливість надавати персоналізовані рекомендації, які відображають індивідуальні інтереси користувачів. І зменшити процес коли вподобання теперішнього користувача обмежені, тому що він наприклад новий користувач сайту, і по матриці рейтингів можна знайти об'єкти, які будуть підходити користувачу.

З недоліків можна відмітити про ту ж матрицю рейтингів, яка полягає в проблемі масштабованості, тому що при збільшенні кількості користувачів та предметів зацікавлення, матриця рейтингів може стати дуже об'ємною, що призведе до затрати великих ресурсів для обчислення результатів.

1. Колаборативна фільтрація предмет-предмет (item-to-item collaborative filtering) [29] – це метод рекомендаційної системи, який базується на подібності між елементами контенту. За допомогою цього методу проводиться аналіз подібності між предметами на основі взаємодії користувачів з ними і рекомендує подібні предмети на основі вибору користувача.

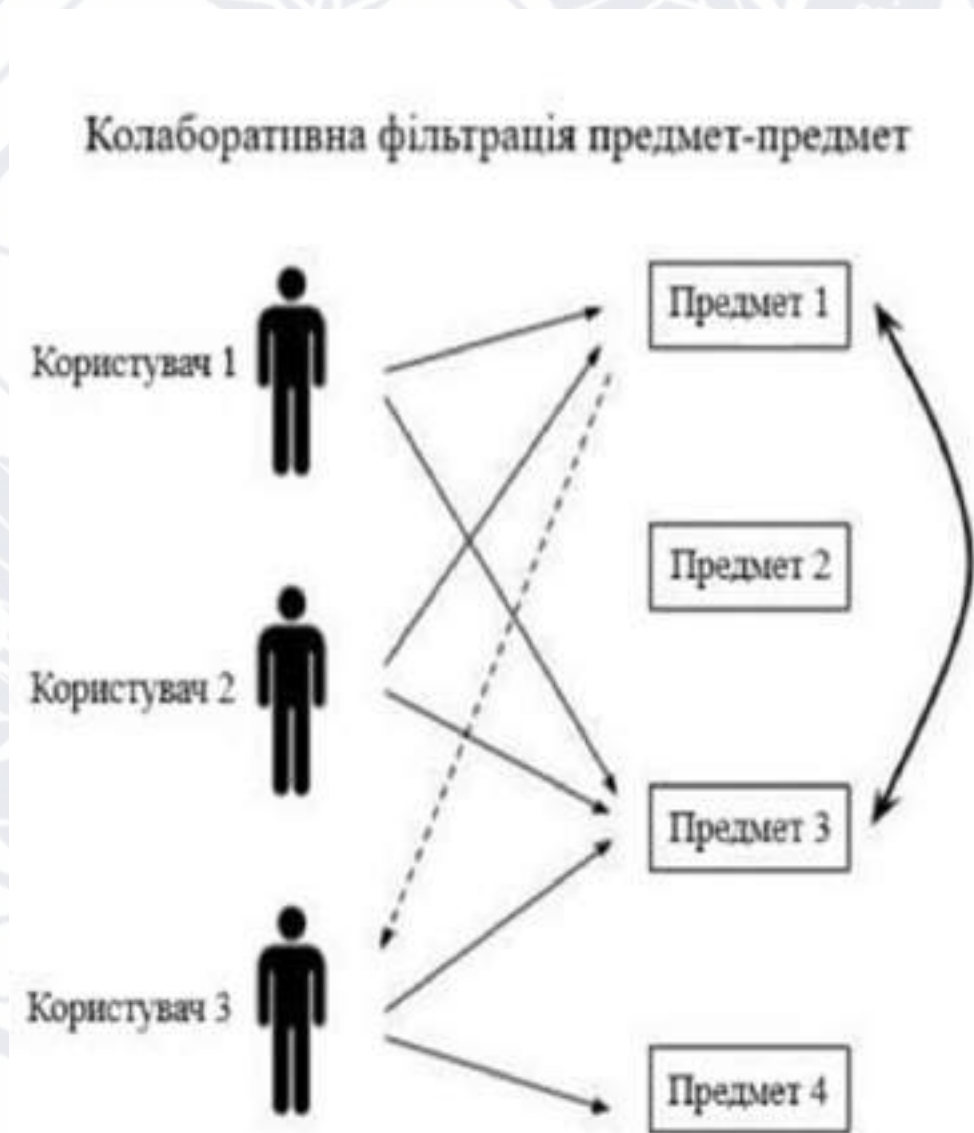


Рисунок 1.3 – Приклад колаборативної фільтрації на основі предметів.

Як видно з рисунку 1.3, найбільш підходящими підходящим для предмету 1, буде рекомендувати предмет 3, відповідно коли користувач відвідає сайт, і переглядатиме предмет 3, йому з впевненістю можна рекомендувати предмет 3, і з великою ймовірністю, йому сподобається цей предмет.

В своїй основі схожий до методу колаборативної фільтрації користувач-користувач, і має матрицю рейтингів, однак на відміну, в даному випадку рядки представляють користувачів, а стовпці – предмети. Також якщо порівнювати з попереднім методом фільтрації, даний метод протилежно знаходить подібність між предметами, а не користувачами. Тобто, до прикладу, коли користувач дивиться на певний предмет (товар в онлайн магазині) система рекомендує подібні до нього предмети, які інші користувачі також раніше переглядали.

Колаборативна фільтрація предмет-предмет ставить акцент безпосередньо на схожості між предметами, замість схожості між користувачами, і ці рекомендації зазвичай здійснюються в рамках певного предмету, який переглядає користувач.

Даний метод часто вважають більш простим в реалізації за попередній, і ще до переваг можна віднести те, що він може надати більш точні рекомендації, оскільки він враховує схожість саме предметів. Але внаслідок того, що відбувається пошук по предметам, новим користувачам, які переглядали мало об'єктів, буде важко щось порекомендувати. Також дуже складно обчислювати, при збільшені кількості предметів.

Отож, даний метод надає можливість для зручних і персоналізованих рекомендацій, але має свої власні обмеження.

Щодо особливостей використання, то все повторюється як з методом колаборативної фільтрації користувач-користувач – стрімінгові платформи, онлайн магазини і веб-контент.

І останній з розглянутих типів рекомендаційних систем – це гібридний підхід. Він полягає в поєднанні попередніх двох методів, для досягнення більш точних та ефективних рекомендацій. Тобто, це означає використання фільтрації, як на основі контенту, так і колаборативного фільтрування.

Гібридний підхід дозволяє використовувати переваги кожного з методів, та компенсувати їхній недоліки. Наприклад, фільтрація на основі контенту може бути корисною в ситуаціях, де інформація про користувачів обмежена або недоступна, в той час як колаборативний підхід може забезпечити більш точніші рекомендації.

Гібридний метод може бути реалізований по різному:

1. Комбінація методів – поєднання різних методів рекомендацій в одній системі для отримання більш точних та різноманітних рекомендацій.
2. Адаптивні підходи – використання алгоритмів, які динамічно адаптуються до зміни у вхідних даних або вимог користувачів.
3. Взаємодія між методами – використання взаємодії між методами для покращення точності або для додавання варіативності рекомендацій.

Гібридні підходи до рекомендаційних систем дозволяють створювати більш потужні та адаптивні системи, які будуть краще задовольняти потреби користувачів.

### **1.3 Огляд аналогів та існуючих технологій**

Задля того, щоб визначити інтерфейс, та дослідити переваги та недоліки кожного варіанту, які вже використовуються користувачами для пошуку маршрутів для подорожей.

1. Сайт [kraina-ua.com](http://kraina-ua.com) – пропонує список турів Україною, з описом, датою та вартістю кожної з локацій. Також можна глянути маршрут і транспорт, за допомогою якого буде відбуватись відвідування локацій, та реальні відгуки від інших людей. На детальній сторінці кожного туру описано

детальний маршрут подорожей для кожного дня. Також доступний зручний календар, за допомогою якого можна обрати дату, яка підходить користувачеві для подорожей, і вже після цього обрати тур для подорожей. З переваг, на сайті можна зареєструватись, обирати зацікавлені маршрути і внаслідок цього для користувача автоматично будуть додаватись зацікавлені, або схожі за інтересами маршрути. Також однією з ключових переваг сайту, є зручний та виконаний в чіткому і плавному вигляді інтерфейс сайту.

2. Сайт [visitukraine.today](http://visitukraine.today) – інтернет-ресурс, який створений для популяризації України серед іноземців. Але також українці можуть з легкістю ним користуватись. Пропонує стандартний пошук туру за такими параметрами: регіон, тривалість та вибір дати коли почнеться і закінчиться тур. Після вибору вищеописаних фільтрів, з'являються деталізовані параметри пошуку такі як: тип туру, інтереси, сезон, вибір валюти та вартість. З мінусів можна відзначити що відсутня можливість створення профілю, внаслідок чого і немає можливості рекомендувати користувачу системи за вподобаннями та ті, які б йому можливо сподобались.
3. Онлайн-платформа [wondersholidays.com](http://wondersholidays.com) призначена для екскурсій та подорожей Україною. На вибір дають обрати декілька регіонів України такі як: Київ, Львів, Одеса, Карпати, Поділля, Полісся та Волинь, Південь України та Запоріжжя. Також окремо від попередніх параметрів можна обрати відпочинок саме за типом, тобто за інтересами користувача. Після обрання параметрів, які зацікавили користувача, на сайті з'являються можливі варіанти відпочинку з тривалістю туру, та ціною. З мінусів, неможливо завести акаунт, відповідно до цього відсутня можливість рекомендувати користувачу будь які варіанти, в залежності від його вподобань.

Отже, порівняння цих трьох сайтів дозволяє виділити ключові аспекти, на які слід звернути увагу при виборі платформи для пошуку туристичних маршрутів. Kraina-ua.com виграє завдяки своєму зручному інтерфейсу, детальній

інформації про тури та можливості персоналізації. Visitukraine.today і wondersholidays.com також пропонують корисні функції та широкий вибір параметрів пошуку, але їх недоліком є відсутність можливості створення профілю, що обмежує можливості персоналізованого користування.

Вибір конкретного ресурсу залежить від індивідуальних потреб користувача, але загалом можна стверджувати, що кожен з розглянутих сайтів має свої сильні сторони і може бути корисним інструментом для планування подорожей Україною.

### **Висновок до першого розділу**

У першому розділі було детально розглянуто концепцію рекомендаційних систем, їхню історію, типи та особливості застосування. Рекомендаційні системи відіграють ключову роль у сучасному інформаційному середовищі, оскільки вони надають користувачам персоналізовані пропозиції на основі їхніх уподобань і взаємодій із контентом. Завдяки розвитку технологій, зокрема машинного навчання та штучного інтелекту, стає можливим створення більш точних, ефективних та адаптивних рекомендаційних систем, які здатні підлаштовуватися під зміни в поведінці користувача.

Було виокремлено кілька основних методів, таких як фільтрація на основі контенту та колаборативна фільтрація, кожен з яких має свої переваги та недоліки. Фільтрація на основі контенту зосереджується на аналізі характеристик об'єктів, щоб створити рекомендації, що відповідають інтересам користувача. Колаборативна фільтрація використовує дані про схожі вподобання інших користувачів, щоб пропонувати нові об'єкти. Гібридні підходи, що поєднують ці два методи, демонструють найкращі результати у забезпеченні точних рекомендацій.

Огляд існуючих технологій у сфері туристичних сервісів показав, що більшість платформ не використовують потенціал персоналізації для покращення користувацького досвіду. Це підкреслює потребу в розробці більш інтуїтивно зрозумілих та адаптивних рішень, які можуть ефективно аналізувати дані про користувачів і надавати їм релевантні рекомендації. Загалом, аналіз

рекомендаційних систем демонструє їхній потенціал у покращенні процесу прийняття рішень користувачами та оптимізації їхнього досвіду в різних сферах, зокрема в туристичній індустрії. Це може суттєво підвищити задоволеність користувачів і ефективність платформ, сприяючи їхньому розвитку.



## РОЗДІЛ 2

### ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМИ ТА ПОБУДОВА МОДЕЛІ ПРОГРАМНОГО ПРОДУКТУ

#### 2.1 Визначення ключового функціоналу додатку

Рекомендаційні системи є невід’ємною частиною сучасних технологій, які знаходять застосування в найрізноманітніших сферах: від електронної комерції до соціальних мереж, різноманітних платформ та мобільних додатків. Їх основна функція полягає у допомозі користувачам знайти релевантний контент чи інформацію серед величезного обсягу доступних даних.

У контексті нашого проекту, рекомендаційна система інтегрується як один із ключових елементів веб-додатка для пошуку туристичних маршрутів в Україні, однак сама по собі є окремим програмним компонентом із власними завданнями, структурою та алгоритмами.

Головною ідеєю створення рекомендаційної системи є забезпечення зручного доступу до маршрутів, які будуть цікавими та корисними для конкретного користувача.

Якщо веб-додаток виконує функцію загального інтерфейсу для пошуку та взаємодії, то рекомендаційна система виконує роль так званого помічника, який аналізує вподобання користувачів і пропонує персоналізовані варіанти. Таким чином, ці два компоненти працюють у тісній взаємодії, але їх функціональні задачі суттєво відрізняються одна від одної.

Створення рекомендаційної системи стало важливим етапом у процесі розвитку нашого проекту, адже воно було обумовлено потребою не лише покращити загальну взаємодію користувачів із платформою, але й зробити цей процес максимально зручним, персоналізованим та ефективним.

У сучасному світі, де користувачі мають доступ до великої кількості інформації, особливо важливо забезпечити інструменти, які допоможуть швидше орієнтуватися серед безлічі варіантів та обирати те, що найбільше відповідає їхнім уподобанням та потребам.

Рекомендаційна система, інтегрована у нашу платформу, має кілька ключових завдань. Перш за все, вона спрямована на забезпечення користувачів персоналізованими пропозиціями маршрутів, які відповідають їхнім інтересам, уподобанням або минулим запитам. Це дозволяє значно скоротити час на пошук необхідної інформації та зробити взаємодію з платформою більш комфортною.

Ще однією метою створення рекомендаційної системи є підвищення рівня задоволеності користувачів сайту, адже в умовах високої конкуренції якісний та зручний сервіс відіграє вирішальну роль. Завдяки аналізу даних про поведінку користувачів, їхні пошукові запити та дії на платформі, система пропонує найактуальніші варіанти, що відповідають їхнім очікуванням.

Таким чином, основними цілями нашої рекомендаційної системи є:

- Підвищення ефективності пошуку маршрутів – оскільки буде працювати автоматичний аналіз великої кількості даних, він буде дозволяти пропонувати лише релевантні варіанти.
- Персоналізація користувацького досвіду – забезпечення унікального підходу до кожного користувача на основі його інтересів, попередніх взаємодій та вподобань.
- Збільшення часу взаємодії з платформою в цілому – завдяки цікавим та релевантним рекомендаціям, користувачі будуть більше часу проводити на платформі та користуватись нею.
- Виявлення нових інтересів – рекомендаційна система може запропонувати варіанти, які користувач міг не врахувати або не шукати, але які потенційно зможуть зацікавити. Це особливо корисно для маловідомих маршрутів.

В той же час, необхідно окреслити технічні характеристики майбутньої системи:

- Рекомендаційна система реалізована як окремий мікросервіс, який взаємодіє з базою даних і веб-додатком через API [30]. Такий підхід дозволяє легко масштабувати систему та адаптувати її до нових потреб за необхідності.

- Рекомендаційна система буде використовувати контентно-орієнтований алгоритм, який буде аналізувати властивості маршрутів, про які ми поговоримо детальніше в наступному розділі.
- Дані, які будуть використовуватись це прямі дії користувача (пошуки, запам'ятовування маршрутів, взаємодія з маршрутами).

Потрібно також змодельовати поведінку рекомендаційної системи, як для нових користувачів, так і для постійних, які вже давно зареєстровані на сайті, та взаємодіють із ним:

- Якщо користувач вперше заходить на платформу, системі потрібен певний час, для «знайомства» з користувачем. Необхідно щоб користувач взаємодіяв з системою, але також можна зробити «холодний старт» і пропонувати популярні маршрути, не базуючись на вподобаннях користувача, якщо не вистачає даних для рекомендацій.
- Якщо ж користувач вже зареєстрований давно, і активно користується платформою, то система буде шукати схожі вподобання на основі типу відпочинку (якщо користувач обирає екстримальний відпочинок, то можна запропонувати відпочинок в горах), або локації (якщо він активно шукає маршрути в житомирській області, то також можна рекомендувати сусідні до цієї області)
- Також можна винести як окремий пункт те, що можна враховувати сезонність, коли користувач взаємодіє з платформою, і наприклад зимою пропонувати лижні маршрути або відповідно маршрути для зимового відпочинку. Влітку – акцент на прогулянках в горах, або іншій схожій активності.

Також необхідно детальніше звернути увагу на один з основних, але в той же час допоміжний атрибут – веб-додаток. Саме через нього буде відбуватись взаємодія користувача з сайтом. За допомогою нього планується зберігати всю взаємодію, за допомогою якої будуть працювати рекомендації. Отож, перед

створенням власного продукту, важливо ретельно оцінити, які технології найкраще використовувати в процесі розробки.

Отже, сайт повинен мати такі основні переваги та функціонал:

1. Реєстрація та авторизація. Забезпечення безпечного доступу користувачів до персональних даних і можливість створення облікових записів.
2. Взаємодія з формами. Можливість зручного введення і обробки даних користувачів через форми.
3. Мінімальна і швидка стилізація сайту, яка буде відповідати сучасним вимогам і повинна бути не гірша ніж стилізація інших схожих сайтів аналогів.
4. Адаптивна верстка. Забезпечення коректного відображення сайту на різних пристроях, включаючи комп'ютери, планшети та смартфони.
5. Підтримка маршрутизації. Зручна навігація по сайту з використанням маршрутизації, що дозволяє користувачам легко переміщуватися між сторінками.
6. Перемальовування DOM-дерева [31, 32] без оновлення сторінки. Використання технологій, які дозволяють оновлювати вміст сторінки без її повного перезавантаження, що покращує користувацький досвід та швидкість взаємодії з сайтом.
7. Можливість користуватись server-side-rendering, для того щоб не було проблем з впровадженням SEO і подальшого прискорення роботи сайту.

Окремо хотілось би зупинитись на одній з найбільш актуальних і важливих проблем сучасних підходів, які використовують різноманітні фреймворки та бібліотеки для відмалювання контенту безпосередньо на стороні клієнта (client-side-rendering) [33].

Цей підхід стосується таких популярних технологій як React [34, 35, 36], Vue [37, 38], Angular [39, 40] та інші, які значною мірою залежать від рендерингу контенту безпосередньо в браузері користувача. Хоча даний підхід і має

безперечно свої переваги, такі як швидкість реакції додатку після першого завантаження, він також приносить низку проблем з точки зору SEO[41, 42], які варто обговорити детальніше.

Проблема полягає в тому, що під час першого запиту браузера до сервера для отримання сторінки сервер часто повертає HTML-шаблон [43], що містить загальну структуру сторінки. Основний контент може бути завантажений асинхронно через JavaScript [44], який відповідає за динамічне оновлення та рендеринг інтерфейсу програми. Це характерно для односторінкових додатків (SPA) [45], де вміст часто отримується через API-запити після первинного рендерингу.

У більшості випадків це відбувається асинхронно, тобто спочатку завантажуються лише скрипти, а потім він "підтягує" необхідні дані і рендерить їх безпосередньо у браузері.

Однак цей підхід має істотний недолік: оскільки пошукові системи не можуть повноцінно індексувати JavaScript-код, особливо якщо він завантажуються асинхронно, це створює значні труднощі для індексації контенту. І саме цей недолік породжує велику кількість подальших проблем, для яких потрібно знайти оптимальне рішення:

1. Боти пошукових систем, можуть просто не побачити динамічний контент, що з'являється лише після виконання JavaScript-скриптів, що негативно впливає на видимість сайту в пошукових системах. Це означає, що весь вміст сайту, який рендериться лише на стороні клієнта, може залишитися невидимим для пошукових систем, а це знижує ймовірність того, що сайт буде коректно проіндексований та отримуватиме потрібний трафік переглядів і користування з пошуку.
2. Окрім цього, велика проблема полягає у відсутності можливості індексувати важливі метадані. Метадані [46], такі як заголовки сторінок, описи та ключові слова, відіграють ключову роль в оптимізації сайту для пошукових систем. Якщо ці дані генеруються динамічно за допомогою JavaScript, а не передаються безпосередньо у HTML під час першого

завантаження сторінки, то пошукові системи можуть не побачити їх взагалі. Як наслідок, навіть якщо вміст сторінки цікавий і корисний, вона може не отримати належної уваги через відсутність належної індексації, що призведе до меншої кількості відвідування сайту.

3. Іншою важливою проблемою є швидкість завантаження та рендерингу сторінки. Оскільки у випадку `client-side-rendering` спочатку завантажується лише базовий HTML і скрипт, а весь контент з'являється лише після виконання цього скрипту, час завантаження сторінки може збільшуватися, що теж впливатиме на місце сайту коли користувач захоче знайти сайт.
4. Це негативно впливає на досвід користувачів, особливо на мобільних пристроях, де швидкість інтернет-з'єднання може бути меншою. Більше того, швидкість завантаження сторінки є одним з важливих факторів ранжування сайтів у пошукових системах. Якщо сторінка завантажується занадто довго, це може призвести до поганих показників SEO, що негативно вплине на видимість сайту у пошуку, що є негативним показником.
5. Ще одна проблема полягає у динамічному контенті, який змінюється в залежності від взаємодії користувача зі сторінкою (наприклад, кліки на елементи інтерфейсу або підвантаження нових даних після певних дій). Якщо цей контент з'являється не одразу, а лише після взаємодії користувача, боти пошукових систем можуть не побачити його взагалі. Це особливо критично для контенту, що містить ключову інформацію для індексації, таку як описи послуг або ключові місця для відвідування.
6. Окрім того, використання хеш-роутінгу, де в URL використовуються символи на кшталт #, може створити додаткові труднощі для пошукових систем. Більшість пошукових роботів не сприймають URL-адреси з хешами як окремі сторінки, тому контент, що доступний за такими адресами, може залишитися невидимим для індексації. Це особливо важливо для сайтів, які мають багато підсторінок або динамічний роутінг, де кожен окремий розділ сайту повинен бути доступний для індексації.

У зв'язку з тим, що наш майбутній веб-додаток буде використовуватися великою кількістю користувачів і має потребу в оптимізації для пошукових систем, ми повинні приділити особливу увагу покращенню індексації. Одним із найбільш ефективних рішень цієї проблеми є використання серверного рендерингу (server-side-rendering) [47, 48].

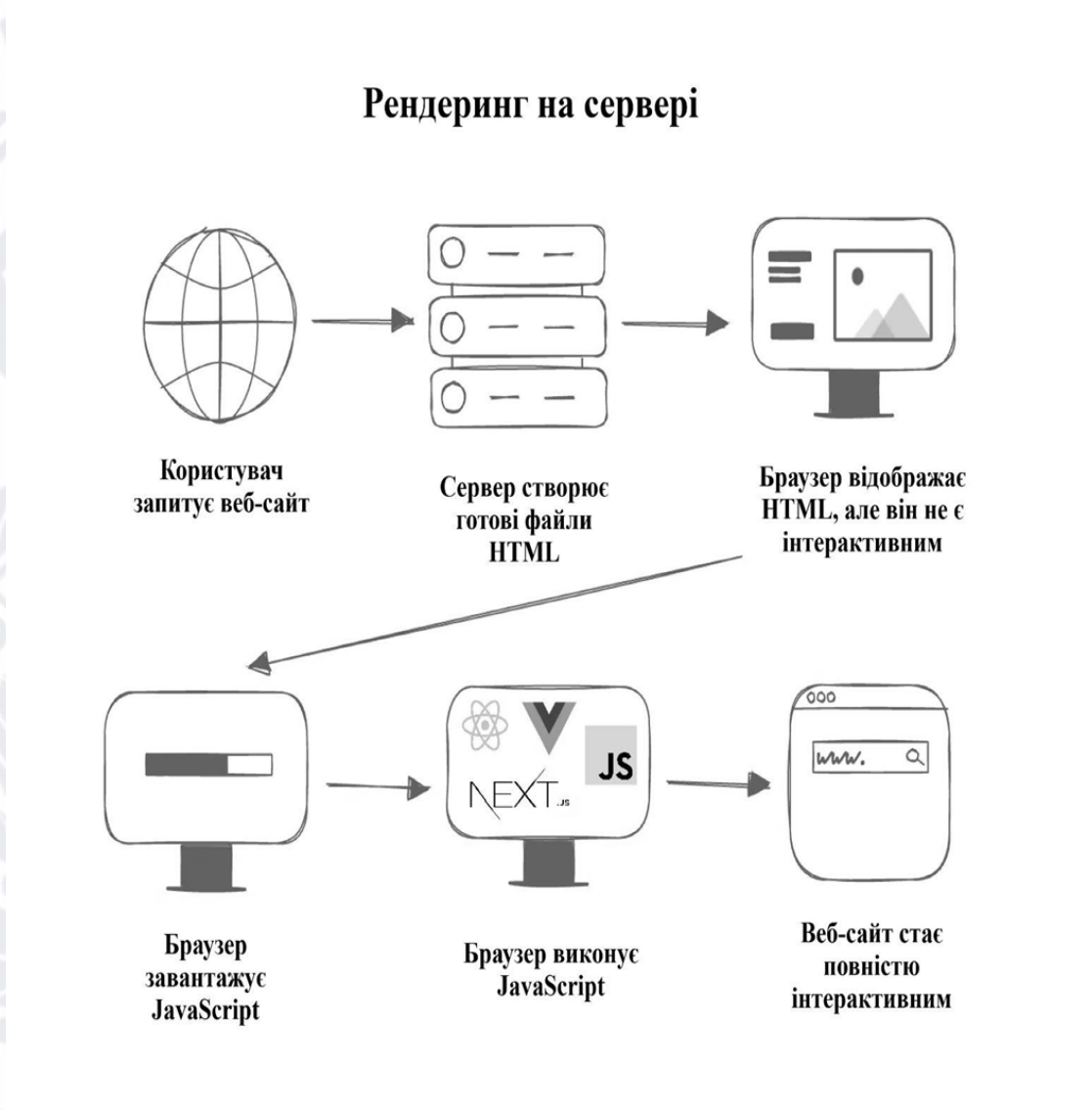


Рисунок 2.1 - Рендеринг на сервері.

На схемі, що наведена на рисунку 2.1, показано як працює рендеринг на стороні серверу, потрібно детальніше це розглянути:

1. Користувач запитує сайт. Коли користувач відкриває сайт, відправляється запит на сервер. Цей запит, може бути зроблений як для отримання динамічної сторінки, так і для отримання статичної сторінки. Сервер отримує цей запит і починає обробку.

2. Сервер створює готові HTML файли. Сервер виконує важливу частину роботи: він генерує HTML-код сторінки. Це може відбуватися різними способами, наприклад, з використанням таких фреймворків. Сервер збирає всі необхідні дані для сторінки, інтегрує їх у шаблон, і відправляє браузеру повністю сформований HTML. У цей момент важливий аспект полягає в тому, що HTML вже містить всю інформацію для відображення — включно з даними, текстом, зображеннями, але сторінка ще не інтерактивна, оскільки відсутні підключені скрипти.
3. Браузер рендерить HTML, але сторінка поки що, не є інтерактивною. Браузер отримує цей готовий HTML-код і відразу рендерить його. Це означає, що користувач побачить весь контент сторінки (текст, зображення, кнопки, тощо) майже одразу, навіть якщо JavaScript ще не завантажено. Однак на цьому етапі взаємодія зі сторінкою обмежена: кнопки, форми та інші інтерактивні елементи ще не працюють, оскільки JavaScript, який керує цією взаємодією, не виконаний.
4. Браузер завантажує JavaScript. Далі браузер починає завантажувати файли JavaScript, які відповідають за поведінку та інтерактивність сторінки. В цей момент браузер отримує всі необхідні скрипти, які можуть бути динамічними або заздалегідь підготовленими на сервері.
5. Браузер виконує JavaScript: Після завантаження JavaScript браузер починає його виконувати, що є важливим етапом в процесі відображення з веб-ресурсами. Це може включати різні фреймворки або бібліотеки, які відповідають за рендеринг компонентів на стороні клієнта, додавання логіки взаємодії, обробку подій користувача, виклики до API тощо. На цьому етапі сторінка оживає: кнопки стають інтерактивними, форми можна заповнювати, меню розкриватися, а динамічний контент — оновлюватися.
6. Сайт повністю інтерактивний: Нарешті, після виконання JavaScript, сайт стає повністю інтерактивним. Всі елементи, які залежать від скриптів, тепер функціонують, і користувач може взаємодіяти зі сторінкою так, як задумано розробниками. Наприклад, форми

відправляють дані, модальні вікна відкриваються, а фільтри на сторінках товарів або статей починають працювати.

Цей підхід дозволяє генерувати повноцінний HTML-контент на сервері перед його відправкою до клієнта. Це забезпечує можливість для ботів пошукових систем коректно індексувати сторінку, а також знижує час завантаження, оскільки користувач одразу отримує готовий HTML, що містить повний контент сторінки.

## **2.2 Збір даних для відображення можливих варіантів**

Збір даних для відображення можливих варіантів для туристичних подорожей – це надзвичайно важливий процес, який охоплює безліч аспектів, від збору базових даних про доступні послуги до аналізу поведінки користувачів. У сучасному світі, де технології постійно розвиваються, а конкуренція у сфері туристичних послуг зростає, необхідність у точному, актуальному та всебічному представленні інформації стає ключовим чинником для успіху будь-якого веб-додатка, спрямованого на туристів.

Основними джерелами інформації є системи бронювання, які формують базу даних про доступність готелів, рейсів, туристичних маршрутів, а також різноманітних заходів, доступних у регіонах. Багато з цих систем працюють на основі API (інтерфейс програмування додатків), які забезпечують автоматизований доступ до актуальної інформації. Наприклад, в системах бронювання можна отримати дані про готелі, що включають їхні характеристики (кількість зірок, наявність Wi-Fi, умови скасування), ціни, рейтинги та відгуки користувачів. Така інформація дозволяє створювати зручні фільтри, які допомагають користувачам знаходити ідеальні варіанти, що відповідають їхнім вимогам.

Додатково до інформації про готелі, важливо отримати дані про транспортні засоби, такі як авіарейси, поїзди, автобуси, а також можливості оренди автомобілів. У цьому контексті дані можуть включати маршрути, ціни, розклади, тривалість подорожі, а також інформацію про наявність місць. Наприклад, якщо користувач планує подорож з одного міста в інше, система

може запропонувати кілька варіантів транспорту, з детальним описом часу в дорозі, кількості пересадок та загальної вартості поїздки. Це надає користувачеві змогу порівнювати різні варіанти та обирати найбільш зручний і вигідний.

Сучасні веб-додатки активно використовують аналітику та алгоритми машинного навчання для забезпечення персоналізації даних. Це дозволяє системі аналізувати поведінку користувачів та пропонувати індивідуалізовані варіанти подорожей на основі їхніх попередніх запитів, історії бронювань і навіть уподобань. Наприклад, якщо користувач постійно шукає варіанти відпочинку на пляжі, система може в першу чергу пропонувати йому варіанти курортів із пляжами, які мають схожі характеристики, навіть якщо вони розташовані в інших країнах. Це не лише спрощує процес пошуку, але й робить його більш приємним і ефективним.

Крім того, алгоритми можуть враховувати сезонні зміни та події, які відбуваються в певних регіонах, щоб запропонувати користувачам цікаві та актуальні варіанти подорожей. Наприклад, під час великих фестивалів або святкувань, система може акцентувати увагу на доступних варіантах подорожей до цих міст, заохочуючи користувачів відвідати ці заходи. Такі рекомендації не тільки допомагають користувачам знайти нові можливості для подорожей, але й створюють у них бажання скористатися сервісом.

Крім традиційних систем бронювання, важливо враховувати й зовнішні джерела даних, які можуть впливати на плани подорожей. Це може бути інформація про візові вимоги, необхідні щеплення, правила безпеки в певних регіонах, а також різноманітні культурні нюанси. Надання такої інформації може суттєво підвищити довіру користувачів до сервісу. Наприклад, мандрівники можуть отримувати сповіщення про зміни в умовах в'їзду до країни, які можуть вплинути на їхню подорож, що дозволяє їм вчасно коригувати свої плани. Однак, оскільки наш додаток спеціалізується лише на туристичних подорожах в межах території України, на цьому аспекті не потрібно робити надто великий акцент, але все ж важливо бути в курсі основних тенденцій. Але оскільки наш додаток

спеціалізується лише для туристичних подорожей в межах території України, на цьому пункті сильно загострювати увагу не потрібно.

Додатково, важливо включити в систему відгуки та рекомендації від інших мандрівників. Це може бути реалізовано через інтеграцію з соціальними мережами або платформами відгуків. Наприклад, користувачі можуть отримувати поради про найкращі місця для відвідування, ресторани, екскурсії, на основі досвіду інших туристів. Така інформація може допомогти уникнути розчарувань і зробити подорож більш комфортною.

Підсумовуючи можна схематично зобразити як відбувається процес збору даних на рисунку 2.2 для кращого розуміння.

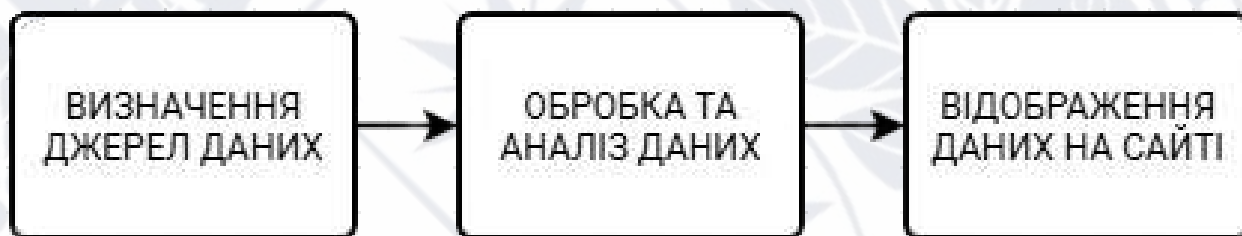


Рисунок 2.2 – Схема збору даних

Забезпечення швидкості та ефективності збору даних є критично важливим завданням для веб-додатків. По-перше, система повинна бути здатною обробляти величезні обсяги запитів одночасно, оскільки популярність туристичних платформ може призводити до пікових навантажень, особливо під час високого сезону. Використання технологій кешування може суттєво зменшити навантаження на сервери та пришвидшити доступ до найбільш запитуваних даних. Кешування дозволяє зберігати інформацію, що часто запитується, таким чином, щоб при наступному запиті система могла швидко надати вже збережену інформацію, без повторного запиту до бази даних.

По-друге, автоматизація збору даних, що включає в себе парсинг веб-сторінок, може значно спростити процес отримання актуальної інформації про ціни та наявність послуг. Однак тут виникає ряд етичних і юридичних питань, пов'язаних із правами на дані. Веб-додатки повинні дотримуватися етики та законодавства, щоб уникнути можливих правових наслідків.

### 2.3 Взаємодія користувача з сайтом.

Сайт для пошуку туристичних подорожей є складним веб-додатком, який має на меті забезпечити користувачам інтуїтивно зрозумілий і зручний інтерфейс для швидкого знаходження найкращих варіантів подорожей. Взаємодія користувача з сайтом включає в себе кілька етапів, кожен з яких спрямований на забезпечення максимальної зручності і ефективності.

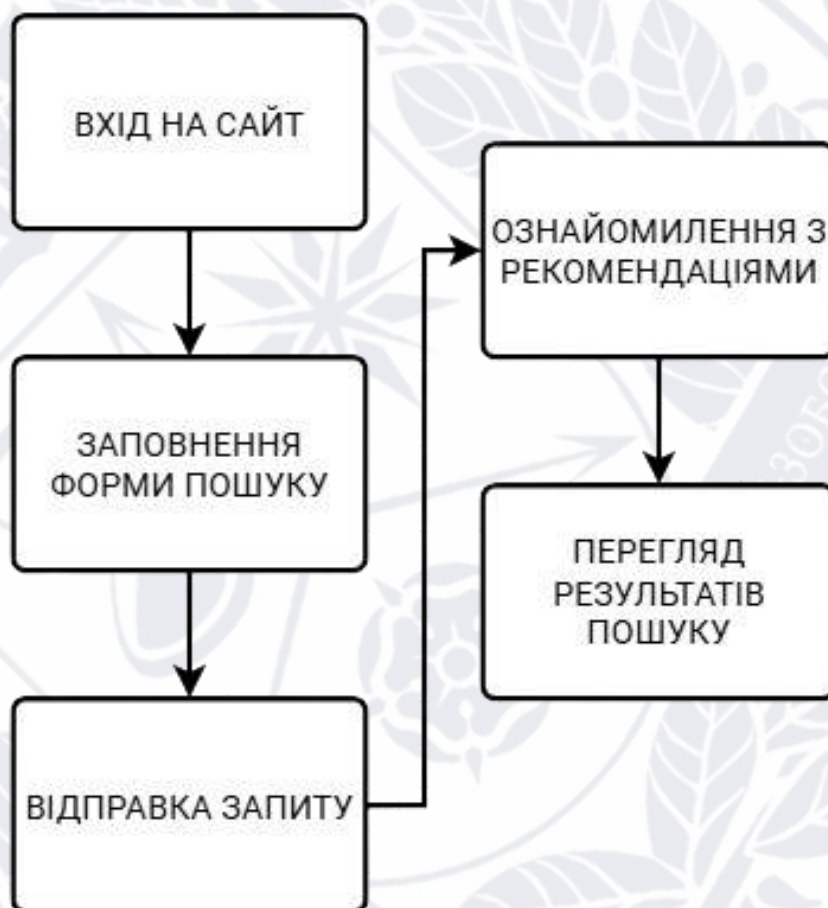


Рисунок 2.3 – Схема взаємодії користувача з сайтом

Розглянемо докладніше кожен із цих етапів, які зображені на рисунку 2.3:

1. Головна сторінка. Користувач потрапляє на головну сторінку сайту після першого відвідування або після входу в систему. Ця сторінка має бути максимально простою та інформативною, щоб не перевантажувати користувача надмірною кількістю деталей, але водночас надати всі необхідні інструменти для пошуку подорожей. Важливо, щоб інтерфейс мав приємний дизайн і був функціональним. Зазвичай, для цього

використовуються сучасні рішення в UX/UI дизайні [49], які гарантують легкість у використанні навіть для новачків. На головній сторінці розміщена форма для пошуку подорожей, яка є основним елементом взаємодії з користувачем. Форма повинна бути чіткою і мати достатню кількість полів для того, щоб користувач міг звужити пошук за своїми критеріями, не переглядаючи зайві варіанти.

2. Поля для фільтрів. У формі на головній сторінці користувач знайде кілька важливих полів, що допомагають конкретизувати пошук:

- Місце призначення. Одним із перших кроків є вибір міста або регіону, куди користувач хоче вирушити. Це поле може бути реалізоване у вигляді випадаючого списку або інтерактивного поля, де користувач може почати вводити назву місця, і система автоматично запропонує варіанти для вибору.
- Дата відправлення та повернення. Поля для вибору дати дозволяють користувачеві обрати часові рамки для своєї подорожі. Сучасні системи підтримують інтерактивні календарі, що спрощують процес вибору дати. Це особливо важливо для туристів, які шукають точні дати для подорожі або мають обмеження по часу.
- Тип транспорту. Наступним етапом є вибір транспорту. Користувач може вибрати один або кілька варіантів, таких як потяг, автобус або інший варіант. Це дозволяє знайти найбільш зручний і економічний варіант для подорожі.
- Ціновий діапазон. Фінансова складова є важливою для багатьох користувачів, тому в формі є повзунок для вибору мінімальної і максимальної ціни за подорож. Це допоможе відібрати тільки ті варіанти, які відповідають бюджету користувача.

- Кількість подорожуючих. Це поле дозволяє вибрати кількість людей, які вирушають у подорож. Воно корисне, якщо йдеться про групові поїздки або сімейні відпустки.
- Тип проживання. Для деяких подорожуючих важливий тип проживання – чи це буде готель, хостел або апартаменти. Додавши це поле у форму, можна значно спростити процес вибору для користувача.

Крім цих основних полів, можна також додати додаткові параметри для ще більшої деталізації пошуку, як, наприклад, наявність зручностей у готелі (басейн, тренажерний зал), тип харчування або рейтинг закладу.

3. Обробка форми і відправка запиту. Після заповнення форми користувач натискає на кнопку "Пошук", що запускає процес відправки запиту на сервер. Це є важливий етап, оскільки система повинна швидко обробити запит і повернути відповідні результати.

Запит містить всю інформацію, яку ввів користувач: вибране місце призначення, дати подорожі, транспорт, тип проживання та інші фільтри. Ці дані відправляються на сервер, де запускається пошук у базі даних. Саме в базі зберігається вся необхідна інформація про доступні варіанти подорожей, рейси, готелі, їхні ціни та наявність місць.

4. Взаємодія з базою даних. У процесі пошуку запит проходить через кілька рівнів системи. Спочатку сервер обробляє дані користувача, перевіряє їх на валідність, після чого відправляє запит до бази даних. База даних у даному випадку – це велика система, яка містить різноманітну інформацію, необхідну для формування результатів пошуку:

- Туристичні маршрути. Зберігаються всі можливі напрямки, їхні характеристики, опис маршруту, зображення, відгуки та рейтинг. Це дає змогу показати користувачу різні варіанти подорожей.

- Транспорт. Інформація про доступні рейси, зокрема авіа, потяг, автобус або оренда автомобіля. Дані можуть містити розклади, ціни, тривалість поїздки та наявність місць.
- Проживання. Зберігаються дані про доступні готелі, хостели, апартаменти, а також їхні характеристики, як-от кількість зірок, ціни, наявність Wi-Fi або умов скасування.
- Історія пошуків. Для зареєстрованих користувачів також зберігаються попередні пошуки, щоб можна було швидко повторити пошук або порівняти варіанти.

5. Отримання результатів пошуку. Після обробки запиту користувач отримує результати, які відображаються у вигляді карток подорожей. Кожна картка містить деталі, які допомагають користувачеві зробити вибір:

- Зображення. На картці є головне фото напрямку або готелю, що дає користувачу перше візуальне враження.
- Назва маршруту. Назва подорожі або напрямку, яка коротко описує основну суть туру або маршруту.
- Ціна. Ціна за людину або загальна вартість поїздки.
- Опис. Короткий текстовий опис, який може включати основні моменти подорожі, що роблять її привабливою для користувача.
- Кнопка "Переглянути інформацію детальніше". Кнопка, що дозволяє переглянути більше інформації про обраний варіант подорожі.

6. Рекомендаційна система. Для зареєстрованих користувачів на сторінці результатів пошуку додається блок "Можливо вас зацікавлять подорожі", який показує персоналізовані пропозиції на основі минулих запитів або історії пошуків. Наприклад, якщо користувач часто шукав тури на море, система може показати додаткові варіанти, пов'язані з пляжним відпочинком.

7. Додаткові функції у хедері сайту. Хедер сайту має іконку користувача, яка відкриває випадаюче меню з кількома важливими функціями:

- Профіль. Користувач може переглянути свій профіль, змінити налаштування, переглянути дані про бронювання або вподобання.
- Історія пошуків: Для зручності користувач може переглянути всі свої попередні запити та швидко повернутися до них.
- Улюблені: Список маршрутів або подорожей, які користувач зберіг у закладках для подальшого перегляду або бронювання.
- Вийти з облікового запису. Кнопка, яка буде дозволяти користувачу вийти з облікового запису для зміни запису або будь чого іншого.

В цілому, взаємодія користувача з сайтом будується на простих, зрозумілих етапах, що полегшують процес пошуку, вибору та бронювання подорожі. Сайт поєднує зручний інтерфейс, інтуїтивно зрозумілу навігацію, ефективну пошукову систему та персоналізовані рекомендації, що робить його корисним і привабливим інструментом для туристів і просто для людей які люблять подорожувати.

Хотілось би більшу увагу акцентувати на важливості рекомендаційної системи в нашому випадку.

Рекомендаційна система – один із ключових елементів, який здатен значно підвищити цінність сервісу для користувача та зробити його досвід максимально персоналізованим. У сфері туристичних послуг, де вибір подорожей, напрямків та варіантів перебування надзвичайно широкий, рекомендації допомагають користувачам знаходити цікаві для них пропозиції, навіть якщо вони про них не знали. Така система працює на основі аналізу попередньої поведінки користувача, його вподобань та тенденцій вибору.

Рекомендаційна система слугує не лише для того, щоб показати можливі варіанти, а й значно підвищує зручність користувацького досвіду, дозволяючи туристам знаходити нові напрямки або типи відпочинку, які б інакше

залишилися поза їхньою увагою. Це не лише заощаджує час, але й створює враження, що веб-додаток адаптований під потреби кожного окремого користувача, зменшуючи так звану "втому від вибору" (пошук серед величезної кількості інформації, яка може бути неактуальною для конкретного користувача).

Також слід врахувати, що в умовах зростання конкуренції між туристичними платформами рекомендаційна система може виступати як ключовий фактор вибору платформи як пріоритетної. Користувачі, які отримують персоналізовані пропозиції, швидше за все, повернуться на платформу і будуть більш лояльними до неї, що в довгостроковій перспективі підвищує користувацький інтерес.

Можна коротко поговорити про те як саме буде працювати рекомендаційна система в рамках нашого веб-додатку. Рекомендаційна система зазвичай будується на кількох ключових механізмах, які дозволяють відстежувати, обробляти і використовувати дані про користувачів для формування релевантних і цікавих пропозицій. Основні етапи роботи системи можна зобразити схематично, однак варто зазначати, що це суб'єктивна репрезентація, після чого додати пояснення до кожного з пунктів.

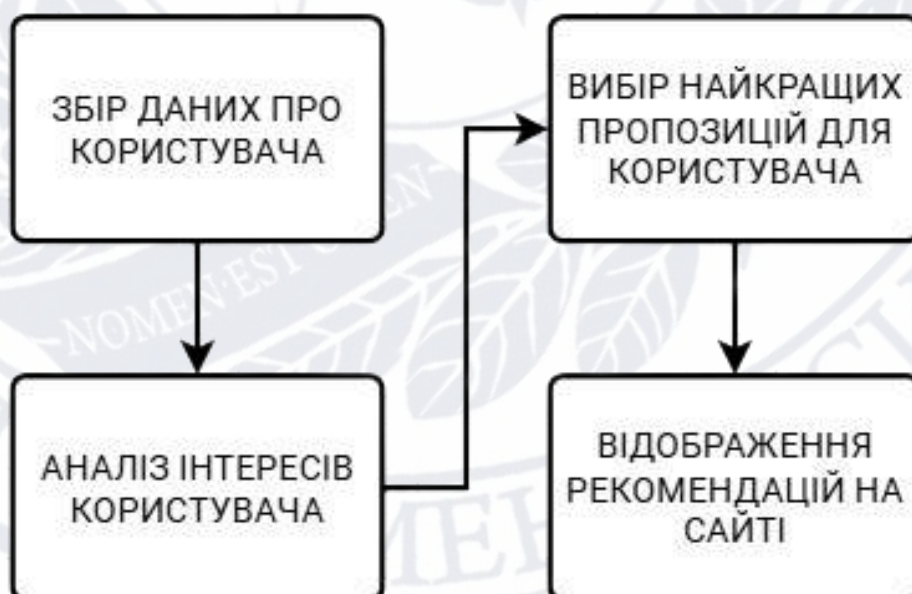


Рисунок 2.4 – Схема роботи рекомендаційної системи на сайті

Після зображеної схеми на рисунку 2.4, слід детальніше розглянути кожен з етапів для покращення розуміння роботи рекомендаційної системи на сайті

1. Збір даних про користувача. Система збирає дані про дії користувача, включаючи його попередні пошуки, обрані фільтри, переглянуті сторінки та додані в закладки певні сторінки. Наприклад, якщо користувач регулярно переглядає активний відпочинок, або певну якусь локацію, це сигнал для системи, що його пріоритети можуть включати схожий вид відпочинку, або схожі за місцезнаходженням локації. Ці дані накопичуються з часом і стають основою для аналізу.
2. Аналіз інтересів користувача. Після збору інформації, система аналізує наявні дані та формує профіль користувача, який відображає його потенційні інтереси. На основі цього профілю можна здійснити класифікацію та впевнено пропонувати найбільш релевантні напрямки, види відпочинку чи до прикладу види транспорту. Наприклад, користувач, який часто шукає варіанти активного відпочинку, буде отримувати рекомендації на кшталт гірських маршрутів, екстремальних екскурсій чи кемпінгових баз.
3. Вибір найкращих пропозицій для користувача. Використовуючи такі методи, як фільтрація за спільними інтересами, система створює унікальний список рекомендацій. Наприклад, для користувача, який переглядає історичні місця, система може пропонувати додаткові варіанти міст із багатою історією або музеї. Це допомагає зробити пропозицію персоналізованою та відображає взаємозв'язок між інтересами користувачів.
4. Відображення рекомендацій на сайті. Як вже було сказано раніше, буде Після основного результату пошуку система виводить блок із релевантними варіантами. Розташування такого блоку підвищує імовірність того, що користувач зверне на нього увагу. Наприклад, після вибору стандартного пошуку курортів у певному регіоні, користувачеві

можуть пропонуватись інші цікаві напрямки в межах країни або ті, що відповідають його інтересам.

Нижче наведено структуру зберігання даних в нашій базі даних. Вона включає деталі маршрутів, які відображаються, а також основні дані для ідентифікації та взаємодії користувача із системою:

|                     |                                                                                        |
|---------------------|----------------------------------------------------------------------------------------|
| <b>Маршрути:</b>    | Унікальний ідентифікатор, зображення, заголовок, опис, рейтинг, локація, тип маршруту. |
| <b>Користувачі:</b> | Електронна пошта, пароль, переглянуті маршрути.                                        |

Таблиця 2.1 – Збереження даних в базі даних.

Також після всіх описаних ключових функціональностей, необхідно відобразити на діаграмі, як будуть взаємодіяти три наші основні компоненти додатку:

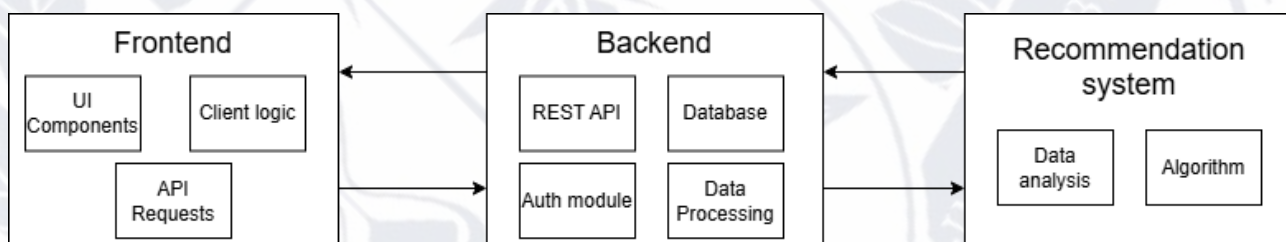


Рисунок 2.5 – Взаємодія компонентів.

Наведена схема на рисунку 2.5 демонструє взаємодію трьох основних частин нашого додатку. Компоненти пов'язані між собою для забезпечення коректної роботи системи, кожен компонент відповідає за свою область призначення.

### Висновок до другого розділу

У другому розділі було визначено ключовий функціонал веб-додатку, який є основою для розробки продукту, що відповідає потребам користувачів і конкурентним вимогам ринку. Основна увага приділялася вибору технологій, які забезпечать ефективність, швидкість і зручність використання сайту.

Визначення основних функцій, таких як реєстрація та авторизація, взаємодія з формами, адаптивна верстка, підтримка маршрутизації та швидке оновлення DOM без перезавантаження сторінки, демонструє, що веб-додаток повинен бути не лише функціональним, а й зручним у використанні. Особливо

важливою є проблема рендерингу, яка суттєво впливає на SEO та користувацький досвід.

Було детально розглянуто проблеми, пов'язані з клієнтським рендерингом, такі як недоступність контенту для пошукових систем, що може негативно вплинути на видимість сайту. На основі цих викликів було обґрунтовано важливість застосування серверного рендерингу (SSR) як рішення для оптимізації SEO і підвищення швидкості завантаження. Цей підхід забезпечує готовий HTML-код, який можна безпосередньо індексувати, зменшуючи час, необхідний для рендерингу, та покращуючи загальний досвід користувачів.

Таким чином, належне визначення функціоналу та вибір технологій є критично важливими етапами у створенні веб-додатку, що відповідає сучасним вимогам і задовольняє потреби кінцевих користувачів.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ДОДАТКУ

#### 3.1 Реалізація front-end частини.

У рамках розробки веб-додатку для туристичних подорожей, фронтенд [50] частина грає ключову роль у забезпеченні взаємодії користувача з системою. Відповідно до вимог, визначених у попередніх розділах, основною метою фронтенду є створення зручного та інтуїтивно зрозумілого інтерфейсу, що дозволяє користувачам легко шукати, переглядати та планувати подорожі.

Щодо вибору необхідних технологій, потрібно виокремити найнеобхідніші в нашому випадку.

Однією з основних технологій, яка буде використовуватися для реалізації проекту, є бібліотека React. Ця бібліотека є одним із найпопулярніших інструментів для розробки веб-додатків у світі, завдяки чому вона отримала широку підтримку як з боку розробників, так і великих компаній. Однією з ключових переваг React є наявність чудової документації, що надає розробникам детальні інструкції та рекомендації щодо використання різних функцій бібліотеки.

Бібліотека React дозволяє розробникам створювати веб-додатки швидко та ефективно завдяки використанню компоненти-зорієнтованого підходу. Однією з ключових особливостей бібліотеки є використання розмітки JSX [51], яка поєднує JavaScript з HTML-подібним синтаксисом, що спрощує створення та використання власних компонентів.

Компоненти в React є повторно використовуваними, що дозволяє їх легко комбінувати з іншими компонентами для створення складних та багатофункціональних веб-додатків. Така модульність не тільки спрощує розробку, але й робить додатки більш гнучкими та масштабованими.

React має значні переваги у контексті управління оновленнями інтерфейсу користувача. Зокрема, бібліотека використовує концепцію віртуального DOM, що дозволяє оновлювати тільки ті елементи інтерфейсу, які дійсно змінилися, без необхідності повного перерендерингу сторінки.

Це значно підвищує продуктивність додатків, оскільки зменшується кількість операцій із оновленням DOM. Завдяки цьому, користувачі можуть насолоджуватися швидким відгуком додатків навіть при великій кількості динамічних елементів.

Ще однією важливою перевагою є інтеграція з бібліотекою `react-router-dom` [52], яка дозволяє ефективно управляти маршрутизацією в сучасних веб-додатках.

React Router DOM є однією з ключових бібліотек, яка інтегрується з React для управління маршрутизацією в веб-додатках, що мають декілька сторінок або розділів. Завдяки цьому інструменту розробники можуть ефективно створювати багатосторінкові додатки, де кожен розділ або сторінка відображається за певним URL.

Важливим аспектом є те, що React Router DOM дозволяє розробникам організувати плавну навігацію між розділами без повного перезавантаження сторінки, що значно покращує користувацький досвід.

Таким чином, використання React у поєднанні з React Router DOM створює міцний фундамент для розробки сучасних, динамічних веб-додатків. Інтеграція маршрутизації дозволяє організувати логічну структуру додатка, забезпечуючи плавну навігацію між сторінками та динамічне завантаження контенту. Це не тільки покращує продуктивність, але й забезпечує високий рівень зручності для користувачів.

Оскільки для розробки веб-додатку ми використовуємо бібліотеку React, і нам дуже важлива зручна та швидка розробка форм і також необхідна подальша підтримка для роботи з формами, в цьому допоможе бібліотека `react-hook-form` [53].

Бібліотека `react-hook-form` - це популярний та потужний інструмент для роботи з формами в React-додатках, який забезпечує ефективне управління станом форми і її валідацією без перерендерювання компонентів, що дає пришвидшення для роботи сайту в цілому.

Вона забезпечує простий в налаштуванні і інтуїтивно зрозумілий API, який дозволяє розробникам, які використовують, або хочуть використовувати цю бібліотеку, швидко та якісно інтегрувати форми в додаток. У порівнянні з іншими схожими рішеннями, react-hook-form виділяється своєю продуктивністю та легкістю у використанні.

Однією з ключових переваг бібліотеки є те, що вона, наскільки це можливо, мінімізує кількість повторних рендерингів компонентів при введенні даних користувачем у форму на сайті. Це досягається через використання нативних елементів форм та управління станом цієї форми за допомогою хуків в вищезазначеній бібліотеці React.

Також варто зазначити ще одну вагому перевагу react-hook-form, яка полягає в вбудованій підтримці валідації полів форми. Тобто, під час того, як користувач вводить певні дані в формах на сайті, можна легко і зручно слідкувати за цими даними, і в разі неправильних даних, повертати користувачу помилку валідації, що безперечно покращує досвід користування сайтом, і не дає користувачеві відправити неправильні дані на сервер, для запобігання помилок в майбутньому.

Підсумовуючи про роботу з формами, яка є ключовою можливістю для нашого сайту, ми обрали чудову бібліотеку, яка пропонує відмінну продуктивність, легкість у використанні та гнучкість. Завдяки простому API, можливостям для валідації та підтримці динамічних форм, вона стає незамінним інструментом для розробників. Вона дозволяє скоротити кількість коду, забезпечити стабільну роботу додатків і спростити управління форм, що робить її ідеальним вибором для будь-яких проектів, в тому числі і для нашого теж.

Наступним не менш важливим етапом є вибір інструменту для стилізації, який б допоміг швидко та якісно надати потрібні стилі для проекту.

Tailwind CSS [54] — це утилітарний фреймворк для CSS, який дозволяє будувати сучасні веб-інтерфейси без необхідності писати власні стилі або створювати окремі файли CSS. Його головна особливість полягає у

використанні набору готових класів, які відповідають конкретним властивостям CSS.

Tailwind надає можливість швидко створювати стильові рішення, не відволікаючись на написання великої кількості кастомного CSS, що робить його зручним для роботи з React.

Також варто зазначити, що використання Tailwind CSS в даний час дуже популярне і все більше набирає обертів. В парі з React використання стало дуже популярним завдяки тому, що він дозволяє зосереджуватись розробникам на створенні та підтримці компонентів, не думаючи надто багато щодо стилізації цих компонентів. Tailwind дуже добре підходить для веб-додатків на React, оскільки кожен складову можна швидко стилізувати, додаючи потрібні класи безпосередньо в JSX-код.

Отже, після ретельного вибору відповідних технологій, а також врахування особливостей взаємодії користувача з веб-сайтом, постає необхідність створення щонайменше трьох основних сторінок. Ці сторінки виконуватимуть ключові функції, такі як реєстрація нових користувачів, аутентифікація вже існуючих, а також забезпечення можливості комфортного користування сайтом, зокрема, для пошуку маршрутів.

Перша сторінка буде присвячена реєстрації користувачів. Вона повинна бути простою і зрозумілою, щоб нові відвідувачі могли без труднощів заповнити необхідні поля для створення облікового запису. Тут важливо забезпечити валідацію введених даних, щоб користувачі отримували зворотний зв'язок про успішність чи помилки під час реєстрації.

Друга сторінка зосередиться на аутентифікації. На цій сторінці користувачі зможуть вводити свої облікові дані, такі як логін і пароль, для входу в систему. Важливо забезпечити надійний механізм захисту даних, щоб гарантувати безпеку облікових записів.

Третя сторінка стане основним інтерфейсом для користувачів, які шукають маршрути. Тут необхідно створити зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє легко вводити запити, переглядати результати пошуку і фільтрувати

їх за різними критеріями. Ця сторінка повинна також містити рекомендаційний блок, що підказує користувачам можливі маршрути на основі їхніх уподобань і історії пошуку.

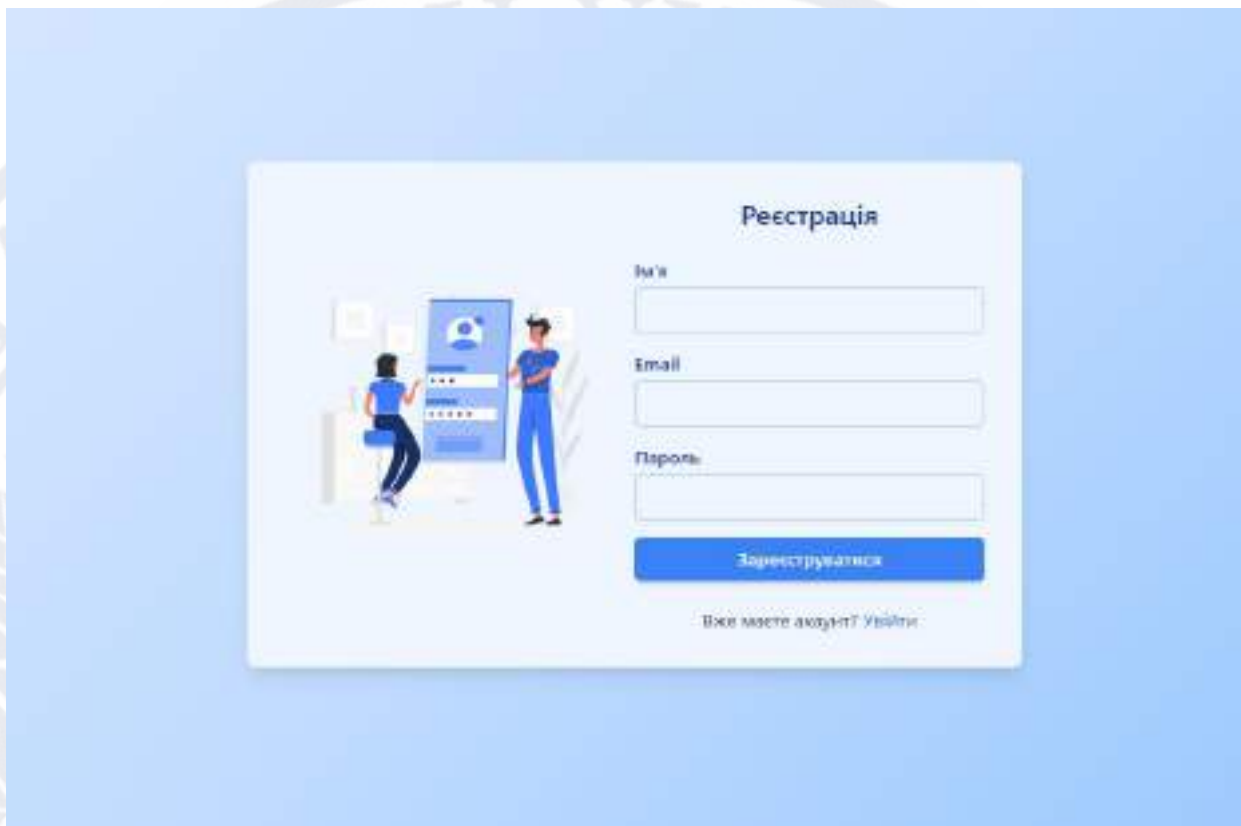


Рисунок 3.1 – Сторінка реєстрації

Сторінка реєстрації зображена на рисунку 3.1 є важливим елементом веб-додатку і містить форму, яка розроблена з урахуванням сучасних стилістичних вимог. Дизайн цієї форми виконаний у приємних кольорах, що сприяють комфортному сприйняттю, та враховує основні елементи, необхідні для введення інформації користувачем. Серед цих елементів обов'язковими є поля для введення Ім'я, електронної пошти та пароля, що забезпечують зручність та зрозумілість при заповненні форми.

Крім того, на сторінці реалізовано можливість переходу на сторінку автентифікації, що є корисним для тих користувачів, які вже мають обліковий запис і бажають увійти в систему. Це полегшує навігацію та підвищує зручність використання веб-додатку.

У разі неправильного введення даних система забезпечує зворотний зв'язок, показуючи користувачу повідомлення про помилку, що дозволяє

уникнути непорозумінь і швидко виправити помилки. Якщо ж користувач правильно заповнить усі поля, сайт перенаправить його на сторінку автентифікації. Тут користувачеві знову буде запропоновано ввести свої дані для входу, що завершує процес реєстрації та відкриває доступ до можливостей сайту.

Таким чином, сторінка реєстрації не лише забезпечує функціональність, але й створює позитивне враження від взаємодії з додатком завдяки продуманому дизайну та зручному користувацькому інтерфейсу.

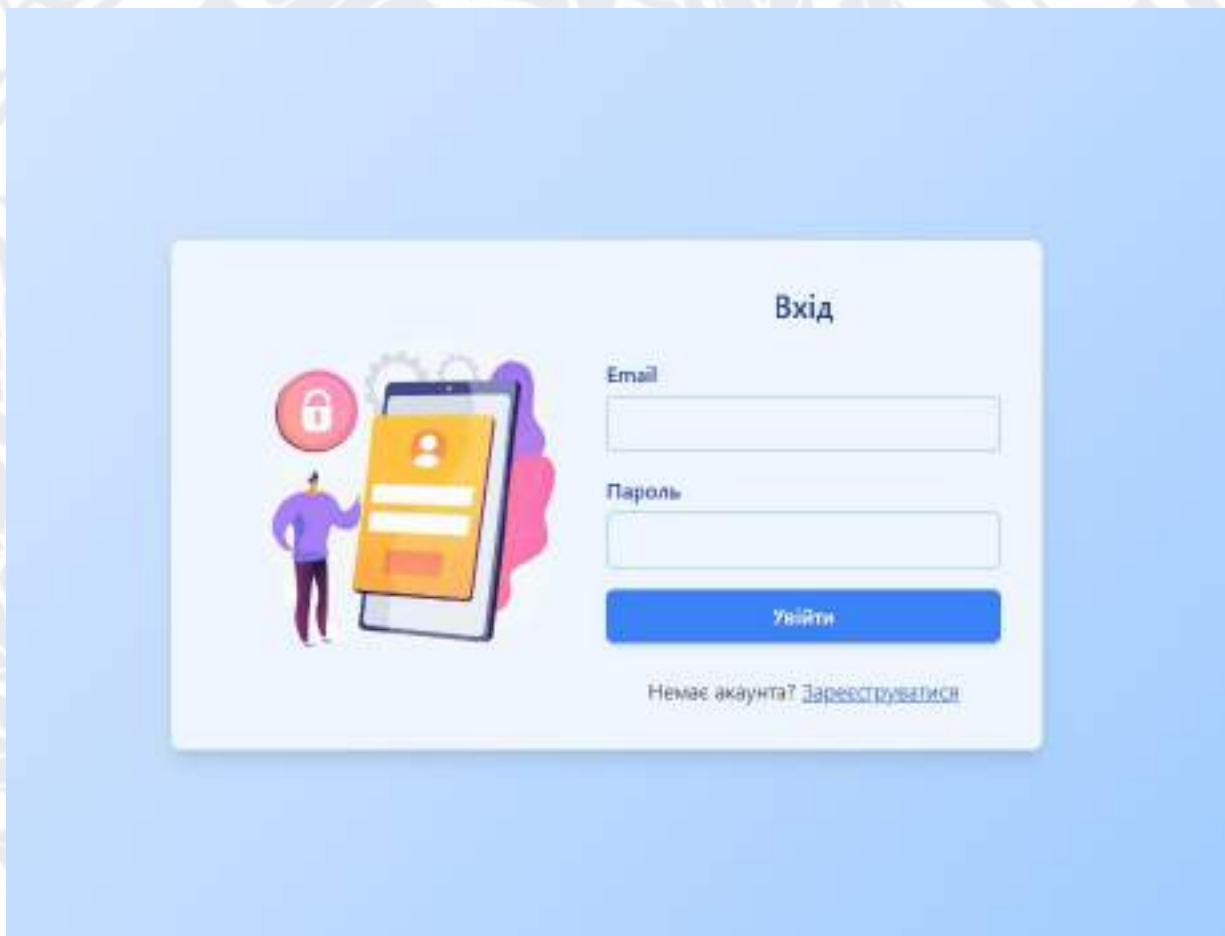


Рисунок 3.2 – Сторінка автентифікації

В цілому, сторінка автентифікації, яка зображена на рисунку 3.2 має подібний дизайн до сторінки реєстрації, але з кількома ключовими відмінностями. По-перше, вона не містить поля для введення імені, оскільки в цьому випадку користувач вже має обліковий запис. Сторінка також забезпечує можливість переходу на сторінку реєстрації для тих, хто ще не зареєструвався, що спрощує навігацію і дозволяє новим користувачам швидко знайти потрібну інформацію.

Після того, як користувач успішно введе свої дані, включаючи електронну пошту та пароль, система перенаправить його на головну сторінку. Це є важливим етапом, оскільки на головній сторінці користувач зможе в повній мірі скористатися можливостями сайту, такими як пошук маршрутів, перегляд рекомендацій та взаємодія з іншими функціями платформи.

На головній сторінці слід детально розглянути кожен компонент окремо, щоб забезпечити чітке розуміння алгоритму роботи сайту в цілому. Це дозволить користувачеві краще орієнтуватися у функціях платформи та сприятиме інтуїтивному використанню додатку.

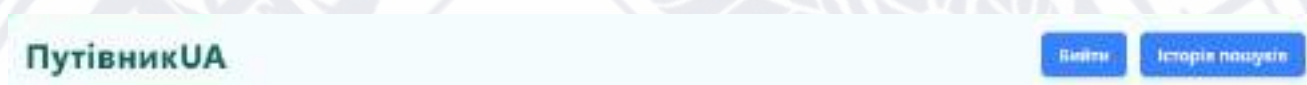


Рисунок 3.3 – Шапка сайту

Почати опис головної сторінки можна з шапки сайту, яка зображена на рисунку 3.3, яка має стандартний вигляд для більшості сучасних веб-додатків. Вона виконує роль основного навігаційного елемента і містить важливі функції для швидкого доступу до ключових розділів.

Ліворуч розміщено логотип сайту – у нашому випадку це назва «ПутівникUA». Логотип створений у лаконічному стилі, який одразу відображає суть веб-додатку. Завдяки назві «ПутівникUA» користувачеві з першого погляду стає зрозуміло, що сайт присвячений подорожам по Україні та допомагає в плануванні маршрутів.

Праворуч у шапці знаходяться дві важливі кнопки:

- Кнопка "Вийти" – призначена для забезпечення можливості зміни облікового запису. Користувач може вийти з поточного облікового запису, а при потребі увійти в інший. Це зручно для тих, хто має кілька облікових записів або користується сайтом із загального пристрою.
- Кнопка "Історія пошуків" – надає користувачеві швидкий доступ до всіх його попередніх запитів. Ця функція корисна для перегляду та відновлення маршрутів, які вже були досліджені, дозволяючи

повторно подивитися цікаві місця або маршрути. Це значно покращує користувацький досвід, адже дає змогу зберігати важливу інформацію про попередні пошуки, щоб не вводити дані знову.

Також важливо зазначити, що обидві ці кнопки не доступні для неавторизованих користувачів.

Таким чином, шапка сайту є не лише красивим, але й функціональним елементом. Вона дозволяє користувачеві швидко і зручно взаємодіяти з платформою, маючи під рукою основні функції для ефективної навігації та керування обліковим записом.

Наступним елементом на головній сторінці є форма пошуку туристичних маршрутів.

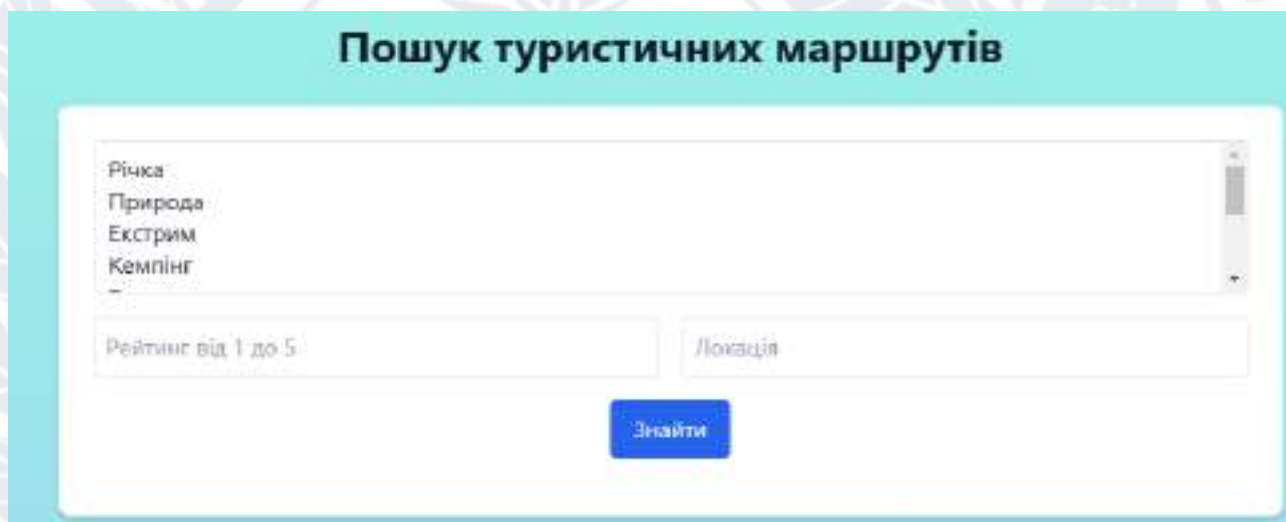


Рисунок 3.4 – Форма пошуку подорожей

Форма пошуку, яку зображено на рисунку 3.4, представлена у мінімалістичному стилі, який водночас забезпечує повний необхідний функціонал. Розташована одразу під шапкою сайту, вона залишається доступною для всіх користувачів, незалежно від того, чи вони авторизовані. Це рішення дозволяє швидко і легко розпочати пошук маршрутів.

У формі представлені поля для введення тексту та випадаючі списки, які обмежують вибір варіантів. Такий підхід не лише спрощує процес введення, а й дозволяє користувачеві задавати більш конкретні критерії для пошуку, підвищуючи гнучкість та точність результатів. Наприклад, користувач може

ввести місце відправлення, пункт призначення або обрати інші специфічні параметри маршруту.

Основним елементом форми є кнопка "Пошук", яка ініціює запит до бази даних. Після натискання кнопки введені дані проходять валідацію – перевірку на коректність та повноту. Якщо всі поля заповнені правильно, запит відправляється на сервер, де відбувається обробка запиту. У разі успішної обробки сервер відповідає списком доступних маршрутів, що відповідають заданим параметрам. Таким чином, форма пошуку забезпечує користувача простим інструментом для доступу до необхідної інформації, а її мінімалістичний дизайн допомагає зосередитися на основному завданні – виборі маршрутів для подорожей.

Як вже було зазначено вище, після успішного запиту, від серверу повертаються результати. На сайті вони знаходяться нижче форми пошуку, і те як вони виглядають, можна переглянути на рисунку, який наведено нижче.

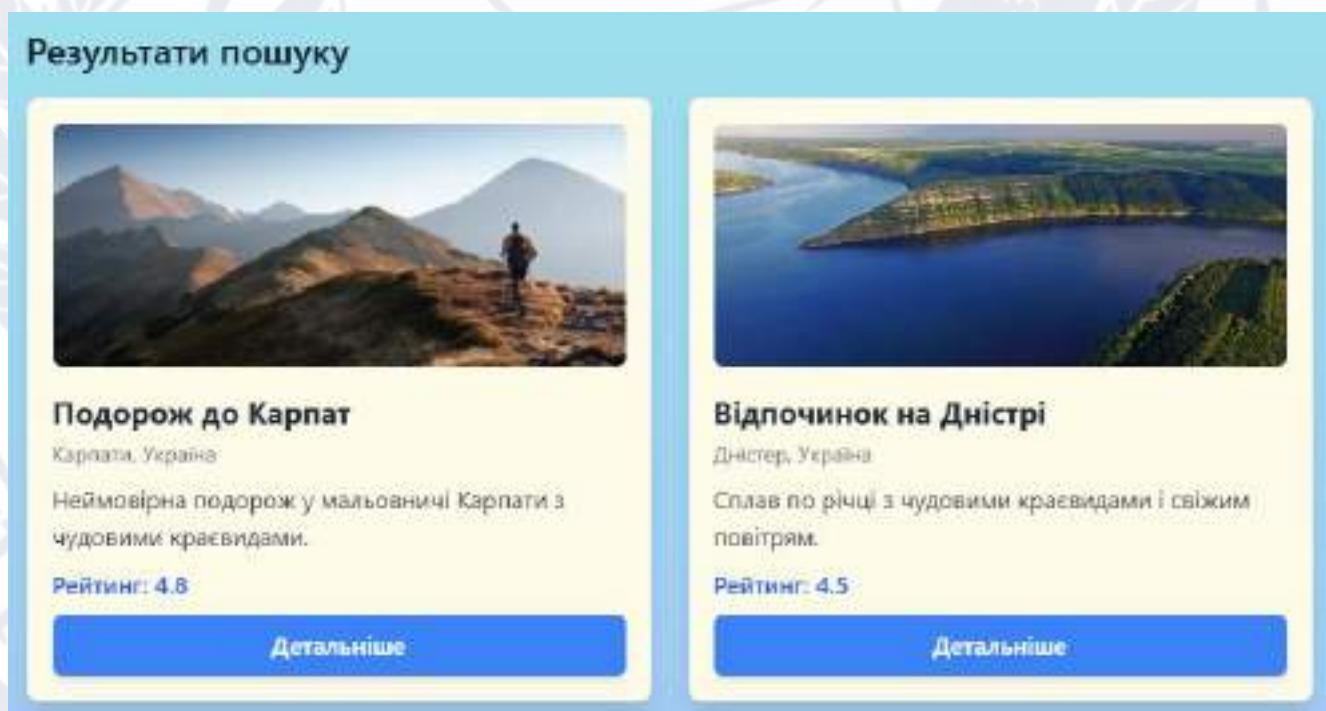


Рисунок 3.5 – Перелік результатів маршрутів для подорожей

Список результатів маршрутів, на рисунку 3.5, представлений у вигляді горизонтально ряду карток, де кожна картка містить інформацію про один маршрут. Такий підхід дозволяє компактно та зручно розмістити маршрути,

полегшуючи вибір та візуальне сприйняття. Якщо ж за заданими фільтрами від користувача даних не знайшлось – на сайті буде відповідний надпис.

На кожній картці розміщено ключову інформацію про маршрут:

- Назва подорожі – короткий, але інформативний заголовок, що дозволяє користувачу швидко орієнтуватися серед запропонованих варіантів.
- Опис подорожі – короткий текст, що надає загальне уявлення про маршрут, включаючи основні деталі, які можуть зацікавити користувача.
- Кнопка "Детальніше" – при натисканні на цю кнопку користувач переходить на сторінку з більш розгорнутою інформацією про маршрут. Це корисно для детального ознайомлення, якщо короткий опис зацікавив користувача.
- Зображення – надає візуальне уявлення про маршрут або локацію, до якої можна дістатися. Завдяки цьому користувач може одразу зрозуміти, про який тип подорожі йдеться, чи відповідає маршрут його інтересам.

Список адаптований під різні розміри екранів: в залежності від ширини екрану картки автоматично перебудовуються, забезпечуючи зручне відображення як на широких моніторах, так і на мобільних пристроях. На великих екранах користувач може бачити одночасно до шести карток у рядку, що дозволяє охопити більше маршрутів, тоді як на мобільних пристроях картки можуть бути розміщені по одній у рядку для зручного перегляду.

Такий адаптивний дизайн підвищує комфорт користувача під час вибору маршруту, а картковий інтерфейс забезпечує швидке та інтуїтивне сприйняття доступних варіантів подорожей.

Завершуючи огляд головної сторінки, потрібно розповісти про останній, але тим не менш дуже важливий блок «Можливо вас зацікавить».

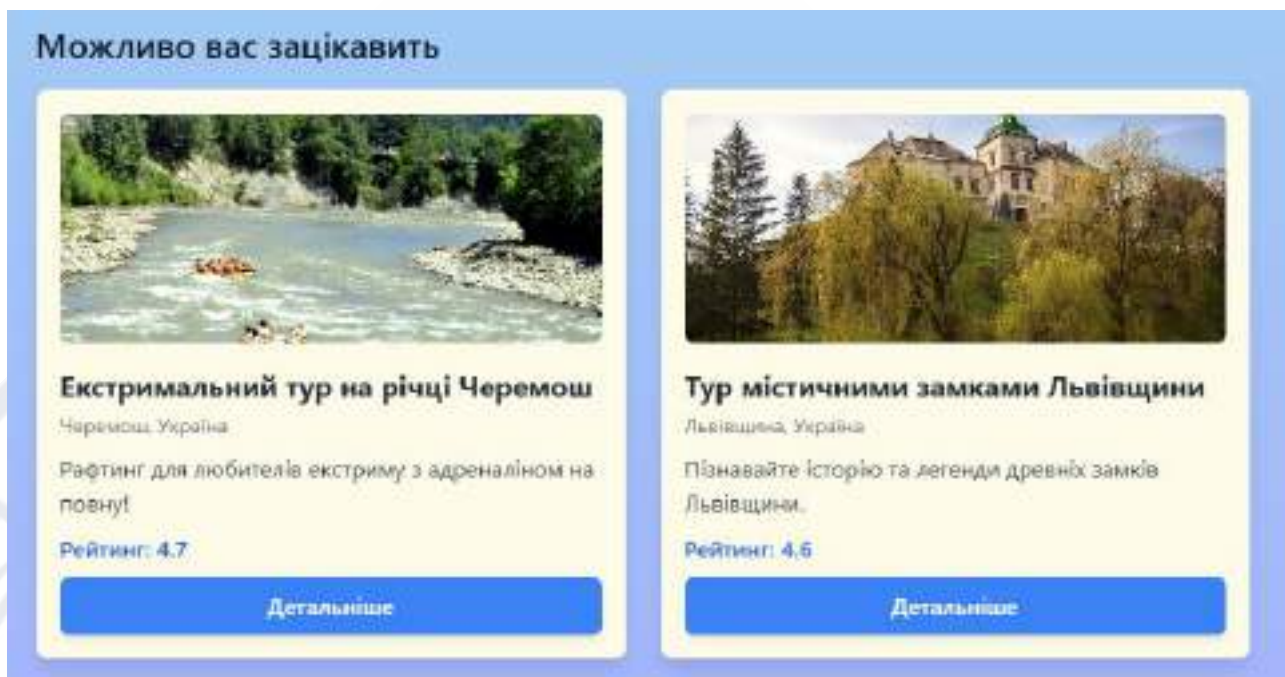


Рисунок 3.6 – Блок «Можливо вас зацікавить»

Цей блок, який зображено на рисунку 3.6, відображення маршрутів аналогічний до попереднього блоку з результатами пошуку, але виконує функцію рекомендаційної системи. Він призначений для відображення персоналізованих пропозицій подорожей, які можуть зацікавити користувача на основі його минулих пошуків та вподобань, а також на основі виборів інших користувачів зі схожими інтересами.

Рекомендаційний блок доступний тільки авторизованим користувачам, що дозволяє формувати точніші пропозиції. Також важливо зазначити, що його робота залежить від наявності достатньої кількості даних у базі для аналізу вподобань користувачів. Якщо ж мінімальна необхідна кількість даних відсутня, цей блок не буде відображатися, щоб не створювати користувачеві зайвого навантаження і не показувати випадкових маршрутів.

Таким чином, цей блок служить інструментом для більш персоналізованого користувацького досвіду, демонструючи релевантні рекомендації та стимулюючи користувача відкривати для себе нові цікаві маршрути.

### 3.2 Реалізація back-end частини

Архітектура нашого бекенду базується на використанні Express.js [55] для обробки HTTP-запитів [56], MongoDB [57] для зберігання даних і JWT [58] для авторизації користувачів. Завдяки цьому ми отримуємо легку, гнучку та безпечну систему для керування користувачами та даними про подорожі. Кожен компонент працює у тісній взаємодії з іншими, забезпечуючи стабільність і масштабованість.

Далі потрібно детальніше ознайомитись з кожною технологією, яка використовується.

Можна почати з Express.js - це популярний серверний фреймворк для Node.js, який допомагає створювати веб-додатки і API з високою продуктивністю та гнучкістю. Його легка вага та мінімалізм дозволяють налаштовувати архітектуру додатку під конкретні вимоги, додаючи лише потрібні компоненти. Express.js дозволяє зручно організувати код за допомогою:

- Маршрутизації: Створення окремих шляхів для кожного ресурсу (наприклад, /register та /login для автентифікації) значно спрощує управління функціоналом. Express підтримує методи HTTP, такі як GET, POST, PUT, DELETE, для взаємодії з різними ресурсами.
- Middleware: Express дозволяє використовувати middleware (проміжні функції), які обробляють запити перед їх потраплянням на основну кінцеву точку. Це може бути корисно для таких функцій, як перевірка автентифікації, обробка помилок та логування.

Express є чудовим варіантом для побудови REST API завдяки простоті налаштування та широкій спільноті розробників, які підтримують фреймворк, завдяки чому в разі потреби не складно знайти вирішення проблеми з якою можна стикнутись під час розробки.

MongoDB - це база даних, яка зберігає дані у форматі JSON-подібних [59] документів. Це означає, що дані структуруються у гнучкі документи, що робить MongoDB відмінним вибором для додатків, які постійно змінюються або розширюються.

Основні особливості MongoDB, що роблять її зручною для бекенду нашого додатку:

- Гнучка структура даних: MongoDB дозволяє зберігати дані з різною структурою в одній колекції, що особливо корисно для зберігання даних, які можуть змінюватися. Наприклад, дані користувача можуть мати різні поля для різних типів подорожей, а MongoDB легко обробляє такі зміни.
- Швидкість роботи: Завдяки відсутності складних зв'язків між таблицями, MongoDB є швидшою у порівнянні з реляційними базами даних. У випадку додатку, який може зберігати тисячі даних про подорожі, швидкість доступу до них має критичне значення.
- Масштабованість: MongoDB забезпечує горизонтальне масштабування, що дозволяє розподілити базу даних по кількох серверах для забезпечення високої доступності та надійності. Це важливо для додатку, який повинен обслуговувати велику кількість користувачів одночасно.

Також слід зазначити основні недоліки цієї бази даних, задля кращого розуміння цілісної картини:

- Відсутність транзакцій на рівні SQL: MongoDB має підтримку багатодокументних транзакцій, але вони менш ефективні та складніші у налаштуванні, ніж транзакції в SQL.
- Проблеми з цілісністю даних: Через відсутність строгих схем дані можуть зберігатися в різних форматах або містити помилки. Це ускладнює обробку й аналіз, наприклад, коли один маршрут дублюється або дані про ціни записані в різних форматах.
- Проблеми з уніфікованістю даних: Гнучкість структури документів може призвести до невідповідностей у назвах полів чи відсутності деяких даних. Це ускладнює виконання запитів та розробку функцій, які залежать від чіткої структури даних.

Таким чином, MongoDB є потужним інструментом для створення масштабованих додатків, що дозволяє швидко змінювати структуру даних відповідно до потреб проекту.

Mongoose [60] — це бібліотека для роботи з MongoDB у Node.js, яка надає інструменти для створення схем і моделей даних. Завдяки Mongoose ми можемо чітко визначити структуру наших даних, валідувати їх і виконувати основні CRUD-операції (створення, читання, оновлення, видалення) з мінімальними зусиллями.

- Схеми: Mongoose дозволяє визначити схеми, що чітко структурують наші дані. Наприклад, схема для користувача може включати поля для імені, електронної пошти, пароля тощо. Кожне поле можна валідувати, наприклад, переконатися, що адреса електронної пошти є унікальною.
- Моделі: Після визначення схеми, Mongoose дозволяє створити модель, яка забезпечує доступ до конкретної колекції MongoDB.
- Валідація: Mongoose має вбудовані механізми валідації даних, що забезпечує надійність додатка. Наприклад, можна задати обов'язкові поля або мінімальні значення, щоб уникнути помилок на стороні користувача.

Після цього можна впевнено стверджувати, що Mongoose робить взаємодію з MongoDB більш організованою та структурованою, що полегшує розробку та подальше підтримання додатку.

JWT token [61] — це сучасний і надійний метод для передачі даних між клієнтом і сервером у вигляді токенів, що активно використовується у веб-додатках. Основна ідея JWT полягає у тому, що клієнт отримує токен після первинної авторизації і надалі використовує його для доступу до ресурсів, не потребуючи повторного введення логіну та паролю. Такий підхід дозволяє зменшити навантаження на сервер і знизити ризик витоку конфіденційних даних під час постійної передачі ідентифікаційної інформації.

Токен JWT складається з трьох важливих частин, що формують його структуру:

- Заголовок. Ця частина токена містить метадані про токен, зокрема тип (в даному випадку це JWT) та алгоритм шифрування. Заголовок допомагає серверу швидко визначити, як саме він має працювати з токеном і як його

декодувати, що важливо для коректної обробки та збереження цілісності даних.

- Дані. Ця частина часто називається "корисним навантаженням", оскільки вона містить самі дані, необхідні для автентифікації та авторизації користувача. Сюди можна додати інформацію про користувача, таку як унікальний ідентифікатор користувача, права доступу (наприклад, роль адміністратора чи звичайного користувача) і навіть додаткові атрибути, що можуть знадобитися для обробки запиту. Дані не шифруються, але їх захист забезпечується шляхом генерації підпису, що робить дані недоступними для сторонніх, навіть якщо сам токен буде перехоплений.
- Підпис. Це, мабуть, найважливіша частина JWT, що відповідає за безпеку. Підпис формується шляхом хешування даних заголовка та корисного навантаження із застосуванням секретного ключа, відомого лише серверу. Цей підпис гарантує цілісність і незмінність токена — будь-яка спроба змінити токен призведе до його невалідності, оскільки підпис більше не збігатиметься з даними у токені. Лише сервер, володіючи секретним ключем, може перевірити цей підпис, що підвищує безпеку передачі.

JWT особливо популярний для авторизації у веб-додатках завдяки своїй універсальності та простоті інтеграції з фронтендом. Саме тому це одна з причин чому ми його використовуємо в нашому додатку.

Клієнт може зберігати токен у локальному сховищі браузера або у cookie, [62] що дозволяє використовувати його для повторних запитів до сервера. Таким чином, JWT забезпечує безперервність сеансу користувача і значно спрощує побудову масштабованих та безпечних систем, де безпека даних має критичне значення.

Оскільки ми використовуємо JWT Token в нашому додатку, можна розглянути, як буде відбуватись реєстрація та авторизація для користувача.

Після введення даних для реєстрації, сервер хешує пароль користувача для безпеки і зберігає дані у базі даних. Як тільки новий користувач створений, йому

можна повернути JWT токен для автоматичного входу, або просто підтвердити успішну реєстрацію.

Коли користувач вводить свої дані для входу або авторизації, сервер перевіряє наявність користувача в базі, порівнює введений пароль із хешованим паролем у базі. Якщо паролі збігаються, сервер генерує токен JWT, який потім передається клієнту. Цей токен дозволяє здійснювати подальші запити від імені користувача без потреби заново вводити логін і пароль.

### **3.3 Результати роботи рекомендаційної системи**

Наша система рекомендацій для туристичних маршрутів — це сучасний, інтуїтивно зрозумілий веб-додаток, який поєднує розширені можливості пошуку з аналізом користувацької поведінки, дозволяючи надавати користувачам персоналізовані рекомендації. Основною метою нашої рекомендаційної системи є не лише допомога користувачам у знаходженні відповідних маршрутів, але й заохочення їх досліджувати нові, несподівані варіанти, які могли б їх зацікавити на основі попередніх пошукових запитів і взаємодії з додатком.

Потрібно поговорити про те як працює наша рекомендаційна система. Наша рекомендаційна система базується на принципах двох ключових підходів — фільтруванні на основі заданих критеріїв та врахуванні історії переглядів. Цей розподіл дозволяє нам комбінувати поточні вподобання користувачів із їхніми минулими інтересами, щоб надавати найбільш релевантні та різноманітні рекомендації. Даваймо детально розглянемо ці підходи.

1. Фільтрація на основі пошукових критеріїв. При здійсненні пошуку користувач має можливість задавати певні параметри: тип відпочинку, рейтинг маршруту та бажану локацію. Коли користувач вводить ці параметри, система відразу зберігає їх як певний набір критеріїв для подальшого формування рекомендацій. У базі даних здійснюється пошук маршрутів, які найбільше відповідають зазначеним критеріям, з використанням відповідних запитів. Наприклад, якщо користувач обирає типи "гори" і "екстрим", рекомендаційна система обирає подібні маршрути, що мають відповідні теги та відповідають заданим умовам

рейтингу. Важливо, що така система дозволяє одночасно застосовувати кілька фільтрів — у цьому полягає її гнучкість і потужність. Якщо один із критеріїв (наприклад, локація) залишається порожнім, система не обмежує пошук, що дозволяє уникнути обмеження варіантів і надає користувачеві ширший спектр пропозицій.

2. Врахування історії переглядів користувача. Система також веде облік маршрутів, які переглядав користувач. Це дозволяє зрозуміти його приховані уподобання, що не завжди виражаються через поточні пошукові критерії. Наприклад, якщо користувач частіше переглядає маршрути з високим рейтингом або певними характеристиками (як-от конкретний тип відпочинку або популярні теги), ці дані зберігаються для подальшого аналізу. Цей підхід дозволяє будувати рекомендації на основі довгострокових вподобань. Якщо користувач переглядає декілька маршрутів певного типу, система зможе запропонувати йому подібні маршрути навіть після зміни поточних пошукових критеріїв. Наприклад, користувач, який неодноразово переглядав "екстримальні" маршрути, буде частіше бачити такі варіанти в блоці рекомендацій "Можливо вас зацікавить", навіть якщо його поточний пошук зосереджений на інших параметрах.

Система забезпечує дві зони, де відображаються результати. Перша зона — це результати основного пошуку, які відповідають заданим критеріям і з'являються одразу під формою. Це прямі результати на основі вибраних фільтрів. Друга зона — це блок "Можливо вас зацікавить". Тут відображаються результати на основі рекомендаційної системи, яка аналізує історію взаємодії користувача із сайтом.

Цей підхід до рекомендацій дозволяє користувачам розширювати власні горизонти, надаючи їм доступ до маршрутів, які вони могли б упустити через початкові обмеження в пошуку. Рекомендації ґрунтуються на раніше переглянутих варіантах і актуальних вподобаннях, що створює баланс між новими пропозиціями та звичними для користувача вподобаннями.

Також варто поговорити і про те, що система не відразу може надавати рекомендації, оскільки наш підхід вимагає накопичення інформації, система проходить кілька стадій перед тим, як почати показувати рекомендації:

1. Первинна фаза: у перші моменти використання система не має достатніх даних, щоб надати рекомендації. Користувачу показуються результати, які відповідають пошуковим критеріям, але без додаткових персоналізованих пропозицій.
2. Стадія накопичення: система збирає дані при кожному пошуку і кожному переході на детальну сторінку маршруту. Чим більше користувач взаємодіє із сайтом, тим більше система розуміє його вподобання.
3. Активація рекомендаційного блоку: після певної кількості пошуків та відвідувань детальних сторінок система починає генерувати персоналізовані пропозиції у блоці "Можливо вас зацікавить". Вона порівнює поточні інтереси користувача з іншими доступними маршрутами і визначає, які з них мають потенціал зацікавити користувача.

Цей метод дозволяє створювати рекомендації, які базуються не тільки на одному запиті або перегляді, а на комплексному профілі користувача. Наприклад, якщо користувач регулярно обирає маршрути з високим рейтингом та такими тегами як "гори" і "екстрим", система надає пріоритет маршрутам зі схожими параметрами. Це також запобігає надмірній кількості нерелевантних рекомендацій на ранніх етапах взаємодії.

Схематично це все доволі легко зобразити для кращого розуміння нижче, на рисунку 3.7:

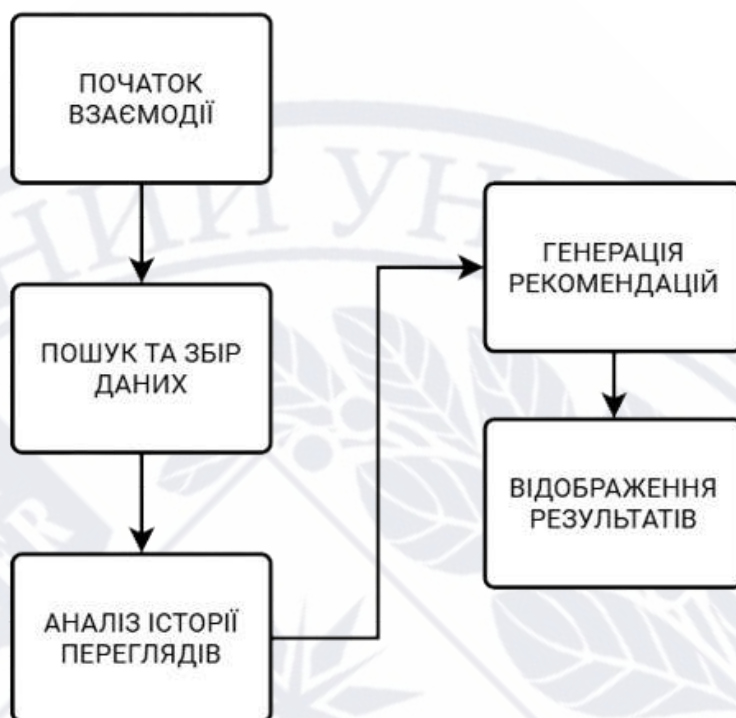


Рисунок 3.7 – Схема роботи рекомендаційної системи.

У підсумку, наша рекомендаційна система є потужним інструментом для підвищення користувацького досвіду на туристичному вебсайті. Вона поєднується з пошуковими фільтрами та аналізом історії переглядів, дозволяючи користувачам знаходити найбільш відповідні маршрути на основі їхніх уподобань та попередніх дій. Система не лише пропонує рекомендації, а й адаптується до користувацької взаємодії, забезпечуючи більш персоналізовані результати з часом.

Завдяки гнучкості та масштабованості, вона готова до подальшого вдосконалення, зважаючи на зростання бази даних та збільшення кількості користувачів, що дозволяє впроваджувати нові методи аналізу та розширення функціоналу.

### **Висновок до третього розділу**

У третьому розділі було детально розглянуто реалізацію фронтенд і бекенд частин веб-додатку для туристичних подорожей, а також результати роботи рекомендаційної системи. Реалізація фронтенд частини, який повністю опирався в реалізації на React, дозволила створити інтерактивний та зручний інтерфейс,

що відповідає потребам користувачів. Компонентна архітектура забезпечила високу модульність, що спростило підтримку та розширення функціональності додатку.

Бекенд частина була реалізована на Node.js, що дозволило організувати ефективну обробку запитів та взаємодію з базою даних. Це забезпечило зручну інтеграцію між фронтендом і бекендом, дозволяючи безперешкодно отримувати дані про подорожі та управляти інформацією про користувачів.

Результати роботи рекомендаційної системи продемонстрували її ефективність у підборі подорожей, що відповідають інтересам користувачів. Завдяки аналізу попередніх запитів і поведінки користувачів система змогла надавати релевантні пропозиції, що сприяло покращенню досвіду користувачів та підвищенню ймовірності їх задоволеності від використання додатку.

Загалом, реалізація як фронтенд, так і бекенд частини забезпечила створення цілісного і функціонального веб-додатку, що виконує поставлені завдання і відповідає сучасним вимогам до розробки програмного забезпечення. Рекомендаційна система, у свою чергу, додала значну цінність до загального користувацького досвіду, що підкреслює важливість інтеграції аналітичних механізмів у сучасних веб-додатках.

## ВИСНОВОК

Виконана робота охоплювала кілька важливих етапів. Спочатку було здійснено збір даних по заданій темі та описано цей процес. Далі відбулося ознайомлення з різними технологіями, що включало їх опис, використання, а також класифікацію експертних та рекомендаційних систем. Було проведено пошук аналогів і здійснено огляд схожих сайтів, які функціонують як рекомендаційні системи для складання маршрутів подорожей.

В першому розділі було розглянуто експертні системи, хронологію виникнення експертних систем, з вказанням реальних прикладів, які з'являлись в кожен із зазначених періодів часу. Було розглянуто будову експертних систем, як використовується, як їх класифікують, і де вони найактивніше використовуються в наш час.

Також було розглянуто рекомендаційні системи, більш детально досліджено їхню класифікацію, та особливості використання в наш час. В процесі роботи було виокремлено основні методи класифікації, їх особливості та переваги і недоліки кожної з цих класифікацій. Також було визначено, що рекомендаційні системи дуже популярні в наш час, і сайти дуже активно використовують їх.

В цьому ж розділі було проведено аналіз та порівняння декількох сайтів-аналогів, та в процесі дослідження вдалось виокремити основні переваги та недоліки кожного з веб-додатків, що в майбутньому дасть перевагу перед створенням власного сайту.

Другий розділ зосереджено на архітектурних рішеннях, що включають визначення ключового функціоналу веб-додатку, збір даних для відображення можливих варіантів подорожей та проектування взаємодії користувача з сайтом. Визначення основних вимог і функцій, які мають бути реалізовані, дозволило створити структуру, що забезпечує зручність та ефективність використання додатку. Це сприяло формуванню цілісного бачення проекту, яке відображає потреби цільової аудиторії.

Третій розділ був присвячений практичній реалізації розробленого веб-додатку, що включала створення фронтенд та бекенд частин, а також реалізацію рекомендаційної системи. Використання сучасних технологій, таких як TypeScript та Node.js, дозволило створити інтерактивний інтерфейс та забезпечити ефективну обробку запитів. Результати реалізації показали, що розроблений додаток відповідає визначеним вимогам, а рекомендаційна система успішно підвищує якість користувацького досвіду.

Тому, виконане дослідження та реалізація веб-додатку для туристичних подорожей підтверджують важливість інтеграції рекомендаційних механізмів у сучасні онлайн-системи. Результати роботи демонструють, що відповідний підхід до аналізу, проектування та реалізації може значно полегшити процес вибору подорожей для користувачів, покращуючи їхній загальний досвід.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Експертна система [Електронний ресурс] // Wikipedia. – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%B0\\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0](https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)
2. Циганов О. В. Основи проектування систем штучного інтелекту [Електронний ресурс]. – 2017. – 5 с. – Режим доступу: [http://lib.khnu.km.ua/konfer\\_HNU/2018/int\\_pot/Part\\_2all.pdf](http://lib.khnu.km.ua/konfer_HNU/2018/int_pot/Part_2all.pdf)
3. Що таке штучний інтелект: історія, види та складові [Електронний ресурс] // GigaCloud. – Режим доступу: <https://gigacloud.ua/blog/navchannja/scho-take-shtuchnij-intelekt-istorija-vidi-ta-skladovi>
4. Dendral [Електронний ресурс] // Wikipedia. – Режим доступу: <https://en.wikipedia.org/wiki/Dendral#:~:text=Dendral%20was%20a%20project%20in,formation%20and%20discovery%20in%20science>.
5. MYCIN [Електронний ресурс] // Britannica. – Режим доступу: <https://www.britannica.com/technology/MYCIN>
6. XCON [Електронний ресурс] // Wikipedia. – Режим доступу: <https://en.wikipedia.org/wiki/Xcon>
7. CAD [Електронний ресурс] // NCBI. – Режим доступу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1665219/>
8. IBM Watson Health [Електронний ресурс] // Wikipedia. – Режим доступу: [https://en.wikipedia.org/wiki/IBM\\_Watson](https://en.wikipedia.org/wiki/IBM_Watson)
9. Ada health [Електронний ресурс] // Ada. – Режим доступу: <https://ada.com/>
10. База знань [Електронний ресурс] // Wikipedia. – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0\\_%D0%B7%D0%BD%D0%B0%D0%BD%D1%8C](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D1%8C)
11. Що таке інтерфейс. Різновиди інтерфейсів [Електронний ресурс] // Cases.media. – Режим доступу: <https://cases.media/article/sho-take-interfeis-riznovidi-interfeisiv>
12. Циганов О. В. Машина логічного пояснення // Основи проектування систем штучного інтелекту [Електронний ресурс]. – 2017. – С. 94–105. – Режим доступу: [https://www.google.com.ua/books/edition/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8\\_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F\\_%D1%81/ZkZNIY0\\_FbUC?hl=en&gbpv=1](https://www.google.com.ua/books/edition/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D1%81/ZkZNIY0_FbUC?hl=en&gbpv=1).
13. Ладанюк А. П. Машина виведення // Основи систематичного аналізу [Електронний ресурс]. – 2016. – С. 123–124. – Режим доступу: [https://pdf.lib.vntu.edu.ua/books/2023/Ladanjuk\\_2004\\_176.pdf](https://pdf.lib.vntu.edu.ua/books/2023/Ladanjuk_2004_176.pdf)
14. Кіяновська, Н. М., Рашевська, Н. В., Семеріков, С. О. (2016). Теоретико-методичні засади використання інформаційно-комунікаційних технологій у навчанні вищої математики студентів інженерних спеціальностей у

- Сполучених Штатах Америки. *Інформаційні технології і засоби навчання*, С. 171.
15. International Science Group. Advancing in research, practice and education. 2022
  16. Експертні системи тиражування знань експертів [Електронний ресурс] // SlideShare. – Режим доступу: <https://www.slideshare.net/slideshow/ai-6-75922488/75922488>
  17. International Science Group. Actual priorities of modern science, education and practice. 2022. 365 с.
  18. Загальні принципи діагностування електронних систем керування автомобіля / за ред. М. Б. Копитчука. – 89 с.
  19. Нечипуренко П. П., Семеріков С. О., Томіліна Л. І. Теоретико-методичні засади використання інформаційно-комунікаційних технологій як засобу формування дослідницьких компетентностей старшокласників у профільному навчанні хімії. – 2018. – 91 с. Режим доступу: [https://www.khadi.kharkov.ua/fileadmin/P\\_vcheniy\\_secretar/ABTOM\\_TPAH\\_CII/AE/2020R/AT\\_AE\\_chorne\\_kreslo.pdf](https://www.khadi.kharkov.ua/fileadmin/P_vcheniy_secretar/ABTOM_TPAH_CII/AE/2020R/AT_AE_chorne_kreslo.pdf)
  20. Ткачук В. В., Семеріков С. О. Теорія та методика використання мобільних технологій навчання інформатичних дисциплін у підготовці інженерів-педагогів з цифрових технологій. – Київ: Інститут педагогічної освіти і освіти дорослих імені Івана Зязюна, 2021. – 127 с. – Режим доступу: [https://lib.iitta.gov.ua/id/eprint/726901/1/my\\_mono\\_v1.pdf](https://lib.iitta.gov.ua/id/eprint/726901/1/my_mono_v1.pdf)
  21. Системи інтерпретації даних [Електронний ресурс] // Підручники. – Режим доступу: [https://pidru4niki.com/1469092447773/informatika/klasifikatsiya\\_ekspertnih\\_sistem](https://pidru4niki.com/1469092447773/informatika/klasifikatsiya_ekspertnih_sistem)
  22. Бабак В. П., Берегун В. С., Бурова З. А., Воробйов Л. Й., Декуша Л. В., Декуша О. Л., Запорожець А. О., Ковтун С. І., Красильников О. І., Назаренко О. О., Полобюк Т. А. Апаратно-програмне забезпечення моніторингу об'єктів генерування, транспортування та споживання теплової енергії. – 2022. – 20 с.
  23. Антонов Ю., Смоктій К. Використання експертних систем для аналізу відповідей у автоматизованих системах контролю знань // Вісник Хмельницького національного університету. С. 142–147. URL: <https://doi.org/10.31891/2307-5732-2024-339-4-51>
  24. Рекомендаційна система [Електронний ресурс] // Wikipedia. – Режим доступу: [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](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)

- 25.Негре, Ельза. Інформаційні та рекомендаційні системи. 2020. С. 2–5. – Режим доступу: [https://www.google.com.ua/books/edition/Information\\_and\\_Recommender\\_Systems/dGezCgAAQBAJ?hl=en&gbpv=1](https://www.google.com.ua/books/edition/Information_and_Recommender_Systems/dGezCgAAQBAJ?hl=en&gbpv=1)
- 26.Фільтрація на основі контенту [Електронний ресурс] // Upwork. – Режим доступу: <https://www.upwork.com/resources/what-is-content-based-filtering>
- 27.Колаборативна фільтрація [Електронний ресурс] // Medium. – Режим доступу: <https://medium.com/@evelyn.eve.9512/collaborative-filtering-in-recommender-system-an-overview-38dfa8462b61>
- 28.Колаборативна фільтрація користувач-користувач [Електронний ресурс] // Google Books. – Режим доступу: [https://www.google.com.ua/books/edition/Collaborative\\_Filtering\\_Recommender\\_System/vVwu5B\\_1q-IC?hl=en&gbpv=0](https://www.google.com.ua/books/edition/Collaborative_Filtering_Recommender_System/vVwu5B_1q-IC?hl=en&gbpv=0)
- 29.Колаборативна фільтрація предмет-предмет [Електронний ресурс] // GeeksforGeeks. – Режим доступу: <https://www.geeksforgeeks.org/item-to-item-based-collaborative-filtering/>
- 30.API Design Patterns. 2021. С. 30–31.
- 31.DOM-дерево [Електронний ресурс] // JavaScript.info. – Режим доступу: <https://uk.javascript.info/dom-nodes>
- 32.Rascia, Tania. Understanding the DOM — Document Object Model. 2020. С. 20-22.
- 33.Client-side-rendering [Електронний ресурс] // Web.dev. – Режим доступу: <https://web.dev/articles/rendering-on-the-web#:~:text=Client%2Dside%20rendering%20means%20rendering,its%20own%20set%20of%20tradeoffs.>
- 34.React [Електронний ресурс] // React.dev. – Режим доступу: <https://react.dev/>
- 35.Boduch, Adam. React and React Native. 2017. С. 34–45.
- 36.Thakkar, Mohit. Building React Apps with Server-Side Rendering: Use React, Redux, and Next to Build Full Server-Side Rendering Applications. 2020. С. 57.
- 37.Vue [Електронний ресурс] // Vue.js. – Режим доступу: <https://vuejs.org/>
- 38.Shavin, Maya. Learning Vue: Core Concepts and Practical Patterns for Reusable, Composable, and Scalable User Interfaces. 2023. С. 50–53.
- 39.Angular [Електронний ресурс] // Angular.dev. – Режим доступу: <https://angular.dev/>
- 40.Gechev, Minko. Getting Started with Angular. 2017. С. 8.
- 41.SEO [Електронний ресурс] // Search Engine Land. – Режим доступу: <https://searchengineland.com/guide/what-is-seo#:~:text=SEO%20stands%20for%20%E2%80%9Csearch%20engine,Services%20you%20provide.>
- 42.Cybellium Ltd. Mastering SEO. 2020. С. 152.
- 43.Гьокхан Гюрлесн. HTML 5.0. 2022. С. 1.

- 44.Сучасний підручник з JavaScript [Електронний ресурс] // JavaScript.info. – Режим доступу: <https://uk.javascript.info/>
- 45.Що таке SPA у програмуванні? [Електронний ресурс]. – Режим доступу: <https://foxminded.ua/spa-u-prohramuvanni/>
- 46.Померанц, Джеффри. Metadata. 2015.
- 47.Server-side-rendering [Електронний ресурс] // EPAM Solutions Hub. – Режим доступу: <https://solutionshub.epam.com/blog/post/what-is-server-side-rendering>
- 48.Geers, Michael. Micro Frontends in Action. 2020. С. 68.
- 49.Що таке UI та UX дизайн? [Електронний ресурс] // ITSTEP. – Режим доступу: <https://te.itstep.org/blog/ui-and-ux-design>
- 50.Розробка з боку Front-end? [Електронний ресурс] // Dan IT. – Режим доступу: <https://dan-it.com.ua/uk/blog/rozrobka-z-boku-front-end-shho-ce-take-i-chim-vidriznjaietsja-vid-back-end/>
- 51.JSX [Електронний ресурс] // React.dev. – Режим доступу: <https://react.dev/learn/writing-markup-with-jsx>
- 52.React router dom [Електронний ресурс] // React Router. – Режим доступу: <https://reactrouter.com/en/main>
- 53.React hook form [Електронний ресурс] // React Hook Form. – Режим доступу: <https://react-hook-form.com/>
- 54.Tailwind CSS [Електронний ресурс] // Tailwind CSS. – Режим доступу: <https://tailwindcss.com/>
- 55.Node.js Web Development: Server-side Web Development Made Easy with Node 14 Using Practical Examples [Електронне джерело]. 2024. 147 с. – Режим доступу: [https://www.google.com.ua/books/edition/Node\\_js\\_Web\\_Development/4\\_30DwAAQBAJ?hl=en&gbpv=1](https://www.google.com.ua/books/edition/Node_js_Web_Development/4_30DwAAQBAJ?hl=en&gbpv=1)
- 56.HTTр-запит (HTTр Request) [Електронне джерело] // База знань QALight. – Режим доступу: <https://qalight.ua/baza-znaniy/http-zapyt-http-request/>
- 57.Mongodb Complete Guide: Develop Strong Understanding of Administering Mongodb, CRUD Operations, Mongodb Commands / М. Шарма. – 2021. – 21 с.
- 58.Як використовувати JSON Web Tokens (JWT) для автентифікації [Електронне джерело] // DevZone. – Режим доступу: <https://devzone.org.ua/post/iak-vykorystovuvaty-json-web-tokens-jwt-dlia-avtentyfikatsiyi>
- 59.JSON [Електронний ресурс] // MDN Web Docs. – Режим доступу: <https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/JSON>
- 60.Векслер, Дж. Get Programming with Node.js. 2023. С. 141.
- 61.Варгезе, III. Web Development with Go: Building Scalable Web Apps and RESTful Services. 2022. С.89.

62.Що таке файли cookie? [Електронне джерело] // Cloudflare. – Режим доступу: <https://www.cloudflare.com/learning/privacy/what-are-cookies/>



## ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

*Повністю вказується ПІБ та статус (освітня (освітньо-наукова) програма – для здобувачів вищої освіти, назва кваліфікаційної роботи)*

що нижче підписалась/підписався, розуміючи та підтримуючи загальновизнані засади справедливості, доброчесності та законності,

### **ЗОБОВ'ЯЗУЮСЬ:**

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

### **ПІДТВЕРДЖУЮ:**

що мені відомі положення статті 42 Закону України «Про освіту»;

що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

### **УСВІДОМЛЮЮ:**

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

\_\_\_\_\_  
(дата)

\_\_\_\_\_  
(підпис)