Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Програмна система для рекомендації та підбору актуальних культурних подій. Front-end, AI.
(тема)

Виконав:
студент 4 курсу, групи ПЗПІ-20-3

Сафошин В.В.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія
(повна назва освітньої програми)

Керівник доц. кафедри ПІ Побіженко І.О.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти перший (бакалаврський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми Освітньо-професійна
Освітня програма Програмна Інженерія
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Сафошину Володимиру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Програмна система для рекомендації та підбору актуальних культурних подій. Front-end, AI.
Затверджена наказом по університету від 20.5.2024р. № 471 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 17.06.2024
3. Вихідні дані до роботи Розробити клієнтську частину з використанням фреймворку React.js та ШІ чат-бота з використанням Python та фреймворку LangChain для програмної системи для рекомендації та підбору актуальних культурних подій.
4. Перелік питань, що потрібно опрацювати в роботі
Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	8.04.2024	*виконано*
2	Створення специфікації ПЗ	12.04.2024	*виконано*
3	Проектування ПЗ	15.04.2024	*виконано*
4	Розробка ПЗ	10.05.2024	*виконано*
5	Тестування ПЗ	30.05.2024	*виконано*
6	Оформлення пояснювальної записки	01.06.2024	*виконано*
7	Підготовка презентації та доповіді	11.06.2024	*виконано*
8	Попередній захист	15.06.2024	*виконано*
9	Нормоконтроль, рецензування	12.06.2024	*виконано*
10	Здача роботи у електронний архів	17.06.2024	*виконано*
11	Допуск до захисту у зав. кафедри	20.06.2024	*виконано*

Дата видачі завдання 8 квітня 2024р.

Студент (ка)_____ Сафошин В.В.
(підпис)

Керівник роботи _____ доц. кафедри ПІ Побіженко І.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра: 74 с., 19 рис., 11 джерел.

КЛІЄНТСЬКА ЧАСТИНА, ПРОГРАМНА СИСТЕМА, СИСТЕМА ДЛЯ РЕКОМЕНДАЦІЇ КУЛЬТУРНИХ ПОДІЙ, ШТУЧНЙИ ІНТЕЛЕКТ, CHATGPT, LANGCHAIN, REACT.JS

Об’єкт розробки – клієнтська частина та ШІ чат-бот для програмної системи для рекомендації та підбору актуальних культурних подій.

Мета розробки – розробка клієнтської частини та ШІ-частини програмної системи для рекомендації та підбору актуальних культурних подій.

Метод рішення – для розробки клієнтської та ШІ-частини було обране середовище розробки Visual Studio Code, мова програмування JavaScript та фреймворк React.js для клієнтської частини та Python та фреймворк LangChain для ШІ чат-боту.

У результаті розробки створено клієнтську частину програмної системи для рекомендації та підбору актуальних культурних подій.

FRONTEND, SOFTWARE SYSTEM, SYSTEM FOR RECOMMENDING CULTURAL EVENTS, ARTIFICIAL INTELLIGENCE, CHATGPT, LANGCHAIN, REACT.JS

The object of development is the client part and AI chatbot for the software system for recommending and selecting relevant cultural events.

The purpose of the development is to develop a frontend of software system for recommending and selecting relevant cultural events.

The solution method is the Visual Studio Code development environment, the JavaScript programming language, and the React.js framework for a frontend, and Python with use of LangChain for an AI chat bot.

As a result of the development, the client part of the software system was created for recommending and selecting relevant cultural events.

Я, Сафошин Володимир Володимирович, студент гр. ПЗПІ-20-3, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програмна система для рекомендації та підбору актуальних культурних подій. Front-end частина, AI частина.», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Розвиток цифрових технологій відіграє ключову роль у формуванні новітніх способів взаємодії з культурою та дозвіллям. У контексті глобалізації та зростаючої мобільності населення, потреба у персоналізованій та інтелектуальній системі рекомендацій культурних заходів стає дедалі важливішою. “Eventify” – це програмний продукт, який вирішує актуальну проблему вибору культурних заходів, адаптованих до індивідуальних уподобань та геолокації користувачів.

Зі збільшенням кількості культурних подій, таких як концерти, виставки, ярмарки та фестивалі, виникає потреба в ефективній системі, яка б не просто інформує про заходи, але й допомагає користувачам знайти найбільш релевантні події, засновані на їхніх минулих вподобаннях та актуальному розташуванні. “Eventify” використовує передові алгоритми машинного навчання для надання персоналізованих рекомендацій, що робить процес вибору заходів зручнішим та інтуїтивно зрозумілим.

Система “Eventify” запропонована як відповідь на зростаючі очікування сучасної аудиторії, яка цінує якість та персоналізацію у своєму культурному досвіді. Завдяки використанню інноваційних технологічних рішень, сервіс здатний забезпечити оперативне інформування про заходи, їх фільтрацію та пошук, а також сприяє зручному бронюванню квитків та взаємодії з ШІ-асистентом для вибору подій.

Основними завданнями “Eventify” є забезпечення високої точності та відповідності рекомендацій, ефективне управління користувацькими даними, а також відповідність найсучаснішим стандартам безпеки та приватності. Розробка цієї системи відіграє ключову роль у забезпеченні культурної інтеграції та розвитку, враховуючи індивідуальні потреби користувача в швидко змінюваному світі.

Таким чином, “Eventify” є продуктом, який сприяє більшій доступності та насолоді від культурних заходів, роблячи цей процес максимально особистісним та задовільним.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Предметна галузь програмної системи “Eventify” охоплює індустрію розваг та культурних заходів, яка включає організацію та пошук різноманітних подій, таких як концерти, виставки, ярмарки та фестивалі. Культурні заходи служать важливим засобом соціальної інтеракції та особистісного розвитку, забезпечуючи платформу для вираження культурної ідентичності та сприяння міжкультурному обміну. Ця галузь характеризується широкою географією розподілених заходів, великою кількістю учасників, а також різноманіттям інтересів аудиторії.

Учасники культурних заходів представляють людей різних вікових груп та соціальних верств, які прагнуть знайти відпочинок, відповідний їхнім інтересам та географічному розташуванню. Вони стикаються з проблемою великої кількості доступних заходів і перевантаженням інформацією, що ускладнює процес вибору потрібної події. Така ситуація створює зростаючу потребу у розробці ефективного інструменту, який би допомагав у фільтрації, рекомендації та персоналізації культурних заходів для користувачів.

Сфера культурних заходів включає широкий спектр діяльності від планування та організації до маркетингу та продажу квитків. Організатори подій і куратори культурних просторів постійно шукають способи залучення аудиторії, вдосконалення взаємодії з відвідувачами та покращення досвіду учасників.

У цій галузі, потреба в персоналізації досвіду стає все більш вираженою, адже сучасні споживачі бажають відвідувати заходи, що максимально відповідають їхнім інтересам та вподобанням. Використання технологій для аналізу даних про попередні відвідування, уподобання та геолокацію користувачів дозволяє створити рекомендаційні системи, які забезпечують високу релевантність та персоналізацію інформації про заходи.

Технологічний розвиток, зокрема в областях мобільних застосунків, соціальних мереж, та інтерактивних платформ, відкриває нові можливості для креативного підходу до організації та просування культурних заходів. “Eventify”

інтегрує ці технологічні новинки для створення ефективної, інтуїтивно зрозумілої та доступної системи підбору заходів.

На ринку присутні різноманітні сервіси для бронювання та купівлі квитків, проте лише невелика кількість з них пропонує глибоку персоналізацію та інтеграцію з інтерактивними картами та ШІ-помічниками для поліпшення користувацького досвіду.

Для аналізу було обрано афішу concert.ua. На цьому сайті присутня можливість для перегляду подій, замовлення квитків для концертів, театру, фестивалів але неможливо подивитись усі ці заходи на мапі, а також немає ніякого інструменту для інтерактивних рекомендацій (див. рис 1.1).

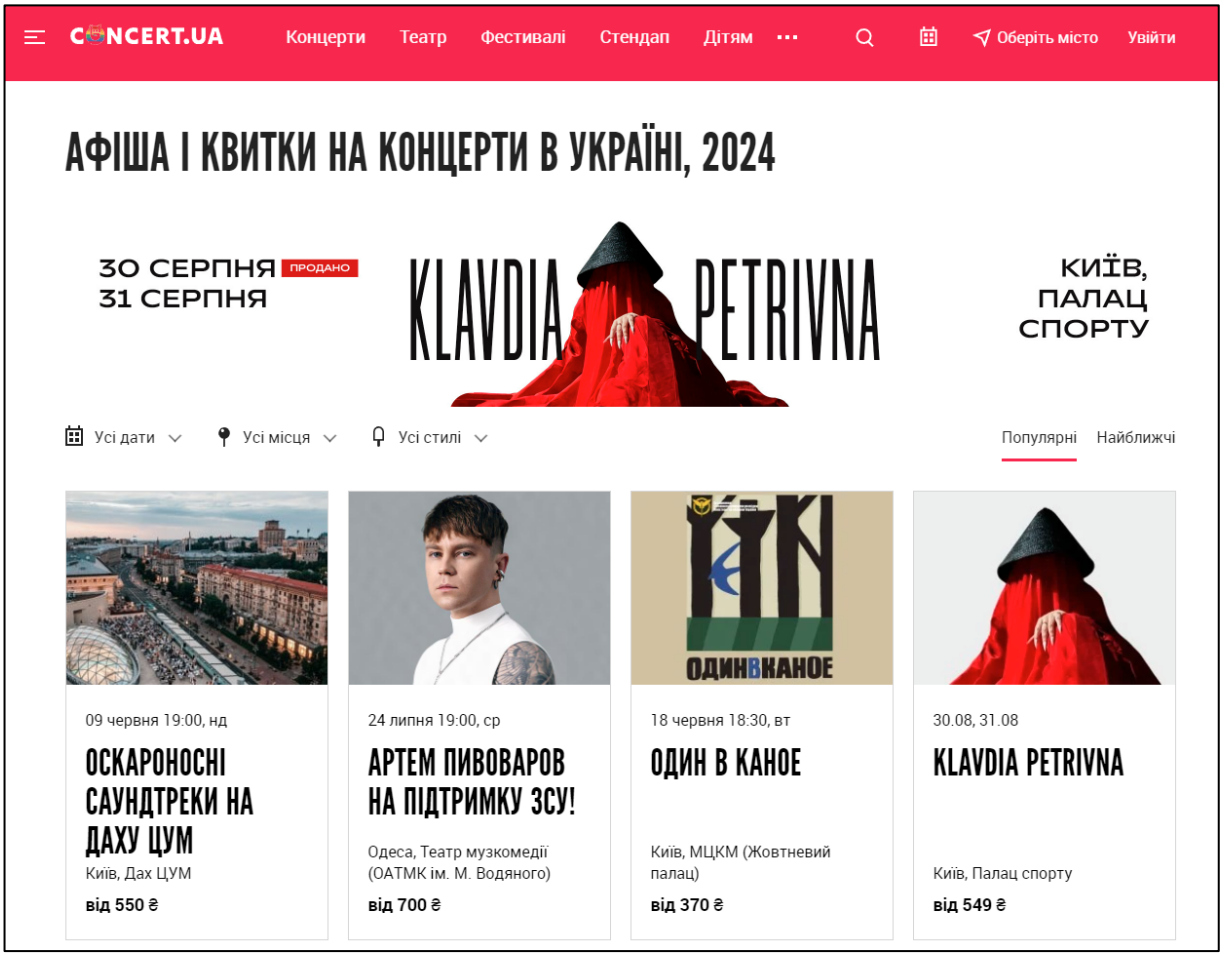


Рисунок 1.1 – Афіша з квитками на концерти Concert.ua (за даними [1])

Другою системою-аналогом є афіша подій KARABAS. На сайті Karabas.com користувачі можуть переглядати афіші майбутніх заходів, включаючи концерти, театральні вистави, клубні вечірки та фестивалі. Відвідувачі мають можливість

замовити квитки на ці події. Проте також як і в першому випадку відсутня функція перегляду усіх заходів на карті або інтерактивного інструменту для рекомендацій.

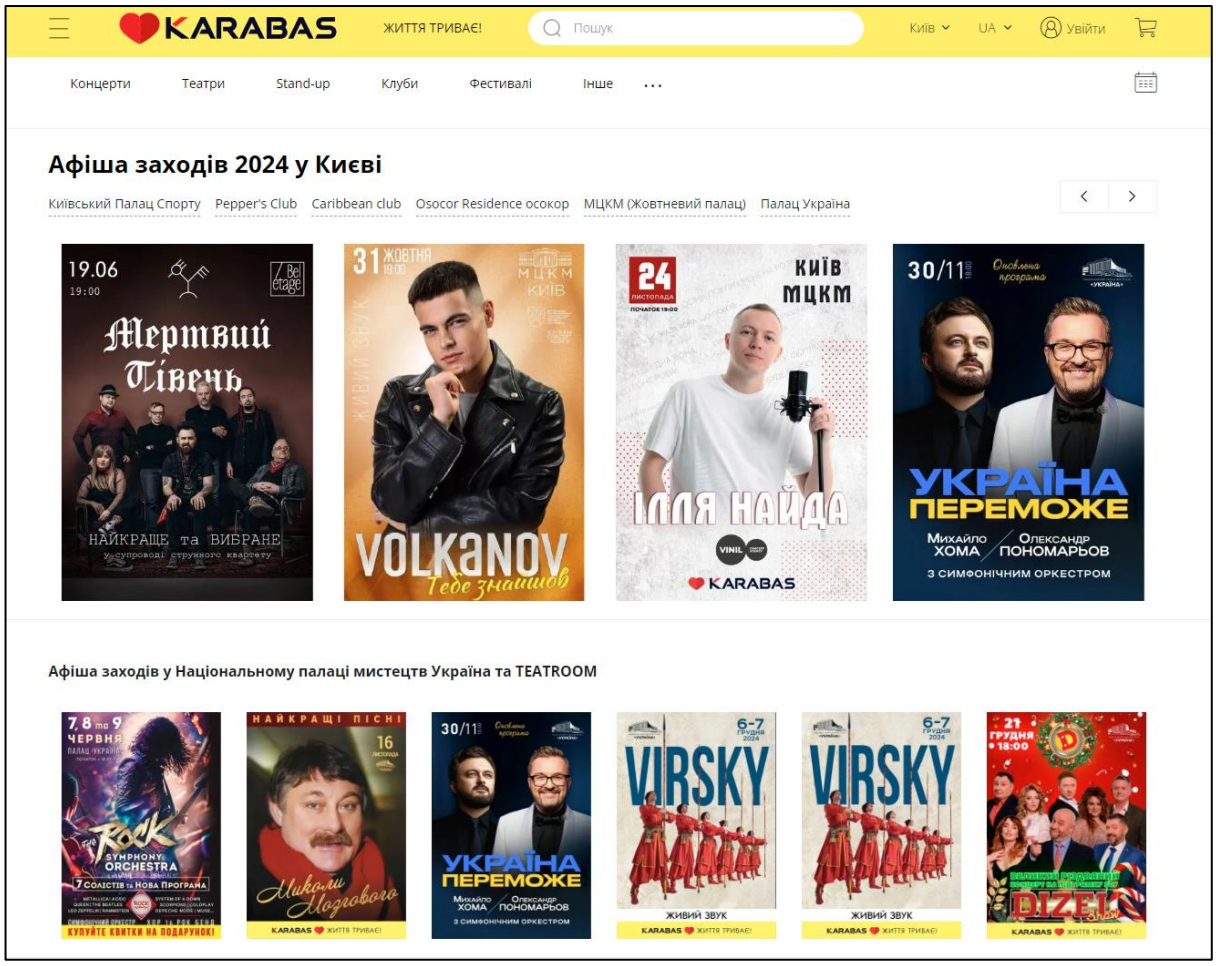


Рисунок 1.2 – Афіша подій karabas.com (за даними [2])

Загалом, індустрія культурних заходів вимагає сучасних цифрових рішень, які б допомагали користувачам орієнтуватися у широкому асортименті заходів, підбираючи найбільш релевантні заходи на основі особистих уподобань та попереднього досвіду. Ця система відповідає на зростаючий попит на індивідуалізований культурний досвід, допомагаючи користувачам знаходити заходи, які найбільше відповідають їхнім вимогам, та підвищуючи загальну задоволеність і залученість аудиторії. Серед основних функцій програмної системи можна виділити наступні:

– замовлення квитків – необхідно зробити інтеграцію з платформами представників заходів для зручного бронювання квитків та запрошень на різноманітні заходи безпосередньо через інтерфейс системи, для зручного

отримання інформації про квитки;

- перегляд подій на карті – необхідно зробити можливість перегляду подій на інтерактивній карті місцевості, що полегшує вибір заходів залежно від їх місцезнаходження;
- спілкування з ШІ-асистентом – необхідно зробити інтеграцію штучного інтелекту для надання допомоги у виборі заходів, відповідей на запитання та забезпечення додаткової інформації про заходи.

1.2 Виявлення проблем та актуалізація рішень

У системі “Eventify”, призначеній для рекомендації культурних заходів, можуть бути реалізовані функціональності для виявлення проблем у сфері вибору культурних подій та актуалізація рішень. Ось декілька прикладів таких функціональностей:

- персоналізація досвіду користувача – система може збирати дані про минулі відвідування та переваги користувачів, щоб точніше підбирати рекомендації, забезпечуючи більш особистісний досвід;
- аналіз та фільтрація інформаційного шуму – велика кількість подій створює інформаційний шум, який може бути складно обробити користувачам. Система може використовувати розширені алгоритми фільтрації для відсіву нерелевантних подій;
- інтеграція з інтерактивними картами для локалізації заходів – користувачі можуть мати складнощі з локалізацією подій, тому інтеграція з картами допоможе їм легше планувати свій час та маршрути;
- виявлення та реагування на зміни уподобань користувачів – система може аналізувати зміни в поведінці користувачів і автоматично адаптуватися, пропонуючи нові види заходів згідно з оновленими інтересами;
- підтримка в реальному часі через ШІ-асистента – ШІ-асистент може надавати користувачам оперативну підтримку, відповідаючи на запитання та рекомендуючи заходи на основі актуальних запитів та локації.

Ці функції спрямовані на вирішення конкретних проблем, які виникають у користувачів при виборі культурних заходів, та допомагають актуалізувати рішення для забезпечення кращого користувацького досвіду.

Ось пропозиції щодо кроків для актуалізації рішень у системі “Eventify” для покращення взаємодії користувачів із сервісом та забезпечення більш ефективного вибору культурних заходів:

- збір та аналіз даних користувача – імплементація інструментів для збору даних про попередні пошуки, відвідування та відгуки користувачів для створення детальних профілів уподобань;
- розвиток персоналізованих рекомендацій – використання машинного навчання для аналізу зібраних даних та автоматизація процесу рекомендацій, що базуються на індивідуальних інтересах і місцеположенні користувачів;
- оптимізація інтерфейсу користувача – розробка інтуїтивно зрозумілого користувацького інтерфейсу з легким доступом до функцій фільтрації, пошуку та перегляду заходів на карті;
- інтеграція з інтерактивними картами – забезпечення візуалізації подій на мапі для підвищення зручності планування відвідувань, включаючи маршрутизацію та інформацію про доступність;
- розробка системи сповіщень – впровадження системи сповіщень, яка інформує користувачів про нові заходи або зміни в уже планованих подіях, засновані на їхніх уподобаннях;
- застосування ШІ-асистента для підтримки у реальному часі – інтеграція розмовного ШІ-асистента, який може надавати миттєві відповіді на запитання користувачів, допомагаючи у виборі заходів або розв'язанні проблем.

1.3 Постановка задачі

Задача даної роботи полягає у розробці програмної системи “Eventify”, яка призначена для рекомендації та персоналізації культурних заходів. Система має забезпечувати ефективне збирання, зберігання, обробку, та аналіз даних, що відносяться до інтересів і поведінки користувачів у контексті відвідування заходів.

Основні задачі роботи:

- система повинна автоматично збирати дані про уподобання користувачів, їх геолокації та історію відвідувань заходів;
- необхідно розробити надійну базу даних для ефективного збереження великого обсягу інформації про користувачів та заходи;
- система повинна обробляти зібрані дані для визначення відповідності заходів уподобанням користувачів;
- необхідно розробити алгоритм для персоналізації рекомендацій, що враховують попередні дані та поточні запити користувачів;
- необхідно впровадження функціоналу для візуалізації заходів на інтерактивній мапі, забезпечуючи користувачам легке планування відвідувань;
- необхідно інтегрувати ШІ-асистента, який може в реальному часі консультувати користувачів, відповідати на їх запитання та сприяти вибору заходів;
- необхідно імплементувати заходи захисту даних для запобігання несанкціонованому доступу та забезпечення конфіденційності інформації.

Мета роботи полягає у створенні ефективного та надійного інструменту, що дозволить користувачам знаходити культурні заходи, які відповідають їхнім інтересам та потребам, тим самим підвищуючи загальну задоволеність та взаємодію з культурним контентом.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Мова програмування та технології:

- реалізація клієнтської частини нашого додатку має бути здійснена за допомогою фреймворку React.js [3]. Цей популярний JavaScript фреймворк дозволяє створювати динамічні та інтерактивні користувацькі інтерфейси з використанням компонентного підходу. Компоненти можуть бути повторно використані, що сприяє модульності та спрощує підтримку коду. React.js надає ефективний механізм рендерингу, який підвищує продуктивність додатку, зменшуючи кількість операцій.
- для управління станом додатку ми будемо використовувати React Redux. Це бібліотека управління станом, яка допомагає централізовано зберігати і керувати станом усіх компонентів додатку. Redux дозволяє чітко визначити, як і де зберігаються дані, спрощує їх оновлення і забезпечує передбачуваність поведінки додатку. Використання одно направленого потоку даних в Redux полегшує налагодження та масштабування додатків;
- використання Bootstrap та CSS для розробки адаптивного веб-дизайну та інтерфейсу користувача.

Взаємодія з серверною частиною:

- реалізація клієнтського API для забезпечення комунікації між клієнтською частиною та сервером;
- використання бібліотек Axios для виконання HTTP запитів до сервера, що дозволяє отримувати, відправляти та обробляти дані з сервера у форматах JSON;
- реалізація обробників запитів, які взаємодіють з серверною частиною через REST, а також обробка відповідей сервера для оновлення UI відповідно до отриманих даних.

Інтеграція чат-бота:

- використання моделі ChatGPT для обробки природної мови, забезпечення зрозумілої та ефективної взаємодії з користувачами;

– інтеграція LangChain[4] для підключення чат-бота до різних джерел даних та сервісів, що дозволяє збирати актуальну інформацію про культурні заходи;
– реалізація функцій пошуку за різними параметрами через інтерфейс чату.

Функціональність чат-бота:

– надання персоналізованих рекомендацій заснованих на інтересах та попередній історії взаємодії користувача;
– відповіді на запитання про деталі заходів, такі як час початку, тривалість, вікна обмеження, вартість квитків тощо;
– підтримка взаємодії в реальному часі, забезпечуючи швидке та точне обслуговування запитів користувачів.

Безпека та конфіденційність:

– забезпечення безпеки даних користувачів та виконання регулятивних вимог щодо обробки особистої інформації;
– імплементація безпечних методів авторизації для доступу до функціональності чат-бота, що дозволяє забезпечити конфіденційність інформації.

Користувацький інтерфейс:

– розробка реактивного та інтуїтивно зрозумілого інтерфейсу, який адаптується під різні пристрої;
– імплементація функціональності пошуку, фільтрації та сортування культурних заходів з використанням компонентів UI бібліотеки.

Обробка даних та інтерактивність:

– використання React Hooks для управління станом компонентів і забезпечення гнучкої взаємодії користувача з додатком;
– реалізація сторінок з детальною інформацією про культурні заходи, включаючи описи, фотографії та інтерактивні елементи для бронювання квитків.

Розробка нашого програмного забезпечення потребує використання Visual Studio Code (VS Code), потужного редактора коду від Microsoft. Ця середовище розробки підтримує численні мови програмування і фреймворки. Завдяки модульній структурі з додатковими плагінами та розширеннями, VS Code є гнучким інструментом, який можна легко налаштувати під потреби розробника. Окрім цього, він забезпечує зручне інтегроване управління Git.

Щоб ефективно співпрацювати між розробниками та керувати версіями коду нашого програмного забезпечення, ми будемо використовувати GitHub. Ця платформа для контролю версій дозволяє нам відслідковувати зміни в коді, вирішувати конфлікти, а також розробляти нові функції в окремих гілках. GitHub надає інструменти для рецензій коду, що покращує його якість. Завдяки GitHub Actions ми можемо автоматизувати рутинні процеси, такі як тестування і розгортання, що значно підвищує продуктивність нашої команди. Крім того, ми використовуємо Azure як основне середовище для розгортання наших додатків. Azure надає широкий набір хмарних сервісів, які дозволяють нам гнучко масштабувати наші рішення відповідно до потреб.

3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 UML проектування ПЗ

Діаграма варіантів використання (Use Case Diagram) є ключовою UML діаграмою, що використовується для моделювання функціональних вимог системи з точки зору зовнішніх акторів. Вона дозволяє визначити основні взаємодії між акторами та системою, виокремлюючи можливості системи, які повинні бути реалізовані.

У контексті системи “Eventify”, діаграми варіантів використання допомагають уточнити потреби користувачів та описати функціональність системи. Розглянемо кілька діаграм, які показують взаємодію різних типів користувачів з системою (див. рис. 3.1 – 3.3).

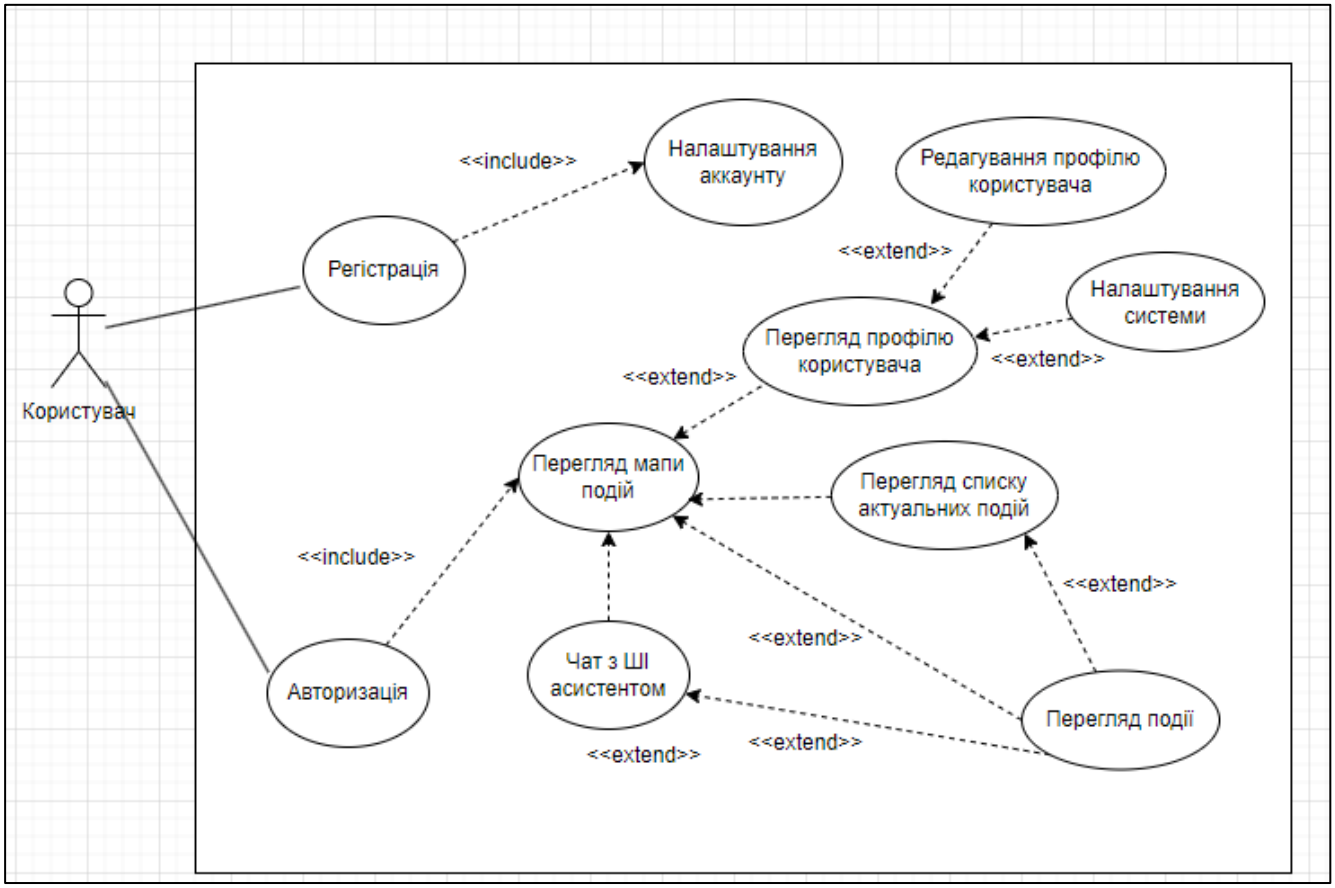


Рисунок 3.1 – Use-Case діаграма для Користувача (рисунок виконано самостійно)

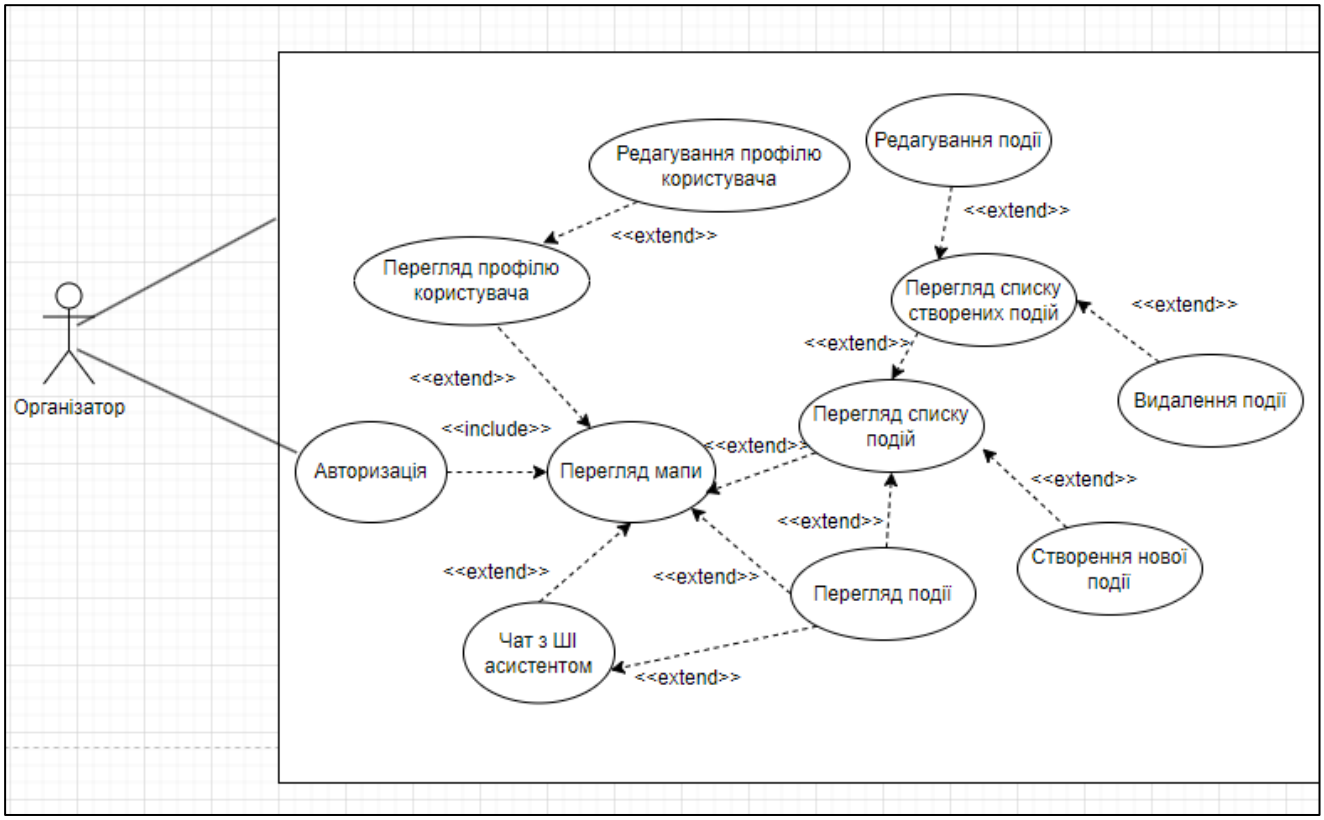


Рисунок 3.2 – Use-Case діаграма для Організатора (рисунок виконано самостійно)

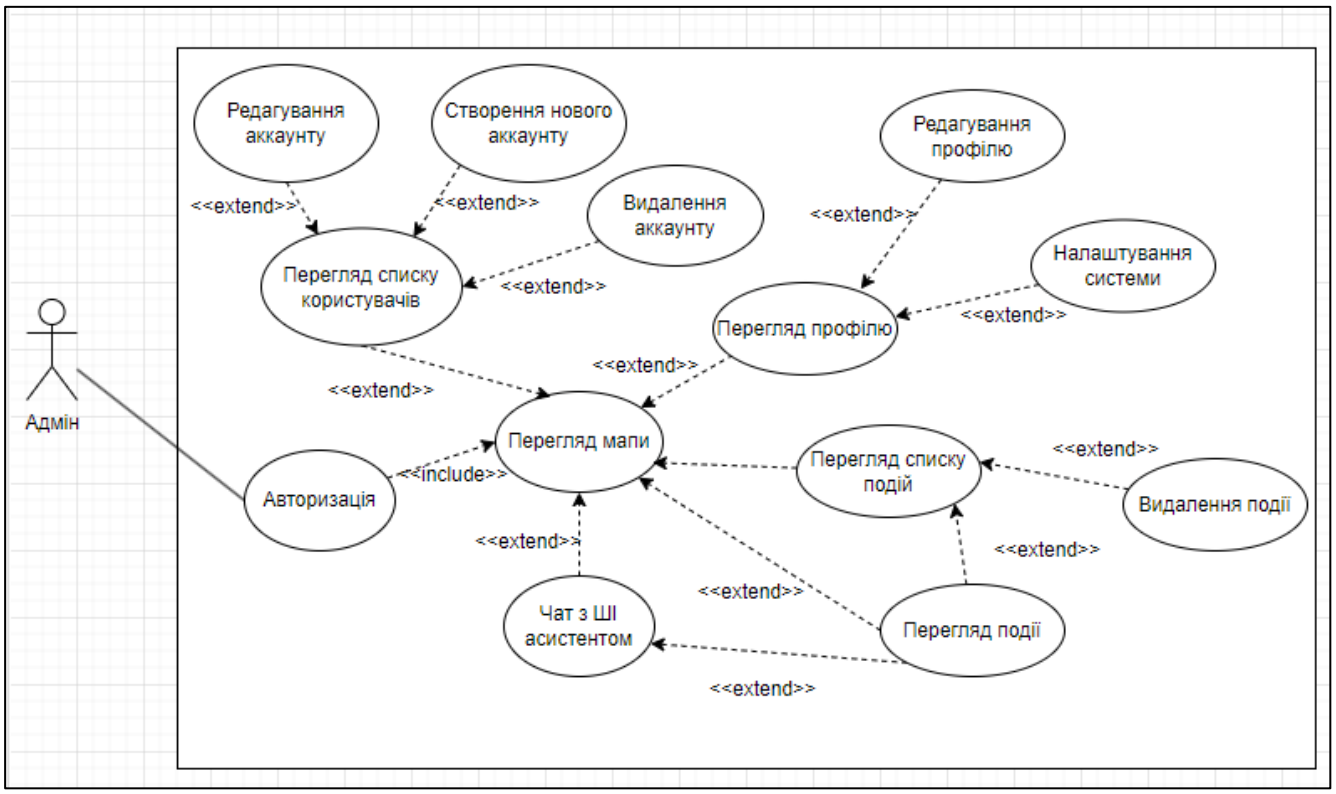


Рисунок 3.3 – Use-Case діаграма для Адміністратора (рисунок виконано самостійно)

Розглянемо акторів програмної системи “Eventify”:

- користувачі – це звичайні відвідувачі, які шукають інформацію про події та взаємодіють з різними функціями системи для планування свого дозвілля. Вони можуть реєструватися та авторизуватися, створюючи особисті облікові записи; переглядати мапу подій для легшого вибору заходів і навігації; спілкуватися з ШІ асистентом для отримання допомоги чи порад; переглядати та редагувати свій профіль; а також мати доступ до списку подій, які можна фільтрувати за власними критеріями (див. рис. 3.1).
- організатори – це представники, які організовують заходи та використовують систему для управління подіями. Вони можуть створювати нові події, додаючи їх до системи, а також редагувати вже існуючі події, вносячи зміни до інформації, яку вони самі опублікували (див. рис. 3.2).
- адміністратори – це особи, які керують всією системою, включаючи управління користувачами, забезпечення стабільності роботи платформи та контентом. Вони можуть додавати, видаляти і модифікувати користувацькі облікові записи, а також розглядати й затверджувати події, створені організаторами (див. рис. 3.3).

3.2 Проектування архітектури ПЗ

Клієнт-серверна архітектура є основою для розподілення функцій між клієнтською (фронтенд та мобільна частини) та серверною (бекенд) частинами системи “Eventify”. Взаємодія між цими складовими відбувається через мережеві протоколи, забезпечуючи користувачам можливість взаємодії з веб-інтерфейсом та мобільними застосунками. До архітектури також входить інтегрований ШІ компонент, що підтримує розуміння природної мови та персоналізоване обслуговування користувачів, вносячи в систему функції розширеного аналізу та адаптивних рекомендацій.

Клієнтська сторона:

- фронтенд – розроблений на React.js, фронтенд забезпечує інтерактивний користувацький інтерфейс, приймає користувацькі введення та відображає дані отримані через API;
- мобільний додаток – розроблений з використанням Swift[5] для iOS, забезпечує адаптований інтерфейс для мобільних користувачів і теж взаємодіє з сервером через API;
- взаємодія з API – клієнтська сторона отримує та відправляє дані до сервера через RESTful API, що забезпечує гнучкість у взаємодії з бекендом.

Серверна сторона (бекенд):

- бекенд – виконаний на ASP.NET[6] із використанням C#[7] як основної мови програмування, бекенд керує бізнес-логікою, обробкою даних, взаємодією з базою даних, і надає дані фронтенду через API;
- база даних – використовується Microsoft SQL Server[8] (MS SQL) як реляційна СУБД для зберігання та управління даними. Вона забезпечує швидкий доступ та надійне зберігання інформації про користувачів, заходи, відгуки та інші системні дані.

ШІ Частина Системи:

а) технології – штучний інтелект у системі “Eventify” втілено за допомогою LangChain, яка розроблена на Python[9]. Ця технологія використовується для реалізації ШІ-асистента, здатного вести діалог з користувачами, надавати рекомендації щодо заходів, а також відповідати на запитання щодо деталей подій;

б) функціональні можливості ШІ:

1) персоналізовані рекомендації – асистент аналізує переваги та історію взаємодій користувачів для надання персоналізованих рекомендацій, заснованих на їхніх інтересах;
2) обробка природної мови – ШІ-асистент здатний розуміти та обробляти запитання та команди, задані природною мовою, завдяки можливостям LangChain;

3) взаємодія в реальному часі – асистент підтримує діалоги з користувачами в режимі реального часу, швидко реагуючи на запити та зміни у контексті розмови.

в) Інтеграція з іншими компонентами системи:

1) Взаємодія з бекендом – ШІ-асистент взаємодіє з серверною частиною через API для отримання та оновлення даних, необхідних для відповідей на запитання користувачів або надання рекомендацій;
2) Синхронізація з базою даних – всі інтеракції з користувачами, їхні переваги та відгуки аналізуються та зберігаються для вдосконалення алгоритмів ШІ.

Комунікація у системі :

– система використовує HTTP-протоколи які забезпечують стандартний метод комунікації між клієнтською і серверною частинами через інтернет. Також вони використовується для передачі даних між фронтендом і бекендом;
– система використовує RESTful API для обміну даними між фронтендом і бекендом. Це забезпечує простий та зрозумілий спосіб організації мережевих запитів, які включають створення, читання, оновлення та видалення даних. RESTful API дозволяє клієнтській частині відправляти запити до бекенду та отримувати від нього необхідні дані у форматі JSON, що сприяє швидкій та ефективній взаємодії.

В архітектурі програмного забезпечення “Eventify” ми використовуємо сучасні підходи та патерни для оптимізації взаємодії між сервером та клієнтами:

На бекенді ми використовуємо API-орієнтований підхід, що дозволяє забезпечити гнучкість та масштабованість в обробці запитів [10]. API служить як єдиний вхідний пункт для обробки бізнес-логіки та взаємодії з базою даних, гарантуючи таким чином єдність та цілісність системи. Це розмежування дозволяє ізолювати бізнес-логіку від користувацького інтерфейсу, що сприяє більшій безпеці та легшій підтримці коду.

Фронтенд “Eventify” розроблено як одно сторінковий застосунок (англ. single-page application, SPA) на платформі React.js, що включає в себе динамічне завантаження контенту без перезавантаження сторінки [11]. Це забезпечує високу швидкість відгуку і знижує час завантаження, що особливо важливо для підвищення задоволеності користувачів. SPA використовує JavaScript для динамічної взаємодії з користувачем, підвищуючи відчуття "живого" інтерфейсу та плавності переходів між різними частинами застосунку.

Використання SPA разом з API дозволяє ефективно розділяти фронтенд та бекенд функціональність, що сприяє легкості управління кодом та масштабуванню застосунку. Це також дозволяє більш легко інтегрувати нові функції та вдосконалення без впливу на існуючу інфраструктуру системи.

Розробимо діаграму розгортання системи “Eventify”, яка буде відображати структуру та взаємозв'язки між основними компонентами системи (див. рис. 3.4).

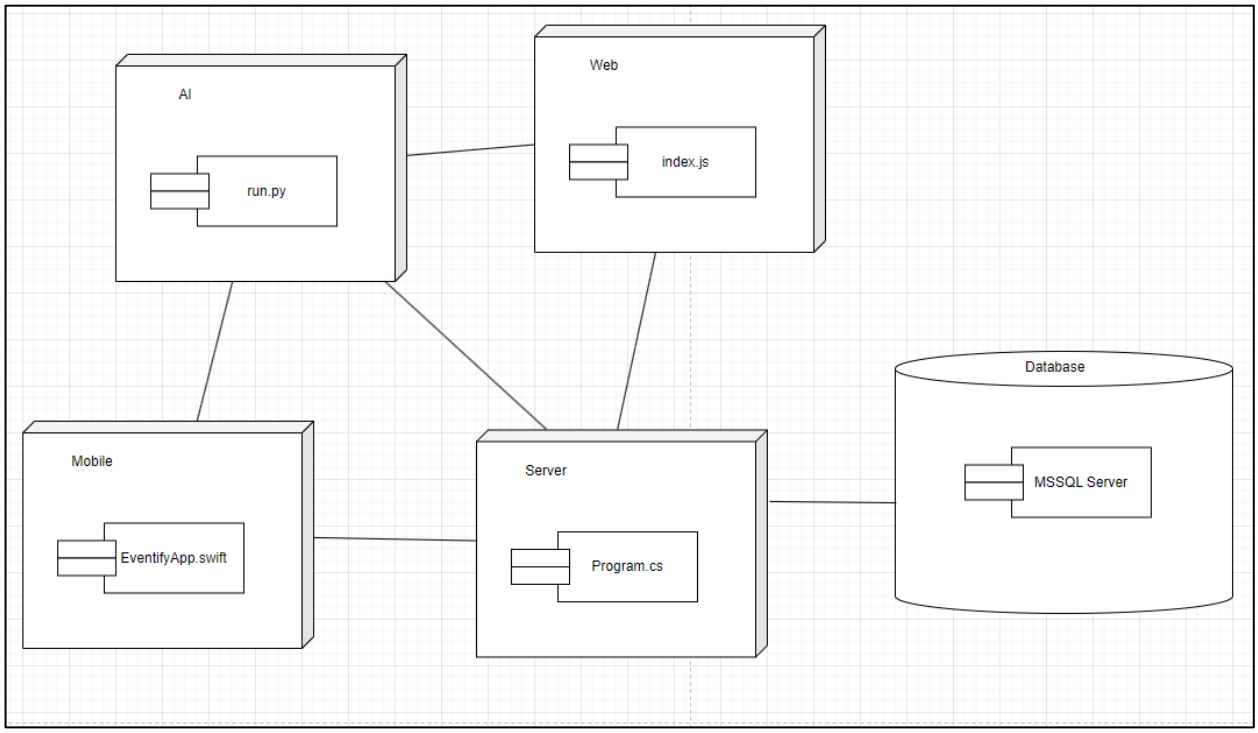


Рисунок 3.4 – Діаграма розгортання (рисунок виконано самостійно)

Головною метою представленої діаграми розгортання є демонстрація фізичної структури системи, яка охоплює апаратні та програмні засоби, мережеві

з'єднання та інші технічні ресурси. Вона надає зручний спосіб зрозуміти розташування та взаємодію компонентів системи між собою.

3.3 Проектування бази даних

Для створення бази даних системи рекомендацій подій важливо детально зрозуміти бізнес-процеси, які повинні бути відтворені у структурі даних. Вибір Microsoft SQL Server як основної системи управління базами даних відповідає потребам у сучасному, надійному і адаптивному рішенні, що підтримує високий рівень продуктивності та масштабування. MS SQL славиться своєю стабільністю, швидкістю обробки даних та здатністю інтегруватися з іншими продуктами Microsoft, що робить його ідеальним для систем на базі Windows. Завдяки потужним інструментам для обробки даних, включно з управлінням, аналізом та формуванням звітів, а також завдяки сильній підтримці та активній спільноті користувачів, MS SQL стає прекрасним вибором для проектів будь-якого розміру.

SQL Server Management Studio (SSMS) - це інтегроване середовище для управління будь-якими компонентами SQL Server, від баз даних до звітів і аналізу. Це програмне забезпечення, розроблене Microsoft, яке надає зручний інтерфейс для адміністрування і роботи з базами даних на платформі SQL Server. З його допомогою можна створювати та редагувати бази даних, виконувати SQL-запити, розгортати бази даних, керувати безпекою, налаштовувати резервне копіювання та відновлення даних, а також проводити моніторинг та оптимізацію продуктивності. SSMS дуже популярний серед адміністраторів баз даних і розробників, які працюють з продуктами SQL Server.

Таблиці бази даних, що використовуються сервером (див. рис. 3.5):

- таблиця «User» використовується для зберігання основної інформації про користувача в системі. Вона містить його особисті та облікові дані;
- таблиця «Event» зберігає інформацію основну інформація про подію, таку як назву час, хто створив;

– таблиця «Tag» містить інформацію про теги, що використовуються щоб відрізнити між собою події. Також надають можливість користувачам підписатися на появу нових події;
– таблиця «Location» зберігає інформацію про місцезнаходження події на карті;
– таблиця «ViewHistory» використовується для зберігання списку події, що зацікавили користувача, для більш ефективного підбору рекомендацій в майбутньому;
– таблиця «Settings» зберігає налаштування системи для користувача;
– таблиця «Roles» містить дані про ролі в системі. Вони допомагають виокремити доступи в залежності від користувача;
– таблиця «Tag-Event» виступає проміжною таблицею між «Tag» і «Event», так як в нас подія може відповідати багатьом тегам і навпаки один тег відноситися до багатьох подій;
– таблиця «User-Tag» є проміжною таблицею для «User» та «Tag» так як в нас один користувач може бути підписаний на декілька тегів і навпаки на один тег може бути підписано багато користувачів.

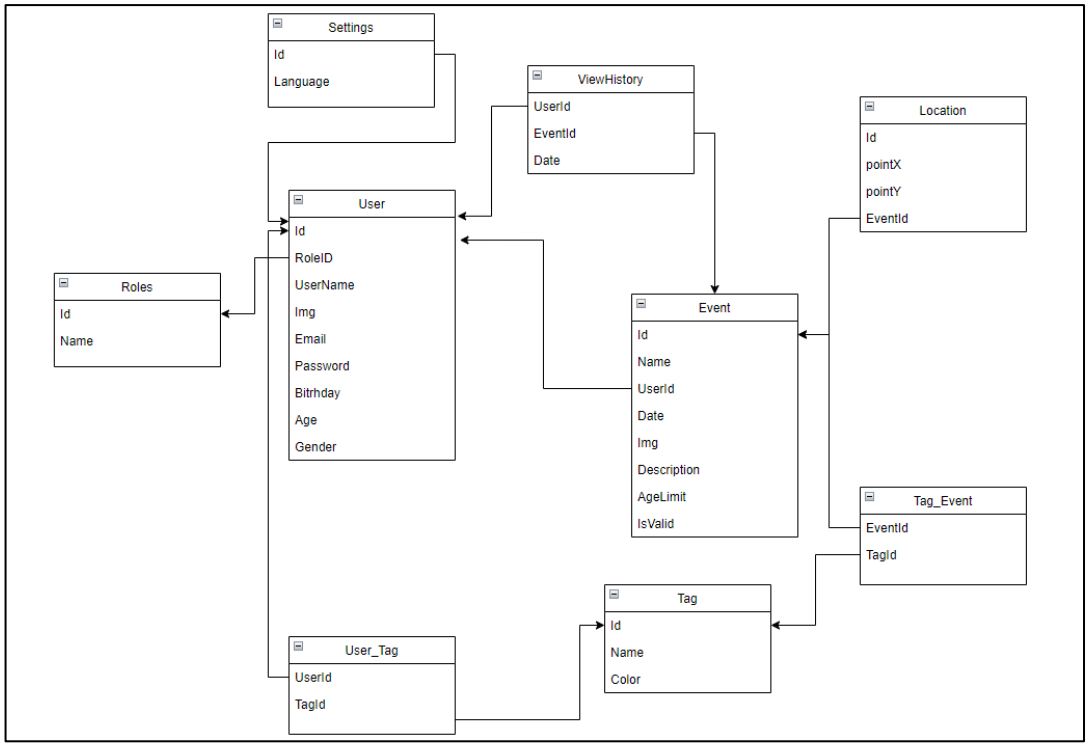


Рисунок 3.5 – ER діаграма бази даних (рисунок виконано самостійно)

База даних є в третій нормальній формі, що підтверджує її нормалізованість. Однією з переваг нормалізованої структури бази даних є відсутність повторюваної інформації та зручність у зберіганні та обробці даних. Це сприяє ефективному використанню обсягу пам'яті, особливо при роботі з великими обсягами даних.

3.4 Огляд алгоритмів та методів

На мою думку, одним з найцікавіших алгоритмів у системі “Eventify” є алгоритм рекомендації подій користувачам. Після отримання запиту від користувача, ключовим етапом є визначення та рекомендація події, яка найкраще відповідає його інтересам. Алгоритм детально описує процес вибору події:

- створення запиту на пошук події – це перший крок, де чат-бот аналізує вхідний запит користувача, розпізнаючи ключові слова, пов'язані з часом, типом події та локацією. Наприклад, якщо користувач хоче відвідати "рок концерт завтра недалеко від Києва", система ідентифікує ці параметри як основні критерії пошуку;
- створення критеріїв пошуку – за допомогою отриманої інформації формуються критерії пошуку. Ці критерії допомагають фільтрувати події в базі даних, враховуючи запитані дату, час та локацію;
- пошук події – система використовує сформульовані критерії для пошуку в базі даних, вибираючи події, які відповідають вимогам користувача;
- відправлення повідомлення з рекомендацією – успішно знайдені події відображаються користувачеві через інтерфейс чат-бота, з наданням деталей про подію та можливістю забронювати або отримати більше інформації.

Цей алгоритм не тільки ефективно використовує дані для забезпечення користувачам персоналізованих рекомендацій, але й дозволяє системі адаптуватися до різноманітних запитів і побажань користувачів, забезпечуючи високий рівень задоволеності від використання системи “Eventify”.

3.5 Створення UI / UX дизайну системи

UI/UX дизайн системи “Eventify” виконаний з урахуванням сучасних стандартів зручності та естетичної привабливості, що забезпечує високий рівень користувацького досвіду та функціональності. Основні вікна системи включають екран входу, мапу подій, список подій, чат з ШІ асистентом та профіль користувача, кожне з яких було ретельно опрацьоване для максимізації зручності і візуальної привабливості. Для даного додатку існує дві версії з англійською та українською локалізаціями, далі усі елементи UI будуть відображатися для англомовної версії.

Екран входу розроблений з фокусом на простоту і інтуїтивність. Велика центральна форма входу дозволяє користувачам швидко знайти поля для введення електронної пошти та пароля. Використання зелених відтінків для кнопки входу і акцентних елементів підкреслює важливі дії, сприяючи швидкому розпізнаванню і легкості взаємодії. Додатково реалізована опція для реєстрації нових користувачів, яка розташована під формою входу, що сприяє зручності навігації для нових користувачів (див. рис. 3.6).

Рисунок 3.6 – Екран входу в “Eventify” (рисунок виконано самостійно)

Мапа подій є центральним елементом системи, що дозволяє користувачам легко орієнтуватися серед великої кількості подій. Інтерактивна мапа відображає місця проведення заходів із використанням кастомних маркерів, що забезпечує чітке розрізнення між різними типами подій. Кнопки для вибору режиму відображення карти і пошукове поле розміщені в верхній частині екрану, забезпечуючи легкий доступ до функцій навігації і пошуку (див. рис. 3.7).

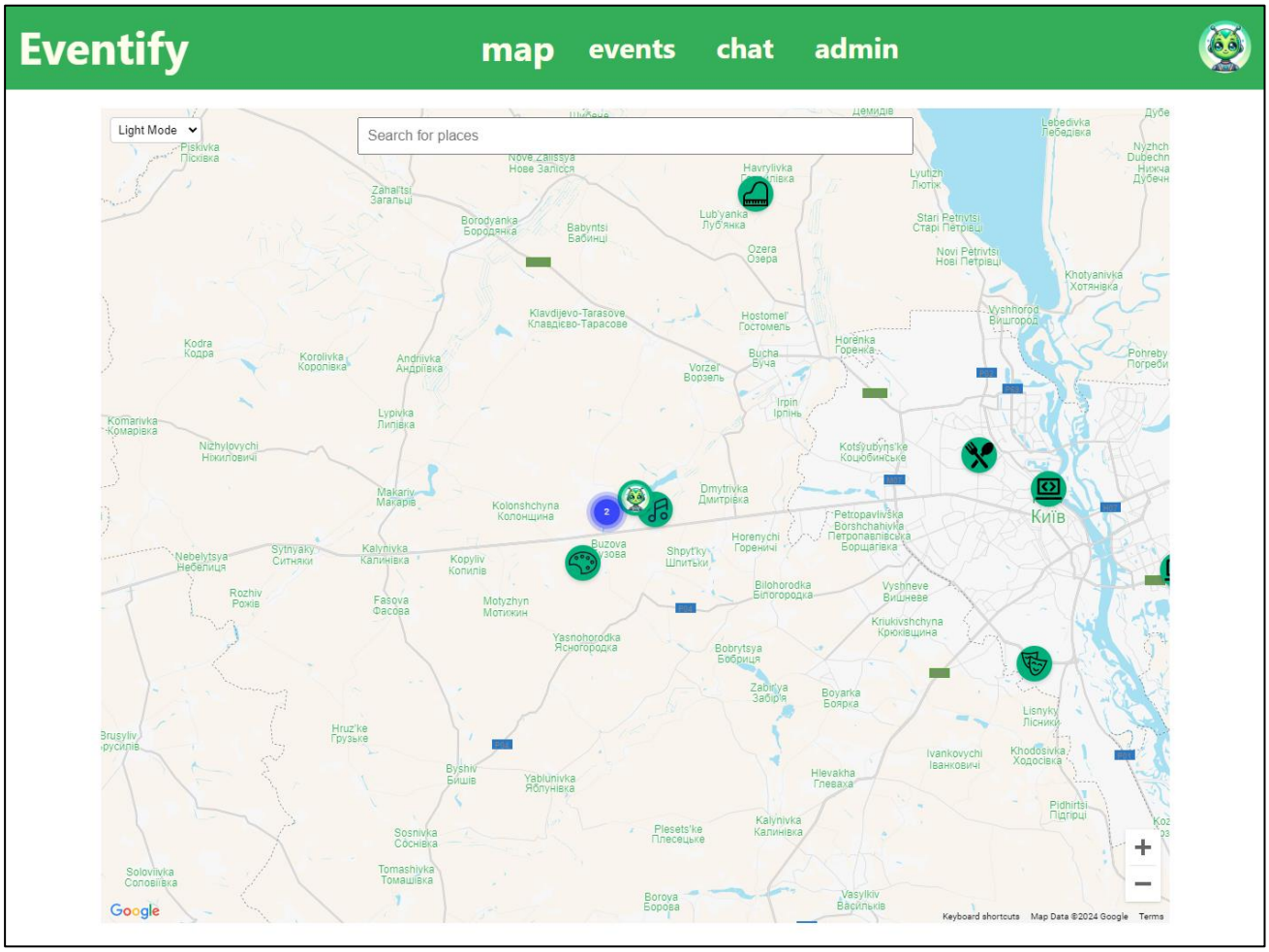


Рисунок 3.7 – Інтерфейс карти подій (рисунок виконано самостійно)

Список подій розроблений з акцентом на візуальну привабливість і функціональність. Картки подій відображають основну інформацію про заходи, включаючи назву, дату, місце і категорії, що дозволяє користувачам швидко знайти релевантні для них заходи. Фільтри для категорій і тегів розташовані у верхній частині екрану, що сприяє швидкій і ефективній фільтрації подій (див. рис. 3.8).

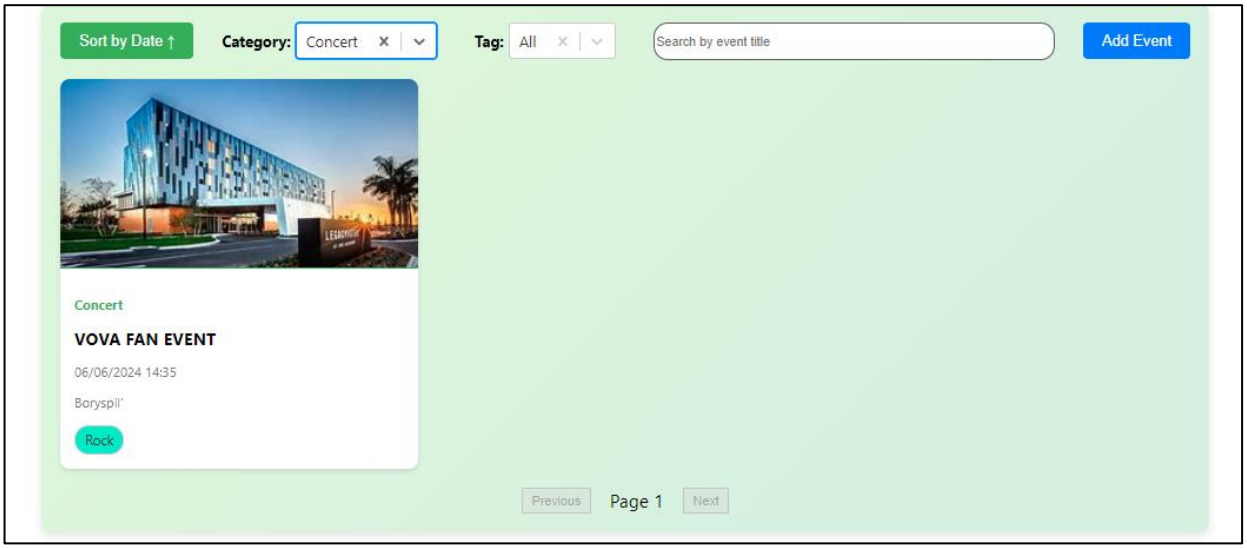


Рисунок 3.8 – Інтерфейс списку подій (рисунок виконано самостійно)

Чат з ШІ асистентом забезпечує інтерактивну підтримку користувачів. Інтерфейс чату розроблений з фокусом на максимальну зручність взаємодії: велика область для відображення повідомлень і зручне поле введення тексту дозволяють користувачам легко отримувати і надсилати повідомлення. Червона кнопка "Reset Chat" забезпечує можливість швидкого скидання чату в разі потреби (див. рис. 3.9).

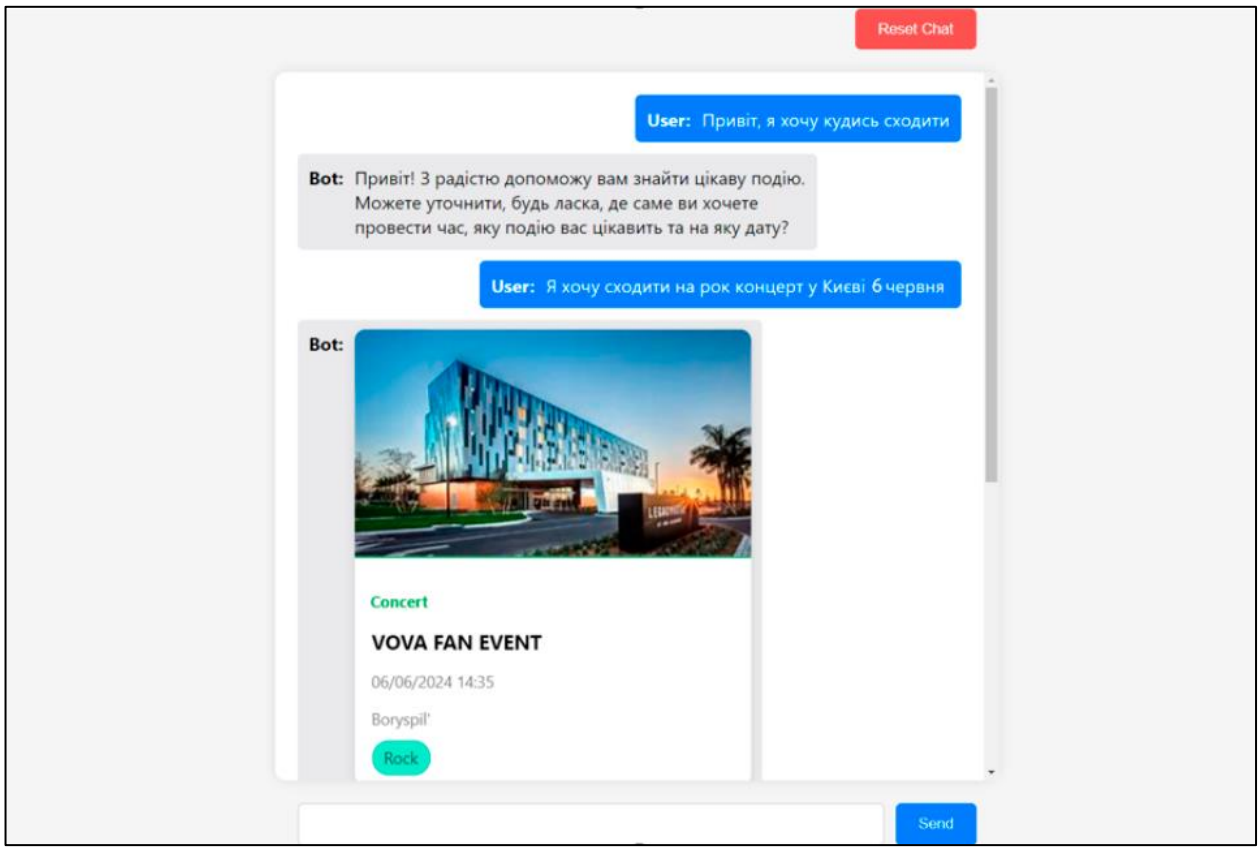


Рисунок 3.9 – Інтерфейс чату з ШІ асистентом (рисунок виконано самостійно)

Профіль користувача надає можливості для перегляду та управління особистими даними. Всі основні елементи профілю, включаючи інформацію про користувача, його зображення, вподобані заходи та налаштування, представлені в структурованому і візуально приємному форматі. Кнопка для редагування профілю розташована на видному місці, що полегшує доступ до функцій управління даними (див. рис. 3.10).

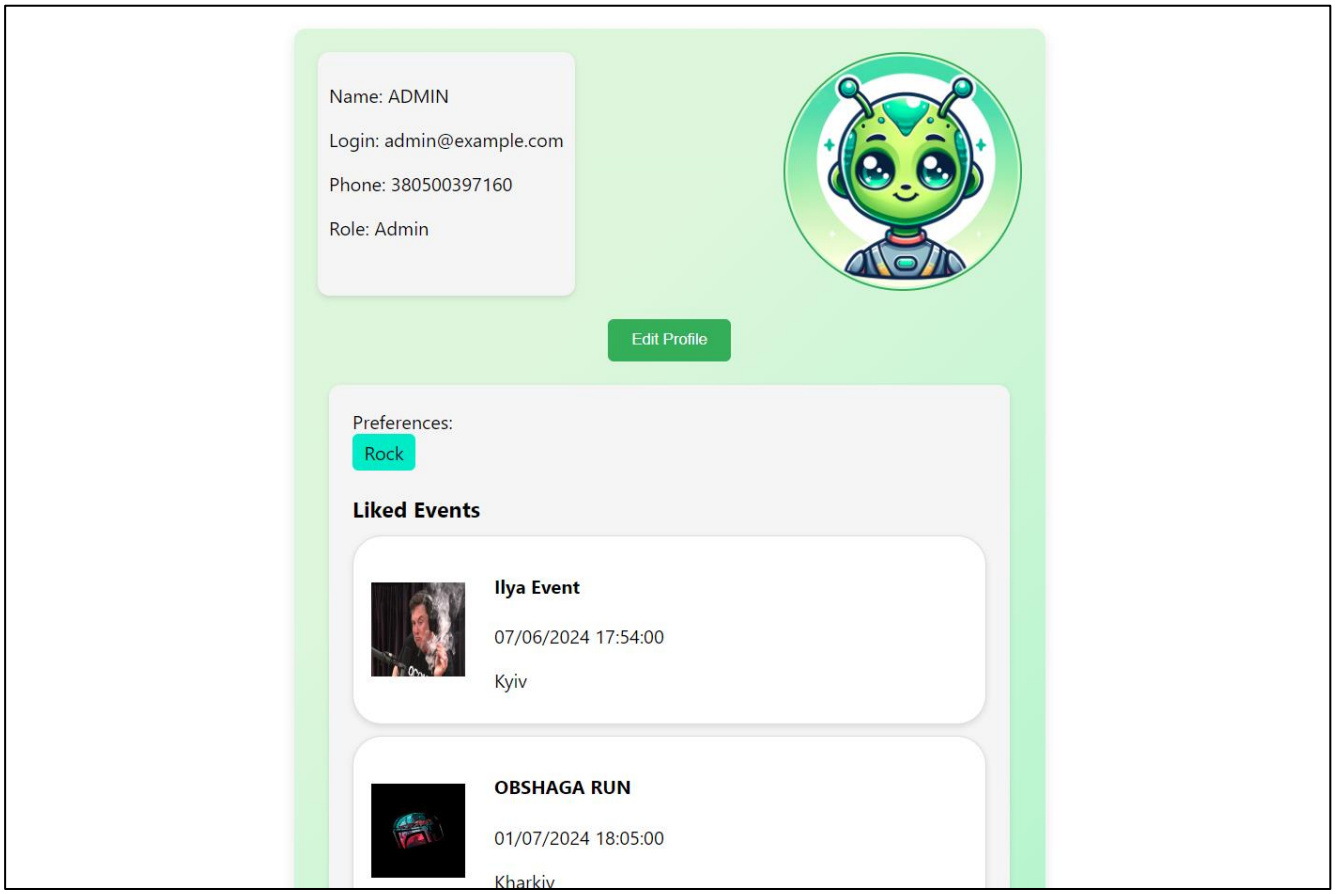


Рисунок 3.10 – Інтерфейс профілю користувача (рисунок виконано самостійно)

Дизайн системи “Eventify” був розроблений з урахуванням принципів зручності користування, інтуїтивності та естетики. Усі основні вікна забезпечують користувачам легкий доступ до інформації та функцій системи, що сприяє покращенню загального досвіду взаємодії з додатком. Інтерфейс системи поєднує в собі елементи сучасного дизайну та функціональність, що дозволяє користувачам ефективно використовувати всі можливості платформи.

4. ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Інтеграція з Google Maps API

Для забезпечення інтерактивної карти в клієнтській частині використовується Google Maps API, який надає потужні інструменти для роботи з картографічними даними. Це дозволяє користувачам переглядати події на карті, шукати місця за допомогою автозаповнення, а також отримувати інформацію про події безпосередньо на карті.

Google Maps API інтегрований (див. рис. 4.1) за допомогою бібліотеки @react-google-maps/api, що дозволяє завантажувати необхідні компоненти і використовувати їх у React-компоненті для відображення геолокаційних даних і взаємодії з ними.

```
const libraries = ["places", "marker"];
const { isLoaded, loadError } = useJsApiLoader({
  googleMapsApiKey: process.env.REACT_APP_GOOGLE_MAPS_API,
  libraries,
});
```

Рисунок 4.1 – Інтеграція Google Maps API для відображення подій на карті (рисунок виконано самостійно)

Для ефективного відображення великої кількості маркерів на карті використовується кластеризація (див. рис. 4.2). Це дозволяє групувати близько розташовані маркери в один кластер, що зменшує навантаження на систему і покращує продуктивність інтерфейсу.

Кластеризація маркерів здійснюється за допомогою MarkerClusterer, що забезпечує згруповане відображення маркерів, зменшуючи кількість елементів, що потрібно рендерити на карті одночасно.

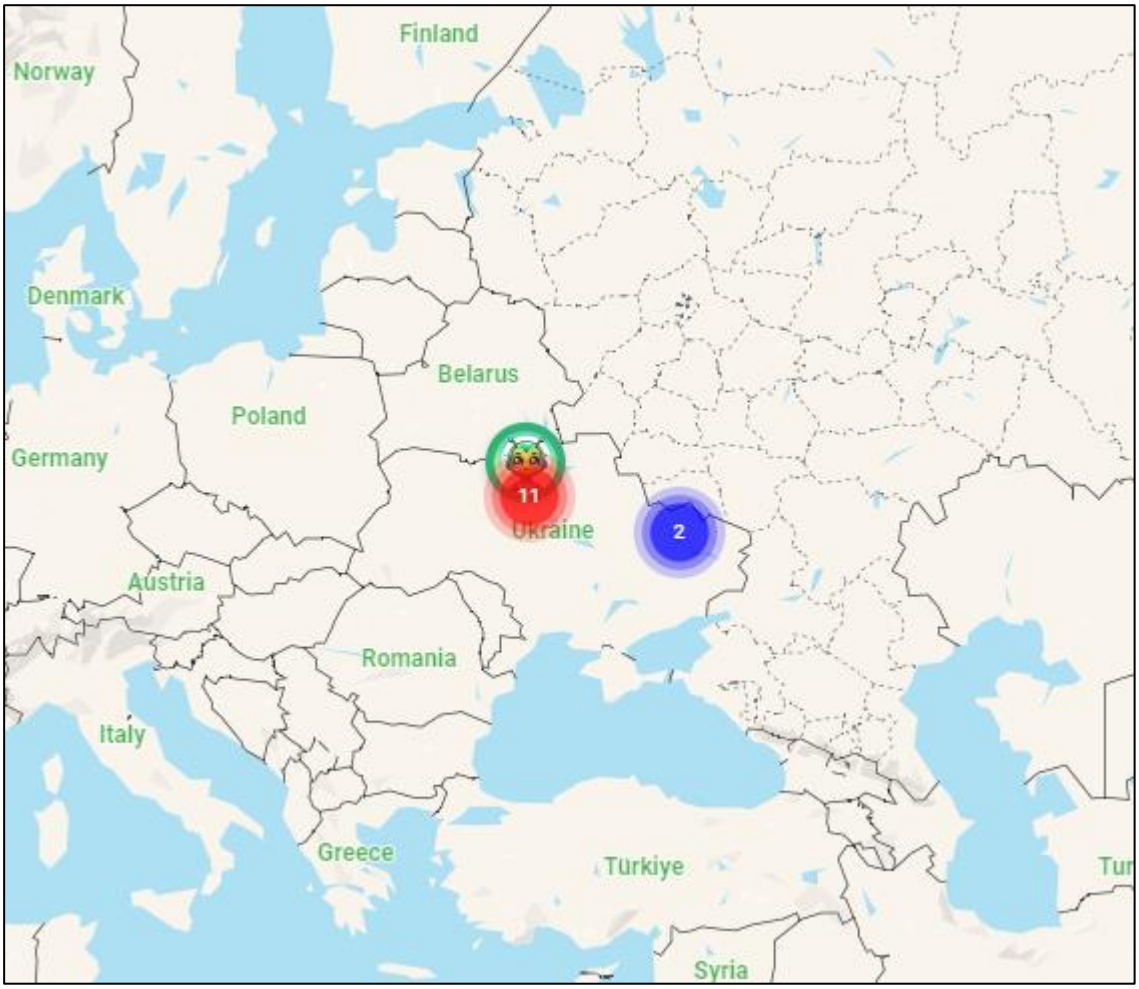


Рисунок 4.2 – Кластеризація маркерів для оптимізації рендерингу великої кількості подій на карті (рисунок виконано самостійно)

Для покращення візуального сприйняття та ідентифікації різних типів подій використовуються кастомні іконки для маркерів на карті. Кожен маркер має іконку, що відповідає типу події, таку як музика, театр, технології тощо.

Це рішення забезпечує чітке розрізнення подій на карті, полегшуючи користувачам процес ідентифікації і вибору заходів.

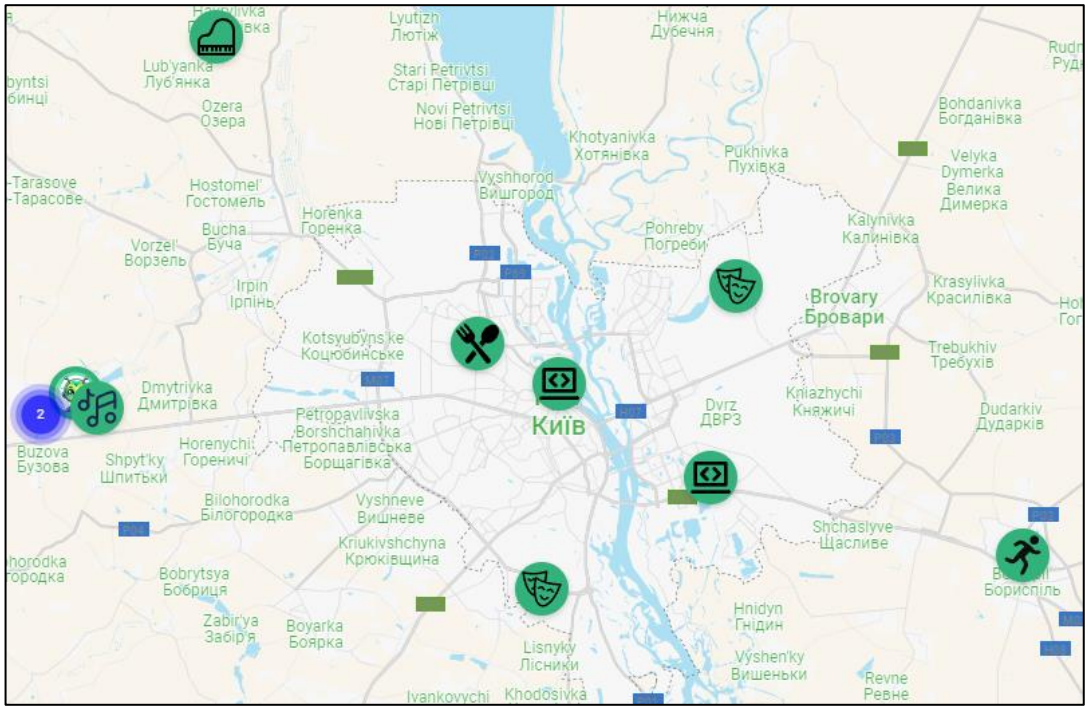


Рисунок 4.3 – Відображення маркерів з кастомними іконками на основі типу події (рисунок виконано самостійно)

Клієнтська частина підтримує інтерактивність маркерів, дозволяючи користувачам натискати на маркери для перегляду детальної інформації про подію. Це забезпечує зручний доступ до додаткової інформації через модальні вікна, що відкриваються при натисканні на маркер (див. рис. 4.4).

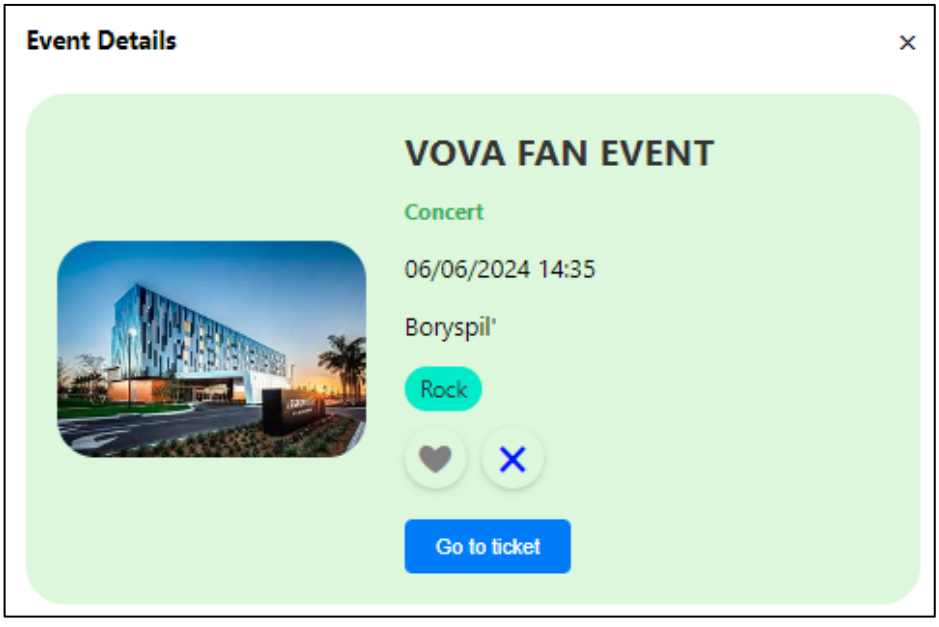


Рисунок 4.4 – Інтерактивність маркерів: відкриття модального вікна з деталями події (рисунок виконано самостійно)

Це значно покращує користувацький досвід, дозволяючи швидко і зручно отримувати інформацію про події без необхідності переходу на інші сторінки.

4.2 Інтеграція з LangChain і OpenAI API для ШІ чат-бота

Для обробки природної мови і генерації відповідей на запити користувачів використовується LangChain і OpenAI API. Це забезпечує підтримку розуміння природної мови і генерацію відповідей за допомогою моделі GPT-4o, що значно підвищує ефективність і точність ШІ чат-бота.

LangChain зберігає історію розмов у пам'яті (ConversationBufferMemory), що дозволяє ефективно управляти контекстом діалогу і підтримувати цілісність сесії (див. рис. 4.5).

```python
def set_chat_environment(history=[]):
    system_instructions = get_prompt_from_source(constants.SYSTEM_INSTRUCTION)

    template = system_instructions
    prompt = ChatPromptTemplate.from_messages(
        [
            SystemMessagePromptTemplate.from_template(template),
            MessagesPlaceholder(variable_name="history"),
            HumanMessagePromptTemplate.from_template("{input}"),
        ]
    )

    model = "gpt-4o-2024-05-13"
    callback_manager = CallbackManager([StreamingStdOutCallbackHandler()])
    llm = ChatOpenAI(
        temperature=0.3, model=model, callback_manager=callback_manager, streaming=True
    )
    memory = ConversationBufferMemory(memory_key="history", return_messages=True)

    memory.chat_memory.messages = [dict_to_message(msg) for msg in history]

    conversation = ConversationChain(memory=memory, prompt=prompt, llm=llm)
    return conversation
```

Рисунок 4.5 – Інтеграція з LangChain і OpenAI API для обробки природної мови (рисунок виконано самостійно)

4.3 Управління станом клієнтської частини за допомогою Redux Toolkit

Для управління станом клієнтської частини використовується Redux Toolkit, який забезпечує централізоване і ефективне управління станом користувача, токенами аутентифікації та іншими даними. Стор (англ. store) забезпечує централізоване зберігання стану додатка і інтеграцію з Redux DevTools для відладки, що спрощує керування станом, робить код більш організованим і сприяє ефективній роботі з даними (див. рис. 4.6).

```
import {combineReducers, configureStore} from "@reduxjs/toolkit";

import authReducer from "../reducers/auth.reducer";

const rootReducer = combineReducers({
    auth : authReducer,
})

export default configureStore({
    reducer: rootReducer,
})
```

Рисунок 4.6 – Конфігурація стору (рисунок виконано самостійно)

Користувацькі дані, такі як профіль та токен автентифікації, зберігаються у локальному сховищі браузера. Якщо токен автентифікації прострочений, він автоматично видаляється, забезпечуючи актуальність даних і безпеку. Аутентифікація здійснюється за допомогою JWT, які зберігаються в локальному сховищі браузера. При вході, виході, зміні профілю і оновленні зображення профілю, стан оновлюється і синхронізується з локальним сховищем для забезпечення безперервності між сесіями. Цей підхід забезпечує надійне управління аутентифікацією і профілем користувача, підвищуючи безпеку і зручність використання додатка. У додатку Д наведено приклад програмного коду для управління станом користувача та аутентифікацією за допомогою Redux Toolkit.

4.4 Локалізація додатку за допомогою i18next

Для забезпечення багатомовної підтримки в додатку використовується бібліотека i18next (див. рис. 4.7). Це дозволяє користувачам перемикатися між мовами і отримувати інтерфейс на обраній мові.

```
import enTranslation from './locales/en/translation.json';
import i18n from "i18next";
import { initReactI18next } from "react-i18next";
import uaTranslation from './locales/ua/translation.json';

export const LANG_KEY = 'LANG_KEY'

const lang = localStorage.getItem(LANG_KEY) ?? "en"

i18n
    .use(initReactI18next)
    .init({
        resources: {
            en: { translation: enTranslation },
            ua: { translation: uaTranslation },
        },
        lng: lang,
        fallbackLng: lang,
        interpolation: {
            escapeValue: false
        }
    });

export default i18n;
```

Рисунок 4.9 – Локалізація додатку за допомогою i18next (рисунок виконано самостійно)

Використання i18next забезпечує завантаження і зберігання мовних налаштувань в локальному сховищі, що дозволяє додатку автоматично підлаштовуватися під обрану мову користувача і надавати локалізований інтерфейс.

5. ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для виявлення помилок під час розробки веб-застосунку застосовувалося мануальне тести. Це означає, що перевірка програмного забезпечення на наявність помилок здійснювалася вручну, без використання автоматизованих тестів.

Щоб розпочати тестування, потрібно скласти план, де буде визначено мету тестування та перелік функцій для перевірки на правильність роботи.

Виходячи з цього плану, виконувалося тестування створення, редагування та видалення необхідних даних. Також важливою частиною тестування була валідація вхідних даних, що сприяє належному функціонуванню веб-застосунку. Приклади проведених тест-кейсів:

а) тест-кейс №1 – створення нового заходу:

1) послідовність роботи:
   - увійти під аккаунтом Організатора;
   - увійти на сторінку усіх заходів;
   - натиснути на кнопку «Додати захід»;
   - ввести дані та натиснути «Додати захід».
2) використані дані:
   - назва = Рок-Концерт;
   - опис заходу = Опис концерту;
   - тип заходу = Концерт;
   - зображення заходу = image.png
   - дата заходу = 20.06.2024;
   - посилання на квиток = ”url-to-ticket”;
   - місце проведення заходу = {lat: 50.47, lng: 30.02}.
3) результат, що очікується – модальна форма закривається та захід додається до загального списку;
4) результат, який було отримано – модальна форма закрилась та захід додався до загального списку;
5) висновок – тест пройдено.

б) тест-кейс №2 – оновити інформацію заходу:

1) послідовність роботи:
   - увійти під аккаунтом Організатора;
   - увійти на сторінку заходів створених Організатором;
   - натиснути на кнопку «Оновити захід»;
   - ввести дані та натиснути «Оновити захід».
2) використані дані:
   - назва = Рок-Концерт;
   - опис заходу = Опис концерту;
   - тип заходу = Концерт;
   - зображення заходу = image.png
   - дата заходу = 20.06.2024;
   - посилання на квиток = ”url-to-ticket”;
   - місце проведення заходу = {lat: 50.47, lng: 30.02}.
3) результат, що очікується – модальна форма закривається та захід оновиться у загальному списку;
4) результат, який було отримано – модальна форма закрилась та захід оновився у загальному списку;
5) висновок – тест пройдено.

в) тест-кейс №3 – оновити дані користувача:

1) послідовність роботи:
   - увійти під аккаунтом Користувача;
   - увійти на сторінку профілю, натиснувши на аватар користувача в навігаційному меню;
   - натиснути на кнопку «Редагувати профіль»;
   - ввести дані та натиснути «Зберегти та вийти».
2) використані дані:
   - ім'я користувача = Вова;
   - телефон = +380500000000;

- зображення = image.png
- теги = Рок.

3) результат, що очікується – після натискання на кнопку «Оновити» дані оновлюються;
4) результат, який було отримано – після натискання на кнопку «Оновити» дані оновилися;
5) висновок – тест пройдено.

г) тест-кейс №4 – оновити дані про тег:

1) послідовність роботи:
   - увійти під аккаунтом Адміністратора;
   - увійти на сторінку адміністрування тегів, натиснувши на опцію Адмін в навігаційному меню;
   - обрати тег натиснути на кнопку «Оновити»;
   - ввести дані та натиснути «Зберегти та вийти».
2) використані дані:
   - ім'я = Рок;
   - колір = #FFFFFF.
6) результат, що очікується – після натискання на кнопку «Оновити» дані оновлюються;
3) результат, який було отримано – після натискання на кнопку «Оновити» дані оновилися;
4) висновок – тест пройдено.

д) тест-кейс №5 – оновити роль користувача:

1) послідовність роботи:
   - увійти під аккаунтом Адміністратора;
   - увійти на сторінку адміністрування користувачів, натиснувши на опцію Адмін в навігаційному меню;
   - обрати користувача натиснути на кнопку «Змінити роль»;
   - ввести дані та натиснути «Зберегти та вийти».

2) використані дані:
   - роль = Адмін;
3) результат, що очікується – після натискання на кнопку «Зберегти» дані оновлюються та у користувача з'являться нові можливості;
4) результат, який було отримано – після натискання на кнопку «Зберегти» дані оновилися та у користувача з'явилися нові можливості;
5) висновок – тест пройдено.

е) тест-кейс №6 – видалити користувача:
1) послідовність роботи:
   - увійти під аккаунтом Адміністратора;
   - увійти на сторінку адміністрування користувачів, натиснувши на опцію Адмін в навігаційному меню;
   - обрати користувача натиснути на кнопку «Видалити»;
   - Підтвердити видалення.
2) використані дані:
   - id користувача;
3) результат, що очікується – після натискання на кнопку «Видалити» профіль видалиться;
4) результат, який було отримано – після натискання на кнопку «Видалити» видалилися;
5) висновок – тест пройдено.

ж) тест-кейс №7 – перевірка роботи сортування, фільтрації та пошуку заходів:
1) послідовність роботи:
   - увійти під аккаунтом користувача;
   - увійти на сторінку заходів, натиснувши на опцію «Заходи» в навігаційному меню;
   - обрати потрібні фільтри для сортування, пошуку та вибору категорій і типів.

2) використані дані:
   - сортування за датою = за зростанням;
   - категорія = концерт;
   - тег = рок;
   - пошукова рядок = Фан івент.
3) результат, що очікується – після введення даних події автоматично з'являється на сторінці;
4) результат, який було отримано – після введення даних події автоматично з'являлися на сторінці;
5) висновок – тест пройдено.

з) тест-кейс №8 – спілкування з ШІ чат-ботом:
1) послідовність роботи:
   - увійти під аккаунтом користувача;
   - увійти на чату, натиснувши на опцію «Чат» в навігаційному меню;
   - написати повідомлення у полі та натиснути кнопку «Відправити».
2) використані дані:
   - повідомлення = привіт, я хочу сходити на рок-концерт у Києві 20 червня;
3) результат, що очікується – відповідь від чат-бота з заходом який шукав користувач;
4) результат, який було отримано – відповідь від чат-бота з заходом який шукав користувач;
5) висновок – тест пройдено.

Зробивши мануальне тестування усіх ключових компонентів клієнтської частини я можу зробити висновок, що все працює правильно та не було виявлено ніяких помилок.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра була розроблена клієнтська та ШІ частини програмної системи “Eventify”, яка спрямована на рекомендацію та підбір актуальних культурних подій, адаптованих до індивідуальних уподобань та геолокації користувачів.

Розробка здійснювалася на основі сучасних технологій, зокрема, фреймворку React.js для клієнтської частини, та платформи LangChain для інтеграції чат-бота з підтримкою штучного інтелекту. У процесі роботи була реалізована інтерактивна карта подій за допомогою Google Maps API, що дозволяє користувачам зручно переглядати та фільтрувати події за місцем їх проведення.

Використання Redux Toolkit для управління станом клієнтської частини стало важливим елементом у забезпеченні централізованого зберігання даних та їх ефективного оновлення. Це рішення дозволило спростити роботу з даними, підвищуючи продуктивність та забезпечуючи зручність управління різними аспектами додатка.

Чат-бот на базі моделі GPT-4o, інтегрований через LangChain та OpenAI API, забезпечує високоякісну підтримку природної мови, дозволяючи надавати персоналізовані рекомендації та відповідати на запитання користувачів в реальному часі.

Розроблена система “Eventify” не лише відповідає сучасним вимогам до програмного забезпечення, але й демонструє значний потенціал для подальшого розвитку та оптимізації. Дотримання принципів сучасної розробки, таких як компонентність і модульність, а також інтеграція передових технологій забезпечують гнучкість, масштабованість та продуктивність системи. Цей підхід дозволяє легко адаптувати систему до змін у потребах користувачів та розвитку технологій, забезпечуючи її надійність та ефективність у вирішенні завдань з підбору культурних подій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Афіша і квитки Concert.ua [Електронний ресурс] – URL: https://concert.ua/uk/catalog/dnipro/concerts (дата звернення: 09.06.2024).
2. Афіша подій KARABAS [Електронний ресурс] – URL: https://kyiv.karabas.com/ua/ (дата звернення: 09.06.2024).
3. React | Getting Started. [Електронний ресурс] – URL: https://legacy.reactjs.org/docs/getting-started.html (дата звернення: 09.06.2024).
4. Introduction | LangChain. [Електронний ресурс] – URL: https://python.langchain.com/v0.1/docs/get_started/introduction/ (дата звернення: 09.06.2024).
5. Swift.org – Documentation. [Електронний ресурс] – URL: https://www.swift.org/documentation/ (дата звернення: 09.06.2024).
6. ASP.NET Tutorial [Електронний ресурс] – URL: https://dotnet.microsoft.com/en-us/learn/aspnet/hello-world-tutorial/intro (дата звернення: 09.06.2024)
7. C# docs - get started, tutorials, reference. Microsoft Learn: Build skills that open doors in your career. [Електронний ресурс] – URL: https://docs.microsoft.com/en-us/dotnet/csharp/ (дата звернення: 09.06.2024).
8. Мазурова О. А., Широкопетлева М. С., Черепанова Ю. Ю. Информационные системы. [Електронний ресурс] – URL: https://dl.nure.ua/pluginfile.php/509/mod_resource/content/2/01.pdf (дата звернення: 09.06.2024).
9. Python 3.12.3 documentation. [Електронний ресурс] – URL: https://docs.python.org/3/ (дата звернення: 09.06.2024).
10. Лавріщева К. М. Програмна інженерія / К. М. Лавріщева. – К. : Академперіодика, 2008. – 319 с.

11. Nguyen H. Single-page application and front-end testing methods: built with React and React Router, tested with Jest and Cypress. – 2022. [Електронний ресурс] – URL: https://www.theseus.fi/bitstream/handle/10024/745098/Nguyen_Huong.pdf (дата звернення: 09.06.2024).

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

UNICHECK
by Turnitin

Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016341986

Дата перевірки:
10.06.2024 12:10:06 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
10.06.2024 14:30:22 EEST

ID користувача:
100012353

Назва документа: 2024_Б_ПІ_ПЗПІ_20_3_Сафошин_В_В_скорочений

Кількість сторінок: 35 Кількість слів: 5364 Кількість символів: 42262 Розмір файлу: 1.66 MB ID файлу: 1016143265

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.88%
Схожість

Найбільша схожість: 0.86% з джерелом з Бібліотеки (ID файлу: 1016105122)

Пошук збігів з Інтернетом не проводився

1.88% Джерела з Бібліотеки 74 ..... Сторінка 37

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

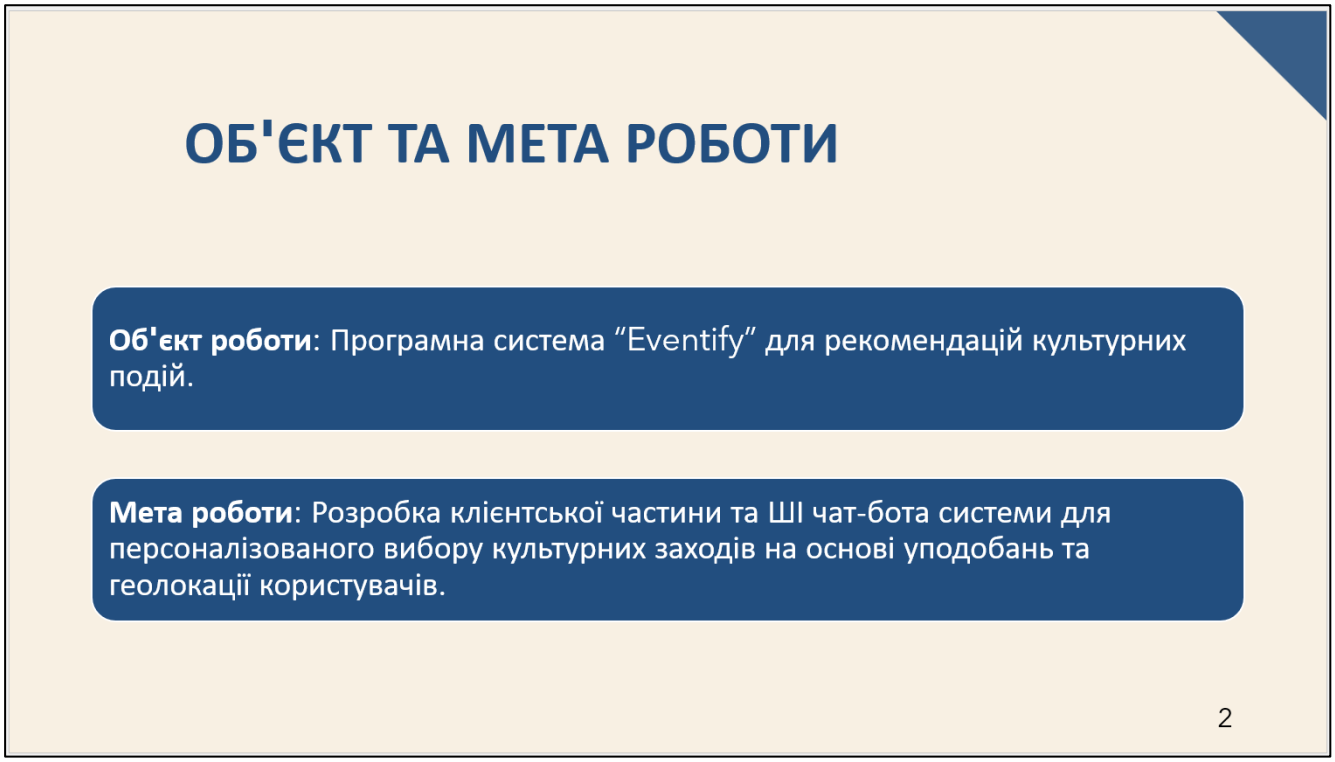
Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 7 сторінок

ДОДАТОК Б

Слайди презентації

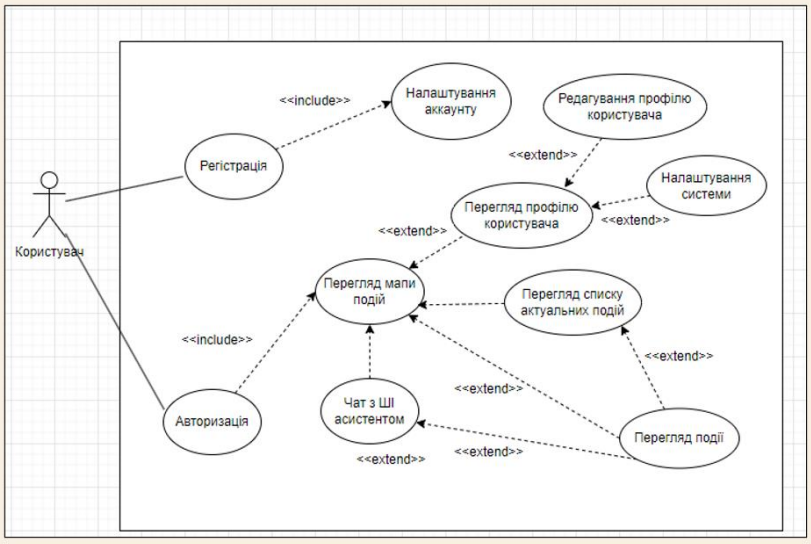


ПОСТАНОВКА ЗАДАЧІ

- Автоматичний збір даних про уподобання користувачів.
- Розробка зручного інтерфейсу користувача.
- Візуалізація подій на інтерактивній мапі.
- Інтеграція з ШІ-асистентом для допомоги у виборі подій.

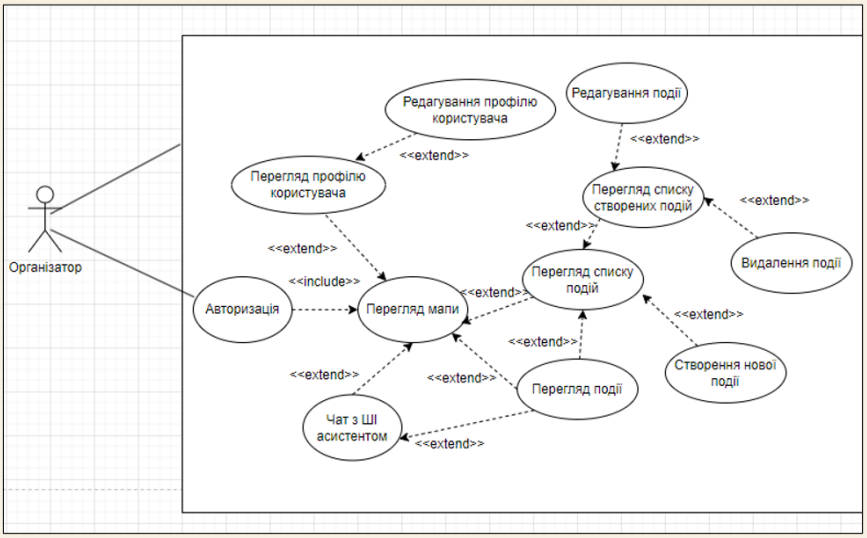
3

USE CASE ДЛЯ КОРИСТУВАЧА



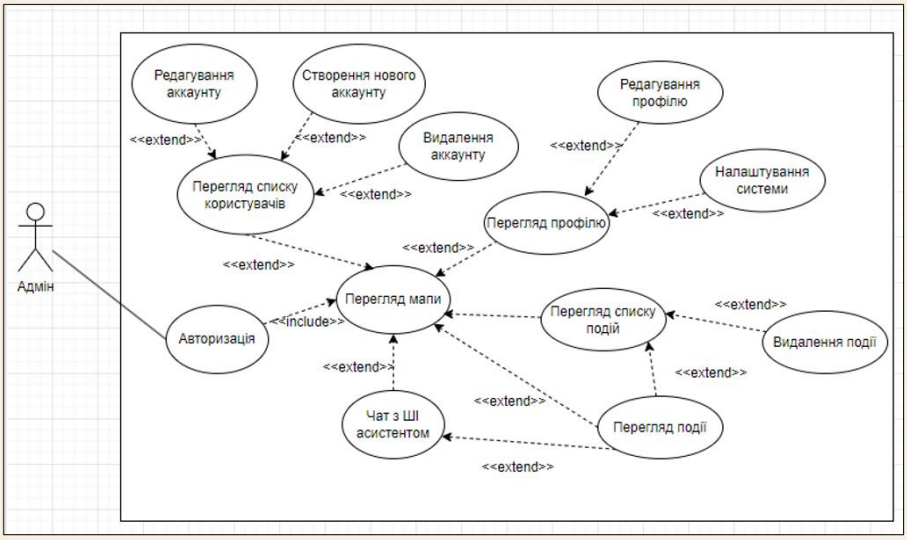
5

USE CASE ДЛЯ ОРГАНІЗАТОРА



6

USE CASE ДЛЯ АДМІНІСТРАТОРА



7

АРХІТЕКТУРА

Односторінковий додаток (SPA) є центральною частиною архітектури фронтенд частини “Eventify”.

Переваги :

1. Швидка відповідь
2. Зменшене навантаження на сервер
3. Поліпшена користувацька взаємодія

7

ЧОМУ Я ОБРАВ REACT.JS?

React.js був обраний завдяки його гнучкості, високій продуктивності та підтримці компонентного підходу.

Переваги :

1. Компонентний підхід
2. Віртуальний DOM
3. Гнучка інтеграція
4. Широка підтримка спільноти

React

8

ЧОМУ Я ОБРАВ PYTHON ТА LANGCHAIN ДЛЯ ШІ ЧАТБОТУ?

Переваги:

1.Легкість використання.

2.Багата екосистема бібліотек.

3.Гнучкість інтеграції.

4.Масштабованість.

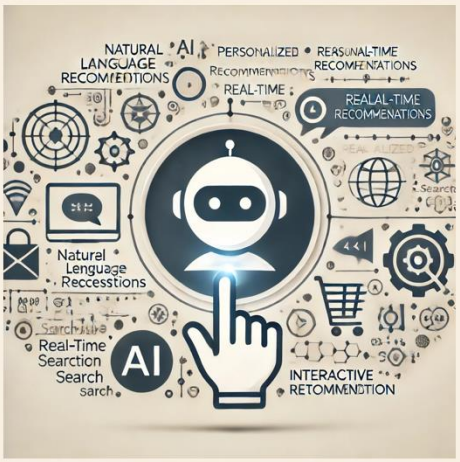
5.Широка підтримка.

9

ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ШІ ЧАТ-БОТА

Функціональні можливості:

1. Персоналізовані рекомендації
2. Обробка природної мови.
3. Підтримка в реальному часі.
4. Інтерактивний пошук.



10

ТЕХНОЛОГІЇ

11

ТЕСТУВАННЯ

Було проведене мануальне тестування, ось основні тест-кейси:

1. Створення нового заходу
2. Оновлення інформації про захід
3. Оновлення даних користувача
4. Оновлення ролі користувача
5. Видалити користувача
6. Перевірка роботи сортування, фільтрації та пошуку заходів
7. Спілкування з ШІ чат-ботом

12

АПРОБАЦІЯ

- Сафошин В. В., Хамінов І. О., Дегтяр В. Е., Побіженко І. О. Eventify: інноваційний підхід до організації та пошуку заходів з використанням ШІ. XXVIII міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті», конференція «Інформаційні інтелектуальні системи» - с. 544.

13

ВИСНОВКИ

- Було визначено мету теми та поставлені задачі, що стоять перед метою;
- Проаналізовані сучасні підходи в розробці;
- Розроблено клієнтську частину й ШІ чат-бота та проведено їх тестування.

14

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК В

Специфікація вимог до програмного продукту

ВСТУП

1.1 Огляд продукту

Сервіс "Eventify" з інтегрованим back-end рішенням представляє собою комплексний інструмент для користувачів, зацікавлених у відвідуванні та організації різноманітних заходів, включаючи концерти, виставки, ярмарки, фестивалі. Цей продукт включає в себе широкий спектр функцій для ефективного пошуку, відбору та участі в заходах.

Автентифікація та авторизація користувачів є ключовими елементами безпеки сервісу, що гарантує захист особистої інформації та контроль доступу до функцій залежно від ролі користувача. Ці можливості дозволяють користувачам створювати персональні профілі, налаштовувати інтереси та отримувати персоналізовані рекомендації.

"Eventify" спрощує процес відкриття та реєстрації на заходи завдяки інтуїтивно зрозумілому інтерфейсу, де користувачі можуть переглядати детальну інформацію про події, включаючи час, дату, місце проведення, а також реєструватися на них в кілька кліків. Це забезпечує зручний доступ до культурних та соціальних заходів.

Система також має функцію керування подіями, де організатори можуть створювати нові заходи, редагувати існуючі, управляти реєстраціями та відгуками відвідувачів. Це дає змогу ефективно організовувати та планувати заходи, а також збирати зворотний зв'язок для покращення майбутніх подій.

Додатково, сервіс включає можливість спілкування з ШІ асистентом, який допомагає користувачам у виборі подій на основі їхніх переваг, історії пошуку та оцінок, підвищуючи особистісний досвід використання платформи.

Загалом, "Eventify" є інноваційним рішенням для тих, хто шукає активне культурне життя, надаючи зручні інструменти для відкриття, планування та участі

в заходах, а також для організаторів, щоб ефективно управляти та просувати свої події.

1.2 Мета

Основною метою проекту "Eventify" є створення універсального сервісу, який спрощує процес пошуку, вибору та участі в різноманітних заходах, таких як концерти, виставки, ярмарки, фестивалі, для користувачів. Цей сервіс надає можливість перегляду актуальних подій на карті місцевості, їх фільтрації за інтересами та пошуку, а також рекомендації заходів, які можуть бути цікавими для конкретного користувача.

Додатково, "Eventify" забезпечує верифікацію користувачів, можливість для переходу на сторінку для замовлення квитків та запрошень на заходи, виставлення оцінок. Сервіс також включає інтеграцію зі штучним інтелектом, який виступає в ролі асистента для користувачів, допомагаючи у виборі заходів на основі їх переваг.

Таким чином, мета проекту "Eventify" полягає у забезпеченні зручного, інтуїтивно зрозумілого інструменту для відкриття нових можливостей дозвілля, сприяння культурному розвитку та соціалізації користувачів через участь у заходах, що відбуваються в їхній місцевості або в інших регіонах.

1.3 Межі

У контексті проекту "Eventify", межі визначаються функціональними та технічними аспектами системи. Проект зосереджений на наданні користувачам інструментів для пошуку, фільтрації, рекомендації та участі у заходах, а також на інтеграції з платформами для замовлення квитків та спілкуванні з ШІ асистентом. Важливими аспектами є верифікація користувачів, виставлення оцінок подіям.

Технічні обмеження системи включають обмеження щодо ресурсів сервера, які можуть вплинути на її масштабованість та продуктивність при великій кількості одночасних запитів. Це також стосується обмежень на кількість одночасних користувачів та подій, які можуть бути оброблені системою без зниження її ефективності.

З точки зору безпеки, проект має забезпечити захист даних користувачів, включаючи аутентифікацію, авторизацію, шифрування даних та захист від несанкціонованого доступу. Це вимагає використання сучасних протоколів безпеки та методів шифрування, а також постійного моніторингу та оновлення системи безпеки.

Проект "Eventify" не займається безпосередньою організацією заходів, а лише надає платформу для їх пошуку та рекомендації. Тому, відповідальність за точність інформації про заходи, їх виконання та якість лежить на організаторах подій та представниках заходів.

1.4 Посилання

Проект "Eventify" прагне автоматизувати та оптимізувати процеси вибору та участі в культурних заходах, мінімізуючи необхідність ручного пошуку актуальних подій у різних джерелах. Інтегровані функції верифікації користувачів, бронювання квитків, виставлення оцінок, а також відображення подій на карті надають користувачам зручний і інтуїтивно зрозумілий доступ до необхідної інформації. Це значно спрощує процес взаємодії з подіями та їх відвідування, покращуючи загальний досвід користувача.

"Eventify" також забезпечує організаторам ефективний канал для залучення нової аудиторії та просування своїх заходів. Сервіс дозволяє їм швидко оновлювати інформацію про події, управляти реєстраціями та отримувати зворотний зв'язок від учасників. Це створює можливості для покращення організації заходів та підвищення їх якості.

Таким чином, "Eventify" пропонує цілісний підхід до взаємодії з культурними подіями, сприяючи культурному розвитку користувачів та ефективному управлінню подіями, що покращує досвід як відвідувачів, так і організаторів

1.5 Означення та абревіатури

Backend – це серверна частина програмного забезпечення, яка відіграє ключову роль у обробці даних, виконанні бізнес-логіки та забезпеченні безпеки. Вона управляє взаємодією з базою даних, виконує процеси аутентифікації та авторизації користувачів і забезпечує виконання операцій, важливих для функціонування системи.

Frontend - це клієнтська частина програмного забезпечення, відповідальна за інтерфейс та взаємодію з користувачем. Вона включає в себе дизайн, розмітку та скрипти сторінок, які користувачі бачать та з якими взаємодіють в браузері або через мобільні додатки.

API (Application Programming Interface) – це інтерфейс програмування застосунків, що дозволяє різним програмним компонентам спілкуватися між собою. API виступає як місток, що визначає правила та методи, за допомогою яких можна доступатися до функціоналу або даних однієї програми з іншої.

ORM (Object-Relational Mapping) – технологія, що дозволяє мапінг реляційних баз даних на об'єктно-орієнтовану модель програми. Завдяки ORM, робота з базою даних стає більш інтуїтивно зрозумілою для розробників, оскільки вони можуть оперувати об'єктами замість SQL-запитів.

HTTP (HyperText Transfer Protocol) – основний протокол передачі даних у Всесвітній мережі, який використовується для завантаження веб-сторінок з сервера на клієнтський браузер. HTTP визначає спосіб, яким повинні бути сформовані та передані запити та відповіді між клієнтом та сервером.

CRUD (Create, Read, Update, Delete) – це концепція, що описує чотири основні дії, які використовуються при роботі з даними: створення нових записів, читання (вибірка) існуючих даних, оновлення (модифікація) даних та видалення записів. Ці операції є фундаментальними для будь-яких систем, які працюють з базами даних.

LLM (Large Language Model) – це велика мовна модель, що використовує алгоритми глибокого навчання для обробки та генерації природної мови, здатна

виконувати широкий спектр задач, від автоматичного перекладу до створення текстового контенту. LLM використовуються для покращення здатності комп'ютерних систем розуміти та взаємодіяти з людською мовою, надаючи можливості для створення більш продвинутих і зручних користувацьких інтерфейсів, а також для аналізу великих обсягів текстових даних.

2. ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Перспективи програмної системи “Eventify” – надання рекомендацій та підбору актуальних заходів (концертів, виставок, ярмарок, фестів) є обнадійливими і обіцяють багато можливостей для подальшого розвитку та розширення.

З урахуванням зростаючого інтересу до культурних подій та розваг, а також зростаючої потреби у персоналізованих та зручних інструментах для вибору та участі в заходах, "Eventify" може знайти своє місце в різних галузях та сприяти розвитку сфери розваг та культурно-масових заходів.

Перспективи продукту включають розширення функціональності та додавання нових модулів, щоб задовольнити специфічні потреби організацій та користувачів. Наприклад, можливості планування і розподілу ресурсів, система відгуків, інтеграція з іншими платформами або соціальними мережами, розширена аналітика та звітність.

Крім того, вдосконалення аналітичних засобів для оцінки успішності різних заходів та рекламних кампаній. Розробка розширених звітів для організаторів та партнерів для підвищення ефективності подій. Розширення географічного охоплення для підтримки подій у різних регіонах та країнах. Активне залучення до міжнародних культурних заходів та фестивалів для розширення аудиторії.

Загалом, "Eventify" має великий потенціал стати ключовим гравцем у сфері організації та участі в культурних та розважальних заходах. Розширення функціональності, посилення географічного охоплення та впровадження новітніх технологій можуть вивести продукт на новий рівень та забезпечити йому стабільне місце на ринку.

2.2 Функції продукту

Основні функції цього продукту включають:

- FE-1: управління користувачами;
- FE-2: управління заходами;

- FE-3: безпека та доступ до даних;
- FE-4: пошук і фільтрація заходів;
- FE-5: взаємодія з ШІ асистентом;
- FE-6: перегляд подій на карті.

2.3 Характеристика користувачів

Характеристики користувачів програмної системи “Eventify” – надання рекомендацій та підбору актуальних заходів (концертів, виставок, ярмарок, фестів) можуть бути різноманітними, оскільки система взаємодіє з різними типами користувачів. Основні характеристики користувачів включають:

- користувачі є основними в системі. Це люди будь-якого віку та статі. Їх характеристики можуть включати їхні інтереси, їх вік та локацію де вони мешкають;
- менеджери компаній відповідають за керування подіями. Вони мають розширені права доступу та можуть створювати, редагувати та видаляти події, керувати користувачами, аналізувати звітність та здійснювати інші адміністративні функції;
- адміністратори системи відповідають за керування користувачами та менеджерами.

2.4 Загальні обмеження

У програмній системі “Eventify” – надання рекомендацій та підбору актуальних заходів (концертів, виставок, ярмарок, фестів) існують загальні обмеження, які можуть впливати на її функціонування та використання. Деякі з цих обмежень включають.

Система повинна дотримуватися вимог безпеки, щоб захистити дані користувачів та запобігти несанкціонованому доступу. Обмеження безпеки можуть включати криптографічні вимоги, захист від атак, правила доступу до даних та контроль прав користувачів.

Також система повинна бути здатна масштабуватись для впорядкування зростаючого обсягу користувачів, проектів та завдань. Обмеження масштабованості можуть впливати на продуктивність та швидкодію системи при збільшенні навантаження.

2.5 Припущення й залежності

Наявність стабільного та надійного Інтернет-з'єднання є важливою умовою для оптимальної роботи системи. Система взаємодіє з мережею для обміну даними, автентифікації, надсилання сповіщень тощо, тому постійне з'єднання з Інтернетом є необхідним. Впевненість в стабільності та швидкості Інтернет-з'єднання є важливою умовою для ефективної роботи системи.

Доступ до бази даних: Система налагоджується від доступу до бази даних для зберігання та отримання інформації про користувачів, проекти, завдання та інші дані. Наявність та налаштування бази даних виконуються через Microsoft SQL Server. Ця база даних є критичною для функціонування системи, тому важливо впевнитися в її належному функціонуванні та оптимальному налаштуванні.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейс користувача програмної системи рекомендації та відбору актуальних подій з back-end реалізацією розроблений з метою забезпечення зручності, ефективності та легкості використання для користувачів. Він пропонує інтуїтивно зрозуміле та привабливе середовище, яке дозволяє взаємодіяти з системою та виконувати необхідні завдання.

Інтерфейс користувача може бути представлений у вигляді веб-додатку або мобільного додатка, залежно від потреб користувачів та доступних платформ. Він має чітку структуру, навігаційні елементи, що дозволяють користувачам легко орієнтуватися та виконувати різноманітні дії.

Основні характеристики інтерфейсу користувача включають:

- інтуїтивність: Забезпечення легкого розуміння та використання для користувачів без додаткового навчання.
- ефективність: Максимізація продуктивності користувачів та зменшення часу на виконання завдань.
- привабливість: Створення привабливого та естетичного вигляду інтерфейсу для залучення користувачів.
- чітка структура: Організація інформації та функціональності для легкого доступу та сприяння логічному розташуванню елементів.
- навігаційні можливості: Доступні та зрозумілі елементи навігації для швидкого переміщення користувачів по системі.

Ці аспекти сприяють позитивному враженню від використання системи та підвищують задоволення користувачів взаємодією з нею.

3.1.2 Апаратний інтерфейс

Апаратний інтерфейс програмній системі рекомендації та відбору актуальних подій з back-end реалізацією включає необхідні апаратні компоненти та засоби, які

використовуються для забезпечення функціональності системи. Оскільки ця програма зазвичай працює в онлайн-середовищі та доступна через веб-браузер або мобільний додаток, система вимагає наявності серверів для зберігання та обробки даних. Це можуть бути фізичні сервери, віртуальні сервери або хмарні платформи, які забезпечують необхідні обчислювальні та ресурси для зберігання роботи системи.

3.1.3 Програмний інтерфейс

Програмний інтерфейс (API) програмної системи рекомендації та відбору актуальних подій з back-end реалізацією визначає набір правил та протоколів, за допомогою яких різні компоненти програми можуть взаємодіяти між собою. Він дозволяє зовнішнім системам або розробникам використовувати функціональні можливості та отримувати доступ до даних програмної системи.

Програмний інтерфейс може бути реалізований у вигляді RESTful API , який базується на використанні HTTP-протоколу для передачі даних між клієнтами та сервером. Це дозволяє здійснювати стандартизовану та просту взаємодію з системою за допомогою HTTP-запитів (GET, POST, PUT, DELETE) та обміну даними у форматі JSON або XML.

3.1.4 Комунікаційний протокол

Комунікаційний протокол в програмної системи рекомендації та відбору актуальних подій з back-end реалізацією визначає правила та формати обміну даними між різними компонентами системи. Він гарантує, що передача інформації між цими компонентами відбувається швидко, надійно та безпомилково.

Один з найпоширеніших комунікаційних протоколів, який може бути використаний в такій системі, - це HTTP (Hypertext Transfer Protocol). HTTP використовується для передачі даних через мережу Інтернет та побудований на базі клієнт-серверної архітектури.

HTTP визначає формат запитів та відповідей між клієнтом та сервером. Клієнт (наприклад, веб-браузер або мобільний додаток) робить запит до сервера, а сервер обробляє цей запит та повертає відповідь з необхідними даними.

3.1.5 Обмеження пам’яті

Обсяг оперативної пам'яті, доступний для виконання програм та зберігання даних, може бути обмеженим. Залежно від розміру доступної пам'яті, система може вміщувати обмежену кількість користувачів та обробляти лише обмежену кількість даних.

Наявність дискового простору для зберігання даних також може бути обмеженою. Дисковий простір використовується для зберігання файлів, баз даних, резервних копій та інших ресурсів. Обмеження дискового простору може впливати на масштабність системи та можливість зберігання великих обсягів даних.

3.1.6 Операції

Користувачі можуть шукати події та заходи що їх цікавлять. Це включає визначення назви, типу, місця, часу та інших важливих атрибутів події.

Система може надавати можливості для створення компаніям або угрупуванням, що мають права на заходи, нові події та заходи. Також система надає можливість утворення звітності для організаторів.

3.2 Атрибути програмного продукту

3.2.1 Надійність

Надійність в програмній системі рекомендації та відбору актуальних подій з back-end реалізацією є важливою характеристикою, оскільки вона забезпечує безперебійну та стабільну роботу системи. Надійність означає, що система працює так, як очікується, і надійно обробляє запити користувачів, запобігаючи можливим збоям чи відмовам.

Для досягнення надійності, система повинна мати вбудований механізм для обробки помилок та винятків, які можуть виникати під час роботи. Це допомагає

виявляти та вирішувати проблеми, що впливають на роботу системи, забезпечуючи її стабільну та надійну функціональність.

Також важливо мати механізми резервного копіювання, що забезпечують збереження та захист даних системи. Це може включати регулярне резервне копіювання даних, створення резервних копій баз даних та інших важливих ресурсів, що гарантує можливість відновлення системи в разі втрати чи пошкодження інформації.

3.2.2 Доступність

Система повинна мати реактивний дизайн, який автоматично адаптується до різних типів пристроїв, таких як комп'ютери, планшети або смартфони. Це дозволяє користувачам зручно взаємодіяти з системою на будь-якому пристрої, незалежно від їхнього розміру екрану. Такий підхід сприяє покращенню зручності використання та задоволенню користувачів, забезпечуючи оптимальний інтерфейс на різних пристроях.

3.2.3 Безпека

Система повинна володіти механізмами аутентифікації для перевірки ідентичності користувачів та авторизації їх доступу до різних функцій системи. Це може включати використання паролів, рівнів доступу та управління правами користувачів.

З метою захисту конфіденційності даних, особливо під час їх передачі між клієнтом та сервером, використовується шифрування. Використання протоколів шифрування, таких як SSL (Secure Sockets Layer) або TLS (Transport Layer Security), сприяє забезпеченню безпеки під час передачі даних.

Система також повинна мати заходи безпеки для захисту від зламу та зловмисницьких атак, таких як введення шкідливого коду, SQL-ін'єкції, переповнення буфера тощо. Це може включати застосування механізмів фільтрації введення користувачів, валідацію даних та захист від переповнення, що сприяє забезпеченню високого рівня безпеки системи.

3.2.4 Супроводжуваність

Система повинна мати належну документацію, яка точно описує її архітектуру, конфігурацію, процедури установки, налаштування та супроводження. Це сприяє швидкому розумінню та вирішенню проблем розробниками та адміністраторами системи, а також дозволяє внесення необхідних змін.

Крім того, система повинна мати механізми для регулярних оновлень та встановлення патчів з метою виправлення відомих помилок, усунення вразливостей та покращення функціональності. Це забезпечує надійність та стійкість системи, а також дозволяє впровадження нових функцій та вдосконалень.

3.2.5 Переносимість

Переносимість в програмній системі, щодо рекомендацій та відбору актуальних подій у back-end реалізації, передбачає здатність системи працювати на різних платформах та середовищах без значних змін. Це досягається шляхом використання платформо-незалежних технологій, стандартизованих фреймворків та уникання глибокої залежності від конкретних платформ або СКБД. Такий підхід гарантує гнучкість та ефективність у розгортанні системи на різних середовищах без великих зусиль.

3.2.6 Продуктивність

Висока продуктивність у програмній системі рекомендації та відбору актуальних подій з реалізацією back-end є суттєвим компонентом і включає в себе оптимізацію коду, масштабування та ефективне використання ресурсів. Розробка коду, який оптимізовано виконує завдання, разом з масштабуванням системи для ефективного розподілу навантаження та правильного використання ресурсів, сприяє високій продуктивності системи та задоволенню потреб користувачів.

3.2.7 Вимоги бази даних

Для програмної системи рекомендації та підбору актуальних подій із back-end реалізацією вимоги до бази даних включають в себе належну структуру, нормалізацію даних та підтримку необхідних запитів та операцій. Забезпечення безпеки даних є пріоритетом і вимагає використання механізмів авторизації, шифрування та контролю доступу. Оптимізація бази даних для досягнення високої швидкодії та ефективності виконання запитів також є ключовою вимогою.

ДОДАТОК Г

Тези XXVIII міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті»

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ XXVIII МІЖНАРОДНОГО МОЛОДІЖНОГО ФОРУМУ

«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ»

16 – 18 квітня 2024 р.

Том 6

КОНФЕРЕНЦІЯ
«ІНФОРМАЦІЙНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ»
INFORMATION INTELLIGENT SYSTEMS

Харків 2024

28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., – Харків: ХНУРЕ. 2024. – 958 с.

У збірнику представлено матеріали доповідей учасників 28-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті».

Для науковців, викладачів, парктичних працівників, студентів, а також широкого кола читачів, які цікавляться цією проблематикою.

Відповідальність за зміст поданого матеріалу несе його автор.

Видання підготовлено факультетом комп'ютерних наук Харківського національного університету радіоелектроніки

61166, Україна, Харків, просп. Науки, 14
тел./факс: (057) 7021397

E-mail: mref21@nure.ua

ISBN ISBN 978-966-659-396-5
DOI 10.30837/IYF.IIS.2024

Програмний комітет конференції

Федорович О.Є.	д.т.н., проф., зав. каф. Комп'ютерних наук та інформаційних технологій (КНІТ), Національний аерокосмічний університет ім. М.Є. Жуковського "Харківський авіаційний інститут", Лауреат Державної премії України.
Субботін С.А.	д.т.н., проф., зав. каф. Програмних засобів, Запорізький національний технічний університет, Україна.
Петренко М.Г.	д.т.н., проф., Інститут кібернетики імені В.М. Глушкова НАН України.
Стасюк О.І.	д.т.н., проф. Державний економіко-технологічний університет транспорту, Україна.
Єрохін А.Л.	проф., декан ф-ту ХНУРЕ, м. Харків, Україна.
Філатов В.О.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Петров К.Е.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Дудар З.В.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Гребеннік І.В.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Дейнеко Ж.В.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.

УДК 004.89

EVENTIFY: ІННОВАЦІЙНИЙ ПІДХІД ДО ОРГАНІЗАЦІЇ ТА ПОШУКУ ЗАХОДІВ З ВИКОРИСТАННЯМ ШІ

Сафошин В. В., Хамінов І. О., Дегтяр В. Е.
Науковий керівник – к.т.н., доц. Побіженко І. О.
Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна
e-mail: volodymyr.safoshyn@nure.ua, illia.khaminov@nure.ua,
e-mail: vladyslav.dehtiar@nure.ua

In an era of rapid cultural evolution, the challenge of effectively organizing and selecting events that cater to individual preferences has become paramount. "Eventify" revolutionizes event organization in the digital age by leveraging artificial intelligence to tailor cultural event recommendations to individual preferences. This platform not only makes finding events effortless but also signifies a new chapter in personalized cultural experiences, ensuring users are matched with events that resonate with their interests. "Eventify" exemplifies the transformative potential of artificial intelligence [1] (AI) in enriching cultural engagement and streamlining the event selection process.

В епоху стрімкого розвитку культурного життя, актуальним стає питання ефективної організації та відбору заходів, які б відповідали особистим уподобанням кожної людини. Доступними рішеннями є різні телеграм канали, які повідомляють користувачів про різні події. В таких джерелах інформація про події не згрупована за видом та розташуванням, тож користувач може загубити цікаві йому події поміж багатьох інших. Також є різні сайти, що пропонують тільки специфічний спектр подій і не охоплюють всі вподобання користувача, що призводить до того що потрібно використовувати багато різних джерел. В цьому контексті сервіс "Eventify" пропонує революційний підхід, що базується на використанні передових технологій штучного інтелекту (ШІ) для аналізу великих обсягів даних про культурні події, їх учасників та переваги користувачів. Удосконалюючи свою місію щодо пошуку та організації культурних заходів, платформа "Eventify" залучатиме передові методи штучного інтелекту, які включають великі мовні моделі [2] (LLMs), ChatGPT[3].

Дослідження спрямоване на створення інноваційної програмної системи, яка за допомогою веб- та мобільного додатків пропонуватиме користувачам персоналізовані рекомендації щодо різноманітних заходів, виходячи з їхніх переваг. Це дозволить користуватися однією платформою для пошуку всіх видів культурних заходів, а не багатьма різними сторонніми ресурсами.

Впродовж дослідження аналогічний рішень нашого додатку, було виявлено те, що жоден з них не використовує ніякі засоби для оптимізації та комфорту у виборі культурних заходів. Враховуючи це, ми прийшли до

висновку, що використання штучного інтелекту стане ідеальним рішенням для покращення користувацького досвіду.

Великі мовні моделі представляють собою клас алгоритмів машинного навчання, здатних аналізувати та генерувати природну мову [4] з високим рівнем складності та точності. У контексті "Eventify" використовуються LLMs для обробки об'ємних текстових даних, пов'язаних із культурними заходами.

ChatGPT – це одна з передових LLMs, розроблена OpenAI. Ця модель базується на технології трансформерів і здатна генерувати природні відповіді в контексті діалогу з користувачем [5]. Під час розробки нашого додатку ми обрали цю LLM як ключовий компонент, оскільки він пропонує унікальні переваги для покращення користувацького досвіду у виборі культурних подій. ChatGPT відіграє ключову роль у нашому додатку, оскільки здатний до глибокого аналізу діалогів, що дозволяє вловлювати нюанси запитів користувачів та відповідати на них з надзвичайною точністю. Також він здатний до поглибленої взаємодії з користувачем, що буде сприяти більш детальному розумінню їхніх уподобань, що дозволяє надавати влучні рекомендації щодо заходів, які відповідають їхнім інтересам.

Отже, наш проект підкреслює значення інтеграції штучного інтелекту в області організації заходів, демонструючи, як сучасні технології можуть радикально змінити взаємодію між організаторами та учасниками. Також за результатами нашого дослідження було проаналізовано усі доступні рішення цієї проблеми та створено інноваційний підхід її вирішення шляхом розробки додатку “Eventify” з використанням передових технологій штучного інтелекту.

Список використаних джерел:

1. Barstow, David. "Artificial intelligence and software engineering." Exploring artificial intelligence. Morgan Kaufmann, 1988. 641-670.

2. Chang, Yupeng, et al. "A survey on evaluation of large language models." ACM Transactions on Intelligent Systems and Technology (2023).

3. Wu, Tianyu, et al. "A brief overview of ChatGPT: The history, status quo and potential future development." IEEE/CAA Journal of Automatica Sinica 10.5 (2023): 1122-1136.

4. Erdem, Erkut, et al. "Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning." Journal of Artificial Intelligence Research 73 (2022): 1131-1207.

5. Sharonova, N., Kyrychenko, I., Gruzdo, I., Tereshchenko, G. (2022) Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types CEUR Workshop Proceedings, 2022, 3171, pp. 16–26.

М

Н

О

П

Р

С

ДОДАТОК Д

Код для управління станом користувача за допомогою Redux Toolkit

```
1  import { createAction, createReducer } from "@reduxjs/toolkit";
2  import defaultImage from "../resources/default.png";
3  import { parseJwt } from "../clients/auth.client";
4  const storageName = 'auth';
5  const rolePath =
6   "http://schemas.microsoft.com/ws/2008/06/identity/claims/role"
7  let data = JSON.parse(localStorage.getItem(storageName));
8
9  if (data && Date.now() > data.tokenExpirationTime) {
10     console.error("Local Data is deleted");
11     data = null;
12 }
13 let userRole = "";
14
15 if (data?.tokenValue) {
16     const decodedToken = parseJwt(data?.tokenValue);
17 if (decodedToken) {
18     userRole =    decodedToken [rolePath] || "";
19     }
20 }
21 const initialState = {
22     user: {
23         email: data?.user?.email || '',
24         password: data?.user?.password || '',
25         img: data?.user?.img === null || data?.user?.img ===
26             undefined  || data?.user?.img === 0? defaultImage :
27             data?.user?.img,
28         userName: data?.user?.userName || '',
29         phoneNumber: data?.user?.phoneNumber || '',
30         id: data?.user?.id || '',
31         role: userRole,
32         settings: data?.user?.settings || '',
33         likedEvents: data?.user?.likedEvents || '',
34         tags: data?.user?.tags || '',
35     },
36     tokenValue: data?.tokenValue || '',
37     tokenExpirationTime: data?.tokenExpirationTime || 0,
38 };
39
40 export const login = createAction("LOGIN", (profile, token)=> {
41     const expirationTime = Date.now() + 3600000;
42     return {
43         payload: {
44             user: {
45                 ...profile,
46                 img: profile.img === null || profile.img ===
47                         undefined ? defaultImage : profile.img,
48                 role: parseJwt(token)[rolePath]
```

```
            },
            tokenValue: token,
            tokenExpirationTime: expirationTime,
        },
    };
});
export const logout = createAction("LOGOUT");
export const changeProfile=createAction("CHANGE_PROFILE",d)=>{
    return {
        payload: {
            user: {
            ...d,
            img: d.img === null || d.img === undefined ?
                 defaultImage : d.img,
            },
        },
    };
});
export const updateUserImage =
            createAction("UPDATE_USER_IMAGE", (imageUrl) => {
    return {
        payload: imageUrl,
    };
});
export default createReducer(initialState, (builder) => {
    builder
        .addCase(login, (state, action) => {
            state.user = action.payload.user;
            state.tokenValue = action.payload.tokenValue;
            state.tokenExpirationTime =
                action.payload.tokenExpirationTime;
            localStorage.setItem(storageName,
                                    JSON.stringify(state));
            })
        .addCase(logout, (state) => {
            state.user = null;
            state.tokenValue = null;
            state.tokenExpirationTime = null;
            localStorage.clear();
        })
        .addCase(changeProfile, (state, action) => {
            state.user = {...state.user,..action.payload.user};
            localStorage.setItem(storageName,
                                     JSON.stringify(state));
        })
        .addCase(updateUserImage, (state, action) => {
            if (state.user) {
                state.user.img = action.payload;
            }
            localStorage.setItem(storageName,
                                 JSON.stringify(state));
         });
    });
```