

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

ДЛЯ НАВІГАЦІЇ ПО МІСЦЕВИМ ПОДІЯМ

SOFTWARE FOR LOCAL

EVENTS NAVIGATION

Виконав(ла): студент(ка) 4 курсу, групи КНТ-112сп

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ГОЦЬКИЙ О.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ЗАЙКО Т.А.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГОЦЬКОГО Олександра Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для навігації по місцевим подіям. Software for Local Events Navigation

керівник проєкту (роботи) к.т.н., доцент, ЗАЙКО Тетяна Анатоліївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 03 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЗАЙКО Т.А., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “ 28 ” квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Олександр ГОЦЬКИЙ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна ЗАЙКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
82 с., 3 табл., 29 рис., 3 дод., 17 джерел.

JAVASCRIPT, NODEJS, ЗАСТОСУНОК, МОВА ПРОГРАМУВАННЯ,
ПОДІЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для навігації по місцевим подіям.

Предмет дослідження – програмні засоби для навігації по місцевим подіям.

Мета роботи – розробка програмного забезпечення для навігації по місцевим подіям.

Матеріали, методи та технічні засоби: мова програмування JavaScript, середовище NodeJS.

Результати. Розроблено проєктні рішення для створення програмного забезпечення для навігації по місцевим подіям. Створено програмне забезпечення для навігації по місцевим подіям. Проведено тестування розробленої програмної системи для навігації по місцевим подіям.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення для навігації по місцевим подіям за допомогою мови програмування JavaScript та середовища NodeJS.

Галузь використання – туризм.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 82 pages, 3 tables, 29 figures, 3 appendixes, 17 sources.

JAVASCRIPT, NODEJS, APPLICATION, PROGRAMMING
LANGUAGE, EVENT, SOFTWARE.

The object of research is the process of developing software for navigating local events.

The subject of the research is software for navigating local events.

The purpose of this work is to develop the software for navigating local events.

Materials, methods and technical tools: JavaScript programming language, NodeJS environment.

Results. Project solutions for the creation of software for navigating local events has been designed. The software for navigating local events has been developed. The developed software system for navigating local events has been tested.

Conclusions. The goal of the work has been achieved. The software tools for navigating local events using the JavaScript programming language and the NodeJS environment have been developed.

Scope of use – tourism.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Програмне забезпечення для навігації по місцевим подіям	11
1.2 Аналіз програмного забезпечення для навігації по місцевим подіям	21
1.3 Висновки за розділом 1	30
2 Матеріали і методи	32
2.1 Вибір мови програмування	32
2.2 Вибір середовища для створення програмного забезпечення для навігації по місцевим подіям	34
2.3 Висновки за розділом 2	37
3 Опис програми	38
3.1 Структура програмного забезпечення для навігації по місцевим подіям	38
3.2 Функціонування програмного забезпечення для навігації по місцевим подіям	41
3.3 Розробка бази даних програмного забезпечення для навігації по місцевим подіям	43
3.4 Проєктування інтерфейсу програмного забезпечення для навігації по місцевим подіям	45
3.5 Висновки за розділом 3	48
4 Експлуатація, тестування та експериментальне дослідження програми	50
4.1 Призначення й умови застосування програми	50
4.2 Характеристики програми для навігації по місцевим подіям	51
4.3 Інструкція по експлуатації програми	52
4.3.1 Звернення до програми	52
4.3.2 Вхідні й вихідні дані	52
4.3.3 Повідомлення	53

4.4 Виконання програмного забезпечення для навігації по місцевим подіям	53
4.5 Тестування програмного забезпечення для навігації по місцевим подіям	56
4.6 Висновки за розділом 4.....	57
Висновки.....	58
Перелік джерел посилань.....	61
Додаток А Технічне завдання.....	63
Додаток Б Фрагмент тексту програми	67
Додаток В Слайди презентації	76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

MVC – Model-View-Controller;

БД – база даних;

ПЗ – програмне забезпечення.

ВСТУП

Актуальність програмного забезпечення для навігації по місцевим подіям обумовлена сучасними технологічними та соціальними тенденціями. Швидкий ритм життя вимагає зручних інструментів для пошуку та організації дозвілля, а мобільні застосунки та вебсервіси забезпечують оперативний доступ до актуальної інформації [1], [2].

Програмне забезпечення для навігації по місцевим подіям є актуальним, оскільки зростає інтерес до локальних заходів, таких як культурні, спортивні та розважальні події. Сучасні програмні застосунки спрощують пошук необхідної інформації, купівлю квитків і бронювання місць, забезпечуючи користувачам максимальну зручність. Завдяки розвитку мобільних технологій і геолокації, такі програмні сервіси можуть пропонувати персоналізовані рекомендації на основі місцезнаходження та інтересів. Крім того, інтеграція з картами, календарями та соціальними мережами дозволяє легко планувати відвідування заходів і координувати зустрічі [1], [2].

Подібні програмні рішення мають попит серед організаторів подій, які використовують їх для залучення аудиторії, збору аналітики та ефективної реклами. Також такі програмні сервіси сприяють розвитку концепції розумного міста, допомагаючи оптимізувати мобільність і покращувати якість життя мешканців [3]-[5].

Завдяки штучному інтелекту та аналізу даних такі системи можуть не лише інформувати користувачів про події, а й пропонувати персоналізовані рекомендації відповідно до їхніх уподобань. Це підвищує рівень залученості аудиторії та сприяє розвитку локальної культури, бізнесу та соціальної активності [6], [7].

Проте деякі програмні застосунки для навігації по місцевим подіям перевантажені інформацією. Велика кількість подій, особливо в мегаполісах, може ускладнювати навігацію, а алгоритми персоналізації не завжди правильно враховують вподобання користувачів. Деякі застосунки або

вебсайти можуть мати складний інтерфейс або містити надлишкову рекламу, що знижує їхню ефективність. Ще одним недоліком є питання конфіденційності та безпеки. Такі сервіси часто використовують дані геолокації та можуть збирати особисту інформацію для аналізу поведінки користувачів, що створює ризики витоку або зловживання цими даними. Окрім цього, розробка, придбання та підтримка таких програмних сервісів потребує значних ресурсів, а деякі їх функції можуть бути доступні лише у платних версіях або за передплатою [1]-[7].

Тому актуальною є розробка програмного забезпечення для навігації по місцевим подіям. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для навігації по місцевим подіям.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для навігації по місцевим подіям;
- здійснити проєктування програмного забезпечення для навігації по місцевим подіям;
- створити програмне забезпечення для навігації по місцевим подіям;
- виконати тестування розробленого програмного забезпечення для навігації по місцевим подіям.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для навігації по місцевим подіям

Програмне забезпечення для навігації по місцевих подіях забезпечує користувачам доступ до актуальної інформації про заходи, що відбуваються у вибраному регіоні. Воно створене для зручного перегляду подій, полегшення комунікації між учасниками та організаторами, а також сприяння взаємодії всередині спільноти [1].

Такі програмні платформи дозволяють знайти цікаві заходи, швидко дізнаватися про нові події та оптимально планувати дозвілля. Завдяки інтегрованим системам оповіщень користувачі отримують персоналізовані рекомендації, що допомагають обирати найбільш релевантні події. Крім того, наявність зручних інструментів для бронювання квитків та реєстрації спрощує процес участі у заходах [1].

Розробка подібного програмного забезпечення відкриває широкі можливості для бізнесу. Індустрія організації подій постійно розвивається, і створення інноваційної платформи може стати вигідним рішенням, здатним привернути увагу як користувачів, так і партнерів. Вдалий підхід до реалізації такого проєкту дозволяє не лише сформувати ефективний сервіс, а й сприяти його подальшому масштабуванню [1].

Для успішної роботи програмного забезпечення для навігації по місцевих подіях необхідно враховувати ключові аспекти її функціональності. Важливу роль відіграє дизайн інтерфейсу, що має бути інтуїтивно зрозумілим та адаптованим для швидкої навігації. Оптимізація пошукових можливостей, включаючи фільтри за категоріями, місцем проведення та часом, робить платформу зручною для користувачів з різними інтересами [1].

Ефективна система взаємодії з соціальними мережами сприяє залученню нових учасників та розвитку спільноти. Можливість запрошувати друзів, обговорювати події та ділитися враженнями стимулює активність користувачів. Додатково запровадження програм лояльності, спеціальних

пропозицій та бонусних систем підвищує зацікавленість у використанні сервісу та сприяє його популяризації [1].

Розробка сучасної програмної платформи для пошуку місцевих заходів вимагає комплексного підходу та врахування потреб аудиторії. Інтеграція локальних платіжних рішень, автоматизованих рекомендаційних систем та інструментів аналітики допомагає підвищити ефективність сервісу, зробивши його більш конкурентоспроможним [1], [2].

Програмне забезпечення для навігації по місцевих заходах слугує платформою, яка об'єднує різних учасників сфери організацій подій та заходів, забезпечуючи ефективну взаємодію між ними. Такий цифровий інструмент надає можливість отримувати актуальну інформацію про події, організовувати зустрічі та координувати діяльність усіх сторін, задіяних у процесі. Його основна мета — спростити пошук заходів, полегшити доступ до квитків та покращити користувацький досвід завдяки інтегрованим функціям [1], [2].

Для відвідувачів важливо мати швидкий доступ до актуальної інформації про майбутні події, а також можливість зручно отримати деталі про місце проведення. Зручний інтерфейс дозволяє фільтрувати заходи за категоріями, місцем, датою чи форматом, що значно полегшує навігацію. Крім того, система повідомлень допомагає не забувати про заплановані зустрічі, своєчасно нагадуючи про них. Інтеграція з картографічними сервісами сприяє точному визначенню маршруту, що особливо корисно для нових локацій чи масштабних подій [1], [2].

Організатори, використовуючи програмне забезпечення для навігації по місцевих подіях, отримують можливість ефективно керувати заходами, розповсюджувати інформацію про них, координувати продаж квитків і забезпечувати комунікацію з аудиторією. Функціонал програмного забезпечення для навігації по місцевих подіях дозволяє структурувати процеси, запроваджувати маркетингові активності та аналізувати поведінку користувачів, що сприяє покращенню майбутніх подій [1], [2].

Оператор програмного забезпечення для навігації по місцевих подіях виконує роль посередника, що забезпечує якісну взаємодію між організаторами та учасниками. Його завдання — підтримувати стабільність системи, вдосконалювати функціонал та розширювати можливості для користувачів. Успішна реалізація такого програмного рішення відкриває перспективи розвитку ринку організації подій та заходів, створюючи зручний цифровий простір для всіх зацікавлених сторін [1], [2].

Програмне забезпечення для управління подіями допомагає координувати всі аспекти їхньої організації та забезпечує ефективний зв'язок між усіма учасниками процесу. Для успішного проведення заходу необхідно враховувати численні деталі, зокрема логістику, технічне оснащення, координацію команди та взаємодію з аудиторією. Використання цифрових інструментів дає змогу значно спростити ці завдання, дозволяючи організаторам зосередитися на створенні якісного контенту [1], [2].

Формування інформаційного простору для події починається зі створення профілю заходу. Для цього у програмному забезпеченні для навігації по місцевих подіях варто передбачити гнучкі механізми редагування, які дозволять оновлювати опис, змінювати візуальне оформлення та додавати важливі деталі в будь-який момент. Вбудовані можливості персоналізації сприяють кращій презентації заходу та приваблюють більшу кількість потенційних відвідувачів [1], [2].

Ефективні канали комунікації з аудиторією є невід'ємною частиною сучасних програмних засобів для навігації по місцевих подіях. Інтегровані засоби взаємодії дозволяють організаторам підтримувати контакт із користувачами через повідомлення, розсилки, рекомендації та нагадування. Використання інструментів для цільової реклами допомагає розширити охоплення та залучити нову аудиторію [1], [2].

Процес розповсюдження квитків також потребує зручних та безпечних рішень. Автоматизована система обробки замовлень як модуль програмного забезпечення для навігації по місцевих подіях спрощує керування продажами,

а можливість здійснення онлайн оплат підвищує комфорт користувачів. Запровадження комісійних зборів може слугувати джерелом прибутку для платформи, забезпечуючи її фінансову стабільність [1], [2].

Аналіз поведінки користувачів у програмному забезпеченні для навігації по місцевих подіях відіграє ключову роль у вдосконаленні якості сервісу. Дані про інтереси відвідувачів дозволяють організаторам краще розуміти очікування своєї аудиторії та коригувати майбутні заходи відповідно до їхніх потреб. Використання опитувань, зворотного зв'язку та інструментів моніторингу сприяє підвищенню рівня задоволеності учасників та вдосконаленню функціоналу платформи [1], [2].

Методи отримання доходу від програмного забезпечення для навігації по місцевих заходах можуть включати різноманітні механізми монетизації. Одним із найпоширеніших варіантів є стягнення плати за використання сервісу. Організатори можуть безкоштовно публікувати інформацію про свої події, а комісія стягується лише з продажу квитків. Ця комісія може бути фіксованою, відсотковою або комбінованою, залежно від політики платформи. Гнучка система тарифікації дозволяє адаптувати модель до особливостей різних регіонів та потреб користувачів [2], [3].

Ще одним джерелом прибутку є розміщення рекламного контенту. Програмне забезпечення, орієнтоване на місцеві події, може стати ефективним каналом для просування товарів і послуг. Використання рекламних блоків, інтеграція спеціальних повідомлень та персоналізованих рекомендацій дозволяє не лише підвищити впізнаваність брендів, а й створити додаткові можливості для взаємодії з аудиторією [2], [3].

Співпраця з місцевими компаніями відкриває додаткові можливості для розширення функціоналу та отримання прибутку. Інтеграція партнерських сервісів може включати спеціальні пропозиції для користувачів, бонусні програми або додаткові зручності, такі як бронювання транспорту чи замовлення послуг у межах платформи. Це не лише робить застосунок більш

корисним для відвідувачів, а й створює додаткову цінність для бізнесів, які отримують нових клієнтів [2], [3].

Процес створення програмного забезпечення, яке допомагатиме користувачам орієнтуватися серед місцевих заходів, вимагає ретельного планування та послідовного підходу. На початковому етапі важливо розібратися, наскільки актуальним є таке рішення для певного регіону. Дослідження допоможе зрозуміти, які джерела інформації про події є найпопулярнішими, чи існує реальний попит на подібний сервіс і які саме проблеми можуть бути вирішені за допомогою нового продукту. Аналіз конкурентних рішень дозволяє визначити їхні сильні та слабкі сторони, а також знайти унікальні можливості для покращення функціональності [2], [3].

Щоб створити ефективний інструмент у вигляді програмного забезпечення для навігації по місцевих заходах, необхідно зрозуміти, що саме очікують користувачі від такого застосунку. Досвід організаторів подій допоможе виявити, яких інструментів їм не вистачає для зручного керування заходами, а думки відвідувачів дозволять визначити, які функції зроблять використання програми більш привабливим. Узагальнення цієї інформації дає змогу чітко сформулювати концепцію проекту та його подальший розвиток [2], [3].

Розробка програмного забезпечення для навігації по місцевих подіях починається з мінімального набору можливостей, достатніх для першого етапу тестування. Головним завданням такого підходу є створення базового функціоналу, який дозволить користувачам переглядати доступні заходи, дізнаватися основну інформацію про них та швидко знаходити контакти організаторів. Використання цього підходу допомагає перевірити, наскільки продукт відповідає очікуванням аудиторії, та визначити напрямки для подальшого вдосконалення [2], [3].

Після завершення дослідження ринку та визначення проблеми, яку має вирішити програмне забезпечення для навігації по місцевим подіям, можна переходити до його безпосередньої розробки. Спершу формується концепція

зовнішнього вигляду та основної логіки роботи. Створення попереднього макета дає змогу оцінити, наскільки ідеї відповідають очікуванням користувачів і які аспекти потребують удосконалення. Після цього розпочинається детальне опрацювання інтерфейсу та взаємодії з користувачем, що забезпечує зручність і доступність функціоналу [2], [3].

Одним із ключових етапів розробки програмного забезпечення для навігації по місцевим подіям є вибір формату реалізації. Мобільні додатки забезпечують високу швидкість роботи та кращу інтеграцію з пристроями, але їхня розробка потребує врахування вимог операційних систем. Вебверсія надає можливість доступу з різних пристроїв без прив'язки до конкретної платформи, однак може мати певні обмеження у функціональності [3], [4].

Перед тим як зробити продукт у вигляді програмного забезпечення для навігації по місцевим подіям доступним для широкого використання, проводиться тестування, яке допомагає виявити можливі недоліки та усунути їх. Після завершення цього процесу програмне забезпечення публікується на відповідних платформах і стає доступним для користувачів. Щоб про продукт дізналося більше людей, необхідно забезпечити його ефективне просування. Робота над залученням користувачів має розпочатися ще до офіційного запуску і не припинятися після виходу на ринок. Використання різних каналів комунікації, таких як соціальні мережі, онлайн реклама, публікації у блогах та співпраця з впливовими особами, сприяє швидкому поширенню інформації [3], [4].

Враховуючи, що програмне забезпечення для навігації по місцевим подіям орієнтовано на дві основні групи – організаторів та відвідувачів, стратегія популяризації має охоплювати обидві аудиторії. Для залучення організаторів важливо забезпечити платформу, яка допоможе їм ефективно керувати подіями та взаємодіяти з учасниками. Якщо ж основний акцент робиться на відвідувачах, варто зосередитися на тому, щоб вони змогли легко знаходити цікаві заходи та отримувати актуальну інформацію. Поєднання обох підходів дозволяє створити активну екосистему, де всі учасники

отримують користь від взаємодії. Додаткові ідеї для розвитку можна запозичити з інших галузей, адаптуючи найкращі практики для потреб цього сегмента [3], [4].

Розробка програмної платформи для навігації по заходах передбачає створення інструменту, який допомагає користувачам легко знаходити та планувати відвідування місцевих подій. Така система дає змогу організаторам ділитися інформацією про майбутні заходи, а відвідувачам — реєструватися та отримувати актуальні оновлення [3], [4].

Основою цього підходу є інтеграція реєстраційної системи, що спрощує збір інформації про потенційних учасників. Гнучке налаштування форм для збору даних забезпечує організаторам можливість планувати логістику заходу відповідно до очікуваної кількості відвідувачів. Такий підхід допомагає уникнути недоліків у процесі планування та гарантує ефективну взаємодію між сторонами [3], [4].

Цифрові платформи для подій можуть охоплювати різноманітні формати. Ділові зустрічі, професійні конференції та освітні семінари потребують спеціальних інструментів для управління реєстрацією, розкладів та інтеграції з календарями. Розважальні заходи, такі як концерти, фестивалі та спортивні події, потребують зручних механізмів пошуку та купівлі квитків. Закриті або приватні події, що включають сімейні святкування чи корпоративні зустрічі, можуть потребувати обмеженого доступу до інформації та персоналізованого налаштування запрошень [4], [5].

Використання подібних платформ значно спрощує організаційні процеси. Можливість централізовано керувати списками учасників дозволяє уникнути плутанини та оперативно обробляти нові реєстрації. Автоматизація обліку учасників дає змогу своєчасно реагувати на зміни, а онлайн формат суттєво знижує витрати на друковані матеріали та традиційні методи запрошення [4], [5].

Крім того, такі сервіси можуть включати вбудовані маркетингові інструменти, що дозволяють збільшити охоплення аудиторії. Персоналізовані

сторінки подій, рекламні розсилки та інтеграція з соціальними мережами сприяють підвищенню обізнаності про заходи та залученню більшої кількості відвідувачів. Використання сучасних цифрових рішень забезпечує комфортний досвід як для організаторів, так і для учасників, роблячи процес планування подій більш ефективним та зручним [4], [5].

Створення програмного рішення для навігації по заходах передбачає впровадження інструментів, які спрощують процес планування та управління подіями. Функціональність таких платформ може відрізнятися, проте основні етапи організації залишаються подібними. Перед початком роботи необхідно визначити характер заходу. Від його специфіки залежить, чи буде доступний механізм реєстрації, чи потрібно впроваджувати можливість продажу квитків, додаткових опцій для відвідувачів або спеціальних умов для певних категорій учасників. Важливим аспектом є також наповнення інформаційного блоку. До нього входять назва події, категорія, місце проведення, часові межі, а також додатковий контент у вигляді зображень або детального опису [5], [6].

Окрему роль відіграє система реєстрації, яка дає змогу автоматизувати збір даних про потенційних учасників. Параметри форми реєстрації можуть змінюватися залежно від типу події, проте в будь-якому разі передбачають введення контактної інформації. У разі необхідності оплати за участь інтегрується система безпечних онлайн платежів, що гарантує збереження особистих даних користувачів [5], [6].

Для ефективного керування заходами варто враховувати кілька ключових моментів. Інтерфейс має бути зрозумілим, а процес реєстрації — максимально зручним, щоб уникнути складнощів при заповненні форм. Важливо надати можливість учасникам отримати оперативну підтримку у разі виникнення питань. Перед запуском реєстрації необхідно переконатися в коректності всіх деталей, перевірити працездатність усіх елементів та протестувати систему обробки заявок [5], [6].

Програмна платформа для навігації по місцевим подіям також може включати функції, що допомагають підвищити впізнаваність заходу.

Використання індивідуального оформлення, брендів кольорів та логотипів сприяє створенню цілісного візуального стилю, що підсилює довіру аудиторії. Завдяки сучасним технологіям управління подіями стає простішим, а навігація по заходах — зручнішою для кожного користувача [5], [6].

Організація заходів значно спрощується завдяки сучасним програмним рішенням, що дозволяють ефективно керувати інформацією та взаємодією з учасниками. Використання спеціалізованого програмного забезпечення для навігації по місцевих подіях допомагає не лише оптимізувати процес підготовки, а й створити зручний інструмент для відвідувачів, які шукають цікаві заходи поблизу [5], [6].

Вибір відповідного програмного рішення залежить від потреб користувачів та функціональних можливостей системи. Основна мета такого програмного забезпечення – забезпечити швидкий доступ до актуальної інформації про події, включаючи їхній формат, місце проведення та додаткові деталі. Крім того, інтеграція з картографічними сервісами дозволяє легко знаходити локації, а наявність реєстрації або бронювання місць робить участь у заходах зручнішою [5], [6].

Якісний сервіс для організації та пошуку подій має передбачати простий інтерфейс, що не потребує додаткових знань для навігації, а також підтримку безпечних способів оплати в разі необхідності. Окрему увагу варто приділяти персоналізації, щоб користувачі могли отримувати рекомендації відповідно до власних уподобань. Важливо також, щоб програмне забезпечення враховувало безпеку та конфіденційність даних, забезпечуючи захист персональної інформації [5], [6].

Гнучкість і можливість адаптації до різних форматів заходів дозволяють такому програмному забезпеченню залишатися актуальним для широкої аудиторії. Постійний моніторинг ефективності та аналіз відгуків допомагають покращувати функціональність і вдосконалювати користувацький досвід [5], [6].

Програмні системи, розроблені для відстеження місцевих заходів, надають можливість швидко та зручно знаходити актуальну інформацію про події, що відбуваються поблизу. Завдяки такому програмному забезпеченню можна отримати деталізовані дані про місце проведення, розклад, доступні квитки та додаткові послуги, пов'язані із заходом. Це дозволяє користувачам краще планувати свій час і не пропускати цікаві події [6], [7].

Оскільки кожен захід має власні особливості, не існує універсального рішення, яке однаково підходило б для всіх випадків. Система навігації має враховувати різноманітність форматів подій, включно з концертами, виставками, конференціями, спортивними змаганнями та локальними фестивалями. Окрім основних функцій пошуку та відстеження, такі інструменти часто включають можливості інтеграції з іншими цифровими сервісами, що допомагають організаторам ефективніше взаємодіяти з аудиторією [6], [7].

Функціональні можливості подібного програмного забезпечення залежать від потреб користувачів. Важливо передбачити доступ до системи бронювання місць, інтеграцію з платіжними сервісами для придбання квитків, можливість отримання сповіщень про зміни у розкладі або появу нових подій. Також корисним є надання карт із позначенням місць, де проходять заходи, а також рекомендації на основі інтересів користувача [6], [7].

Для організаторів подій такі платформи є важливим інструментом управління, що дозволяє контролювати кількість зареєстрованих учасників, надсилати запрошення, формувати електронні квитки та аналізувати відвідуваність [6], [7].

Визначено, що програмне забезпечення для навігації по місцевих подіях надає користувачам доступ до актуальної інформації про заходи, що відбуваються у вибраному регіоні. Такі програми забезпечують зручний перегляд подій, полегшення комунікації між учасниками та організаторами, а також сприяють взаємодії всередині спільноти, дозволяють знайти цікаві заходи, швидко дізнаватися про нові події та оптимально планувати дозвілля.

1.2 Аналіз програмного забезпечення для навігації по місцевим подіям

Проаналізуємо програмне забезпечення для навігації по місцевим подіям [8]–[13].

Програмне забезпечення Bandsintown (рис. 1.1) дозволяє легко знаходити концерти та виступи у своєму місті, отримуючи актуальну інформацію про заплановані заходи. Платформа дозволяє відстежувати виступи улюблених виконавців і гуртів, надсилаючи сповіщення про майбутні концерти, що відбуватимуться неподалік [9]–[11].

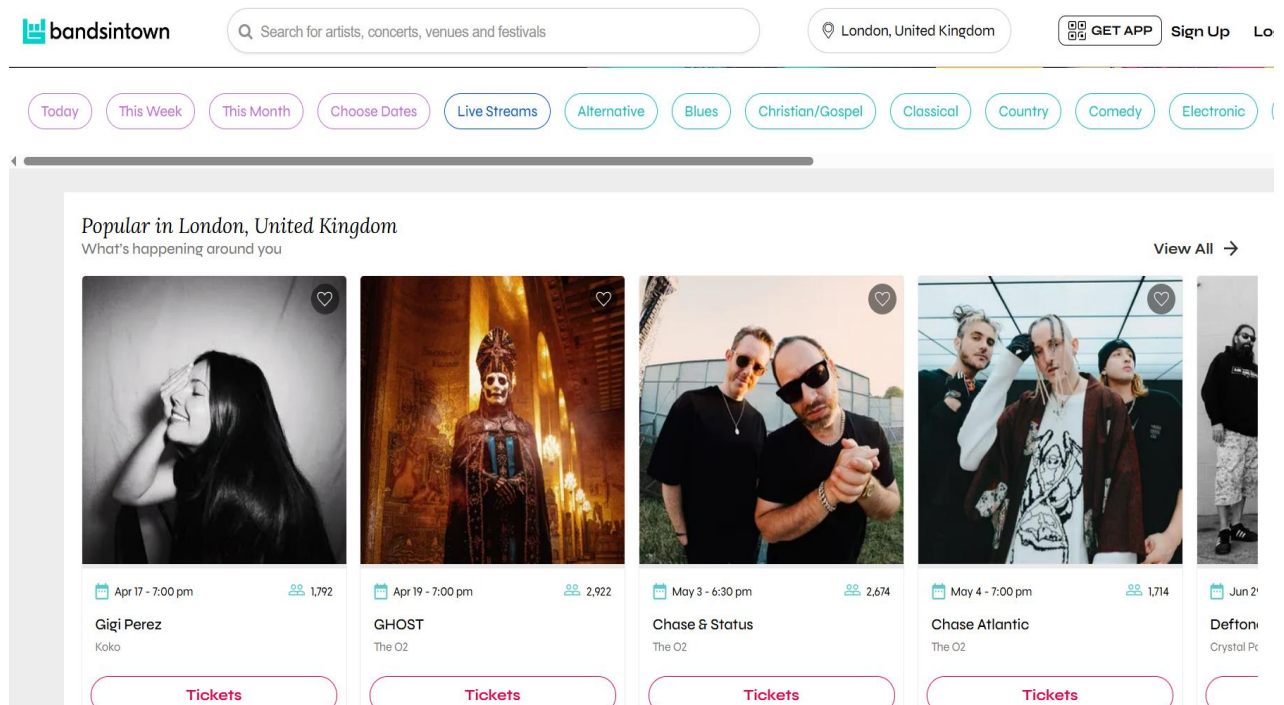


Рисунок 1.1 – Програмне забезпечення Bandsintown [11]

Для більш точних рекомендацій Bandsintown аналізує музичні вподобання, синхронізуючись з популярними стримінговими сервісами. Це гарантує, що користувачі завжди будуть у курсі подій, які відповідають їхнім інтересам. Крім переліку заходів, додаток також пропонує можливість бронювання квитків і збереження концертів у календарі, що спрощує планування відвідувань [9]–[11].

Bandsintown активно сприяє взаємодії між любителями музики, надаючи можливість залишати відгуки та коментарі до подій. Це не просто сервіс для пошуку концертів, а й платформа для спілкування з іншими шанувальниками. Додаток популярний серед незалежних музикантів, а також використовується багатьма відомими виконавцями для анонсування своїх турів [9]–[11].

За допомогою Bandsintown можна не лише слідкувати за концертами, а й запрошувати улюблених виконавців відвідати своє місто. Цей сервіс створено для того, щоб жодна важлива музична подія не залишилася поза увагою, а пошук живих виступів став зручним і швидким [9]–[11].

Програмне забезпечення Bandsintown має такі особливості [9]–[11]:

- зручний пошук концертів: програмне забезпечення Bandsintown дозволяє швидко знаходити музичні події у вибраному місті або регіоні, враховуючи індивідуальні вподобання користувача [9]–[11];

- персоналізовані рекомендації: програмне забезпечення Bandsintown аналізує історію прослуховувань у стримінгових сервісах і пропонує заходи, які відповідають музичним смакам [9]–[11];

- сповіщення про події: програмне забезпечення Bandsintown надсилає користувачам повідомлення, коли улюблені виконавці дають концерт поблизу, допомагаючи не пропустити виступ [9]–[11];

- інтеграція зі стримінговими сервісами: програмне забезпечення Bandsintown синхронізується з обліковими записами Spotify, Apple Music та інших платформ, підбираючи заходи відповідно до вподобань [9]–[11];

- можливість купівлі квитків: програмне забезпечення Bandsintown надає доступ до офіційних продавців квитків безпосередньо через додаток, що спрощує процес бронювання [9]–[11];

- календар подій: програмне забезпечення Bandsintown дозволяє додавати концерти до особистого розкладу, допомагаючи організувати відвідування заходів [9]–[11];

- взаємодія між користувачами: програмне забезпечення Bandsintown містить функції коментування та обговорення подій, створюючи активну спільноту меломанів [9]–[11];

- підтримка незалежних виконавців: програмне забезпечення Bandsintown дає можливість маловідомим артистам анонсувати свої виступи, розширюючи їхню аудиторію [9]–[11];

- запити на концерти: програмне забезпечення Bandsintown дозволяє фанатам надсилати запити на приїзд улюблених музикантів до їхнього міста, впливаючи на тури виконавців [9]–[11];

- багатоплатформність: програмне забезпечення Bandsintown доступне у вигляді мобільного застосунку для iOS та Android, а також має веб-версію для перегляду подій через браузер [9]–[11].

Серед недоліків програмного забезпечення Bandsintown можна відзначити обмежену підтримку менш популярних музичних жанрів і виконавців, через що користувачі можуть не завжди знаходити події, які відповідають їхнім специфічним уподобанням. Деякі користувачі скаржаться на неточність сповіщень, коли додаток рекомендує концерти, які не відповідають їхнім інтересам або відбуваються у віддалених містах. Інтеграція зі стримінговими сервісами іноді працює нестабільно, що може призводити до некоректного відображення персоналізованих рекомендацій. Також зустрічаються проблеми з відображенням оновлень розкладу концертів, що ускладнює своєчасне планування відвідувань заходів. Інтерфейс програми, хоча й зручний, може містити рекламні вставки або партнерські пропозиції, які не завжди є релевантними для користувачів [9]–[11].

Програмне забезпечення Meetup (рис. 1.2) створене для навігації по місцевим подіям, що надає користувачам можливість знаходити заходи та зустрічі відповідно до їхніх інтересів. Програмна платформа Meetup дозволяє долучатися до різних груп за інтересами, що сприяє зручному пошуку місцевих заходів [8], [9], [12].

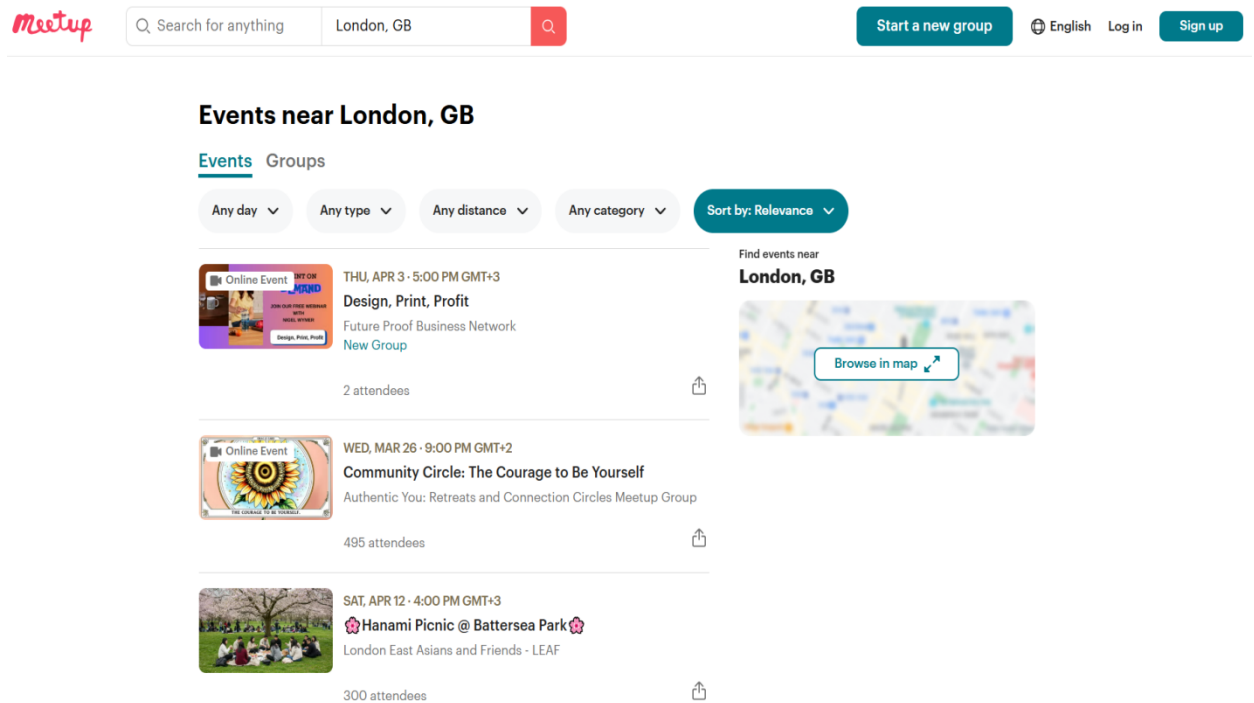


Рисунок 1.2 – Програмне забезпечення Meetup [12]

За допомогою цього програмного забезпечення Meetup можна легко відстежувати події, організовані спільнотами, і брати в них участь, що робить його ефективним інструментом для навігації по місцевим зустрічам та заходам. Програмне забезпечення Meetup забезпечує можливість пошуку подій за категоріями, зокрема за хобі та професійними інтересами, допомагаючи користувачам швидко знаходити відповідні зустрічі у своєму місті. Особливістю є акцент на спільнотах, що дозволяє не лише брати участь у заходах, а й створювати власні події та об'єднувати людей із подібними вподобаннями. Такий підхід робить Meetup ключовим інструментом для навігації по місцевим заходам, надаючи можливість як знаходити події, так і створювати нові активності [8], [9], [12].

Завдяки своєму функціоналу Meetup є одним із найпопулярніших сервісів для навігації по місцевим подіям. Додаток охоплює велику кількість груп по всьому світу, де організовуються різноманітні зустрічі, починаючи від неформальних заходів і закінчуючи професійними семінарами. Користувачам доступний зручний інструмент для перегляду місцевих подій та пошуку активностей у межах своєї спільноти [8], [9], [12].

Основний акцент зроблено на локальних подіях, що дозволяє людям знаходити заходи у своєму місті або регіоні, а також отримувати інформацію про нові події, які можуть їх зацікавити. Meetup виконує важливу функцію для тих, хто хоче орієнтуватися в місцевих заходах, відкриваючи доступ до широкого спектра подій у різних сферах життя [8], [9], [12].

Програмне забезпечення Meetup має такі особливості [8], [9], [12]:

- локальна навігація по подіях: програмне забезпечення Meetup дозволяє знаходити зустрічі та заходи, що проводяться у вибраному регіоні, допомагаючи користувачам швидко дізнаватися про актуальні події у своїй місцевості [8], [9], [12];

- створення та управління групами: програмне забезпечення Meetup надає користувачам можливість організовувати спільноти за інтересами, керувати подіями, запрошувати учасників і координувати активності у межах платформи [8], [9], [12];

- персоналізовані рекомендації: Meetup аналізує інтереси користувача та пропонує події та групи, що найбільше відповідають його вподобанням, підвищуючи зручність використання [8], [9], [12];

- багатофункціональна система сповіщень: програмне забезпечення Meetup надсилає нагадування про майбутні зустрічі, зміни в розкладі та нові події, що відповідають обраним категоріям [8], [9], [12];

- обмежений безкоштовний функціонал: для організаторів подій існують певні обмеження у безоплатній версії, що вимагає оформлення підписки для доступу до розширених можливостей [8], [9], [12];

- інтеграція з календарями: програмне забезпечення Meetup підтримує синхронізацію з Google Calendar та іншими сервісами для зручного відстеження запланованих заходів і кращого управління власним розкладом користувача [8], [9], [12];

- можливість онлайн заходів: окрім фізичних зустрічей, програмне забезпечення Meetup забезпечує підтримку віртуальних подій, що дозволяє організовувати вебінари, лекції та інші онлайн активності [8], [9], [12];

- соціальна взаємодія: програмне забезпечення Meetup сприяє встановленню нових знайомств, розвитку професійних і дружніх контактів, об'єднуючи людей зі спільними інтересами [8], [9], [12];

- доступність у різних країнах: програмне забезпечення Meetup пропонує широкий вибір подій у багатьох регіонах світу, що дозволяє знаходити цікаві заходи під час подорожей чи переїзду [8], [9], [12].

Серед недоліків програмного забезпечення Meetup можна відзначити обмежений доступ до функціоналу без оформлення платної підписки, що може створювати незручності для організаторів подій та звужувати можливості взаємодії в межах платформи. Навігація по місцевим подіям не завжди є інтуїтивно зрозумілою, особливо для нових користувачів, оскільки інтерфейс містить багато налаштувань і вкладених меню. Окрім цього, інколи трапляються проблеми з оновленням інформації про події, що може призводити до відображення застарілих або скасованих заходів. Ще однією проблемою є відсутність інтеграції з деякими популярними соціальними мережами, що ускладнює поширення інформації про події та залучення нових учасників [8], [9], [12].

Програмне забезпечення Eventbrite (рис. 1.3) спеціалізується на навігації по місцевим подіям, надаючи користувачам доступ до широкого спектра заходів, що відбуваються навколо. Завдяки йому можна легко знайти концерти, фестивалі, виставки, майстер-класи та інші події, орієнтуючись на власні інтереси та розташування [8], [9], [13]. Програма Eventbrite дозволяє отримувати актуальну інформацію про заходи, включаючи місце проведення, розклад та доступні опції квитків. Використання геолокації дає змогу швидко переглядати заходи у вибраному регіоні або здійснювати пошук у будь-якому іншому місті. Передбачено можливість збереження цікавих подій у власному акаунті, що спрощує планування вільного часу [8], [9], [13].

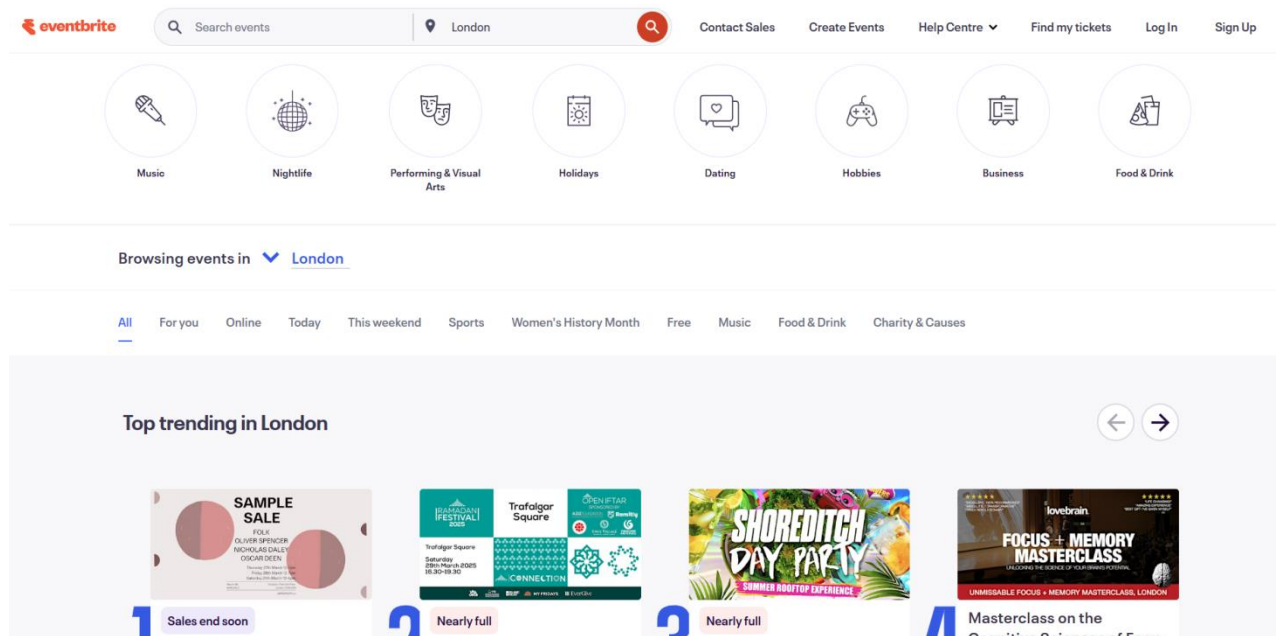


Рисунок 1.3 – Програмне забезпечення Eventbrite [13]

Система пошуку програмного забезпечення Eventbrite налаштована так, щоб користувачі могли знаходити лише актуальні для них події, застосовуючи фільтри за тематиками, датами та іншими параметрами. Вбудований механізм рекомендацій допомагає знаходити заходи, що відповідають вподобанням. Доступна функція обміну подіями, що дозволяє ділитися знайденими заходами з друзями [8], [9], [13].

Програма Eventbrite також підтримує можливість бронювання та придбання квитків безпосередньо через інтерфейс, спрощуючи доступ до обраних заходів. Таким чином, Eventbrite слугує не лише навігатором по місцевих подіях, а й зручним інструментом для планування дозвілля, допомагаючи користувачам завжди залишатися в курсі найцікавіших подій у своєму місті [8], [9], [13].

Основними характеристиками програмного забезпечення Eventbrite є такі [8], [9], [13]:

– призначення: програмне забезпечення Eventbrite створено для навігації по місцевим подіям, що дозволяє знаходити, переглядати та бронювати заходи в різних категоріях [8], [9], [13];

- пошук подій: програмне забезпечення Eventbrite підтримує можливість фільтрації заходів за категоріями, датами, локаціями та ключовими словами, що допомагає швидко знаходити цікаві події [8], [9], [13];
- персоналізовані рекомендації: алгоритми програмного забезпечення Eventbrite аналізують інтереси користувача та пропонують заходи, які можуть відповідати його вподобанням [8], [9], [13];
- бронювання квитків: програмне забезпечення Eventbrite надає функцію купівлі квитків безпосередньо через застосунок, що робить процес зручним та швидким [8], [9], [13];
- інтеграція з картами: для зручності користувачів програмне забезпечення Eventbrite надає можливість перегляду розташування подій на карті, що спрощує планування маршруту [8], [9], [13];
- взаємодія з друзями: програмне забезпечення Eventbrite підтримує функцію обміну подіями через соціальні мережі або месенджери, що сприяє спільному відвідуванню заходів [8], [9], [13];
- управління заходами: організатори можуть створювати події, налаштовувати параметри квитків і відстежувати реєстрації безпосередньо через платформу [8], [9], [13];
- підтримка різних пристроїв: програмне забезпечення Eventbrite доступне для мобільних платформ і вебверсії, що забезпечує його гнучкість у використанні [8], [9], [13];
- можливість збереження подій: користувачі програмного забезпечення Eventbrite можуть додавати заходи в обране, щоб мати до них швидкий доступ у майбутньому [8], [9], [13];
- сповіщення про події: у програмному забезпеченні Eventbrite реалізована система нагадувань і повідомлень про майбутні заходи, що допомагає не пропустити важливі події [8], [9], [13].

Серед недоліків програмного забезпечення Eventbrite можна відзначити залежність від доступу до Інтернету, без якого неможливо переглядати актуальні події та здійснювати бронювання квитків. Також слід звернути увагу

на те, що деякі заходи можуть бути представлені з обмеженою інформацією або без детальних відгуків, що ускладнює вибір. У певних випадках комісія за купівлю квитків може бути досить високою, що впливає на загальну вартість відвідування події. Окрім того, для користувачів, які віддають перевагу менш популярним заходам, вибір може бути обмеженим, оскільки більшість подій у системі зосереджені на масових заходах [8], [9], [13].

Порівняльну характеристику програмного забезпечення для навігації по місцевим подіям наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для навігації по місцевим подіям

Критерій порівняння	Bandsintown [11]	Meetup [12]	Eventbrite [13]
Можливість створення власних заходів	—	+	+
Зручність інтерфейсу	+	+	+—
Рекомендації на основі інтересів	+	+	+—
Можливість купівлі квитків	+	—	+
Наявність реклами та партнерських пропозицій	+—	+—	+
Фокус на професійних подіях	—	+—	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для навігації по місцевим подіям. Проте деякі програмні засоби для навігації по місцевим подіям можуть бути перевантажені інформацією. Велика кількість подій, особливо в мегаполісах, може ускладнювати навігацію, а алгоритми персоналізації не завжди правильно враховують вподобання користувачів. Деякі застосунки або вебсайти можуть мати складний інтерфейс або містити надлишкову рекламу,

що знижує їхню ефективність. Ще одним недоліком є питання конфіденційності та безпеки. Такі сервіси часто використовують дані геолокації та можуть збирати особисту інформацію для аналізу поведінки користувачів, що створює ризики витоку або зловживання цими даними. Окрім цього, розробка, придбання та підтримка таких програмних сервісів потребує значних ресурсів, а деякі їх функції можуть бути доступні лише у платних версіях або за передплатою. Тому актуальною є розробка програмного забезпечення для навігації по місцевим подіям.

При розробці програмного забезпечення для навігації по місцевим подіям необхідно забезпечити такі функціональні вимоги:

- підтримка можливості подання користувачу списку місцевих подій (локальних заходів, що відбудуться у відповідному регіоні);
- можливість надання інформації про кожну подію, зокрема, інформації про місце події та вартість участі у події;
- можливість відображення місця проведення заходу на карті;
- можливість фільтрації подій за типом або датою;
- підтримка можливості реєстрації та авторизації користувачів;
- підтримка можливості зареєстрованих користувачів додавання інформації про нові події.

1.3 Висновки за розділом 1

Визначено, що програмне забезпечення для навігації по місцевих подіях надає користувачам доступ до актуальної інформації про заходи, що відбуваються у вибраному регіоні. Такі програми забезпечують зручний перегляд подій, полегшення комунікації між учасниками та організаторами, а також сприяють взаємодії всередині спільноти, дозволяють знайти цікаві заходи, швидко дізнаватися про нові події та оптимально планувати дозвілля.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для навігації по місцевим подіям.

Проте деякі програмні засоби для навігації по місцевим подіям можуть бути перевантажені інформацією. Велика кількість подій, особливо в мегаполісах, може ускладнювати навігацію, а алгоритми персоналізації не завжди правильно враховують вподобання користувачів. Деякі застосунки або вебсайти можуть мати складний інтерфейс або містити надлишкову рекламу, що знижує їхню ефективність. Ще одним недоліком є питання конфіденційності та безпеки. Такі сервіси часто використовують дані геолокації та можуть збирати особисту інформацію для аналізу поведінки користувачів, що створює ризики витоку або зловживання цими даними. Окрім цього, розробка, придбання та підтримка таких програмних сервісів потребує значних ресурсів, а деякі їх функції можуть бути доступні лише у платних версіях або за передплатою. Тому актуальною є розробка програмного забезпечення для навігації по місцевим подіям.

Сформульовано функціональні вимоги до програмного забезпечення для навігації по місцевим подіям.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для навігації по місцевим подіям було використано мову програмування JavaScript [14], [15].

JavaScript є однією з найпоширеніших мов програмування, що активно використовується для створення вебресурсів, мобільних додатків і серверних рішень. Її універсальність забезпечує можливість розробки на одній мові як клієнтської частини, так і серверної частини, що досягається завдяки середовищу виконання Node.js [14], [15].

Висока продуктивність і зручність роботи з асинхронними запитами дозволяють легко взаємодіяти з базами даних, API та іншими зовнішніми ресурсами. Завдяки великій кількості бібліотек і фреймворків, серед яких React, Angular та Vue.js, процес створення додатків значно прискорюється. Використання JavaScript у мобільній розробці, зокрема через React Native та Ionic, забезпечує кросплатформну підтримку, а для настільних додатків застосовується Electron [14], [15].

Ця мова дає можливість інтегрувати картографічні сервіси, такі як Google Maps та OpenStreetMap, що є особливо важливим для створення програмного забезпечення, орієнтованого на навігацію та локальні події. Завдяки оптимізованій роботі з мережею забезпечується швидкий обмін даними, що особливо корисно для відображення актуальної інформації про заходи [14], [15].

Підтримка єдиного коду для вебверсій і мобільних застосунків значно спрощує процес розробки та скорочує витрати. Використання JavaScript гарантує стабільну продуктивність і сумісність із широким спектром пристроїв. Завдяки великій спільноті розробників доступні численні інструменти, що прискорюють створення програмного забезпечення та розширюють можливості його використання [14], [15].

Інтеграція з серверними технологіями, такими як Node.js, дозволяє обробляти великі обсяги даних у реальному часі, що особливо важливо для додатків, які забезпечують швидкий доступ до інформації про місцеві події. JavaScript є оптимальним вибором для розробки таких систем, оскільки забезпечує ефективну взаємодію між клієнтською та серверною частинами, підтримує масштабованість і надає гнучкі можливості для подальшого розвитку програмного забезпечення [14], [15].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для навігації по місцевим подіям наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для навігації по місцевим подіям

Критерій порівняння мов програмування	Мова програмування		
	JavaScript	Python	Java
Асинхронність та робота в реальному часі	+	+—	+—
Інтерактивність	+	—	+—
Висока продуктивність	+	+—	+
Зручність для створення ПЗ для навігації по місцевим подіям	+	—	—
Велика спільнота	+	+	+—

Отже, для реалізації програмного забезпечення для навігації по місцевим подіям обрано мову програмування JavaScript через її універсальність і широкі можливості у веброботі. JavaScript дозволяє створювати інтерактивні вебзастосунки, що працюють безпосередньо в браузері. Крім того, JavaScript також забезпечує легку інтеграцію з іншими технологіями, включаючи серверні рішення на Node.js, що дозволяє

створювати повноцінні динамічні застосунки з гнучкою обробкою даних на стороні сервера.

2.2 Вибір середовища для створення програмного забезпечення для навігації по місцевим подіям

В якості середовища для створення програмного забезпечення для навігації по місцевим подіям було обрано середовище Node.js [16], [17].

Node.js є середовищем виконання для JavaScript, що ґрунтується на рушії Chrome V8. Його архітектура використовує неблокуючу, подієво-орієнтовану модель введення-виведення, що забезпечує ефективність та швидкість роботи. Завдяки наявності однієї з найбільших екосистем з відкритим кодом, можливості використання значно розширюються [16], [17].

Це середовище створене для розробки вебзастосунків, які працюють в асинхронному режимі. Взаємодія з підключеннями відбувається за допомогою механізму подій, що дозволяє системі залишатися легкою та масштабованою навіть при обробці великої кількості запитів. На відміну від традиційних підходів, які застосовують потокову обробку операційною системою, цей метод є простішим і продуктивнішим. Вузли не потребують додаткових витрат на управління потоками, що дозволяє створювати розподілені системи з високою пропускнуою здатністю [16], [17].

До появи цього середовища JavaScript виконувався виключно у браузерях. Запуск Node.js відкрив нові можливості, дозволивши виконувати код JavaScript не лише у браузерному середовищі, а й на серверах, мобільних пристроях, настільних системах і навіть на мікроконтролерах. Це дало змогу використовувати спільний код на різних платформах, що значно спростило розробку та пришвидшило випуск програмного забезпечення [16], [17].

Вплив цього середовища помітний у мобільній розробці, де стало можливим створювати додатки для iOS та Android з використанням єдиного коду. Однак певні функції, характерні для конкретних операційних систем, не

реалізовані за замовчуванням, тому застосовуються нативні модулі. Це рішення забезпечує високу продуктивність, а також гнучкість у розробці мобільних застосунків, що використовують React Native. Такий підхід дозволяє командам зосередитися на єдиному стеку технологій, спрощуючи процес підтримки та оновлення програмного забезпечення [16], [17].

Node.js надає можливість виконання JavaScript у середовищі реального часу, використовуючи високопродуктивний рушій Google V8, що забезпечує швидку обробку коду. Завдяки асинхронній і подієво-орієнтованій архітектурі забезпечується ефективна оптимізація запитів, що є ключовим для серверних програм, API та додатків із миттєвим оновленням даних [16], [17].

Вбудована система модулів надає великий вибір бібліотек, що значно спрощує процес створення програмних рішень. Це середовище підходить для розробки різноманітних застосунків, включаючи сервери, різні архітектури, системи обробки даних та автоматизовані інструменти [16], [17].

Використання спільної мови JavaScript як для серверної, так і для клієнтської частини усуває необхідність перемикатися між різними технологіями, що значно полегшує роботу розробників. Його популярність зумовлена простотою, зручністю та широкими можливостями масштабування. Ця платформа відповідає вимогам сучасних вебзастосунків щодо високої продуктивності та швидкої реакції на запити користувачів [16], [17].

Висока швидкість обробки досягається завдяки оптимізованій обробці коду рушієм V8, що особливо важливо при виконанні тривалих обчислень або обробці великої кількості одночасних підключень. Використання неблокуючої архітектури разом із механізмом подій дозволяє забезпечити ефективний обмін даними без затримок, що є особливо корисним у системах, де критичною є швидкість оновлення інформації, наприклад у відстеженні місцеперебування або динамічному оновленні контенту [16], [17].

Завдяки підтримці сокетів спрощується реалізація миттєвого обміну даними між клієнтом і сервером, що дозволяє забезпечувати інтерактивність у додатках, які вимагають постійного оновлення інформації в реальному часі.

Ще одним вагомим аргументом на користь цієї технології є можливість використання єдиного технологічного стеку, що прискорює процес створення програмного забезпечення. Це не лише зменшує витрати часу на навчання, а й полегшує інтеграцію різних компонентів [16], [17].

Широкий набір бібліотек у охоплює рішення для роботи з географічними даними, базами даних, API та іншими сервісами, що значно прискорює розробку та розширює можливості взаємодії з іншими системами. Таким чином, Node.js забезпечує швидке виконання запитів, зручність у розробці та легкість інтеграції, що робить його ефективним вибором для створення масштабованих вебзастосунків [16], [17].

Обґрунтування вибору середовища для створення програмного забезпечення для навігації по місцевим подіям наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища для створення програмного забезпечення для навігації по місцевим подіям

Критерій порівняння середовищ	Середовища		
	Node.js	Deno	Electron
Простота розробки	+	+—	—
Гнучкість у розгортанні	+	+—	+—
Асинхронна робота	+	+	—
Підтримка картографічних сервісів	+	+—	—
Швидкість розробки	+	+	+—

Таким чином, для створення програмного забезпечення для навігації по місцевим подіям обрано середовище Node.js через його продуктивність, масштабованість і підтримку асинхронної обробки даних. Node.js забезпечує високу швидкість виконання коду, особливо при роботі з великою кількістю

запитів у реальному часі. Однією з ключових переваг Node.js є неблокуюча архітектура та використання подійного циклу, що дозволяє ефективно обробляти численні одночасні запити, наприклад, при оновленні інформації про події чи зміні геолокації користувачів. Це особливо важливо для інтерактивних сервісів, що потребують швидкого обміну даними між клієнтами та сервером.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для навігації по місцевим подіям обрано мову програмування JavaScript через її універсальність і широкі можливості у веброзробці. JavaScript дозволяє створювати інтерактивні вебзастосунки, що працюють безпосередньо в браузері. Крім того, JavaScript також забезпечує легку інтеграцію з іншими технологіями, включаючи серверні рішення на Node.js, що дозволяє створювати повноцінні динамічні застосунки з гнучкою обробкою даних на стороні сервера.

Для створення програмного забезпечення для навігації по місцевим подіям обрано середовище Node.js через його продуктивність, масштабованість і підтримку асинхронної обробки даних. Node.js забезпечує високу швидкість виконання коду, особливо при роботі з великою кількістю запитів у реальному часі. Однією з ключових переваг Node.js є неблокуюча архітектура та використання подійного циклу, що дозволяє ефективно обробляти численні одночасні запити, наприклад, при оновленні інформації про події чи зміні геолокації користувачів. Це особливо важливо для інтерактивних сервісів, що потребують швидкого обміну даними між клієнтами та сервером.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для навігації по місцевим подіям

Структуру програмного забезпечення для навігації по місцевим подіям наведено на рис. 3.1.



Рисунок 3.1 – Структура програмного забезпечення для навігації по місцевим подіям

Програмне забезпечення для навігації по місцевим подіям має чітко визначену архітектуру, яка включає п'ять основних модулів (серверний модуль, модуль конфігурації, модуль роботи з даними, модуль проміжного програмного забезпечення, модуль інтерфейсу користувача), кожен з яких відіграє ключову роль у функціонуванні системи, а також каталог додаткових бібліотек.

Серверний модуль відповідає за запуск вебзастосунку, визначення доступних ресурсів та інтеграцію GraphQL-сервісу у серверну частину.

Модуль конфігурації містить усі необхідні параметри для підключення до бази даних, управління авторизацією та збереження конфіденційної інформації.

Модуль роботи з даними забезпечує підключення до сховища даних, виконує операції запису, читання та оновлення інформації про події. Також цей модуль містить визначення структури збережених даних, що використовуються у застосунку, та забезпечує їх мапування у формат MongoDB.

Модуль проміжного програмного забезпечення реалізує логіку попередньої обробки HTTP-запитів перед їх передачею до основних обробників застосунку.

Модуль інтерфейсу користувача забезпечує взаємодію з користувачем програмного забезпечення для навігації по місцевим подіям. Також цей модуль включає опис схем даних та механізми обробки запитів, що дозволяє користувачам отримувати та змінювати інформацію за допомогою GraphQL-запитів.

Крім цього, програмне забезпечення для навігації по місцевим подіям використовує додаткові бібліотеки, які зберігаються у спеціальному каталозі, що містить всі встановлені залежності. Це забезпечує швидке налаштування середовища та зручність у підтримці проєкту.

Центральним елементом програмного забезпечення для навігації по місцевим подіям виступає основний файл програми, що відповідає за

ініціалізацію вебзастосунку, визначення доступних ресурсів для обробки запитів та інтеграцію GraphQL-сервісу у вебсервер.

Окремо зберігаються налаштування проєкту, включаючи перелік необхідних бібліотек та команд для розгортання та розробки, що забезпечує узгоджене середовище для роботи з кодом. Конфігураційні параметри програмного забезпечення для навігації по місцевим подіям, такі як параметри з'єднання з базою даних та секретні ключі авторизації, зосереджені у відповідному файлі, що відповідає за керування доступом і безпекою.

Для взаємодії з базою даних у відповідному модулі міститься логіка підключення до сховища даних. Усі операції збереження, читання та обробки інформації виконуються у цьому компоненті, що дозволяє ефективно організувати доступ до даних.

Система обробки запитів використовує проміжне програмне забезпечення, яке працює у зв'язці з Express. Завдяки йому вхідні запити проходять попередню обробку перед тим, як вони потрапляють до основної бізнес-логіки застосунку.

Реалізація структури даних базується на окремому модулі, де зберігаються всі моделі, необхідні для роботи з MongoDB. У цьому компоненті відбувається мапування інформації, що дозволяє взаємодіяти з базою даних через ORM-рівень.

Ще одним важливим елементом є система управління GraphQL-запитами, яка складається з окремих файлів для опису схем та механізмів обробки. Визначення типів даних та їхньої валідації відбувається у спеціальному модулі, тоді як обробка запитів реалізована у іншій частині системи, що відповідає за виконання відповідних функцій відповідно до запитів користувачів.

Таким чином, структура програмного забезпечення створює зручну, масштабовану та ефективну систему для навігації по подіях у режимі реального часу.

3.2 Функціонування програмного забезпечення для навігації по місцевим подіям

Функціонування програмного забезпечення для навігації по місцевим подіям подамо за допомогою схеми, зображеної на рис. 3.2.

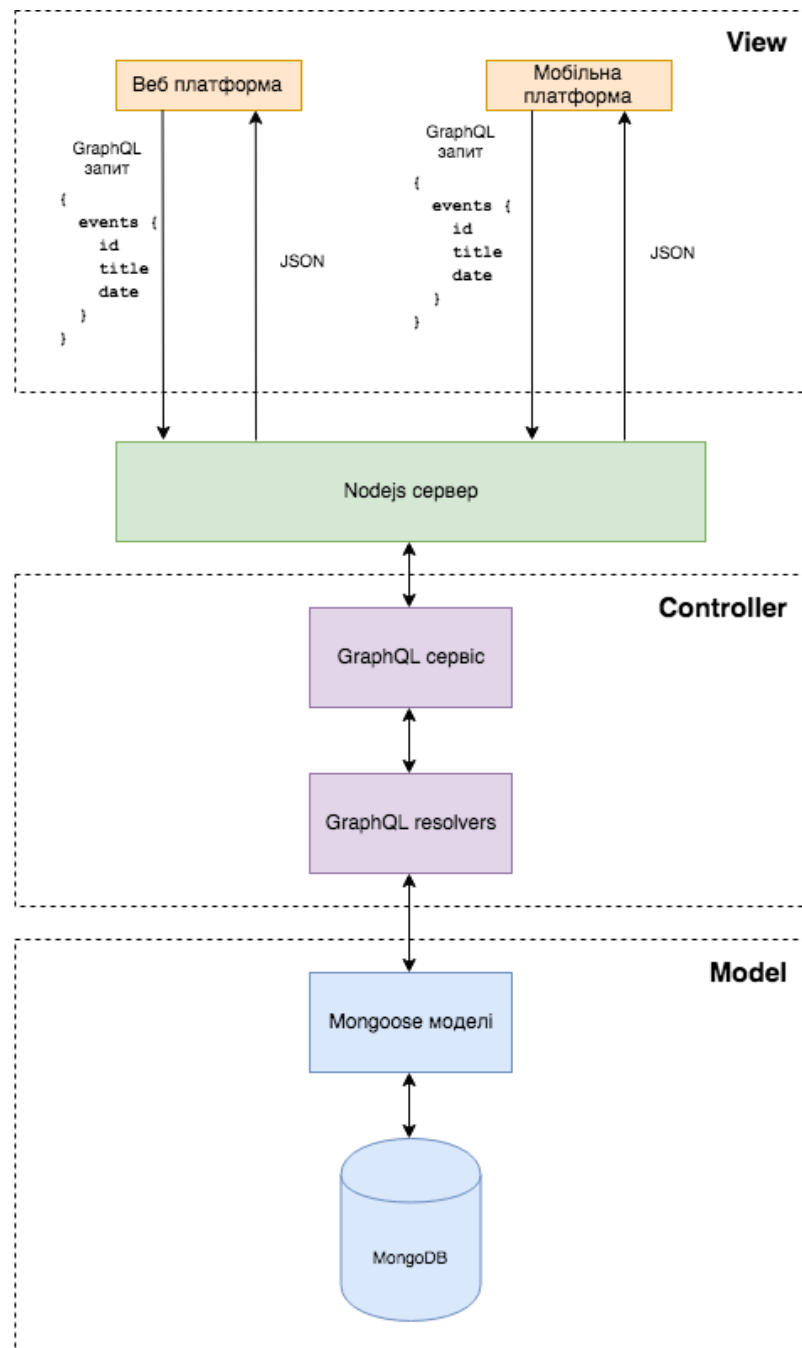


Рисунок 3.2 – Схема функціонування програмного забезпечення для навігації по місцевим подіям

Функціонування програмного забезпечення для навігації по місцевим подіям базується на архітектурному підході Model-View-Controller (MVC), що забезпечує чіткий розподіл відповідальностей між основними компонентами.

Основний принцип роботи полягає в тому, що логіка програми відокремлюється від користувацького інтерфейсу, що дозволяє ефективно керувати даними та взаємодією з користувачем. В основі структури лежать три ключові елементи: модель, представлення та контролер.

Модель відповідає за управління інформацією, забезпечує доступ до бази даних та здійснює обробку запитів на збереження чи зміну даних. Вона не взаємодіє з інтерфейсом користувача безпосередньо, а лише зберігає та обробляє інформацію.

Представлення виконує функцію візуалізації даних, що надаються додатком. Воно не містить бізнес-логіки, а лише відображає отриману інформацію та передає дії користувача далі для обробки. Веб- або мобільні клієнти можуть взаємодіяти із застосунком через Application Programming Interface (API), надсилаючи HTTP-запити, зокрема у форматі GraphQL.

Контролер виконує роль обробника запитів, приймаючи введення від представлення та керуючи взаємодією між інтерфейсом і моделлю. Використовуючи Node.js як серверний середній шар, застосунок передає запити до GraphQL-сервісу, який виконує їх обробку. GraphQL-сервер отримує запити, звертається до відповідних resolvers і забезпечує повернення даних у необхідному форматі.

Resolvers є ключовими функціями, які відповідають за вибірку та обробку даних. Кожне поле у запиті відповідає окремому resolver, який визначає, де і як отримати відповідну інформацію. Використовуючи бібліотеку Mongoose, resolvers звертаються до бази MongoDB, отримують необхідні записи та передають їх назад у GraphQL-сервіс.

Після завершення обробки запиту дані повертаються до Node.js сервера, який трансформує їх у формат JSON та надсилає клієнту. Таким чином, кожен запит розгортається у серію функціональних викликів, які

поетапно забезпечують отримання необхідної інформації та її передачу користувачеві.

3.3 Розробка бази даних програмного забезпечення для навігації по місцевим подіям

База даних програмного забезпечення для навігації по місцевим подіям реалізована за допомогою Mongoose – спеціалізованої бібліотеки для роботи з MongoDB, що дозволяє ефективно маніпулювати даними у вигляді об'єктів JavaScript та підтримує їхню типізацію через GraphQL-схеми. Оскільки в основі зберігання використовується документно-орієнтований підхід, оновлення структури даних відбувається гнучко і не залежить від попередньої схеми.

База даних програмного забезпечення для навігації по місцевим подіям призначена для зберігання даних про користувачів системи та місцеві події, а також для зберігання іншої інформації, необхідної для роботи програмної системи.

База даних, яка використовується для роботи програмного забезпечення з навігації по місцевим подіям, містить чотири основні таблиці. Всі дані організовані у форматі, що дозволяє швидке виконання запитів і забезпечує гнучкість у зміні структури збереження. Таблиці пов'язані між собою, що дозволяє ефективно взаємодіяти з інформацією про користувачів та події.

Перша таблиця містить інформацію про користувачів, які взаємодіють із платформою. Кожен запис у цій таблиці має унікальний ідентифікатор, який призначається при створенні нового профілю. Додатково зберігається ім'я користувача, яке підтримує символи різних мов. Для забезпечення входу використовується електронна пошта, а паролі зберігаються у зашифрованому вигляді, що підвищує безпеку. Крім цього, реєструється час останнього входу в систему, що дає змогу відстежувати активність. Автоматично додається

посилання на аватар, якщо користувач завантажує зображення профілю. У цій же таблиці знаходиться поле, в якому фіксуються улюблені події, додані у список вподобань. Авторизація для певних функцій реалізована через спеціальний токен доступу.

Друга таблиця бази даних програмного забезпечення для навігації по місцевим подіям призначена для збереження інформації про події, доступні в системі. Кожен запис містить унікальний ідентифікатор заходу, а також його назву, яка відображається в інтерфейсі. Дата події подається у форматі, що дозволяє легко виконувати сортування та пошук за часовими параметрами. Кожен захід належить до певного типу, що допомагає класифікувати його. Опис подається у вигляді текстового блоку, який містить усю необхідну інформацію для користувачів. Організатор має можливість додавати теги, що полегшує пошук подій за категоріями. Додатково до запису прив'язується посилання на зображення або рекламний банер. Якщо захід є платним, вказується вартість квитка, а також валюта, в якій здійснюється оплата. Геолокація зберігається у вигляді координат, що дозволяє точно відображати місце проведення на карті.

Третя таблиця бази даних програмного забезпечення для навігації по місцевим подіям визначає набір параметрів, які можуть бути змінені для профілів користувачів. Вона використовується для модифікації даних без ризику змінити ключові параметри, що впливають на безпеку. Тут можна змінювати ім'я користувача, оновлювати посилання на аватар та редагувати список обраних подій. У разі необхідності користувач може змінити пароль, який перед збереженням хешується для захисту даних.

Четверта таблиця бази даних програмного забезпечення для навігації по місцевим подіям відіграє аналогічну роль для подій. Вона дозволяє коригувати тільки певні атрибути заходу, що зменшує ризик помилкових змін. У цій таблиці можна редагувати назву, дату проведення, опис, теги, категорію події та її вартість. Також передбачена можливість оновлення місця проведення заходу шляхом зміни географічних координат.

Зв'язки між таблицями забезпечують ефективну взаємодію між даними. Таблиця користувачів пов'язана з подіями через список вподобань, що дозволяє зберігати обрані заходи для кожного профілю. Кожен користувач може мати кілька обраних подій, що реалізується через збереження унікальних ідентифікаторів заходів у відповідному полі. Таблиця подій у свою чергу містить інформацію, що дозволяє групувати заходи за категоріями та датами. Таблиці для редагування даних виступають допоміжними та гарантують, що зміни відбуваються лише у встановлених межах.

3.4 Проєктування інтерфейсу програмного забезпечення для навігації по місцевим подіям

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для навігації по місцевим подіям виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для навігації по місцевим подіям враховані функціональні вимоги до програми:

- підтримка можливості подання користувачу списку місцевих подій (локальних заходів, що відбудуться у відповідному регіоні);
- можливість надання інформації про кожну подію, зокрема, інформації про місце події та вартість участі у події;
- можливість відображення місця проведення заходу на карті;
- можливість фільтрації подій за типом або датою;
- підтримка можливості реєстрації та авторизації користувачів;
- підтримка можливості зареєстрованих користувачів додавання інформації про нові події.

Інтерфейс головної форми програмного забезпечення для навігації по місцевим подіям наведено на рис. 3.3.

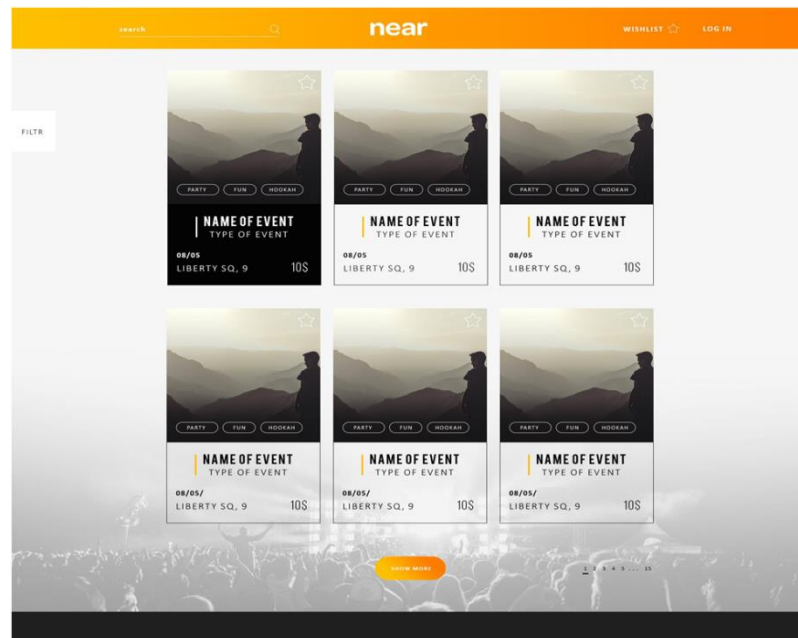


Рисунок 3.3 – Інтерфейс головної сторінки програмного забезпечення для навігації по місцевим подіям

Приклад сторінки заходу або події наведено на рис. 3.4.

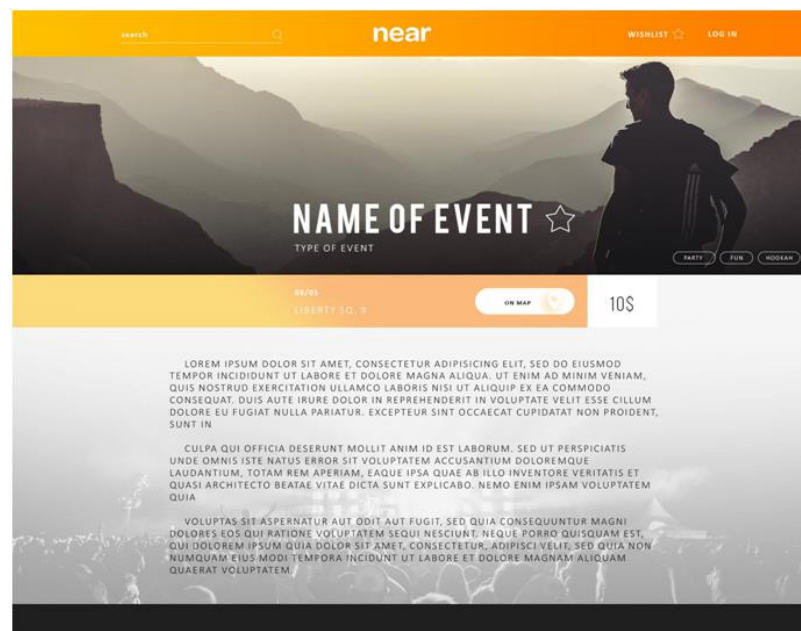


Рисунок 3.4 – Сторінка заходу, що відбувається на місцевому рівні

Форму відображення місця проведення заходу на карті наведено на рис. 3.5.

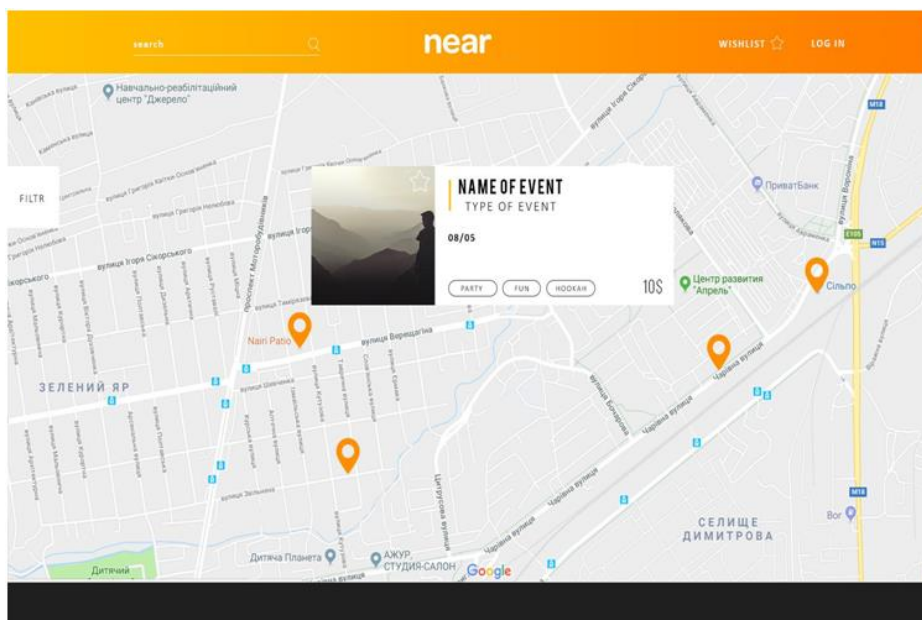


Рисунок 3.5 – Відображення місця проведення заходу на карті

Панель фільтрації заходів наведено на рис. 3.6.

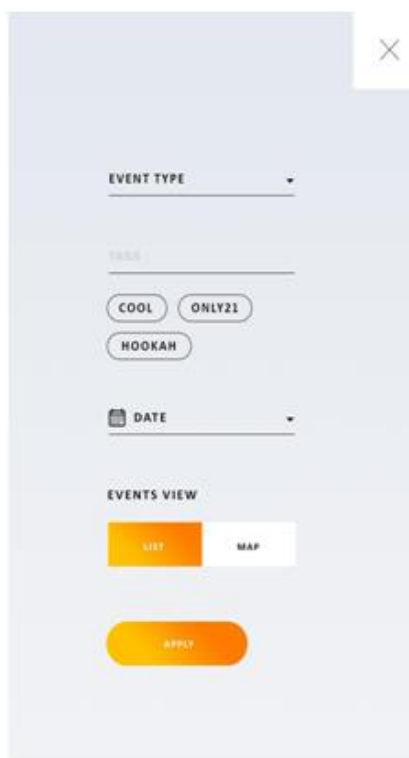


Рисунок 3.6 – Панель фільтрації заходів

Форму авторизації користувача наведено на рис. 3.7.

Рисунок 3.7 – Форма авторизації користувача

Форма створення нового заходу авторизованим користувачем наведено на рис. 3.8.

Рисунок 3.8 – Форма створення нового заходу авторизованим користувачем

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для навігації по місцевим подіям.

Запропоновано структуру програмного забезпечення для навігації по місцевим подіям. Визначено, що програмне забезпечення для навігації по місцевим подіям складається з п'яти основних модулів (серверний модуль, модуль конфігурації, модуль роботи з даними, модуль проміжного програмного забезпечення, модуль інтерфейсу користувача), кожен з яких відіграє ключову роль у функціонуванні системи, а також містить каталог додаткових бібліотек.

Розроблено базу даних програмного забезпечення для навігації по місцевим подіям, яка призначена для зберігання даних про користувачів системи та місцеві події, а також для зберігання іншої інформації, необхідної для роботи програмної системи. Визначено, що база даних, яка використовується для роботи програмного забезпечення з навігації по місцевим подіям, містить чотири основні таблиці (користувачі, місцеві події, а також таблиці для зберігання системної інформації). Всі дані організовані у форматі, що дозволяє швидке виконання запитів і забезпечує гнучкість у зміні структури збереження. Таблиці пов'язані між собою, що дозволяє ефективно взаємодіяти з інформацією про користувачів та події.

Описано функціонування програмного забезпечення для навігації по місцевим подіям.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для навігації по місцевим подіям.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для навігації по місцевим подіям. Програма написана на мові Java, через її універсальність і широкі можливості у веброзробці, оскільки JavaScript дозволяє створювати інтерактивні вебзастосунки, що працюють безпосередньо в браузері. В якості середовища для створення програмного забезпечення для навігації по місцевим подіям обрано Node.js через його продуктивність, масштабованість і підтримку асинхронної обробки даних. Крім того, Node.js забезпечує високу швидкість виконання коду, особливо при роботі з великою кількістю запитів у реальному часі.

Програмне забезпечення для навігації по місцевим подіям забезпечує виконання таких функцій:

- підтримка можливості подання користувачу списку місцевих подій (локальних заходів, що відбудуться у відповідному регіоні);
- можливість надання інформації про кожну подію, зокрема, інформації про місце події та вартість участі у події;
- можливість відображення місця проведення заходу на карті;
- можливість фільтрації подій за типом або датою;
- підтримка можливості реєстрації та авторизації користувачів;
- підтримка можливості зареєстрованих користувачів додавання інформації про нові події.

Для роботи програмного забезпечення для навігації по місцевим подіям потрібні такі технічні та програмні ресурси. Серверна частина повинна мати високопродуктивний процесор з тактовою частотою не нижче 2,5 ГГц (рекомендовано багатоядерний процесор, наприклад, Intel Xeon або AMD EPYC), оперативну пам'ять обсягом не менше 16 ГБ для забезпечення швидкої обробки запитів та взаємодії з базою даних, а також SSD-накопичувач ємністю

від 500 ГБ для швидкого зчитування та запису інформації про користувачів і події. Операційна система повинна бути сумісною з сучасними серверними платформами, такими як Ubuntu Server, CentOS або Windows Server. Клієнтська частина повинна підтримувати сучасні браузери (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge) з актуальними оновленнями. При мобільному використанні програми необхідно мати мобільний телефон або планшет на базі Android 8.0+ або iOS 13+, процесор з тактовою частотою від 1,8 ГГц (наприклад, Qualcomm Snapdragon 600 серії або Apple A10), оперативну пам'ять не менше 2 ГБ.

Функціональні характеристики розробленого програмного забезпечення для навігації по місцевим подіям наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для навігації по місцевим подіям наведено у додатку Б.

4.2 Характеристики програми для навігації по місцевим подіям

Програмне забезпечення для навігації по місцевим подіям складається з п'яти основних модулів (серверний модуль, модуль конфігурації, модуль роботи з даними, модуль проміжного програмного забезпечення, модуль інтерфейсу користувача), кожен з яких відіграє ключову роль у функціонуванні системи, а також містить каталог додаткових бібліотек.

Програмне забезпечення для навігації по місцевим подіям характеризується здатністю забезпечувати швидкий доступ до актуальної інформації про заходи, що проводяться на місцевому рівні. Воно підтримує інтеграцію з картографічними сервісами, що дозволяє користувачам переглядати місця проведення подій на карті та отримувати маршрут для зручного переміщення до заходу.

Інтерфейс реалізовано з урахуванням зручності використання, що забезпечує інтуїтивну навігацію та доступність основних функцій. Передбачено можливість фільтрації подій за критеріями. Взаємодія з

системою здійснюється через мобільні засоби або персональні комп'ютери, що дозволяє користувачам отримувати сповіщення про нові заходи.

Обробка даних виконується у серверній частині, що базується на сучасних технологіях для ефективного керування інформацією. Для збереження відомостей про події, користувачів та їхні вподобання використовується база даних, яка підтримує гнучку структуру та швидкий пошук.

Передбачено механізми ідентифікації та авторизації, що гарантують безпеку особистих даних користувачів. Інформація передається з використанням вбудованих засобів шифрування, що забезпечує захист від несанкціонованого доступу.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для навігації по місцевим подіям на серверній частині розгорнути та запустити Node.js, після чого запустити програму.

4.3.2 Вхідні й вихідні дані

Вхідними даними до серверної частини програмного забезпечення для навігації по місцевим подіям є запити, що надходять з клієнтської сторони. Вхідними даними до клієнтської сторони є інформація щодо критеріїв фільтрування заходів, дані для реєстрації клієнта та ін.

Вихідними даними до серверної частини програмного забезпечення для навігації по місцевим подіям є відповіді на запити, що надходять від клієнтської сторони, та згенеровані сторінки вебсайту.

4.3.3 Повідомлення

Користувач програмного забезпечення для навігації по місцевим подіям може отримати повідомлення про необхідність авторизації, повідомлення про успішно створену подію та ін.

4.4 Виконання програмного забезпечення для навігації по місцевим подіям

Після запуску програмного забезпечення для навігації по місцевим подіям користувачу відображається головна сторінка (рис. 4.1), де знаходиться перелік найближчих подій. Відображається частина найближчих подій, для отримання списку інших подій користувачу необхідно або натиснути на кнопку Show more, або перейти до наступної сторінки за відповідним посиланням.

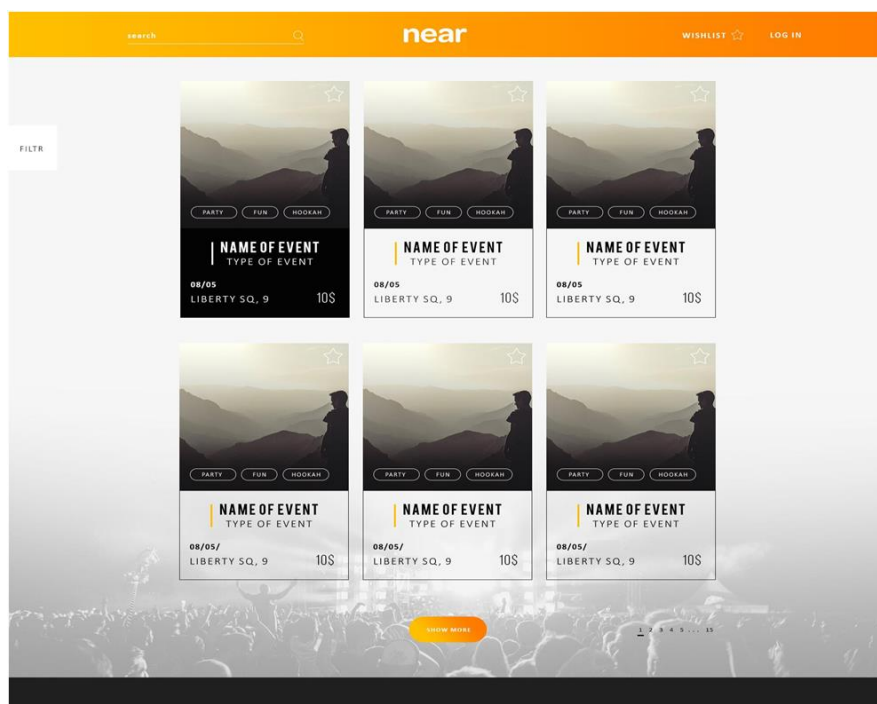


Рисунок 4.1 – Інтерфейс головної сторінки програмного забезпечення для навігації по місцевим подіям

При виборі конкретної події відображається сторінка обраного заходу (рис. 4.2), де подається його опис, ціна та дата проведення.

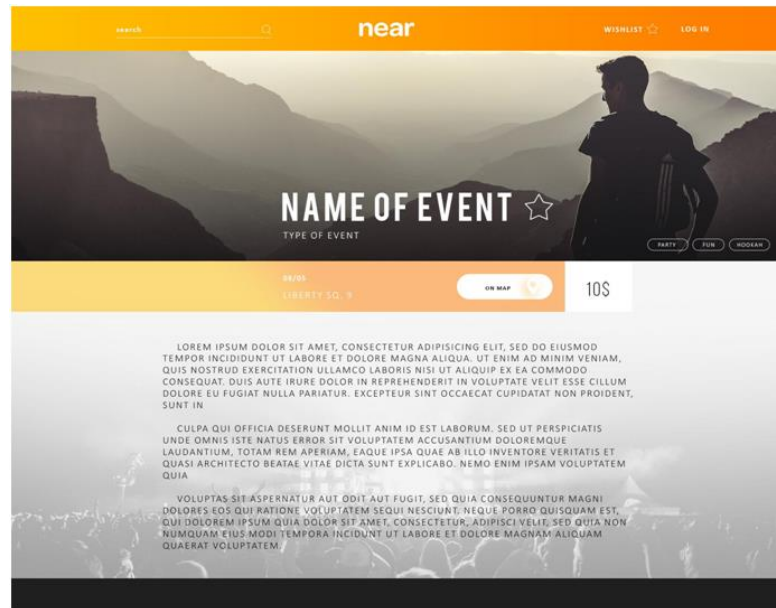


Рисунок 4.2 – Сторінка заходу, що відбувається на місцевому рівні

При натисканні на кнопку On map відображається карта, на якій позначено місце проведення відповідного заходу (рис. 4.3).

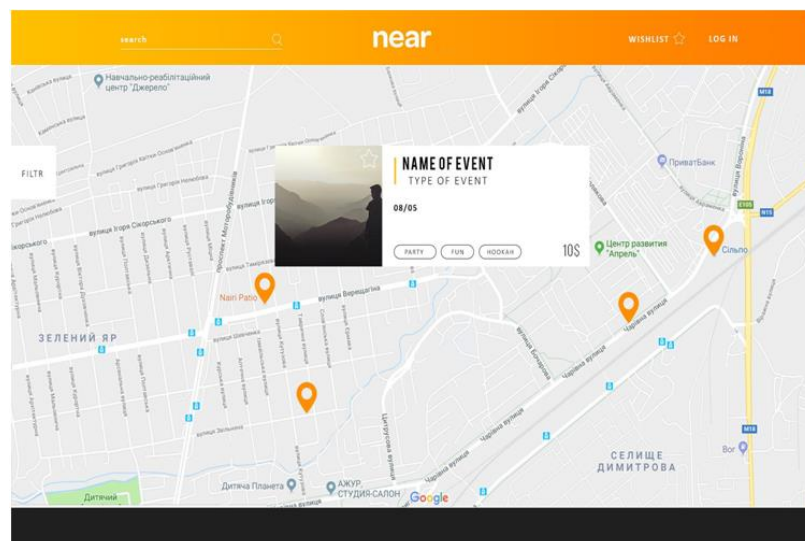


Рисунок 4.3 – Відображення місця проведення заходу на карті

При натисканні на головній сторінці кнопки FILTR відображається форма фільтрації заходів (рис. 4.4), за допомогою якої користувач може

відібрати заходи та типом або датою. Також існує можливість відображення результатів у вигляді списку або спеціальними позначками на мапі.

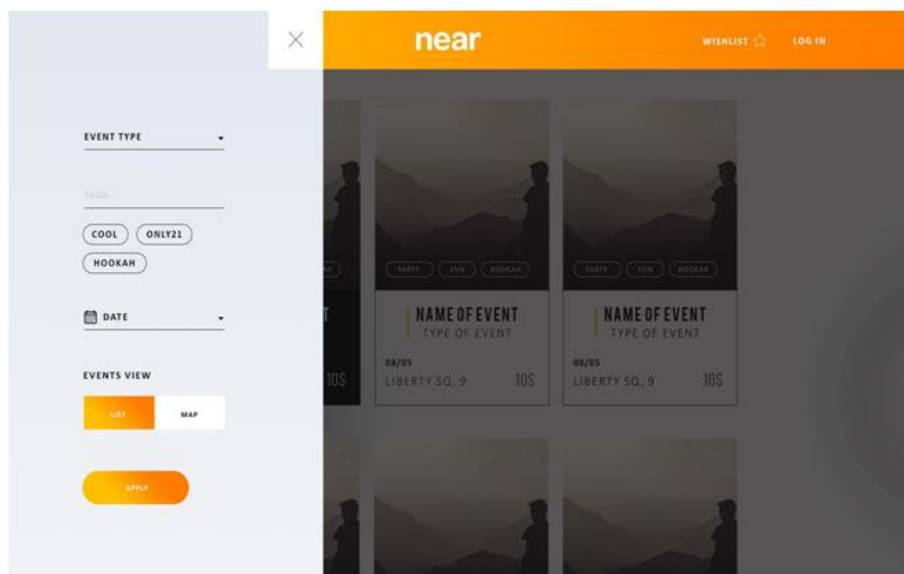


Рисунок 4.4 – Засоби фільтрації заходів

Для можливості додавання інформації про нові заходи, що будуть проходити на локальному рівні, користувач повинен пройти процедуру авторизації (рис. 4.5), ввівши свої реєстраційні дані.

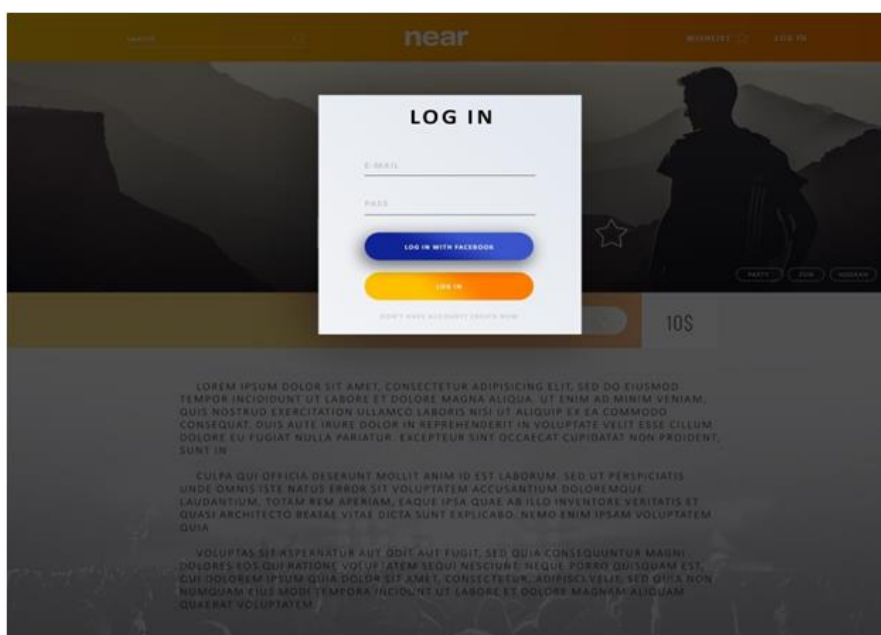


Рисунок 4.5 – Авторизація користувача

Для створення нової події необхідно натиснути відповідну кнопку та у формі, що з'явилася (рис. 4.6), ввести інформацію про захід (його дату, адресу, опис та, за необхідності, фотографії або інші зображення).

The screenshot shows a web application interface for creating a new event. A modal form is displayed in the center, overlaid on a background image of a crowd at night. The form has a white background and a thin border. At the top left of the form is a plus icon and the text 'ADD PHOTO'. Below this is a section with two input fields: 'EVENT TITLE' and 'EVENT TYPE' (which has a dropdown arrow). Underneath these is a 'DATE' field with a calendar icon and a dropdown arrow, followed by a 'TAGS' section with three placeholder tags: 'TAG 1, TAG 2, TAG 3'. Below the tags is an 'ADDRESS' field with a location pin icon. At the bottom of the form is a 'DESCRIPTION' field with a text area. A large orange 'CREATE' button is positioned at the bottom center of the form. The background image shows a crowd of people at night, with a bright light source in the distance. The overall layout is clean and modern.

Рисунок 4.6 – Форма створення нового заходу авторизованим користувачем

4.5 Тестування програмного забезпечення для навігації по місцевим подіям

Виконано тестування програмного забезпечення для навігації по місцевим подіям.

Виявлено, що програма для навігації по місцевим подіям функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу подання інформації про події в регіоні.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з поданням інформації про події в регіоні.

4.6 Висновки за розділом 4

Описано програмне забезпечення для навігації по місцевим подіям. Виконано тестування розробленого програмного забезпечення для навігації по місцевим подіям. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з поданням інформації про місцеві події.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для навігації по місцевим подіям.

Визначено, що програмне забезпечення для навігації по місцевих подіях надає користувачам доступ до актуальної інформації про заходи, що відбуваються у вибраному регіоні. Такі програми забезпечують зручний перегляд подій, полегшення комунікації між учасниками та організаторами, а також сприяють взаємодії всередині спільноти, дозволяють знайти цікаві заходи, швидко дізнаватися про нові події та оптимально планувати дозвілля.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для навігації по місцевим подіям. Проте деякі програмні засоби для навігації по місцевим подіям можуть бути перевантажені інформацією. Велика кількість подій, особливо в мегаполісах, може ускладнювати навігацію, а алгоритми персоналізації не завжди правильно враховують вподобання користувачів. Деякі застосунки або вебсайти можуть мати складний інтерфейс або містити надлишкову рекламу, що знижує їхню ефективність. Ще одним недоліком є питання конфіденційності та безпеки. Такі сервіси часто використовують дані геолокації та можуть збирати особисту інформацію для аналізу поведінки користувачів, що створює ризики витоку або зловживання цими даними. Окрім цього, розробка, придбання та підтримка таких програмних сервісів потребує значних ресурсів, а деякі їх функції можуть бути доступні лише у платних версіях або за передплатою. Тому актуальною є розробка програмного забезпечення для навігації по місцевим подіям.

Сформульовано функціональні вимоги до програмного забезпечення для навігації по місцевим подіям.

Для реалізації програмного забезпечення для навігації по місцевим подіям обрано мову програмування JavaScript через її універсальність і широкі

можливості у веброзробці. JavaScript дозволяє створювати інтерактивні вебзастосунки, що працюють безпосередньо в браузері. Крім того, JavaScript також забезпечує легку інтеграцію з іншими технологіями, включаючи серверні рішення на Node.js, що дозволяє створювати повноцінні динамічні застосунки з гнучкою обробкою даних на стороні сервера.

Для створення програмного забезпечення для навігації по місцевим подіям обрано середовище Node.js через його продуктивність, масштабованість і підтримку асинхронної обробки даних. Node.js забезпечує високу швидкість виконання коду, особливо при роботі з великою кількістю запитів у реальному часі. Однією з ключових переваг Node.js є неблокуюча архітектура та використання подійного циклу, що дозволяє ефективно обробляти численні одночасні запити, наприклад, при оновленні інформації про події чи зміні геолокації користувачів. Це особливо важливо для інтерактивних сервісів, що потребують швидкого обміну даними між клієнтами та сервером.

Розроблено програмне забезпечення для навігації по місцевим подіям.

Запропоновано структуру програмного забезпечення для навігації по місцевим подіям. Визначено, що програмне забезпечення для навігації по місцевим подіям складається з п'яти основних модулів (серверний модуль, модуль конфігурації, модуль роботи з даними, модуль проміжного програмного забезпечення, модуль інтерфейсу користувача), кожен з яких відіграє ключову роль у функціонуванні системи, а також містить каталог додаткових бібліотек.

Розроблено базу даних програмного забезпечення для навігації по місцевим подіям, яка призначена для зберігання даних про користувачів системи та місцеві події, а також для зберігання іншої інформації, необхідної для роботи програмної системи. Визначено, що база даних, яка використовується для роботи програмного забезпечення з навігації по місцевим подіям, містить чотири основні таблиці (користувачі, місцеві події, а також таблиці для зберігання системної інформації). Всі дані організовані у

форматі, що дозволяє швидке виконання запитів і забезпечує гнучкість у зміні структури збереження. Таблиці пов'язані між собою, що дозволяє ефективно взаємодіяти з інформацією про користувачів та події.

Описано функціонування програмного забезпечення для навігації по місцевим подіям.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для навігації по місцевим подіям.

Виконано тестування розробленого програмного забезпечення для навігації по місцевим подіям. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з поданням інформації про місцеві події.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. How to Build Best Local Events App [Electronic resource]. – Access mode: How to Build Best Local Events App Like Eventbrite and YPlan.
2. Event Creation [Electronic resource]. – Access mode: <https://www.g2.com/glossary/event-creation-definition>.
3. Best CRM Software for Event Management: Features, Reviews, & Custom Solutions [Electronic resource]. – Access mode: <https://www.phaedrasolutions.com/blog/best-crm-software-for-event-management>.
4. Best Local Events Apps for Finding Events Near You [Electronic resource]. – Access mode: <https://www.lifewire.com/best-local-events-apps-4048406>.
5. Complete Guide to Event Ticketing Software in the UK [Electronic resource]. – Access mode: <https://ticket-generator.com/blog/complete-guide-to-event-ticketing-software-in-the-uk>.
6. Event creation software: Choose the right tool [Electronic resource]. – Access mode: <https://blog.bikubo.com/en/event-creation-software/>.
7. Software Tools for Events: Comprehensive Guide For All Stages [Electronic resource]. – Access mode: <https://info.eventvesta.com/resources/software-tools-for-events/>.
8. Best Apps for Discovering Events Happening Around You [Electronic resource]. – Access mode: <https://www.makeuseof.com/mobile-apps-for-nearby-events/>.
9. Best Apps for Discovering Events Happening Around You [Electronic resource]. – Access mode: <https://umatechnology.org/the-10-best-apps-for-discovering-events-happening-around-you/>.
10. Apps to Find Upcoming Events & New Places of Interest [Electronic resource]. – Access mode: <https://www.hongkiat.com/blog/apps-events-activities-around-you/>.

11. Bandsintown [Electronic resource]. – Access mode:
<https://www.bandsintown.com/>.

12. Meetup [Electronic resource]. – Access mode:
<https://www.meetup.com/>.

13. Eventbrite [Electronic resource]. – Access mode:
<https://www.eventbrite.co.uk/>.

14. JavaScript Introduction [Electronic resource]. – Access mode:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>.

15. JavaScript Frameworks [Electronic resource]. – Access mode:
<https://help.form.io/developers/javascript-development/frameworks>.

16. Download Node.js [Electronic resource]. – Access mode:
<https://nodejs.org/en/download>.

17. Node.js Modules [Electronic resource]. – Access mode:
https://www.w3schools.com/nodejs/nodejs_modules.asp.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для навігації по місцевим подіям.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для навігації по місцевим подіям», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 р.

А.2 Призначення розробки

Програмний продукт призначений для навігації по місцевим подіям.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для навігації по місцевим подіям забезпечує виконання таких функцій:

- підтримка можливості подання користувачу списку місцевих подій (локальних заходів, що відбудуться у відповідному регіоні);
- можливість надання інформації про кожну подію, зокрема, інформації про місце події та вартість участі у події;
- можливість відображення місця проведення заходу на карті;
- можливість фільтрації подій за типом або датою;
- підтримка можливості реєстрації та авторизації користувачів;
- підтримка можливості зареєстрованих користувачів додавання інформації про нові події.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для навігації по місцевим подіям повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для навігації по місцевим подіям повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для навігації по місцевим подіям необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для навігації по місцевим подіям потрібні такі технічні та програмні ресурси. Серверна частина повинна мати високопродуктивний процесор з тактовою частотою не нижче 2,5 ГГц (рекомендовано багатоядерний процесор, наприклад, Intel Xeon або AMD EPYC), оперативну пам'ять обсягом не менше 16 ГБ для забезпечення швидкої обробки запитів та взаємодії з базою даних, а також SSD-накопичувач ємністю від 500 ГБ для швидкого зчитування та запису інформації про користувачів і події. Операційна система повинна бути сумісною з сучасними серверними платформами, такими як Ubuntu Server, CentOS або Windows Server. Клієнтська частина повинна підтримувати сучасні браузеры (Google Chrome,

Mozilla Firefox, Safari, Microsoft Edge) з актуальними оновленнями. При мобільному використанні програми необхідно мати мобільний телефон або планшет на базі Android 8.0+ або iOS 13+, процесор з тактовою частотою від 1,8 ГГц (наприклад, Qualcomm Snapdragon 600 серії або Apple A10), оперативну пам'ять не менше 2 ГБ.

А.3.6 Вимоги до маркування і пакування

Програма для навігації по місцевим подіям може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для навігації по місцевим подіям».

А.4 Вхідні дані до роботи

Вхідними даними до серверної частини програмного забезпечення для навігації по місцевим подіям є запити, що надходять з клієнтської сторони. Вхідними даними до клієнтської сторони є інформація щодо критеріїв фільтрування заходів, дані для реєстрації клієнта та ін.

Вихідними даними до серверної частини програмного забезпечення для навігації по місцевим подіям є відповіді на запити, що надходять від клієнтської сторони, та згенеровані сторінки вебсайту.

ДОДАТОК Б
Фрагмент тексту програми

```

function chckTkn(rqwst, rslt, nxt) {
  const grphQLOpName = rqwst.body.OpName;

  if (grphQLOpName !== 'Protected') {
    nxt()
  } else {
    const tkn = rqwst.headers['x-access-tkn']

    if (!tkn) {
      return rslt.status(401).send({ osh: 'Not authorized' })
    }

    jrwst.verify(tkn, config.secret, async (err, decd) => {
      if (err) {
        return rslt.status(401).send({ osh: 'Not authorized' })
      } else {
        const krstvch = await Krstvch.findOne({ tkn })

        if (!krstvch) {
          return rslt.status(401).send({ osh: 'Not authorized' })
        }

        rqwst.decd = decd
        nxt()
      }
    })
  }
}

export {
  chckTkn
}

type Krstvch {
  id: ID
  krstvchname: String!
  email: String!
  pswrd: String!
  avtrurl: String
  tkn: String
}

```

```

    lstlgn: Float
    wshlst: [String]
}

input KrstvchInput {
  krstvchname: String
  pswrd: String
  avtrurl: String
  wshlst: [String]
}

```

```
scalar Date
```

```

type Podija {
  id: ID
  ttkl: String!
  date: Date!
  type: String!
  dscriptn: String!
  tgs: [String]
  pstr: String
  cina: Int
  vltn: String
  lagt: Float
  lobn: Float
}

```

```

input PodijaInput {
  ttkl: String
  date: Date
  type: String
  dscriptn: String
  tgs: [String]
  pstr: String
  cina: Int
  vltn: String
  lagt: Float
  lobn: Float
}

```

```
type Zapt {
```

```

    rgstr(email: String!, krstvchname: String!, pswrd: String!):
Krstvch
    lgn(email: String!, pswrd: String!): Krstvch
    lgt: Boolean
    me: Krstvch
    createPodija(podija: PodijaInput): Podija
    getPodija(id: ID!): Podija
    getPodijas(lagt: Float!, lobn: Float!): [Podija]
    getAllPodijas: [Podija]
    dropPodijas: Boolean
    addToWshlst(krstvch_id: ID!, podija_id: ID!): [Podija]
    removeFromWshlst(krstvch_id: ID!, podija_id: ID!): [Podija]
    getWshlst(krstvch_id: ID!): [Podija]
}

```

```
const Schema = mongoose.Schema
```

```

const KrstvchSchema = new Schema({
  krstvchname: String,
  email: String,
  pswrd: String,
  avtrurl: String,
  tkn: String,
  lstlgn: Number,
  wshlst: [String]
})

```

```

const Krstvch = connection.model('Krstvch', KrstvchSchema)
export default Krstvch

```

```

const PodijaSchema = new Schema({
  ttkl: String,
  date: Number,
  type: String,
  dscrptn: String,
  tgs: Array,
  pstr: String,
  cina: Number,
  vltn: String,
  lagt: Number,
  lobn: Number

```

```

}))

const Podija = connection.model('Podija', PodijaSchema)

async function lgn({ email, ps wrd }) {
  const krstvch = await Krstvch.findOne({ email })

  if (krstvch) {
    const ps wrdMatch = await bcrypt.compare(ps wrd,
krstvch.ps wrd);

    if (!ps wrdMatch) {
      throw new Osh('Wrong ps wrd')
    }

    try {
      const pyld = { naruto: 'hokage' }

      const lstlgn = Date.now().valueOf()
      const tkn = jrt.sign(pyld, config.secret)

      krstvch.lstlgn = lstlgn
      krstvch.tkn = tkn

      await Krstvch.update({ _id: krstvch._id }, { lstlgn, tkn
}).exec()
      return krstvch
    } catch (osh) {
      throw new Osh('We can not lgn you, sorry :( Sorry')
    }
  } else {
    throw new Osh('Wrong email')
  }
}

async function lgt(args, rqwst) {
  const tkn = rqwst.get('x-access-tkn');

  const krstvch = await Krstvch.findOne({ tkn })

  if (krstvch) {

```

```

    try {
      await Krstvch.update({ _id: krstvch._id }, { lstlgn: null,
tkn: null }).exec()
      return true
    } catch (osh) {
      throw new Osh('Can not lgt, try again')
    }
  } else {
    throw new Osh('Not authorized');
  }
}

```

```

async function rgstr({ email, krstvchname, pswrd }) {
  const krstvch = await Krstvch.findOne({ email })

  if (krstvch) {
    throw new Osh('Krstvchname with such email already exists')
  } else {
    try {
      const slft = 10;
      const pswrdHash = await bcrypt.hash(pswrd, slft)
      return await Krstvch.create({ email, krstvchname, pswrd:
pswrdHash })
    } catch (osh) {
      throw new Osh('sorry, try again')
    }
  }
}

```

```

async function me(args, rqwst) {
  const tkn = rqwst.get('x-access-tkn')

  if (!tkn) {
    throw new Osh('Not authorized')
  }

  try {
    return await Krstvch.findOne({ tkn })
  } catch (osh) {
    throw new Osh('Not authorized')
  }
}

```

```

}

async function getAllKrstvchs() {
  try {
    return await Krstvch.find({})
  } catch (osh) {
    throw new Osh('Osh getting all krstvchs')
  }
}

async function dropKrstvchs() {
  try {
    await Krstvch.deleteMany({})
    return true
  } catch (osh) {
    throw new Osh('Osh dropping krstvchs')
  }
}

export default {
  lgn,
  rgstr,
  lgt,
  me,
}

async function createPodija({ podija }) {
  try {
    return await Podija.create(podija)
  } catch (e) {
    throw new Osh('Osh creating new podija')
  }
}

async function dropPodijas() {
  try {
    await Podija.deleteMany({})
    return true
  } catch (osh) {
    throw new Osh('Osh dropping podijas')
  }
}

```

```

}

async function getPodija({ id }) {
  let podija

  try {
    podija = await Podija.findOne({ _id: id });
  } catch (e) {
    throw new Osh('Osh getting podija')
  }

  if (!podija) {
    throw new Osh('Podija was not found')
  }

  return podija
}

```

```

async function getPodijas({ lagt, lobn }) {
  const trslthold = 0.5;

  try {
    return await Podija.find({
      lagt: {
        $gte: lagt - trslthold,
        $lte: lagt + trslthold
      },
      lobn: {
        $gte: lobn - trslthold,
        $lte: lobn + trslthold
      }
    })
  } catch (e) {
    throw new Osh('Osh searching for podijas')
  }
}

```

```

async function getAllPodijas() {
  try {
    return await Podija.find({});
  } catch (e) {

```

```
        throw new Osh('Osh getting all podijas')
    }
}

export default {
  createPodija,
  getPodija,
  getAllPodijas,
  dropPodijas,
  getPodijas
}
```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Програмне забезпечення для навігації по місцевим подіям

Виконав: Олександр ГОЦЬКИЙ
ст. групи КНТ-112сп

Керівник: Тетяна ЗАЙКО
к.т.н., доцент

Рисунок В.1 – Слайд 1

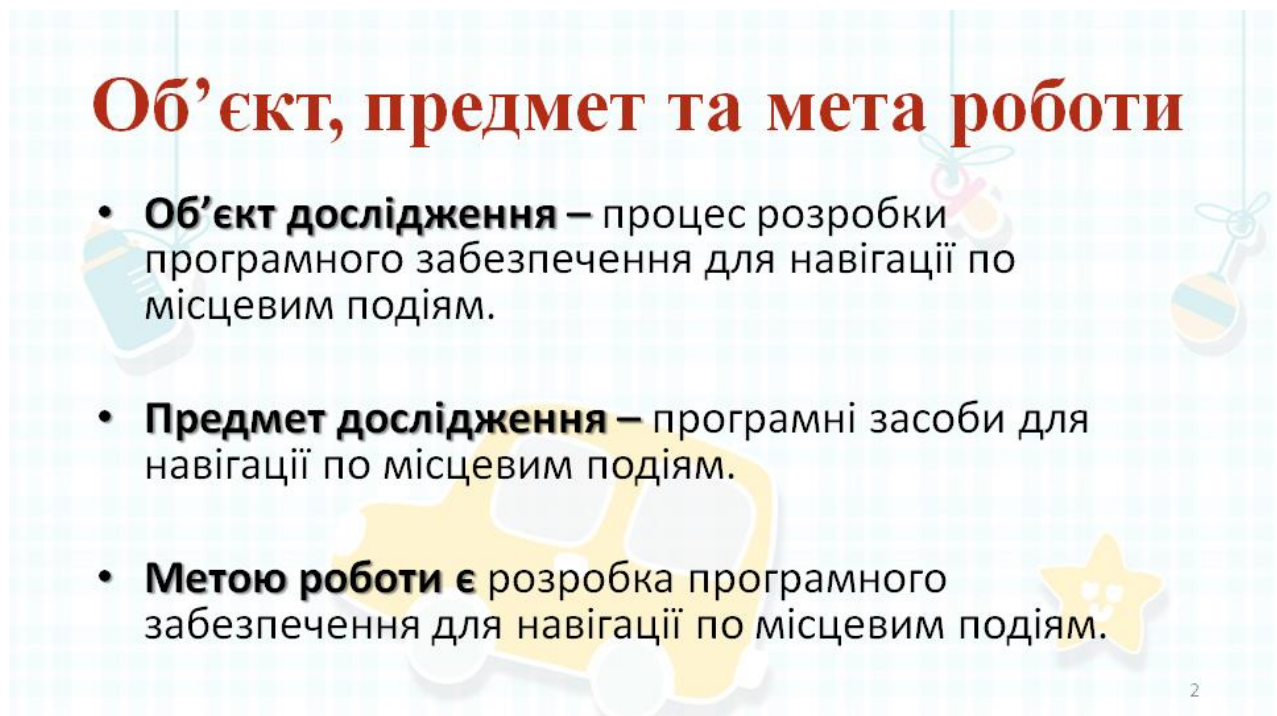


Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для навігації по місцевим подіям;
- здійснити проектування програмного забезпечення для навігації по місцевим подіям;
- створити програмне забезпечення для навігації по місцевим подіям;
- виконати тестування розробленого програмного забезпечення для навігації по місцевим подіям.

3

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Bandsintown	Meetup	Eventbrite
Можливість створення власних заходів	–	+	+
Зручність інтерфейсу	+	+	+–
Рекомендації на основі інтересів	+	+	+–
Можливість купівлі квитків	+	–	+
Наявність реклами та партнерських пропозицій	+–	+–	+
Фокус на професійних подіях	–	+–	+

4

Рисунок В.4 – Слайд 4

Постановка завдання

При розробці програмного забезпечення для навігації по місцевим подіям необхідно забезпечити такі функціональні вимоги:

- підтримка можливості подання користувачу списку місцевих подій (локальних заходів, що відбудуться у відповідному регіоні);
- можливість надання інформації про кожну подію, зокрема, інформації про місце події та вартість участі у події;
- можливість відображення місця проведення заходу на карті;
- можливість фільтрації подій за типом або датою;
- підтримка можливості реєстрації та авторизації користувачів;
- підтримка можливості зареєстрованих користувачів додавання інформації про нові події.

5

Рисунок В.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	JavaScript	Python	Java
Асинхронність та робота в реальному часі	+	+-	+-
Інтерактивність	+	-	+-
Висока продуктивність	+	+-	+
Зручність для створення ПЗ для навігації по місцевим подіям	+	-	-
Велика спільнота	+	+	+-

6

Рисунок В.6 – Слайд 6

Порівняння середовищ

Критерій порівняння середовищ	Середовища		
	Node.js	Deno	Electron
Простота розробки	+	+-	-
Гнучкість у розгортанні	+	+-	+-
Асинхронна робота	+	+	-
Підтримка картографічних сервісів	+	+-	-
Швидкість розробки	+	+	+-

7

Рисунок В.7 – Слайд 7

Структура програми



8

Рисунок В.8 – Слайд 8

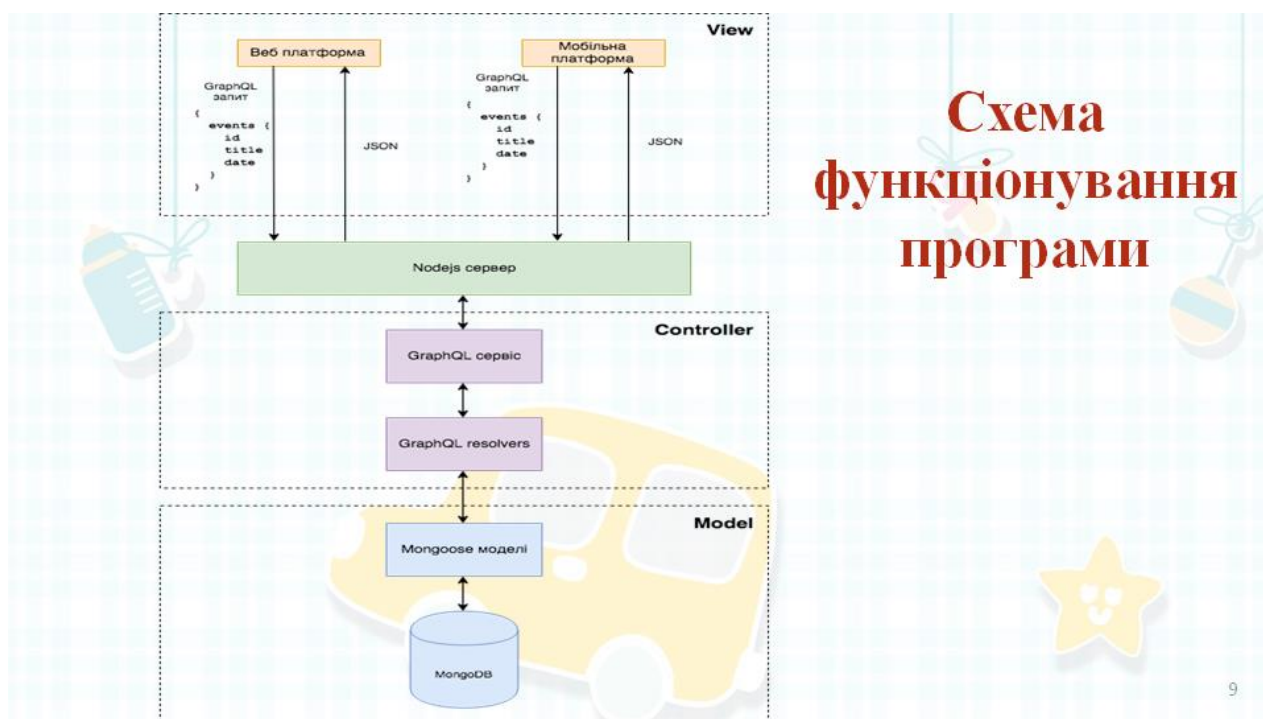
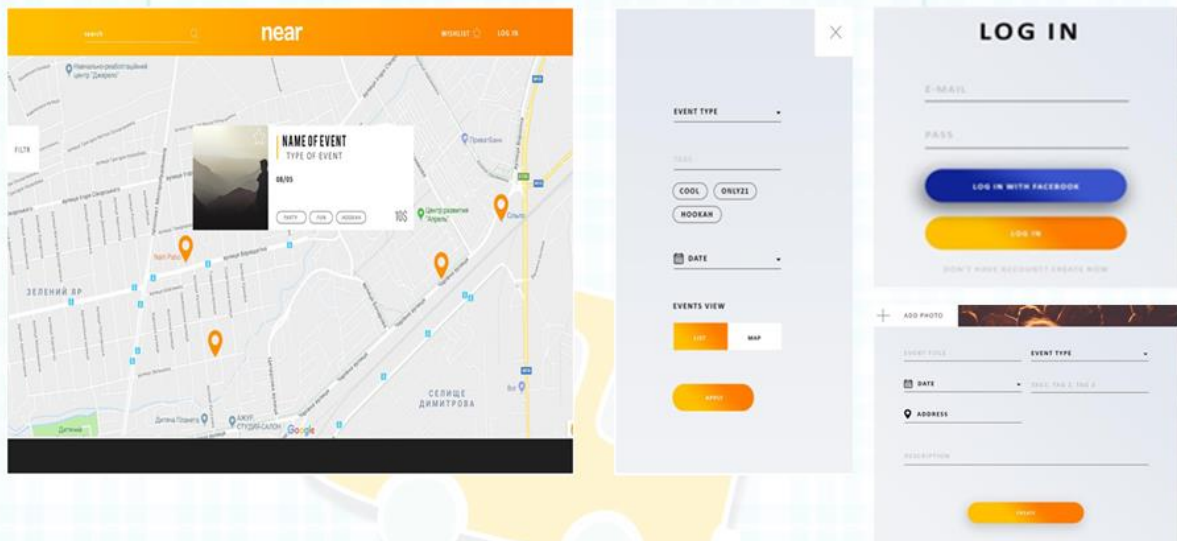


Рисунок В.9 – Слайд 9



Рисунок В.10 – Слайд 10

Інтерфейс застосунку



11

Рисунок В.11 – Слайд 11

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено розробки програмного забезпечення для навігації по місцевим подіям.

Сформульовано функціональні вимоги до програмного забезпечення для навігації по місцевим подіям. Для реалізації програмного забезпечення для навігації по місцевим подіям обрано мову програмування JavaScript та середовище Node.js. Розроблено програмне забезпечення для навігації по місцевим подіям. Запропоновано структуру програмного забезпечення для навігації по місцевим подіям. Визначено, що програмне забезпечення для навігації по місцевим подіям складається з п'яти основних модулів (серверний модуль, модуль конфігурації, модуль роботи з даними, модуль проміжного програмного забезпечення, модуль інтерфейсу користувача), кожен з яких відіграє ключову роль у функціонуванні системи, а також містить каталог додаткових бібліотек. Виконано тестування розробленого програмного забезпечення для навігації по місцевим подіям.

В результаті виконання завдання було отримано програмну систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

12

Рисунок В.12 – Слайд 12