

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ОНЛАЙН-СЕРВІСУ ДЛЯ ПЛАНУВАННЯ ЗАХОДІВ ТА
УЧАСТІ У НИХ НА БАЗІ VUE.JS ТА NODE.JS**

**DEVELOPMENT OF AN ONLINE SERVICE FOR PLANNING AND
PARTICIPATING IN EVENTS BASED ON VUE.JS AND NODE.JS**

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗ-41
Калюжний Дмитро Валерійович


(підпис)

Керівник:
асистент
Потейчук Михайло Іванович


(підпис)

Кваліфікаційну роботу
допущено до захисту
« 8 » червне 2024 р.
к.т.н., доцент
Гарант освітньої програми:
Ліщина Наталія Миколаївна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти: бакалавр
Галузь знань: 12 Інформаційні технології
Спеціальність: 121 Інженерія програмного забезпечення
Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М. М. Миколюк
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Калюжному Дмитру Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка онлайн-сервісу для планування заходів та участі у них на базі Vue.js та Node.js

Керівник роботи: асистент Потейчук Михайло Іванович

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів проектування, опис функціонального наповнення об'єкта проектування, розробка й обґрунтування системного наповнення, оцінка ергономічних та надійнісних параметрів проекрованої системи, функціонально-структурна схема роботи об'єкта проектування, розробка моделі бази даних об'єкта проектування.

5. Перелік графічного матеріалу: 18 рисунків

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Потейчук М. І.		
Специфікації вимог до розробленої системи	Потейчук М. І.		
Розробка об'єкта проєктування	Потейчук М. І.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		6,4 %	
Академічна доброчесність	Яцук А. А.		

7. Дата видачі завдання «2» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ З/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	вик.
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2024 р.	вик.
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	вик.
4	Розробка функціонально-структурної схеми роботи об'єкту проєктування та проєктування бази даних	17.04.2024 р.	вик.
5	Практична реалізація об'єкту проєктування та розробка бази даних	18.05.2024 р.	вик.
6	Тестування та налагодження об'єкту проєктування	01.06.2024 р.	вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	05.06.2024 р.	вик.

Здобувач вищої освіти

Керівник кваліфікаційної роботи

 (підпис)

 Калюжний Д. В.
 (прізвище, ініціали)

 Потейчук М. І.
 (прізвище, ініціали)

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: *Калюжний Дмитро Валерійович*

Тема: *«Розробка онлайн-сервісу для планування заходів та участі у них на базі Vue.js та node.js».*

Коротка характеристика кваліфікаційної роботи та прийнятих рішень:

Кваліфікаційна робота бакалавра складається з трьох розділів, в яких проаналізовані проблематика та поставлені завдання за темою роботи, здійснено теоретичне дослідження та практичну реалізацію онлайн-сервісу для планування заходів та участі у них.

Самостійні розробки і пропозиції автора: *Автор самостійно виконав розробку онлайн-сервісу для планування заходів та участі у них*

Практичне значення роботи *полягає у розробці інтегрованого онлайн-сервісу, який забезпечує інтерактивний та адаптивний користувацький інтерфейс на базі Vue.js, що дозволяє користувачам зручно планувати, редагувати та керувати заходами. Цей онлайн-сервіс стане корисним інструментом як для особистого використання, так і для бізнес-сегменту, допомагаючи ефективно планувати та проводити заходи будь-якого масштабу.*

Недоліки: *Істотних недоліків в роботі не виявлено.*

Загальний висновок: *Кваліфікаційна робота бакалавра здобувача освіти Калюжного Дмитра Валерійовича присвячена розробці сервісу для організації і проведення заходів. У роботі проаналізовано проблематику та поставлені завдання, виконано теоретичне дослідження та практичну реалізацію додатку, а також проведено тестування розробленої системи.*

Автор самостійно виконав розробку додатку, створивши зручний інструмент для організації і проведення заходів.

Кваліфікаційна робота здобувача освіти Калюжного Дмитра Валерійовича рекомендується до захисту експертній комісії та заслуговує позитивної оцінки.

Рецензент кваліфікаційної роботи: Здолбіцька Ніна Василівна, к.т.н., доцент кафедри комп'ютерних наук

(прізвище, ім'я, по-батькові, посада)



« 57 » червня 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач вищої освіти Калюжний Дмитро Валерійович
(прізвище, ім'я, по батькові)
група ІПЗ-41

Тема кваліфікаційної роботи:

Розробка онлайн-сервісу для планування заходів та участі у них на базі Vue.js та node.js

Керівник Потейчук Михайло Іванович

Актуальність теми

У сучасному світі, де швидкість життя постійно зростає, онлайн-сервіси для планування заходів набувають все більшої популярності. Використання технологій веб-розробки, таких як Vue.js та Node.js, дозволяє створювати високопродуктивні та масштабовані рішення для планування та управління заходами.

Об'єкт дослідження процеси організації та управління заходами за допомогою інформаційних технологій

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи:

Автор аналізує різні аналоги, методи та засоби для розробки функціонального сервісу для планування заходів, розробляє власний онлайн-сервіс з використанням Vue.js та node.js. Зміст роботи відповідає обраній темі.

Зауваження та недоліки: Суттєвих недоліків в роботі не виявлено.

Загальний висновок:

Кваліфікаційна робота бакалавра виконана на високому рівні. Робота виконана відповідно до вимог, логічно й послідовно описує хід виконання поставленого завдання. Пояснювальна записка відповідає стандартам до її оформлення. Робота може бути оцінена на високу оцінку

Керівник 
(підпис)

(Потейчук М. І.)
(прізвище, ім'я, по батькові)

«07» 06 2024 р.

АНОТАЦІЯ

Калюжний Д. В. Розробка онлайн-сервісу для планування заходів та участі у них на базі Vue.js та node.js. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану проблеми розробки онлайн-сервісів на базі Vue.js та node.js і сформульовано завдання. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено онлайн-сервіс на базі Vue.js та node.js і здійснено його тестування. У висновках узагальнено інформацію, відображену у попередніх частинах. Результати розробки можуть бути застосовані в практичній діяльності для планування заходів та участі у них.

Ключові слова: онлайн-сервіс, Vue.js, node.js, планування.

ABSTRACT

Kaliuzhnyi D. V. Development of an online service for planning and participating in events based on Vue.js and node.js. Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualifying thesis consists of an introduction, three sections, conclusions and proposals, a list of used sources and annexes.

In the first chapter, an analysis of the current state of the problem of developing online services based on Vue.js and node.js was carried out and the task was formulated. The second section defines the requirements for the developed system, as well as means, methods and development algorithms. In the third section, an online service based on Vue.js and node.js was developed and tested. The conclusions summarize the information presented in the previous parts. The results of the development can be applied in practical activities for planning events and participating in them.

Keywords: online service, Vue.js, node.js, planning.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ СЕРВІСІВ ДЛЯ ПЛАНУВАННЯ ЗАХОДІВ ТА УЧАСТІ У НИХ.....	8
1.1 Аналіз сучасного стану проблеми	8
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	15
Висновки до розділу 1	15
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	17
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	17
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	18
Висновки до розділу 2.....	23
РОЗДІЛ 3 РОЗРОБКА ОНЛАЙН-СЕРВІСУ ДЛЯ ПЛАНУВАННЯ ЗАХОДІВ .	24
3.1 Практична реалізація об'єкта проектування.....	24
3.2 Посібник користувача	27
3.3 Захист інформаційно-комп'ютерної системи	32
3.4 Тестування та налагодження інформаційно-комп'ютерної системи ..	33
Висновки до розділу 3.....	35
ВИСНОВКИ	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39

ВСТУП

Актуальність теми. У сучасному світі, де швидкість життя постійно зростає, онлайн-сервіси для планування заходів набувають все більшої популярності. Це пов'язано з необхідністю оптимізації часу, зручності комунікації та можливістю організовувати події з мінімальними витратами. Використання технологій веб-розробки, таких як Vue.js та Node.js, дозволяє створювати високопродуктивні та масштабовані рішення для планування та управління заходами. Такий підхід забезпечує ефективне управління інформацією, автоматизацію процесів та інтеграцію з іншими сервісами, що підвищує загальну ефективність організаційних заходів.

Метою роботи є розробка онлайн-сервісу для планування заходів та участі у них на базі технологій Vue.js та Node.js, який забезпечить користувачам зручний та інтуїтивно зрозумілий інтерфейс, високу продуктивність і надійність.

Щоб досягнути поставленої мети потрібно вирішити такі завдання:

- здійснити аналіз сучасного стану проблеми та існуючих рішень;
- обґрунтувати вибір технологій;
- розробити архітектуру системи, включаючи фронтенд та бекенд;
- створити інтерфейс користувача за допомогою Vue.js;
- реалізувати основні функціональні можливості онлайн-сервісу для планування заходів;
- реалізувати серверну частину на базі Node.js
- забезпечити безпеку даних користувачів та їх конфіденційність;
- провести тестування розробленого сервісу для планування заходів та участі у них.

Об'єктом дослідження є процеси організації та управління заходами за допомогою інформаційних технологій.

Предметом дослідження є технології та методи розробки онлайн-сервісів для планування заходів, зокрема, використання фреймворку Vue.js для фронтенду та платформи Node.js для бекенду.

Практичне значення роботи полягає у розробці інтегрованого онлайн-сервісу, який забезпечує інтерактивний та адаптивний користувацький інтерфейс на базі Vue.js, що дозволяє користувачам зручно планувати, редагувати та керувати заходами. Цей онлайн-сервіс стане корисним інструментом як для особистого використання, так і для бізнес-сегменту, допомагаючи ефективно планувати та проводити заходи будь-якого масштабу.

РОЗДІЛ 1

АНАЛІЗ СЕРВІСІВ ДЛЯ ПЛАНУВАННЯ ЗАХОДІВ ТА УЧАСТІ У НИХ

1.1 Аналіз сучасного стану проблеми

Предметна область розроблюваного сервісу включає не тільки пошук заходів, але й їх створення. Захід організовується користувачем для групи інших зареєстрованих користувачів, із визначеною тематикою. Захід або подія – це зустріч користувачів за запрошенням організатора.

Для точного визначення вимог до системи, розглянемо існуючі аналоги. Виходячи з поставленої мети та завдань, було визначено наступні критерії оцінки подібних сервісів:

- можливість створювати та реєструвати події, персональний профіль з інформацією про користувача та його події, функція реєстрації на події, можливість спілкування між учасниками заходу, між учасником і організатором, або залишення відгуків про подію;
- інтуїтивно зрозумілий інтерфейс, де основні функції легко знайти, адаптивний дизайн інтерфейсу для телефонів або окремий мобільний додаток, інтерфейс, який коректно відображається на різних розмірах екранів, карта міста із зазначеними на ній подіями.

Далі наведемо коротку характеристику систем для організації і проведення заходів.

vFairs – за словами розробника [1] одна з найкращих платформ для управління подіями, що пропонує рішення для віртуальних, фізичних та гібридних заходів. Серед типів подій, які можна організувати за допомогою vFairs, є виставки, конференції, ярмарки вакансій, дні відкритих дверей, тощо. Платформа відзначається широким спектром інтеграцій із сторонніми сервісами та функціями доступності.

Основні функції даної платформи: реєстрація, продаж квитків, створення індивідуальних сторінок заходу з формою реєстрації, порядком денним і деталями, можливість оплати через платіжні шлюзи, такі як Stripe, PayPal,

управління даними учасників та групова реєстрація зі знижками, перевірка на місці, швидке друкування бейджів з QR-кодами та розпізнаванням облич, ведення статистики в реальному часі (рис. 1.1).

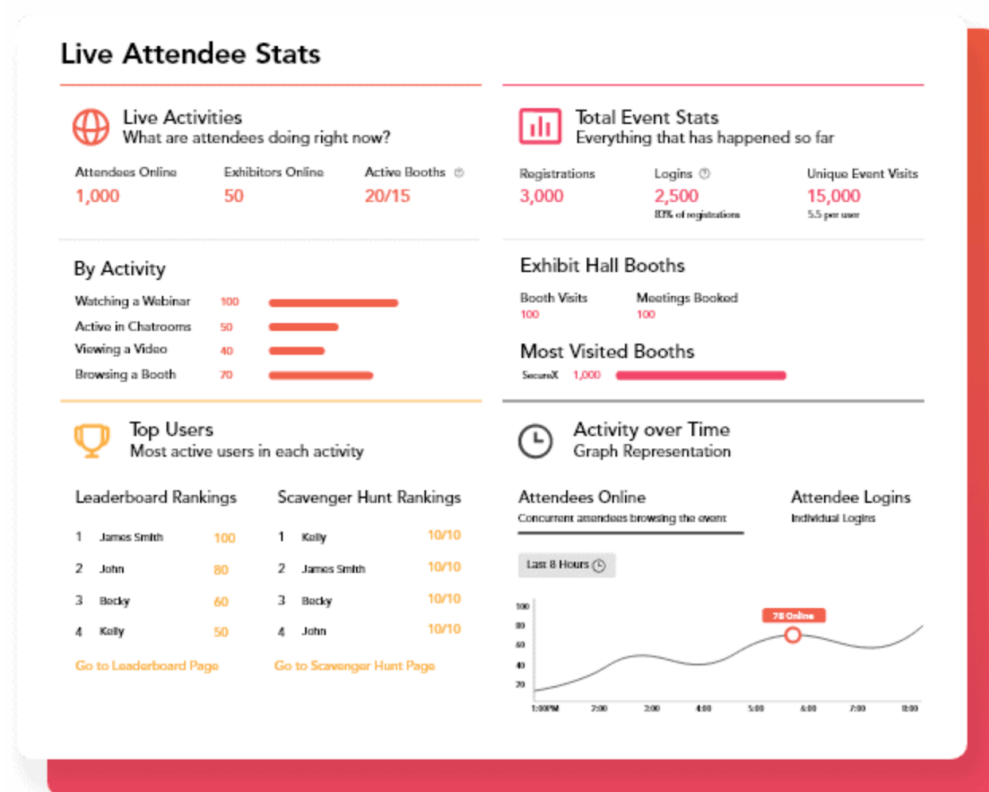


Рисунок 1.1 – Аналітика даних у платформі vFairs [1]

Мобільний додаток для цієї платформи дозволяє отримувати доступ до ресурсів, таких як брошури, відео та презентації, надає можливість додавати сеанси до індивідуального календаря.

Серед недоліків платформи виділяють високу ціну та необхідність додаткових налаштувань для безперебійної роботи під час великих подій.

Альтернативним рішенням є система RingCentral (Hopin) [2], яка характеризується наступними особливостями:

- підтримка до 10,000 учасників;
- можливість потокової трансляції та інтеграції з додатками для підвищення взаємодії учасників;
- висока ціна та недоліки в підтримці під час подій.

Svent – це ще один інструмент, призначений для управління подіями різних масштабів [3]. Він надає широкі можливості інтеграції, але має проблеми з продуктивністю та підтримкою під час живих подій.

Eventbrite (рис. 1.2) – це платформа для продажу квитків та управління платежами. Вона придатна для менших заходів, але потребує додаткових інструментів для повного управління подіями [4].

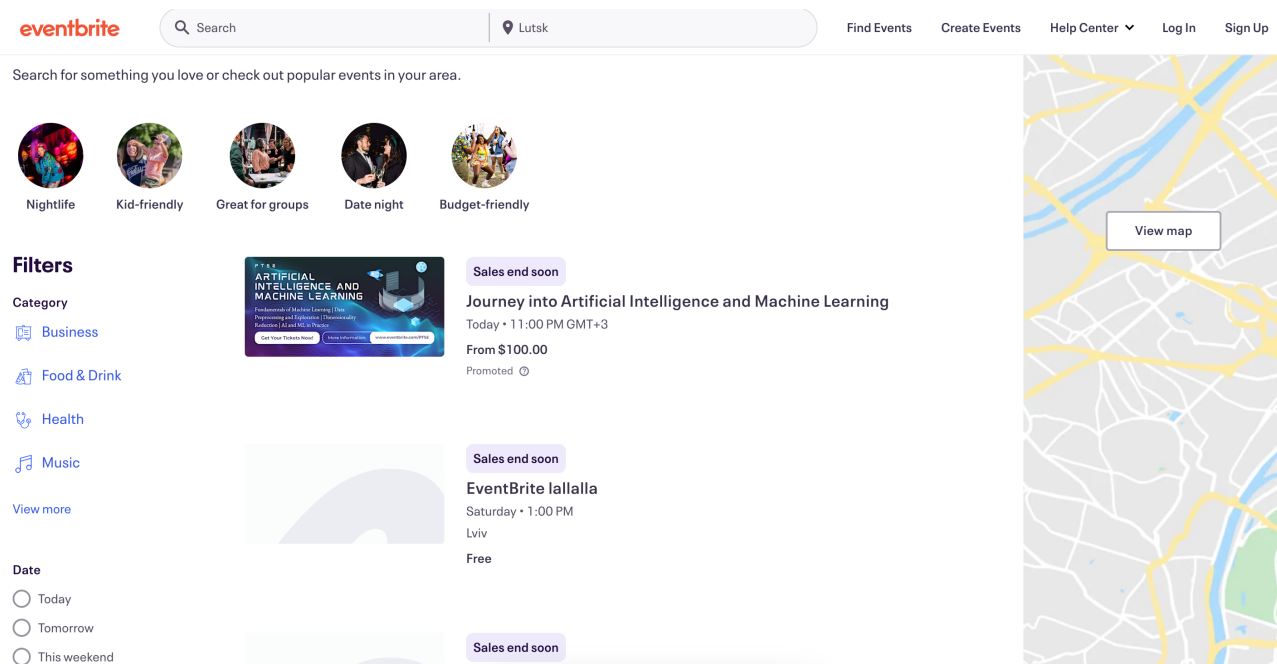


Рисунок 1.2 – Платформа Eventbrite [4]

Сервіс надає можливість користувачам переглядати, створювати та рекламувати місцеві події. Сервіс стягує оплату з організаторів подій в обмін на онлайн-послуги з продажу квитків, коли подія не є безкоштовною. В другій половині 2023 року Eventbrite оновила свої тарифні плани, обмеживши безкоштовні події 25 квитками, перш ніж вони почнуть стягувати комісію з організаторів.

Whova [5] – одна з найкращих платформ для управління подіями, яка допомагає заощадити час та повністю контролювати захід (рис. 1.3).

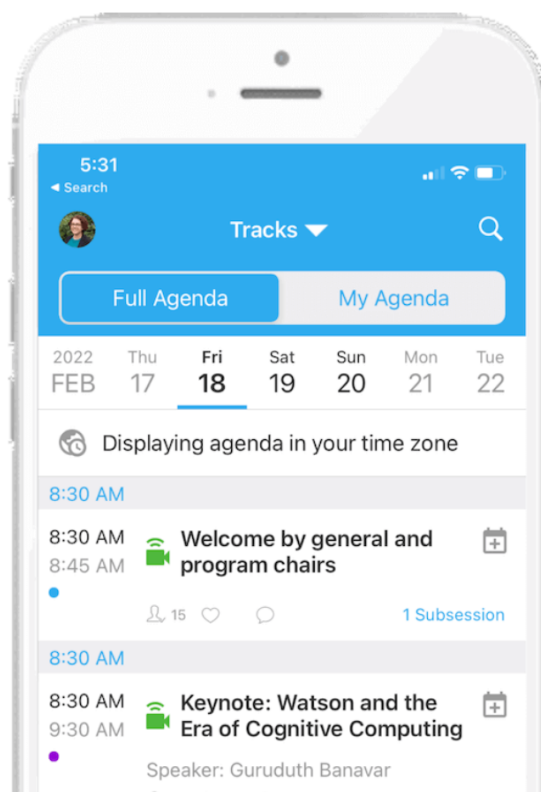


Рисунок 1.3 – Платформа Whova [5]

Платформа пропонує можливості керування спікерами, детальні аналітичні звіти, створення бейджів, управління відвідуваністю тощо. Проте, користувачі часто скаржаться на відсутність попереднього навчання роботи з платформою та живих сервісів підтримки.

Центр для спікерів дозволяє їм самостійно завантажувати інформацію про сесії, біографії, фото та інші матеріали. Платформа надає більше 16 готових шаблонів для сторінок спікерів, що полегшують процес налаштування.

Також доступні шаблони опитувань та опитувальників для зручного збору відгуків у реальному часі. Опитування можуть бути проведені через додаток або електронну пошту.

Whova включає дашборд для перегляду та аналізу даних в режимі реального часу та після заходу. Метрики включають продаж квитків, статус реєстрації учасників, результати опитувань та інші дані.

Серед недоліків системи виділяють відсутність налаштувань та складність в користуванні.

Zoom Events – платформа для проведення одноденних та багатоденних конференцій [6]. В ній відсутні деякі розширені функції, такі як налаштування виставкових залів, гейміфікація та можливість мережових взаємодій.

Ця платформа надає можливість організації безкоштовних та платних заходів різного типу та тривалості, а також підтримку одночасних сесій та їх запису для повторного перегляду.

Zoom Events надає інструменти для налаштування вигляду та стилю заходів відповідно до бренду компанії.

Також є дашборд для перегляду даних про взаємодію учасників, продаж квитків, генерування доходу тощо.

Недоліками даної системи є не найкраща ціна за можливості, які надає система, а також велике споживання ресурсів комп'ютера та випадкові зависання.

Timely Event Ticketing – це платформа, яка допомагає в управлінні квитками та календарями подій [7] (рис. 1.4).

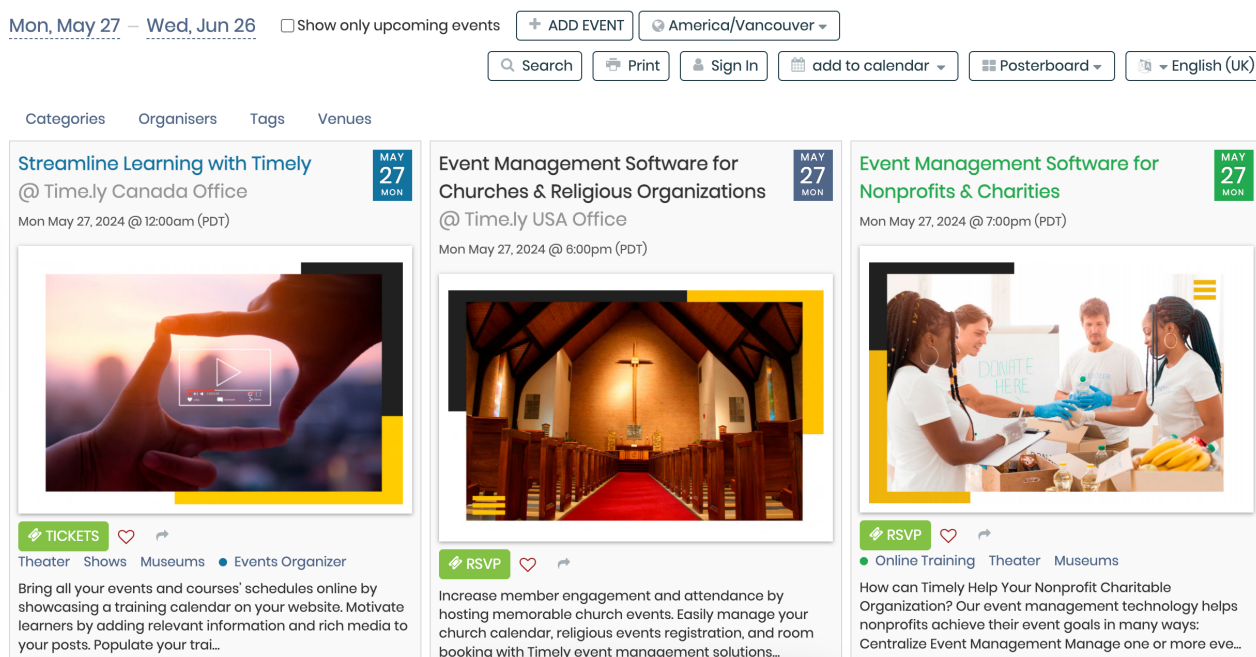


Рисунок 1.4 – Платформа Timely Event Ticketing [7]

Вона підходить для малого бізнесу та великих підприємств, але не пропонує всі можливості комплексного управління заходами.

Основні функції платформи: календар подій, можливість додавання зображень, відео та посилань, інструменти реєстрації, інтеграція з існуючим вебсайтом компанії, система продажу квитків.

Недолік даної платформи – складність процесу налаштування.

Airmeet [8] – платформа для управління подіями з різними рішеннями для віртуальних і гібридних заходів. Вона надає можливості управління квитками, інтеграції з CRM та кастомізації подій.

Основні функції Airmeet: продаж квитків та реєстрація, безпечні платіжні шлюзи для купівлі квитків онлайн, інтеграція з CRM.

Недоліки цієї платформи: тривалий процес входу в систему, обмежені можливості для управління квитками та учасниками.

GoTo Webinar – платформа для управління вебіарами та конференціями, яка пропонує маркетинг через електронну пошту та підтримку учасників. Вона орієнтована на віртуальні конференції та вебіари [9].

Основними функціями GoTo Webinar є автоматизація Email-маркетингу, вбудовані інструменти для управління email маркетинговими кампаніями, інтеграція з CRM для управління контактами та відповідями, управління учасниками, підтримка реєстрацій та продажу квитків онлайн через Stripe, мінімальні можливості для взаємодії учасників між собою.

GoTo Webinar пропонує функції, такі як спеціальні сторінки реєстрації, керування подіями, шаблони вебінарів, обмін відео, опитування, керування аудиторією тощо. Крім того, він дозволяє користувачам персоналізувати матеріал вебінару, використовуючи власні зображення, логотипи та кольори.

Далі можливості вищезазначених платформ, зведені в порівняльній таблиці 1.1.

Таблиця 1.1 – Функціональні можливості платформ для організації заходів

Функція	Платформа								
	vFairs	RingCentral (Hopin)	Cvent	Eventbrite	Whova	Zoom Events	Timely Event Ticketing	Airmeet	GoTo Webinar
Управління спікерами	Так	Так	Так	Ні	Так	Так	Ні	Так	Так
Живі опитування	Так	Так	Так	Так	Так	Так	Ні	Так	Так
Аналітика	Так	Так	Так	Так	Так	Так	Так	Так	Так
Реєстрація та продаж квитків	Так	Так	Так	Так	Так	Так	Так	Так	Так
Мережеві взаємодії	Так	Так	Так	Так	Так	Так	Ні	Так	Ні
Інтеграція з CRM	Так	Так	Так	Ні	Так	Ні	Ні	Так	Так
Email маркетинг	Так	Так	Так	Так	Так	Так	Ні	Так	Так
Кастомізація сторінок подій	Так	Так	Так	Так	Так	Так	Так	Так	Так
Служба підтримки	Так	Так	Так	Так	Так	Так	Так	Так	Так

Усі платформи є комерційними продуктами. Серед розглянутих платформ Timely Event Ticketing та Eventbrite надають можливість безкоштовного з деякими умовами та обмеженнями.

Отже, існує велика кількість платформ для організації і проведення зустрічей, які орієнтовані як на живі зустрічі, так і на вебінари. Недоліком багатьох платформ є їх ціна, складність використання, вимогливість до ресурсів. Розробка простої в користуванні, доступної і високопродуктивної платформи є актуальною проблемою.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Після аналізу сучасного стану проблеми для досягнення поставленої мети необхідно вирішити такі завдання:

- обґрунтувати вибір технологій Vue.js та Node.js для розробки онлайн-сервісу;
- розробити архітектуру системи, включаючи фронтенд та бекенд частини;
- створити інтерфейс користувача за допомогою Vue.js, забезпечивши його інтуїтивність та зручність;
- реалізувати основні функціональні можливості, такі як створення, редагування, видалення заходів тощо;
- реалізувати серверну частину на базі Node.js для обробки запитів користувачів, зберігання даних та управління ними;
- забезпечити безпеку даних користувачів та їх конфіденційність;
- провести тестування розробленого сервісу для планування заходів та участі у них;

Виконання цих завдань дозволить створити повноцінний онлайн-сервіс для планування заходів, який буде відповідати сучасним вимогам користувачів та забезпечувати високу якість наданих послуг.

Висновки до розділу 1

Розробка онлайн-сервісу для пошуку та створення заходів є актуальною задачею, враховуючи високу конкуренцію на ринку платформ для організації подій. Аналіз існуючих рішень показав, що хоча більшість платформ мають широкий функціонал, вони часто характеризуються високою ціною, складністю у використанні та вимогливістю до ресурсів. Це створює необхідність у створенні простої, доступної та продуктивної платформи, яка б відповідала сучасним вимогам користувачів.

Для досягнення цієї мети було обрано технології Vue.js і Node.js, які дозволяють створити інтуїтивний інтерфейс та забезпечити ефективну обробку запитів користувачів. Розробка системи включає створення архітектури з фронтенд та бекенд частинами, реалізацію основних функцій управління заходами, захист даних користувачів та забезпечення надійної взаємодії між компонентами системи. Виконання цих завдань дозволить створити сучасний сервіс для планування заходів, що забезпечуватиме високу якість наданих послуг та задовольнить потреби користувачів.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Сформулюємо наступні функціональні вимоги до розроблюваної системи для створення і проведення заходів:

- можливість користувачів створити обліковий запис на платформі;
- користувачі мають різні ролі: модератор, гість, звичайний користувач;
- аутентифікація користувачів за допомогою паролів;
- можливість фільтрувати події за різними критеріями;
- учасники можуть спілкуватися в чаті після аутентифікації;
- користувачі можуть реєструватися для участі в заходах;
- користувачі можуть управляти своїми профілями та налаштуваннями;
- можливість створювати та організовувати власні заходи;
- відображення місць проведення заходів на інтерактивній карті.

Нова система для менеджменту подій розробляється на основі клієнт-серверної архітектури (рис. 2.1).

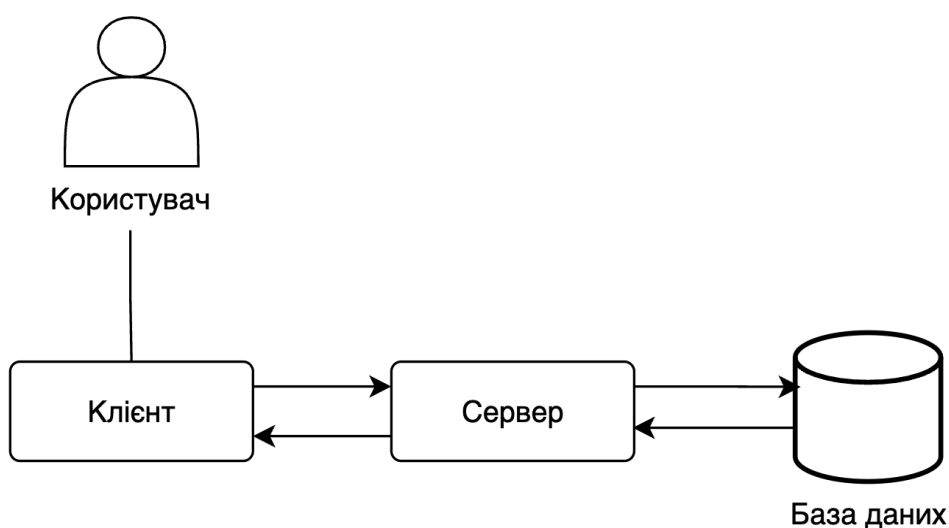


Рисунок 2.1 – Клієнт-серверна архітектура

Обрана трирівнева архітектура включає клієнта, сервер і базу даних. Це дозволяє розділити процеси відправки запитів і надання результатів користувачеві на різних комп'ютерах, що забезпечує їх незалежну роботу. Запити від необмеженої кількості клієнтів передаються на сервери, спеціально підготовлені для обробки цих запитів. У нашій системі використовуватиметься один сервер.

Серед переваг клієнт-серверної архітектури можна виділити централізоване управління даними через єдиний сервер, забезпечення інформаційної безпеки, підвищення спільної продуктивності, масштабованість, що дозволяє збільшувати кількість клієнтів та серверів безкоштовно.

Недоліки клієнт-серверної архітектури: ризик виходу з ладу окремого сервера, можливість перевантаження сервера великою кількістю клієнтів.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для розробки клієнтської частини нашого проєкту було обрано кілька популярних мов програмування та фреймворків, які мають великий функціонал. JavaScript є основною технологією для клієнт-серверної розробки, підтримуючи різні стилі програмування, такі як об'єктно-орієнтоване, імперативне та функціональне програмування [10].

Vue.js – прогресивний фреймворк, який значно спрощує процес створення користувацьких інтерфейсів завдяки своїм численним функціям [11].

HTML служить основою для створення веб-додатків, підтримуючи інтеграцію аудіо, відео, графіки, анімації та геолокації [12]. Щоб спростити роботу з CSS, ми використовуємо SASS (SCSS), який додає гнучкість та розширені можливості [13].

На серверній стороні було обрано Node.js, який розширює можливості JavaScript за межами браузера, дозволяючи писати серверний код для веб-сторінок та додатків [14].

Також використовувався Nuxt.js – фреймворк для створення універсальних додатків на базі Vue.js та Node.js. Він дозволяє використовувати серверний рендеринг, що покращує продуктивність [15].

Для передачі даних між клієнтом і сервером використовується HTTP (Hyper Text Transfer Protocol), який є протоколом прикладного рівня. Він дозволяє клієнту відправляти запити на сервер для отримання інформації або виконання дій, а сервер у відповідь надсилає відповідні повідомлення [16].

Як базу даних ми обрали MongoDB, документно-орієнтовану систему, яка не вимагає опису схеми таблиць [17]. Вона використовує JSON-подібні документи для зберігання даних і пропонує низку переваг, таких як зберігання даних, що не є жорстко пов'язаними, можливість надання прав різним категоріям користувачів, відсутність необхідності в SQL, можливість створення колекцій з документами різної структури та легку масштабованість.

Для проєктування використовується методологія ICONIX, яка включає чотири основні діаграми UML: діаграма варіантів використання, діаграма послідовності, діаграма придатності та діаграма класів [18]. Для нашого проєкту достатньо створити діаграму варіантів використання, модель предметної області та діаграму класів.

У нашій системі існує три типи користувачів: гість, авторизований користувач і модератор з різним рівнем прав доступу до ресурсу і різними можливостями.

Гість, який заходить на сайт має можливість бачити головну сторінку з подіями, а також здійснювати вхід у власний аккаунт або реєструвати новий аккаунт (рис. 2.2). Головна сторінка має доступною для перегляду обмежену кількість подій без можливості їх детального перегляду, коментування і підписування на відвідування. Для того, щоб отримати ці та інші додаткові можливості гість повинен пройти процедуру входу зі своїм логіном і паролем або зареєструватися як новий користувач.

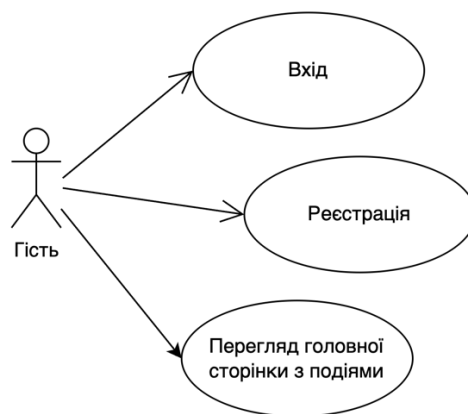


Рисунок 2.2 – Діаграма варіантів використання для гостя

В порівнянні з гостем у авторизованого користувача з'являються нові варіанти використання. Користувач може переглядати власний аккаунт, створювати подію, записуватися на події, створені іншими користувачами, видаляти подію, шукати заходи, обирати категорії, а також залигати коментарі до подій (рис. 2.3).

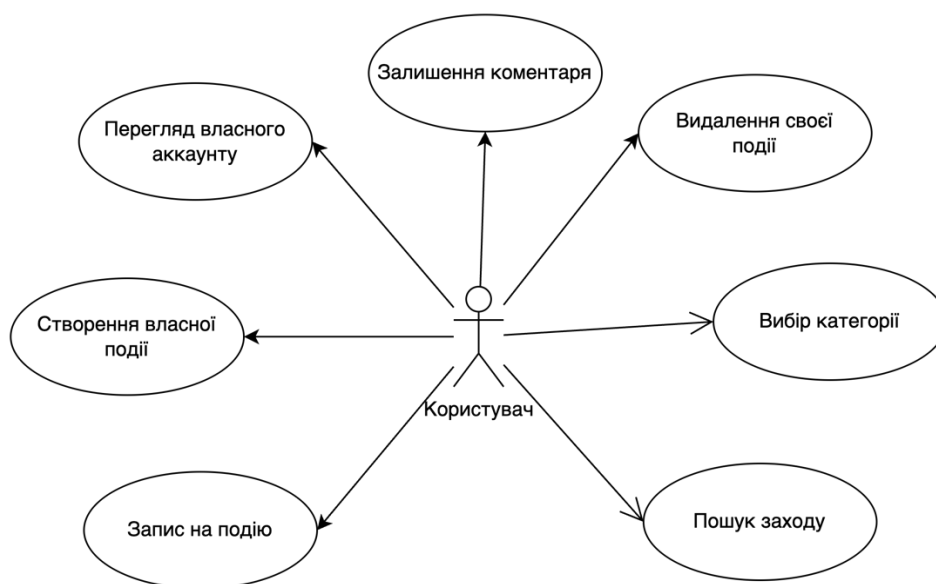


Рисунок 2.3 – Діаграма варіантів використання для користувача

Модератор володіє найбільшими можливостями в порівнянні з двома іншими типами користувачів, зокрема він може видаляти події і коментарі, які були створені іншими користувачами (рис. 2.4).

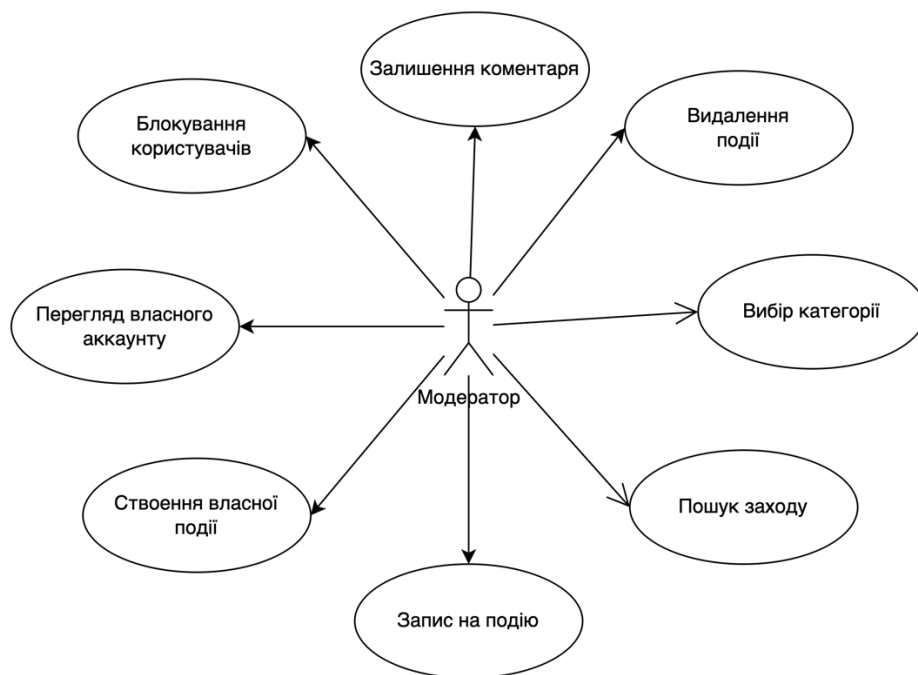


Рисунок 2.4 – Діаграма варіантів використання для модератора

Модель предметної області описує сутності на цьому етапі проектування та очевидні зв'язки включення. Сутності – це екранні вікна, які містять елементи на екрані. На рисунку 2.5 показана діаграма, яка описує модель предметної області.

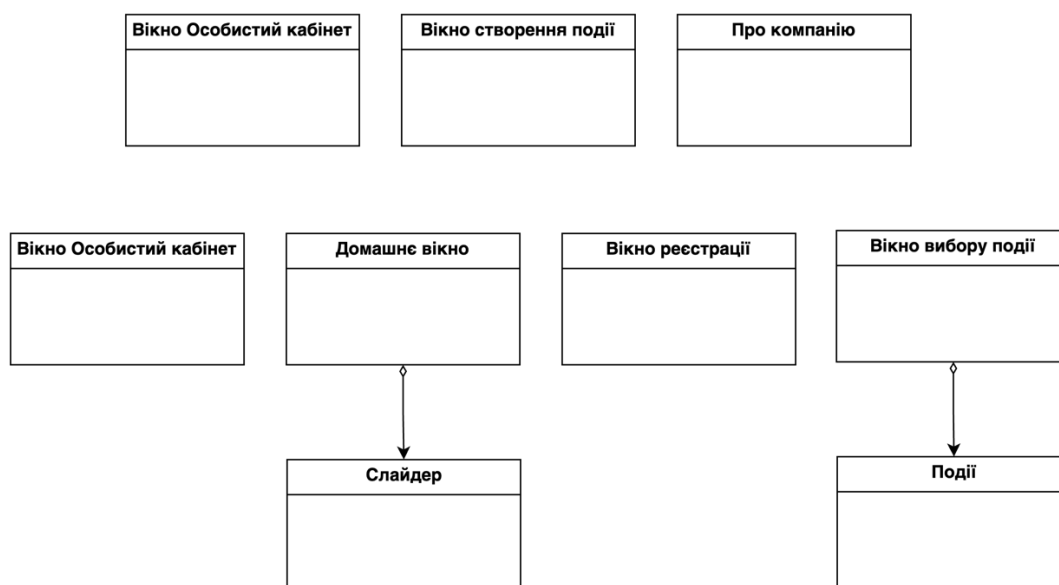


Рисунок 2.5 – Модель предметної області

Діаграма класів показує структуру програми та взаємозв'язок між елементами. У цьому випадку в якості елементів виступають класи. На діаграмі показані змінні та функції, які використовує клас. Діаграма класів для розроблюваного додатку показана на рисунку 2.6.

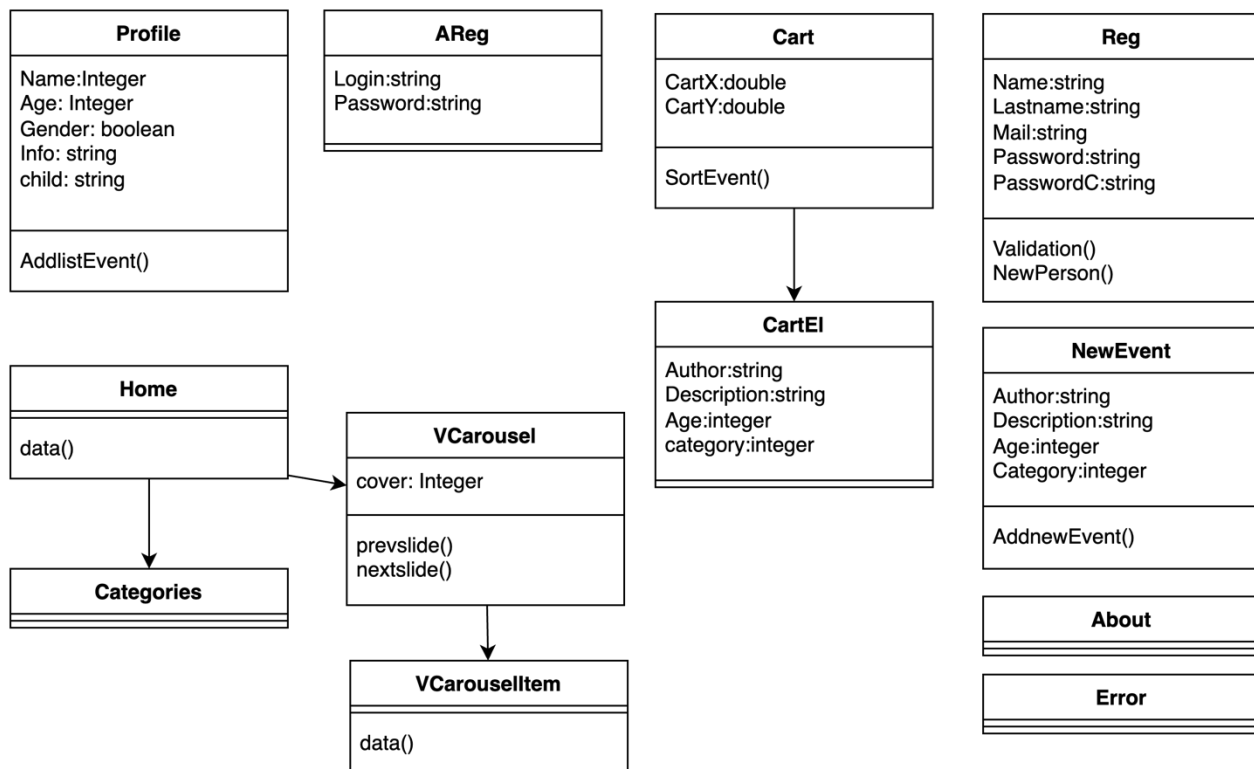


Рисунок 2.6 – Діаграма класів

Логічна модель даних являє собою сукупність сутностей і зв'язків. Оскільки в цій кваліфікаційній роботі використовується документно-орієнтована база даних, то у якості даних є дві сутності: події та користувачі. Події включають інформацію про назву, дату, час, місце проведення, організатора та список учасників. Користувачі, в свою чергу, містять дані про ім'я, контактну інформацію, а також перелік заходів, у яких вони брали участь або планують взяти участь (рис. 2.7).



Рисунок 2.7 – Логічна модель бази даних

Висновки до розділу 2

Було визначено функціональні вимоги до розроблюваної системи для створення і проведення заходів, зокрема можливість створення облікових записів користувачів, підтримка різних ролей, аутентифікація, фільтрація подій, чат для учасників, реєстрація на заходи та інтерактивна карта місць проведення заходів. Обрана клієнт-серверна архітектура дозволяє ефективно розподіляти навантаження між клієнтом, сервером і базою даних, забезпечуючи централізоване управління даними та підвищену продуктивність.

Для реалізації проекту були обрані сучасні технології та інструменти, зокрема Vue.js і Node.js для фронтенд та бекенд частин відповідно, а також MongoDB як базу даних. Методологія ICONIX використовується для проектування системи, що включає створення діаграм UML для візуалізації варіантів використання та структури програми.

В результаті виконання поставлених завдань, очікується створення повнофункціонального, зручного та надійного онлайн-сервісу для планування та проведення заходів.

РОЗДІЛ 3

РОЗРОБКА ОНЛАЙН-СЕРВІСУ ДЛЯ ПЛАНУВАННЯ ЗАХОДІВ

3.1 Практична реалізація об'єкта проектування

Інтерфейс реалізований за допомогою популярного фреймворку «Vue.js». Розробка базується на формуванні структури додатку за допомогою HTML з використанням методології BEM (Block, Element, Modifier), яка є компонентним підходом до веб-розробки. Цей підхід полягає у поділі інтерфейсу на незалежні блоки, що дозволяє легко і швидко створювати інтерфейси будь-якої складності та повторно використовувати існуючий код, уникаючи його дублювання.

Навігація між сторінками здійснюється за допомогою бібліотеки «Vue Router». Для визначення маршрутів використовуються такі компоненти: HomeComponent для головної сторінки, AboutComponent для сторінки про компанію, PageComponent для динамічних сторінок та ProfilComponent для профілю користувача.

У клієнтській частині були реалізовані наступні компоненти:

- About-comprant (сторінка про компанію);
- Areg-companу (сторінка реєстрації);
- Card-companу (сторінка пошуку подій);
- CardEl-companу (сторінка подій);
- Error-companу (сторінка помилки);
- Home-companу (головна сторінка);
- Myprofile-companу (сторінка профілю);
- Reg-companу (сторінка входу);
- V-carousel(фотослайдер);
- V-carousel-item (елементи каруселі).

У даній роботі для обміну даними між браузером (клієнтом) і сервером використовується JSON. Такий формат відмінно підходить для передачі паролів, логінів та іншої інформації користувача, а потім запису її в базу даних. Саме в цьому форматі будуть передаватися дані користувача і події (лістинг 3.1).

Лістинг 3.1 – Дані про користувача в форматі JSON

```
{
  "Users": [
    {
      "id": "0",
      "name": "Dmytro",
      "surname": "Kaluzhnyi",
      "login": "dmytro",
      "mail": "dmytro@demomail.com",
      "password": "*****",
      "city": "Lutsk",
      "gender": "Male",
      "description": "Опис",
      "admrights": "morerator"
    }
  ],

  "Events": [
    {
      "id": "1",
      "city": "Lutsk",
      "category": "Music",
      "organizer": "0",
      "members": ["2", "4", "7"]
      "description": "Опис події"
      "address": "Л.Українки"
    }
  ]
}
```

Кінець лістингу 3.1

Завдяки API Google Maps була реалізована карта з мітками (лістинг 3.2).

Лістинг 3.2 – Приклад використання Google Maps API

```
<!DOCTYPE html>
<html>
  <head>
```

```

<title>Простий приклад карти</title>
<!-- Підключення Google Maps API з вашим API-ключем -
->
<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API
_KEY&callback=initMap" async defer></script>
<script>
    // Функція для ініціалізації карти
    function initMap() {
        // Створення карти, центрованої на Луцьку
        var map = new
google.maps.Map(document.getElementById('map'), {
            center: {lat: 50.7472, lng: 25.3254},
            zoom: 12
        });

        // Додавання першої мітки
        var marker1 = new google.maps.Marker({
            position: {lat: 50.7472, lng: 25.3254},
            map: map,
            title: 'Marker 1'
        });

        // Додавання другої мітки
        var marker2 = new google.maps.Marker({
            position: {lat: 50.7525, lng: 25.3644},
            map: map,
            title: 'Marker 2'
        });
    }
</script>
</head>
<body>
    <!-- Контейнер для відображення карти -->
    <div id="map" style="height: 500px; width:
100%;"></div>
</body>
</html>

```

Функція «initMap()» ініціалізує карту з центром у м. Луцьк та додає дві мітки на карту. Мітки позначаються на карті за допомогою об'єкта «google.maps.Marker».

Для створення серверної частини використовується комбінація Node.js, Express та Nuxt.js. Express є оптимальним фреймворком для розробки на Node.js завдяки своїм перевагам: схожий синтаксис, простота використання, можливість швидкої інтеграції та автоматична обробка кодування і читання файлів. Nuxt.js забезпечує швидке розгортання клієнт-серверного додатку.

Крім того, утиліта командного рядка Nodemon використовується для автоматичного перезавантаження сервера під час розробки.

Складові серверної частини включають:

- registration – додає нових користувачів до бази даних та перевіряє введені дані;
- authorization – звіряє введені дані з базою даних, видає повідомлення про помилку або авторизує користувача;
- drawdata – відображає дані про події на головній сторінці;
- drawactivities – відображає події у вікні вибору події;
- activities – отримує інформацію про захід та обробляє запити на участь у заході;
- account – відображає та змінює інформацію про користувача.

База даних була створена за допомогою MongoDB Compass, що надає візуальне представлення бази даних.

Були створені база даних «DataDip» та дві колекції: «Events» для зберігання інформації про події і «Users» для зберігання інформації про користувачів.

3.2 Посібник користувача

Щоб скористатися сервісом вперше, гість повинен відвідати головну сторінку сайту (рис. 3.1).

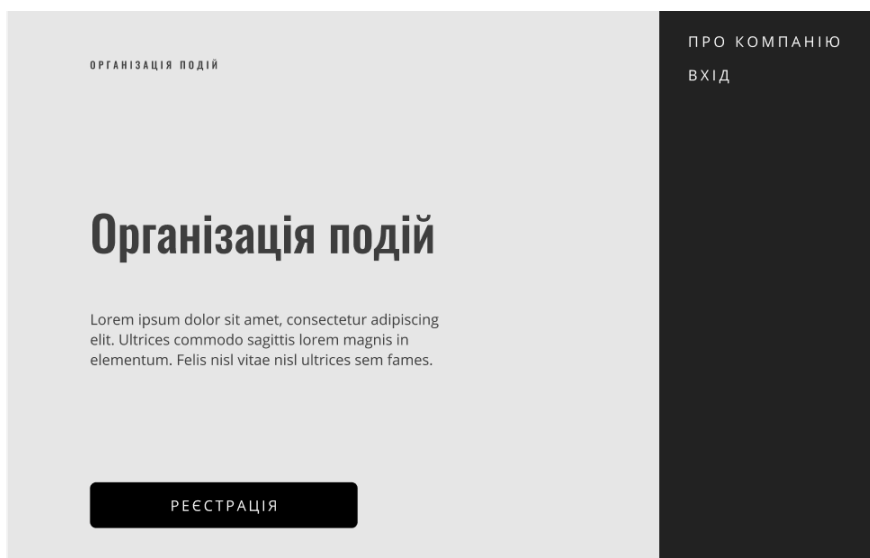


Рисунок 3.1 – Головна сторінка сервісу

На головній сторінці може побачити елементи меню «Про компанію», «Вхід», кнопку «Реєстрація».

Після прокручування сторінки вниз користувач може побачити слайдер з інформацією про сайт, слайдер з топ-5 популярних подій у нижній частині сторінки і популярні категорії подій (рис. 3.2).

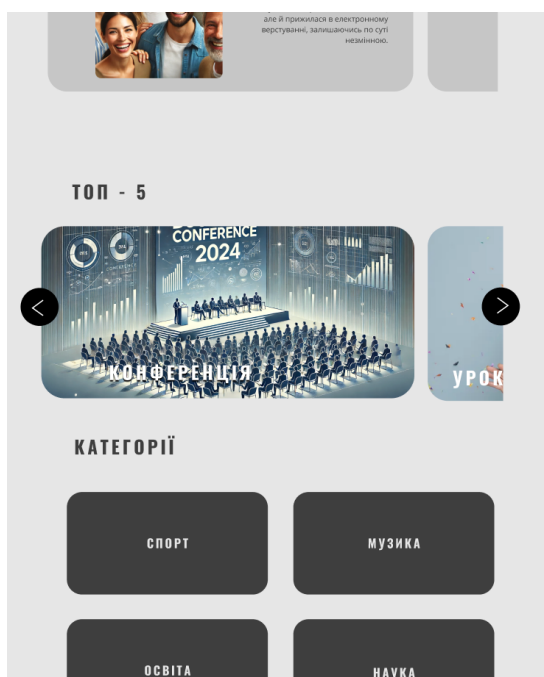
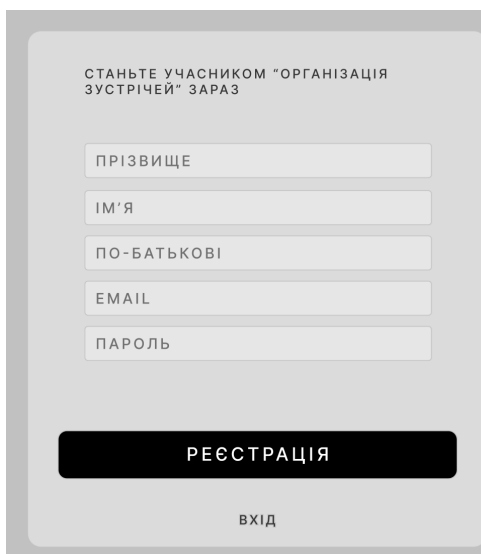


Рисунок 3.2 – Нижня частина головної сторінки

Після перегляду інформації на сторінках, для продовження гість повинен натиснути кнопку «Реєстрація» (рис. 3.3).



СТАНЬТЕ УЧАСНИКОМ "ОРГАНІЗАЦІЯ
ЗУСТРІЧЕЙ" ЗАРАЗ

ПРИЗВИЩЕ

ІМ'Я

ПО-БАТЬКОВІ

EMAIL

ПАРОЛЬ

РЕЄСТРАЦІЯ

ВХІД

Рисунок 3.3 – Форма реєстрації користувача

У цьому вікні гість повинен ввести свої дані і натиснути кнопку «Реєстрація», після цього він потрапляє у вікно входу (рис. 3.4).



ЛОГІН

ПАРОЛЬ

ВХІД

РЕЄСТРАЦІЯ

Рисунок 3.4 – Вікно входу в систему

Після реєстрації гість отримує статус користувача. Авторизований користувач вводить дані, натискає кнопку «Увійти». Після цього він потрапляє у вікно вибору події (рис. 3.5).



Рисунок 3.5 – Вікно вибору події

У цьому вікні користувач бачить наступні елементи:

- кнопка «Місце зустрічі», яка відкриває головне вікно;
- кнопка «Мій обліковий запис» відкриває вікно «Мій профіль»;
- фільтр вибору подій;
- перелік подій.

Для участі в події користувачу потрібно вибрати подію зі списку і натиснути на неї, після чого відкриється вікно обраної події (рис. 3.6).

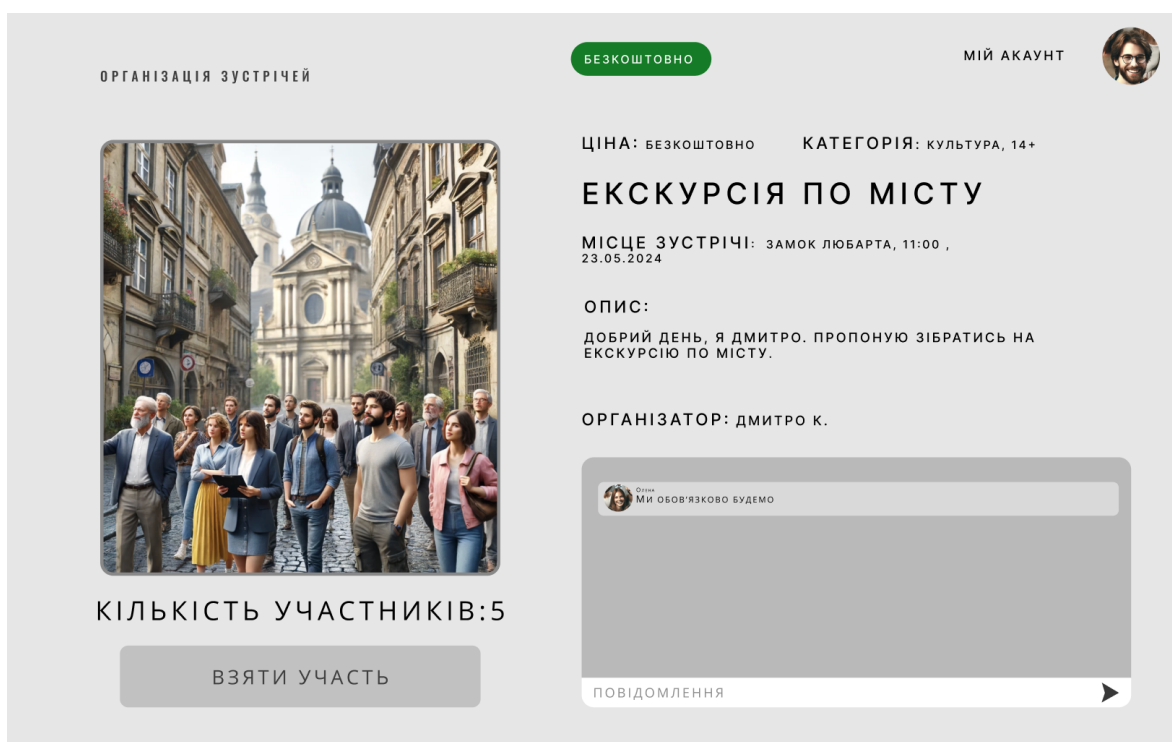


Рисунок 3.6 – Вікно обраної події

У цьому вікні користувач бачить наступні елементи:

- фото, додане автором заходу;
- опис заходу, що включає ціну, категорію, назву, текстовий опис і ім'я організатора;
- можливість спілкуватися з учасниками заходу.

Після ознайомлення з інформацією, користувач може натиснути кнопку «Взяти участь», що додає його до списку учасників заходу, і надає можливість писати в чаті.

Щоб створити власну подію, користувач повинен натиснути кнопку «Мій обліковий запис», після чого він буде перенаправлений до вікна особистого кабінету.

У цьому вікні користувач бачить такі елементи як фото, додане користувачем, інформація про користувача, список подій, на які підписаний користувач, кнопки «Редагувати» та «Створити подію» і список подій, які створив користувач.

З вікна «Особистий кабінет» можна перейти до вікна редагування особистої інформації. Після внесення змін користувач натискає кнопку збереження і повертається до вікна «Мій обліковий запис».

У головному вікні доступна кнопка «Про компанію», яка відкриває вікно з описом проекту.

При спробі перейти на неіснуючу сторінку з'являється вікно помилки (рис. 3.7).



Рисунок 3.7 – Вікно повідомлення про неіснуючу сторінку

У цьому вікні користувач може натиснути кнопку «На домашню», щоб повернутися на головну сторінку сервісу.

3.3 Захист інформаційно-комп'ютерної системи

Захист розроблюваної системи забезпечується через впровадження ряду заходів, що гарантують конфіденційність, цілісність та доступність даних користувачів.

Перш за все, було впроваджено сучасні методи аутентифікації та авторизації для безпечного входу користувачів. Паролі зберігаються у хешованому вигляді з використанням алгоритмів, таких як bcrypt, що ускладнює їх розшифрування у разі компрометації бази даних.

Також реалізовано систему ролей та прав доступу для користувачів, що дозволяє контролювати доступ до різних функціональних можливостей системи.

Дані, що передаються між клієнтом і сервером, шифруються за допомогою SSL/TLS, забезпечуючи захист від перехоплення та маніпуляцій. Чутлива інформація, така як особисті дані користувачів, зберігається у зашифрованому вигляді в базі даних, що унеможлиблює їх використання у разі несанкціонованого доступу.

Система захищена від поширених веб-атак, таких як SQL-ін'єкції, XSS та CSRF, через використання параметризованих запитів, валідації та очищення введених даних. Додатково було впроваджено фаєрвол для веб-додатків (WAF), який моніторить та захищає HTTP-запити від шкідливого трафіку.

Для відстеження підозрілої активності та реагування на можливі інциденти безпеки впроваджено систему логування та моніторингу.

При використанні системи потрібно регулярно проводити аудит безпеки системи, включаючи перевірку коду та вразливостей, щоб забезпечити відповідність сучасним стандартам та вимогам безпеки.

Таким чином, комплексний підхід до захисту інформаційно-комп'ютерної системи гарантує захист даних користувачів та стабільну роботу сервісу.

3.4 Тестування та налагодження інформаційно-комп'ютерної системи

Для тестування розробленого онлайн-сервісу використовувалися різні інструменти та методи. Основними засобами є:

- Jest – фреймворк для тестування JavaScript-коду. Він дозволяє писати тести для окремих компонентів та перевіряти їхню роботу в ізоляції;
- Cypress – інструмент для енд-ту-енд тестування веб-додатків. Він дозволяє автоматизувати тестування користувацького інтерфейсу та взаємодії з веб-сторінками;
- Postman – інструмент для тестування RESTful API. Використовується для перевірки правильності роботи бекенду та взаємодії між клієнтом і сервером
- MongoDB Compass – інструмент для роботи з базою даних MongoDB, який дозволяє перевіряти коректність збережених даних.

У лістингу 3.3 наведено приклад тестування одного з компонентів сервісу за допомогою Jest. Компонент: EventCard.vue – відображає інформацію про окремий захід.

Лістинг 3.3 – Тестування компонентів сервісу за допомогою Jest

```
// EventCard.vue

<template>
  <div class="event-card">
    <h2>{{ event.title }}</h2>
    <p>{{ event.description }}</p>
    <button @click="register">Register</button>
  </div>
</template>

<script>
export default {
  props: {
    event: Object
  },
```

```

methods: {
  register() {
    this.$emit('register', this.event.id);
  }
}
}
</script>

```

Кінець лістингу 3.3

Перевірка коректного рендерингу компонента та обробки події кліку на кнопку показані в лістингу 3.4.

Лістинг 3.4 – Перевірка рендерингу компонента та обробки події кліку

```

// EventCard.spec.js
import { shallowMount } from '@vue/test-utils'
import EventCard from '@components/EventCard.vue'

describe('EventCard.vue', () => {
  it('renders event title and description', () => {
    const event = { title: 'Test Event', description: 'This is a test event' }
    const wrapper = shallowMount(EventCard, {
      propsData: { event }
    })
    expect(wrapper.text()).toContain('Test Event')
    expect(wrapper.text()).toContain('This is a test event')
  })

  it('emits register event with event id when button is clicked', async () => {
    const event = { id: 1, title: 'Test Event', description: 'This is a test event' }
    const wrapper = shallowMount(EventCard, {
      propsData: { event }
    })
    await wrapper.find('button').trigger('click')
  })
})

```

```
expect(wrapper.emitted().register).toBeTruthy()  
expect(wrapper.emitted().register[0]).toEqual([1])  
})  
})
```

Кінець лістингу 3.4

Перший тест перевіряє, чи правильно рендериться компонент з переданими пропсами «title» та «description».

Другий тест перевіряє, чи відправляється подія «register» з правильним «event id», коли користувач натискає на кнопку «Register».

Використання автоматизованих тестів дозволило виявляти та виправляти помилки на ранніх етапах розробки, що значно підвищує якість кінцевого продукту.

Висновки до розділу 3

В процесі розробки було створено онлайн-сервіс для планування заходів з використанням технологій Vue.js та Node.js.

Створено інтуїтивно зрозумілий інтерфейс користувача з використанням Vue.js. Реалізовано основні компоненти інтерфейсу, такі як форма реєстрації, вхід до системи, відображення списку подій, детальна інформація про подію, профіль користувача та інші.

Розроблено серверну частину на основі Node.js та Express, яка забезпечує обробку запитів від клієнтів, аутентифікацію та авторизацію користувачів, роботу з базою даних MongoDB для збереження інформації про користувачів та події.

Впроваджено взаємодію між фронтендом та бекендом через RESTful API. Використання JSON для обміну даними забезпечує ефективну та безпечну передачу інформації.

Застосування Jest, Cypress, Postman та MongoDB Compass допомагає охопити різні аспекти тестування – від модульного до інтеграційного та енд-ту-енд тестування.

ВИСНОВКИ

Проведений аналіз сучасного стану проблеми показав, що існує велика кількість платформ для організації та управління заходами, проте багато з них характеризуються високою вартістю, складністю у використанні та вимогливістю до ресурсів.

Вибір технологій Vue.js та Node.js для розробки онлайн-сервісу було обґрунтовано їхніми перевагами у створенні високопродуктивних, масштабованих та інтуїтивно зрозумілих веб-додатків. Vue.js забезпечує швидку та ефективну розробку інтерфейсу користувача, тоді як Node.js дозволяє створювати продуктивний та масштабований сервер для обробки запитів.

Було розроблено архітектуру системи, яка включає фронтенд на основі Vue.js та бекенд на базі Node.js. Архітектура побудована за принципом клієнт-серверної моделі, що дозволяє забезпечити ефективний розподіл завдань між клієнтською та серверною частинами, покращити продуктивність та масштабованість системи, а також забезпечити централізоване управління даними.

Створення інтерфейсу користувача за допомогою Vue.js дозволило забезпечити високу інтуїтивність та зручність у використанні. Компонентний підхід, запропонований Vue.js, дозволив розділити інтерфейс на незалежні блоки, що спрощує їхню розробку, тестування та повторне використання.

Реалізація основних функціональних можливостей, таких як створення, редагування та видалення заходів, управління списком учасників, була виконана з використанням сучасних методів та підходів.

Реалізація серверної частини на базі Node.js дозволила створити продуктивний та масштабований сервер для обробки запитів користувачів. Використання Express.js спростило розробку серверних маршрутів та обробку запитів, а MongoDB забезпечила надійне зберігання даних про користувачів та події.

Для забезпечення захисту даних користувачів та їх конфіденційності були впроваджені сучасні методи шифрування даних, захисту від атак типу SQL ін'єкцій та XSS, а також використання безпечних протоколів передачі даних.

Проведення тестування системи з використанням таких інструментів, як Jest, Cypress та Postman, дозволило виявити та виправити помилки на ранніх етапах розробки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hassan, S. H. 10 Best Event Management Software in 2024. URL: <https://www.vfairs.com/blog/best-event-management-software/https://www.vfairs.com/blog/best-event-management-software/> (дата звернення: 07.02.2024).
2. RingCentral for Microsoft Teams. URL: <https://www.ringcentral.com/rc-events.html> (дата звернення: 09.02.2024).
3. Cvent | Event Platform for In-person, Virtual, and Hybrid Events & Webinars. URL: <https://www.cvent.com/> (дата звернення: 09.02.2024).
4. Eventbrite. URL: <https://www.eventbrite.com/> (дата звернення: 09.02.2024).
5. Whova: Award-winning Event Apps & Event Management. URL: <https://whova.com/> (дата звернення: 15.02.2024).
6. Transform Your Hybrid & Virtual Event Experiences. Zoom. URL: <https://www.zoom.com/en/products/event-platform/> (дата звернення: 15.02.2024).
7. Event Ticketing Software to Sell More Tickets Online | Timely. URL: <https://time.ly/solutions/event-ticketing-software/> (дата звернення: 15.02.2024).
8. Airmeet | AI powered Webinar platform with deep insights. URL: <https://www.airmeet.com/> (дата звернення: 02.03.2024).
9. Hassle-free webinar software. URL: <https://www.goto.com/webinar> (дата звернення: 02.03.2024).
10. JavaScript | MDN. (n.d.). MDN Web Docs. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата звернення: 12.04.2024).
11. Документація Vuejs. URL: <https://vuejs.org/v2/guide/> (дата звернення: 12.04.2024).
12. HTML і CSS довідник українською. URL: <https://html-css.co.ua/> (дата звернення: 18.04.2024).
13. Sass: Syntactically Awesome Style Sheets. URL: <https://sass-lang.com/> (дата звернення: 18.04.2024).

14. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/en> (дата звернення: 23.05.2024).

15. Nuxt: The Intuitive Vue Framework. URL: <https://nuxt.com/> (дата звернення: 23.05.2024).

16. HTTP | MDN. (2024, March 3). MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP> (дата звернення: 25.05.2024).

17. MongoDB. URL: <https://www.mongodb.com/> (дата звернення: 29.05.2024).

18. ICONIX. Wikipedia. URL: <https://en.wikipedia.org/wiki/ICONIX> (дата звернення: 29.05.2024).