

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування
імені адмірала Макарова
Херсонський навчально-науковий інститут

Кафедра інформаційних технологій та фізико-математичних дисциплін

«Допущений до захисту»
Завідувач кафедри



д.т.н., доцент Гучек П.Й.
«22» червня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
до кваліфікаційної роботи на здобуття ступеня вищої освіти
«бакалавр»

на тему: “Розробка програмного забезпечення web застосунку «Eventor»”

Виконала: студентка IV курсу, групи 4157
спеціальності
121 - Інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми
Інженерія програмного забезпечення
(назва освітньої програми)



Яновська К.О.

(прізвище та ініціали)

Керівник



Гайдаєнко О.В.

(прізвище та ініціали)

Рецензент



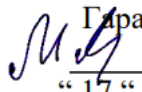
Михелєв І.Л.

(прізвище та ініціали)

м. Миколаїв-Херсон - 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова
ХЕРСОНСЬКИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ

Кафедра інформаційних технологій та фізико-математичних дисциплін
 Галузь знань 12 “Інформаційні технології”
(шифр і назва)
 Спеціальність 121 “Інженерія програмного забезпечення”
(шифр і назва)
 Освітня програма “Інженерія програмного забезпечення”
(назва)

«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми

 Літвінова М.Б.
 “ 17 ” березня 2024р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студентки Яновської Ксенії Олександрівни
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи “Розробка програмного забезпечення web застосунку «Eventor»”

Керівник роботи Гайдаєнко Оксана Володимирівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені розпорядженням по інституту від “04” березня 2024 року № 01

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1 Титульний аркуш, ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ (ДИПЛОМНУ) РОБОТУ, АНОТАЦІЯ (українською, англійською), ЗМІСТ, ПЕРЕЛІК УМОВНИХ ОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ТА ТЕРМІНІВ (при необхідності).

4.2 ВСТУП

4.3 ОПИС ПРЕДМЕТНОЇ ГАЛУЗІ, ПОСТАНОВКА ЗАДАЧІ

4.4 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (ПЗ)(зовнішні специфікації ПЗ, стани програми при виконанні, структура програми, структура даних, потоки даних та управління, проект ПЗ на рівні програмних модулів, випробування ПЗ)

4.5 РЕЗУЛЬТАТИ РОЗРОБКИ

4.6 РОЗДІЛ З ОХОРОНИ ПРАЦІ (ОХОРОНИ ОТОЧУЮЧОГО СЕРЕДОВИЩА) _____

4.7 ВИСНОВКИ

4.8 СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

4.9 ДОДАТКИ (технічне завдання, опис програми, текст програми, програма та методика

випробувань ПЗ, інструкція користувача)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

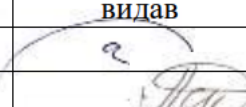
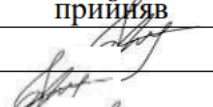
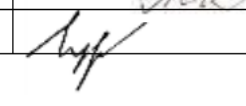
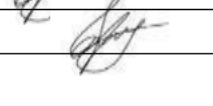
5.1 Діаграма варіантів використання інтерфейсу Evento

5.2 Діаграма компонентів і розгортання Evento

5.3 Діаграма активності сеансу користувача на Evento

5.4 _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4.1-4.3	Дудченко О.М., декан		
4.4-4.5	Латанська Л.О., доцент		
4.6	Щедролосєв О.В., професор		

7. Дата видачі завдання 25 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

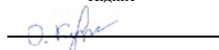
№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1.	Опис та аналіз предметної галузі	28.03.2024 р.	
2.	Постановка та формалізація задачі	06.04.2024 р.	
3.	Ескізний проєкт програмного забезпечення	13.04.2024 р.	
4.	Технічний проєкт програмного забезпечення	20.04.2024 р.	
5.	Робочий проєкт програмного забезпечення	27.04.2024 р.	
6.	Випробування програмного забезпечення	04.05.2024 р.	
7.	Розробка робочої документації	11.05.2024 р.	
8.	Виконання графічних документів	13.05.2024 р.	
9.	Вирішення питань охорони праці	18.05.2024 р.	
10.	Оформлення пояснювальної записки	25.05.2024 р.	
11.	Подання кваліфікаційної роботи на попередній захист	03.06.2024 р.	
12.	Подання кваліфікаційної роботи рецензенту	06.06.2024 р.	
13.	Подання на кафедру ІТ та ФМД тексту остаточного варіанту роботи, підписаного її керівником, разом з заявою щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи	08.06.2024 р.	
14.	Подання на кафедру ІТ та ФМД електронних версії наступних документів у форматі pdf: кваліфікаційної (дипломної) роботи; файлу-опису кваліфікаційної роботи (згідно Застосунку до наказу ректора НУК від 19.05.2020 р. за №287-уч); презентації доповіді; письмового відгуку та рецензії на кваліфікаційну роботу	14.06.2024 р.	
15.	Подання на кафедру ІТ ФМД письмового відгуку та рецензії на кваліфікаційну роботу	22.06.2024 р.	

Студент



Підпис

Керівник роботи



Підпис

Яновська К.О.

Прізвище та ініціали

Гайдаєнко О.В.

Прізвище та ініціали

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці та реалізації виняткової програми, створеної для революції в участі та просуванні заходів.

У ході розробки програмного продукту були проаналізовані статичні данні, щодо програм у сучасному світі, а також розроблено програмне забезпечення, яке надає змогу:

- Підібрати персональні налаштування.
- Створити стабільну підтримку.
- Приєднатися до івентів.
- Спростити ручну реєстрацією та складне оформлення квитків на різні івенти.

Робота викладена українською мовою і виконана на 90 аркушах, містить 10 рисунків, 1 таблицю, 5 додатків. Список використаних джерел містить 14 найменувань.

Ключові слова: платформа управління подіями; цифрова екосистема заходів; централізоване управління ресурсами подій; комунікація учасників подій; сповіщення в реальному часі; інформаційна підтримка заходів; управління контентом подій; взаємодія організаторів та учасників; інформаційна платформа для конференцій та івентів; цифровий сервіс координації подій.

ABSTRACT

The qualification work is devoted to the development and implementation of an exceptional program created to revolutionize the participation and promotion of events.

During the development of the software product, static data was analyzed regarding programs in the modern world, and software was developed that enables:

- Select personal settings.
- Create stable support.
- Join events.
- Simplify manual registration and complex ticketing for various events.

The work is written in Ukrainian and consists of 90 sheets, contains 10 figures, 1 tables, and 5 appendices. The list of used sources contains 14 items.

Keywords: event management platform; digital event ecosystem; centralized event resource management; event participant communication; real-time notifications; event information support; event content management; interaction between organizers and participants; information platform for conferences and events; digital event coordination service.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ.....	10
1.1. Актуальність теми.....	10
1.2. Мета та завдання дослідження.....	10
1.3. Об'єкт та предмет дослідження.....	11
1.4. Методи дослідження.....	11
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА WEB-ЗАСТОСУНКУ	13
2.1 Визначення web-застосунків та їх переваги перед традиційними настільними програмами.....	13
2.2 Типи web-застосунків та їх різноманітне використання.....	14
2.3 Ключові технології, що використовуються для розробки web- застосунків.....	15
2.4 Фреймворки та бібліотеки для створення динамічних та зручних інтерфейсів.....	17
2.5 Серверна логіка та бази даних для зберігання та обробки даних.....	19
2.6 Життєвий цикл розробки web-застосунків.....	21
2.6.1 Методи розгортання, включаючи статичне, динамічне та хмарне розгортання.....	24
2.6.2 Тестування, оптимізація та забезпечення безпеки web- застосунків.....	26
2.7 Огляд існуючих web застосунків для організації подій.....	28
2.7.1 Аналіз їх переваг та недоліків.....	30
2.7.2 Функціональні вимоги.....	32
2.7.3 Нефункціональні вимоги.....	33

РОЗДІЛ 3. ТЕСТУВАННЯ РОЗРОБКИ.....	34
3.1. Архітектура застосунку.....	34
3.1.1. Вибір технологій.....	36
3.1.2. Розробка інтерфейсу користувача.....	36
3.1.3. Опис процесу розробки.....	38
3.2. Тестування та валідація.....	39
3.3. Впровадження web застосунку.....	47
3.4. Оцінка функціональності.....	48
3.5. Порівняння з конкуруючими рішеннями.....	52
3.6. Активність програмного забезпечення.....	56
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	58
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДОДАТОК А - Технічне завдання до застосунку «Evento».....	71
ДОДАТОК Б - Програма і методика випробувань застосунку «Evento».....	75
ДОДАТОК В- Інструкція користувача до застосунку «Evento».....	79
ДОДАТОК Г - Опис застосунку «Evento».....	82
ДОДАТОК Д - Текст виконуючого коду застосунку «Evento».....	84

ВСТУП

В еру цифрового зв'язку організація та просування подій набули абсолютно нового характеру виміру. Ось чому з'явилася Evento, виняткова програма, створена для революції в участі та просуванні заходів.

З Evento участь у подіях ніколи не була такою простою. Програма, яка безпроблемно з'єднує організаторів, відвідувачів і виконавців події, пропонує спрощений досвід роботи з початку до закінчення. Попрощайтеся з ручною реєстрацією та складним оформленням квитків.

Evento спрощує все це, дозволяючи вам без зусиль досліджувати події та приєднуватися до них. Все, що вам потрібно - вибрати свій захід і взяти в ньому участь.

Для організаторів подій Evento надає надійну платформу для просування ваших подій широкій аудиторії. Від подробиць подій і розкладів до інтерактивних функцій і мультимедійного вмісту,

EventConnect дозволяє залучити потенційних відвідувачів і залишити незабутнє враження.

Evento заохочує мережеве спілкування та співпрацю, створюючи динамічну екосистему, ентузіасти подій можуть зібратися разом і процвітати.

Крім того, Evento є центром для ресурсів, пов'язаних із подіями, пропонуючи централізовану платформу, де учасники можуть отримати доступ до матеріалів події, зв'язатися з виконавцями та доповідачами і бути в курсі будь-яких оновлень або змін. Зі сповіщеннями в реальному часі та безперебійними каналами зв'язку, EventConnect тримає всіх, хто бере участь у циклі, забезпечує гладкий та незабутній досвід для всіх.

Незалежно від того, чи ви організатор події, який прагне охопити ширшу аудиторію, чи окрема особа.

Якщо ви прагнете відкривати для себе визначні події та брати участь у них, Evento — ваше рішення. З Evento, ви знаходитеся лише за один клік від підключення, просування та участі в більшості надзвичайних подій нашого часу.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ

1.1. Актуальність теми

Яка мета специфікації вимог до програмного забезпечення? Як це буде виконувати очікування? Які функції має продукт для задоволення потреб усіх користувачів?

Метою цього документа є запропонувати застосунок під назвою «Evento».

Evento – це застосунок, який хоче запропонувати себе в мобільному застосунку та веб-застосунку.

Застосунок має дві мети: пропонує платформу для просування своїх подій і дозволяє своїм користувачам брати участь у них.

1.2. Мета та завдання дослідження

Метою «Evento» є полегшення як участі в події, так і її організації:

- Запропонувати користувачам платформу, соціальну мережу, яка дозволить їм виконувати інтуїтивно зрозумілі операції. І, таким чином, дозволить створити спільноту, засновану на взаємній діяльності.
- Дозволити створення облікового запису для користувачів «Evento» для персоналізованого обслуговування та запис інформації користувача для участі в події. Обліковий запис користувача можна налаштувати і виправити, щоб забезпечити точне обслуговування.
- Створити інтерфейс користувача, який дозволить користувачам публікувати та переглядати події.

- NUI, який дозволяє видавцям обробляти індивідуальні запити та відповідати відвідувачам події, зберігаючи їх інформованість.
- Інтуїтивно зрозумілий графічний інтерфейс користувача, який дозволяє їм виявляти події, які їх зацікавили, та зареєструватися на захід за системою рухомих карток праворуч і ліворуч, які відображають інформацію про події.
- Інтерфейс користувача для власника системи, який підтримує звітування про облікові записи та події, які порушують правила та умови.

1.3. Об'єкт та предмет дослідження

Об'єкт дослідження: Розробка програмного забезпечення web застосунку «Evento».

Предмет дослідження: Проектування, розробка та впровадження web застосунку «Evento», спрямованого на організацію та управління подіями. Включає в себе аналіз вимог користувачів, проектування інтерфейсу користувача, вибір технологій, розробку функціональності, тестування та валідацію, а також впровадження отриманого програмного забезпечення.

1.4. Методи дослідження

1. Аналіз вимог користувачів: проведення співбесід, опитувань або фокус-груп для збору вимог щодо функціональності та інтерфейсу користувача.
2. Дослідження ринку: аналіз конкурентів та потенційних користувачів для визначення унікальних особливостей та можливостей для покращення продукту.
3. Проектування інтерфейсу користувача: використання методів UX/UI дизайну для створення зручного та привабливого інтерфейсу.

4. Технічне тестування: використання методів тестування програмного забезпечення, таких як модульне тестування, інтеграційне тестування та приймальне тестування.

5. Аналіз даних та зворотний зв'язок: моніторинг використання застосунку, збір та аналіз даних користувачів для постійного вдосконалення продукту.

6. Управління проектом: використання методів Agile або Scrum для ефективного керування процесом розробки.

7. Оцінка результатів: проведення оцінки ефективності та задоволеності користувачів для визначення успішності розробленого застосунку.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА WEB-ЗАСТОСУНКУ

2.1. Визначення web-застосунків та їх переваги перед традиційними настільними програмами

Визначення web-застосунків:

Web-застосунки - це програмне забезпечення, призначене для роботи в веб-браузері, а не на окремому комп'ютері. Вони створюються за допомогою веб-технологій, таких як HTML, CSS і JavaScript, і до них можна отримати доступ через Інтернет. На відміну від традиційних настільних програм, web-застосунки не потребують встановлення на кожен окремий комп'ютер, що робить їх більш зручними та доступними для користувачів.

Переваги web-застосунків перед традиційними настільними програмами:

- * Незалежність від платформи: web-застосунки сумісні з будь-яким пристроєм, що має веб-браузер, незалежно від операційної системи чи архітектури процесора; вони є ідеальними для використання в середовищах, де використовуються різні пристрої та платформи.

- * Легке розгортання та оновлення: розгортання web-застосунків є простим процесом, який полягає в завантаженні їх на веб-сервер; оновлення також здійснюються автоматично, що гарантує, що користувачі завжди мають доступ до останньої версії програми.

- * Доступність із будь-якого місця: web-застосунки доступні з будь-якого місця, де є підключення до Інтернету; це робить їх ідеальними для віддаленої роботи, співпраці та доступу до даних на ходу.

- * Зниження витрат: web-застосунки зазвичай дешевші в розробці та обслуговуванні порівняно з традиційними настільними програмами, це пов'язано з

тим, що їм не потрібно створювати окремі версії для різних платформ або купувати ліцензії на програмне забезпечення.

* Безпека: сучасні веб-застосунки використовують різні заходи безпеки для захисту даних користувачів від несанкціонованого доступу та кібератак.

2.2. Типи web-застосунків та їх різноманітне використання

Існує безліч типів web-застосунків, кожен з яких призначений для певної мети або галузі. Деякі з найбільш поширених типів web-застосунків включають:

* електронна комерція (інтернет-магазини та платформи електронної комерції дозволяють користувачам купувати товари та послуги в Інтернеті);

* соціальні мережі (соціальні мережі, такі як Facebook, Twitter і Instagram, дозволяють користувачам спілкуватися один з одним, обмінюватися контентом і створювати спільноти);

* офісні програми (веб-версії офісних програм, таких як текстові процесори, електронні таблиці та програми для презентацій, дозволяють користувачам створювати, редагувати та співпрацювати над документами в Інтернеті);

* хмарні обчислення (веб-сервіси хмарних обчислень надають обчислювальні ресурси, такі як зберігання даних, обробка даних і машинне навчання, через Інтернет);

* освітні програми (веб-застосунки для навчання, такі як інтерактивні курси, вікторини та навчальні матеріали, надають гнучкі та доступні можливості навчання);

* медичні програми (електронні медичні записи, системи телемедицини та інструменти самоконтролю здоров'я є прикладами web-застосунків, які використовуються в медичній галузі).

* ігри (браузерні ігри, багатокористувацькі онлайн-ігри та ігрові платформи дозволяють користувачам грати в ігри через веб-браузер);

* інструменти для розробників (веб-редактори коду, засоби відстеження проблем і платформи для спільної роботи є прикладами web-застосунків, які використовуються розробниками програмного забезпечення).

Розмаїття типів web-застосунків відображає їхню універсальність і здатність задовольняти різні потреби в різних галузях.

2.3. Ключові технології, що використовуються для розробки web-застосунків

Веб-застосунки створюються за допомогою набору взаємопов'язаних технологій, кожна з яких відіграє свою унікальну роль у забезпеченні функціональності та зручності використання програми. Основними технологіями, що використовуються для розробки web-застосунків, є:

HTML (HyperText Markup Language).

HTML є основою всіх веб-застосунків. Це мова розмітки, яка використовується для визначення структури та вмісту веб-сторінок. HTML-теги описують різні елементи веб-сторінки, такі як заголовки, абзаци, списки та зображення.

CSS (Cascading Style Sheets).

CSS використовується для стилізації веб-сторінок. Він дозволяє розробникам контролювати вигляд і відчуття веб-застосунку, включаючи шрифти, кольори, макети та анімацію. CSS відокремлює вміст від представлення, що робить розробку та обслуговування веб-застосунків більш ефективними.

JavaScript.

JavaScript є мовою програмування, яка використовується для додавання динамічної поведінки до веб-застосунків. Вона дозволяє розробникам створювати

інтерактивні елементи, такі як спадні меню, слайдери та форми перевірки. JavaScript також використовується для обробки подій, таких як кліки мишею та натискання клавіш.

Фреймворки JavaScript.

Фреймворки JavaScript, такі як React, Angular і Vue.js, надають набір готових компонентів та інструментів, які спрощують розробку web-застосунків. Вони пропонують структурований підхід до створення інтерфейсів користувача, управління станом і обробки даних.

Серверні технології.

Веб-застосунки часто використовують серверні технології, такі як Node.js, Python або Java, для обробки даних, виконання бізнес-логіки та взаємодії з базами даних. Серверні технології дозволяють веб-застосункам зберігати та отримувати дані, генерувати динамічний контент і забезпечувати безпеку.

Бази даних.

Бази даних використовуються для зберігання та управління даними, які використовуються web-застосунками. Поширені типи баз даних, що використовуються в веб-розробці, включають реляційні бази даних, такі як MySQL і PostgreSQL, а також нереляційні бази даних, такі як MongoDB і Cassandra.

Інструменти керування версіями.

Інструменти керування версіями, такі як Git, допомагають розробникам відстежувати зміни в коді веб-застосунку. Вони дозволяють кільком розробникам працювати над проектом одночасно, а також повертатися до попередніх версій коду у разі потреби.

Інструменти розгортання.

Інструменти розгортання, такі як Jenkins і Docker, автоматизують процес розгортання web-застосунків з середовища розробки в середовище виробництва. Вони допомагають забезпечити безперебійне та ефективне розгортання оновлень та нових функцій.

Ці ключові технології складають основу сучасних веб-застосунків. Розуміння їх ролей та взаємодії є важливим для ефективного розробки та підтримки надійних, масштабованих і зручних у використанні web-застосунків.

2.4. Фреймворки та бібліотеки для створення динамічних та зручних інтерфейсів

Фреймворки та бібліотеки є важливими інструментами для розробників web-застосунків, які прагнуть створювати динамічні та зручні інтерфейси користувача. Вони надають набір готових компонентів, інструментів та шаблонів, які спрощують процес розробки та забезпечують послідовність у зовнішньому вигляді та відчутті веб-застосунку.

Фреймворки JavaScript.

Фреймворки JavaScript, такі як React, Angular і Vue.js, є одними з найпопулярніших інструментів для створення динамічних інтерфейсів користувача. Вони використовують архітектуру на основі компонентів, яка дозволяє розробникам створювати складні інтерфейси шляхом складання багаторазових компонентів.

React є бібліотекою, яка фокусується на створенні декларативних компонентів. Вона використовує віртуальний DOM для ефективного оновлення інтерфейсу користувача.

Angular є фреймворком, який забезпечує повний набір функцій для розробки масштабованих, односторінкових веб-застосунків, він використовує архітектуру модель-вид-контролер (MVC) і двостороннє прив'язування даних.

Vue.js є прогресивним фреймворком, який поєднує простоту з потужними функціями, він використовує реактивну систему шаблонів і віртуальний DOM для створення динамічних інтерфейсів.

Бібліотеки компонентів.

Бібліотеки компонентів, такі як Material UI, Bootstrap і Ant Design, надають набір готових до використання компонентів інтерфейсу користувача, таких як кнопки, введення, навігаційні меню та діалогові вікна. Ці компоненти полегшують розробку спільних елементів інтерфейсу користувача та забезпечують узгодженість зовнішнього вигляду та відчуття по всьому веб-застосунку.

Material UI є бібліотекою компонентів, яка реалізує принципи дизайну матеріалу від Google. Вона забезпечує широкий спектр компонентів для створення сучасних та інтуїтивно зрозумілих інтерфейсів.

Bootstrap є популярною бібліотекою компонентів, яка надає адаптивні компоненти для створення веб-застосунків, які добре виглядають на будь-якому пристрої.

Ant Design є бібліотекою компонентів, яка надає елегантні та гнучкі компоненти для створення складних інтерфейсів користувача.

Інструменти стилізації.

Інструменти стилізації, такі як Sass, Less і Stylus, дозволяють розробникам писати стилі CSS більш ефективно та підтримувано. Вони надають функції, такі як змінні, вкладення і міксини, які полегшують створення та управління стилями CSS.

Sass є мовою препроцесора CSS, яка розширює можливості CSS за допомогою функцій, таких як змінні, міксини та вкладення.

Less є ще однією мовою препроцесора CSS, яка пропонує подібні функції, що і Sass.

Stylus є мовою препроцесора CSS, яка фокусується на простоті та читабельності.

Використання фреймворків, бібліотек та інструментів стилізації може значно покращити якість та ефективність розробки веб-застосунків. Вони надають розробникам набір будівельних блоків та інструментів, які дозволяють їм створювати динамічні, зручні та візуально привабливі інтерфейси користувача.

2.5. Серверна логіка та бази даних для зберігання та обробки даних

Серверна логіка та бази даних є невід’ємними компонентами веб-застосунків, що забезпечують зберігання, обробку та управління даними.

Серверна логіка стосується програмного коду, який виконується на сервері та керує бізнес-логікою та операціями веб-застосунку. Вона відповідає за:

- обробку запитів від клієнтів (веб-браузерів);
- доступ та модифікацію даних у базі даних;
- виконання обчислень та генерацію динамічного контенту;
- керування сесіями користувачів та забезпечення безпеки.

Серверна логіка може бути реалізована за допомогою різних технологій, таких як Node.js, Python, Java або C#. Ці технології дозволяють розробникам створювати масштабовані, надійні та безпечні серверні програми.

Бази даних використовуються для зберігання та управління даними, які використовуються веб-застосунками. Вони організують дані в структурованому форматі, що дозволяє ефективно зберігати, отримувати та оновлювати інформацію.

Існує два основних типи баз даних, які використовуються в веб-розробці:

- Реляційні бази даних, такі як MySQL і PostgreSQL, зберігають дані в таблицях, пов'язаних між собою ключами. Вони підтримують складні запити та транзакції, що робить їх придатними для складних веб-застосунків.
- Нереляційні бази даних, такі як MongoDB і Cassandra, зберігають дані у форматі документів або ключ-значення. Вони пропонують високу масштабованість та гнучкість, що робить їх придатними для обробки великих обсягів даних.

Серверна логіка взаємодіє з базами даних для зберігання та отримання даних, необхідних для роботи веб-застосунку. Ця взаємодія зазвичай здійснюється за допомогою об'єктів доступу до даних (DAO) або шарів доступу до даних (DAL).

DAO та DAL є програмними компонентами, які інкапсулюють операції з базою даних. Вони надають розробникам простий та узгоджений інтерфейс для виконання запитів до бази даних, отримання та збереження даних, що спрощує розробку та обслуговування веб-застосунків.

Захист даних, що зберігаються в базах даних, має важливе значення для безпеки веб-застосунків. Розробники повинні реалізувати належні заходи безпеки, такі як:

- шифрування даних як у базі даних, так і під час передачі;
- застосування контролю доступу для обмеження доступу до даних лише авторизованим користувачам;
- регулярне резервне копіювання даних для забезпечення відновлення у разі втрати даних.

Оптимізація продуктивності бази даних є критичною для забезпечення швидкого та надійного доступу до даних. Розробники можуть використовувати різні техніки оптимізації, такі як:

- індексування таблиць для прискорення пошуку даних;
- використання кешування для зберігання часто використовуваних даних у пам'яті;

- нормалізація даних для зменшення надмірності та покращення цілісності даних.

Проектування та реалізація ефективної серверної логіки та бази даних є важливими аспектами розробки веб-застосунків. Це забезпечує надійне зберігання та обробку даних, що є основою для побудови масштабованих, безпечних і зручних веб-застосунків.

2.6. Життєвий цикл розробки web-застосунків

Життєвий цикл розробки web-застосунків (SDLC) – це структурований процес, який використовується для планування, проектування, розробки, тестування та розгортання веб-застосунків. Це ітеративний та інкрементний процес, який складається з наступних фаз:

1. Планування

Фаза планування включає визначення мети та обсягу веб-застосунку, а також створення дорожньої карти та плану проекту. Вона включає в себе:

- визначення вимог користувачів та бізнес-цілей;
- аналіз ринку та конкурентів;
- визначення функціональних можливостей та технічних вимог;
- створення дорожньої карти та плану проекту.

2. Проектування

Фаза проектування включає створення детального проекту веб-застосунку. Вона включає в себе:

- проектування архітектури веб-застосунку;
- вибір технологій та фреймворків;
- створення прототипів та каркасів;
- розробка інформаційної архітектури та навігації.

3. Розробка

Фаза розробки включає фактичну розробку веб-застосунку. Вона включає в себе:

- написання коду;
- створення інтерфейсу користувача;
- інтеграція з базами даних та іншими системами;
- тестування коду та виправлення помилок.

4. Тестування

Фаза тестування включає перевірку коректності та функціональності веб-застосунку. Вона включає в себе:

- одиничне тестування;
- інтеграційне тестування;
- тестування прийняття користувачами;
- перевірка продуктивності та навантаження.

5. Розгортання

Фаза розгортання включає публікацію веб-застосунку та його доступність для користувачів. Вона включає в себе:

- розгортання веб-застосунку в середовищі виробництва;
- налаштування та конфігурація серверів;
- моніторинг та усунення несправностей на етапі виконання.

6. Обслуговування

Фаза обслуговування включає підтримку та оновлення веб-застосунку після його розгортання. Вона включає в себе:

- виправлення помилок та випуск оновлень;
- додавання нових функцій та покращення;

моніторинг продуктивності та безпеки.

Існує кілька методологій SDLC, які можуть бути використані для розробки web-застосунків. Серед найпоширеніших:

Водоспад: традиційна методологія, яка слідує лінійному процесу;

Скрам: ітеративна та інкрементна методологія, яка фокусується на доставці невеликих частин функціональності за короткий період часу;

Канбан: методологія, яка використовує візуальну дошку для управління потоком роботи та обмеження незакінченої роботи;

DevOps: набір практик та інструментів, які автоматизують та інтегрують процеси розробки та експлуатації.

Вибір відповідної методології SDLC залежить від розміру та складності веб-застосунку, а також від уподобань та можливостей команди розробників.

Документація SDLC є важливою частиною процесу розробки web-застосунків. Вона надає зацікавленим сторонам чітке розуміння вимог, дизайну, архітектури та процесів тестування веб-застосунку. Документація SDLC може включати: план проекту, документ з вимогами, технічні специфікації, документи з дизайну, звіти про тестування, документація з розгортання.

Дотримання добре визначеного життєвого циклу розробки web-застосунків допомагає командам розробників створювати високоякісні, надійні та зручні веб-застосунки в ефективний та своєчасний спосіб.

2.6.1. Методи розгортання, включаючи статичне, динамічне та хмарне розгортання

Після розробки та тестування web-застосунку його необхідно розгорнути, щоб зробити його доступним для користувачів. Існують різні методи розгортання, кожен з яких має свої переваги та недоліки:

Статичне розгортання - передбачає створення статичних файлів, таких як HTML, CSS та JavaScript, з коду веб-застосунку. Ці статичні файли потім копіюються на веб-сервер, і користувачі отримують доступ до них безпосередньо. Серед переваг статичного розгортання: швидке та просте розгортання; висока продуктивність та надійність; низька вартість; підходить для простих веб-застосунків зі статичним вмістом. Недоліками статичного розгортання є: не підходить для динамічних веб-застосунків, які генерують вміст на льоту; складніше розгортати оновлення, оскільки потрібно замінити всі статичні файли.

Динамічне розгортання - передбачає запуск веб-застосунку на веб-сервері, який інтерпретує код веб-застосунку та генерує динамічний вміст у відповідь на запити користувачів. Основні переваги динамічного розгортання: підходить для динамічних веб-застосунків, які генерують вміст на льоту; легше розгортати оновлення, оскільки потрібно оновити лише код веб-застосунку на сервері. Недоліки динамічного розгортання: повільніше та складніше розгортати порівняно зі статичним розгортанням; може бути менш надійним, оскільки залежить від веб-сервера; вища вартість через необхідність більшої обчислювальної потужності.

Хмарне розгортання - передбачає розгортання веб-застосунку на хмарній платформі, такій як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP). Хмарні платформи надають масштабовану та гнучку інфраструктуру, яка дозволяє автоматизувати процес розгортання та керування. До переваг хмарного розгортання віднесено: масштабованість та гнучкість; автоматизація розгортання та керування; висока надійність та доступність; додаткові послуги, такі як бази даних, сховище та машинне навчання. Недоліки хмарного розгортання: більш висока вартість порівняно зі статичним або динамічним розгортанням на фізичних серверах; залежність від стороннього постачальника послуг.

Вибір відповідного методу розгортання залежить від конкретних вимог веб-застосунку, таких як його тип, розмір, складність та бюджетні обмеження.

Процес розгортання web-застосунку зазвичай включає наступні кроки:

1. Підготовка середовища розгортання (веб-сервер або хмарна платформа).
2. Створення пакета розгортання (зазвичай ZIP-файл або контейнер Docker).
3. Завантаження пакета розгортання на сервер розгортання.
4. Розпакування та встановлення пакета розгортання.
5. Налаштування та запуск веб-застосунку.

Для автоматизації процесу розгортання часто використовуються інструменти розгортання, такі як Jenkins, Docker і Kubernetes. Ці інструменти дозволяють розробникам створювати канали розгортання та автоматично розгортати оновлення в різних середовищах, таких як розробка, тестування та виробництво.

Дотримуючись належних практик розгортання, розробники можуть забезпечити швидке, надійне та безпроблемне розгортання своїх web-застосунків.

2.6.2. Тестування, оптимізація та забезпечення безпеки web-застосунків

Після розробки web-застосунку важливо провести його тестування, оптимізацію та забезпечити його безпеку, щоб гарантувати його якість, продуктивність та надійність.

Тестування web-застосунків включає перевірку його функціональності, продуктивності та зручності використання. Існує кілька типів тестування web-застосунків, включаючи:

- одиничне тестування: тестування окремих компонентів коду веб-застосунку;
- інтеграційне тестування: тестування того, як різні компоненти веб-застосунку працюють разом;

- тестування прийняття користувачами: тестування того, чи веб-застосунок відповідає вимогам користувачів та є простим у використанні;

- тестування продуктивності: тестування того, як веб-застосунок працює під навантаженням та в різних мережевих умовах.

Оптимізація web-застосунків передбачає покращення його швидкості, продуктивності та використання ресурсів. Це може включати:

- оптимізація зображень: зменшення розміру зображень без шкоди для їх якості;

- мініфікація та об'єднання коду: видалення непотрібних символів та об'єднання декількох файлів коду в один файл;

- використання мережі доставки вмісту (CDN): розповсюдження вмісту веб-застосунку по всьому світу для зменшення часу завантаження;

- кешування: зберігання часто використовуваних даних у пам'яті для швидшого доступу.

Забезпечення безпеки web-застосунків є критично важливим для захисту даних користувачів та запобігання кібератак. До цієї складової віднесено:

- використання протоколу HTTPS - шифрування зв'язку між веб-застосунком та користувачами;

- захист від міжсайтових сценаріїв (XSS) - запобігання виконанню шкідливого коду на веб-сторінках;

- захист від міжсайтової підробки запиту (CSRF) - запобігання неавторизованим користувачам надсилати запити від імені авторизованих користувачів;

- регулярне оновлення програмного забезпечення - встановлення останніх оновлень безпеки для веб-застосунку та його компонентів.

Існує широкий спектр інструментів, які можуть допомогти у тестуванні, оптимізації та забезпеченні безпеки web-застосунків. Деякі з найпоширеніших інструментів включають: Selenium, Cypress, JMeter (інструменти тестування), Google PageSpeed Insights, GTmetrix, WebPageTest (інструменти оптимізації), OWASP ZAP, Nessus, Acunetix (інструменти забезпечення безпеки).

Тестування та оптимізація web-застосунків - це постійний процес. Оскільки веб-застосунки постійно оновлюються та змінюються, важливо регулярно проводити тестування та оптимізацію, щоб гарантувати, що вони працюють належним чином. Крім того, постійний моніторинг безпеки є важливим для виявлення та усунення будь-яких потенційних вразливостей.

Дотримуючись належних практик тестування, оптимізації та забезпечення безпеки web-застосунків, розробники можуть гарантувати, що їхні веб-застосунки є надійними, швидкими, безпечними та зручними для користувачів.

2.7. Огляд існуючих web застосунків для організації подій

Створення веб-сайтів подій.

Ці застосунки дозволяють користувачам створювати професійні веб-сайти для своїх подій, що містять важливу інформацію, таку як програма, зареєстровані особи та можливості спонсорства.

Реєстрація учасників.

Веб-застосунки забезпечують інтегровані інструменти реєстрації, які автоматизують процес прийому реєстрацій, надання підтверджень та керування списками гостей.

Керування бюджетом.

Застосунки допомагають організаторам подій відстежувати витрати та керувати бюджетами, забезпечуючи прозорість та запобігаючи перевитратам.

Просування подій.

У застосунки вбудовані функції просування, які дозволяють користувачам створювати та розповсюджувати маркетингові матеріали, такі як електронні листи, публікації в соціальних мережах та націлену рекламу.

Взаємодія з учасниками.

Застосунки надають можливості для взаємодії з учасниками, такі як форуми та чати, що дозволяють організаторам спілкуватися з гостями та відповідати на їхні запитання.

Збір даних і аналітика.

Застосунки збирають цінні дані про події, які можна використовувати для аналізу відвідуваності, оцінки ефективності маркетингу та виявлення областей для покращення.

Існує безліч різноманітних веб-застосунків для організації подій. Можна виокремити такі популярні, як Eventbrite, Cvent, Whova, Accelevents, Splash.

Eventbrite є одним з провідних веб-застосунків для організації подій, що пропонує комплексний набір функцій для організації подій різного масштабу. Він включає інструменти для створення веб-сайтів, реєстрації учасників, керування бюджетом, просування подій і взаємодії з учасниками.

Cvent - це конференц-платформа, яка спеціалізується на великих корпоративних подіях. Вона пропонує широкі можливості персоналізації та автоматизації, такі як створення індивідуальних програм для учасників, автоматизовані процеси реєстрації та просунуті інструменти звітності.

Whova - це мобільний застосунок для організації подій, який забезпечує взаємодію учасників у режимі реального часу, мережу та функціональність управління подіями. Він пропонує мобільні програми для учасників, спікерів та організаторів, а також можливості для мережових зустрічей, опитувань та обміну презентаціями.

Accelevents - це віртуальна та гібридна платформа подій, яка дозволяє організаторам транслювати події в прямому ефірі, проводити віртуальні сеанси та створювати віртуальні кабінети для спонсорів. Вона пропонує функції інтерактивності в режимі реального часу, такі як чати, віртуальні зали та можливості для віртуальних зустрічей.

Splash - це веб-застосунок для організації маркетингових подій, який фокусується на просуванні, реєстрації та зборі даних про події. Він пропонує інструменти для створення цільових сторінок, автоматизації електронного маркетингу та збору даних про відвідувачів для аналізу ефективності маркетингу подій.

2.7.1. Аналіз їх переваг та недоліків

Веб-застосунки для організації подій дозволяють економити час та зусилля. Застосунки автоматизують багато завдань, пов'язаних з організацією подій, що дозволяє організаторам заощадити час і зосередитися на інших аспектах планування подій. Також вони мають поліпшену ефективність. Застосунки забезпечують ефективний спосіб управління реєстраціями, комунікацією з учасниками та фінансовими операціями, покращуючи загальну ефективність організації подій. Серед переваг також й підвищена взаємодія учасників. Застосунки надають можливості для учасників взаємодіяти один з одним, ставити запитання і ділитися зворотним зв'язком, що призводить до більш привабливого та інтерактивного досвіду події. За допомогою застосунків відбувається збір цінних даних. Вони збирають дані про події, які можна використовувати для аналізу тенденцій, оцінки ефективності маркетингу та виявлення областей для покращення майбутніх подій.

Вибір найкращого веб-застосунку для організації подій залежить від конкретних потреб і вимог події. Організатори повинні враховувати розмір і масштаб події, цільову аудиторію, бюджет, потрібні функції, інтеграцію з іншими інструментами.

Разом з тим, існуючі веб-застосунки для організації подій мають й певні недоліки. Серед основних: вартість, складність, залежність від Інтернету, обмежена інтеграція. Багато веб-застосунків для організації подій пропонуються

на платній основі або стягують плату за певні функції. Це може бути фінансовим тягарем для організаторів подій з обмеженим бюджетом. Деякі веб-застосунки можуть бути складними у використанні, особливо для тих, хто не має досвіду в організації подій. Складний інтерфейс і складні функції можуть ускладнити процес організації події. Вони вимагають стабільного підключення до Інтернету для роботи. У разі проблем з підключенням організатори можуть втратити доступ до важливих інструментів і даних. Не всі веб-застосунки для організації подій легко інтегруються з іншими інструментами, такими як системи управління відносинами з клієнтами (CRM) або інструменти розсилки електронної пошти. Це може ускладнити синхронізацію даних і автоматизацію процесів.

Також є певні проблеми з функціональних обмежень. Деякі веб-застосунки можуть не мати всіх функцій, необхідних для організації певних типів подій. Наприклад, застосунок, який спеціалізується на віртуальних подіях, може не мати необхідної функціональності для організації очних подій. Проблемою є відсутність персоналізації та недостатня конфіденційність. Деякі веб-застосунки пропонують обмежені можливості персоналізації, що ускладнює організаторам подій налаштування додатків відповідно до своїх конкретних потреб і брендингу. Важливо враховувати проблеми конфіденційності при використанні веб-застосунків для організації подій, оскільки вони збирають і обробляють конфіденційну інформацію про учасників. Організатори подій повинні переконатися, що вибрані ними веб-застосунки відповідають стандартам конфіденційності і захищають дані учасників.

Незважаючи на ці недоліки, веб-застосунки для організації подій залишаються цінними інструментами для організаторів подій. Організатори повинні ретельно оцінити свої потреби та вимоги перед вибором веб-застосунку, щоб мінімізувати можливі недоліки та максимізувати переваги.

Веб-застосунки для організації подій стали невід'ємною частиною сучасного процесу планування та управління подіями. Вони пропонують широкий спектр

функцій для спрощення різних аспектів організації подій, підвищення ефективності, покращення взаємодії з учасниками та збору цінних даних. Вибравши правильний веб-застосунок, організатори подій можуть зекономити час і зусилля, а також забезпечити успішні та незабутні події для своїх учасників.

2.7.2. Функціональні вимоги

- Вхід - неавторизований користувач виконує ідентифікацію за своєю електронною адресою та паролем і стає авторизованим користувачем для використання, що дозволяє йому використовувати функції застосунку.

- Видалити власний обліковий запис - дозволяє авторизованому користувачеві видалити обліковий запис і дані пов'язані з ним.

- Видалити власні події - надає користувачеві можливість видалити подію, яку він створив собі.

- Створити подію — функція, яка дозволяє авторизованому користувачеві створювати подію за посилання на місце, назву події, дату й час тощо, повний опис події (за бажанням).

- Повідомити про помилки - неавторизований користувач і авторизований користувач можуть заповнити звіт про помилку - форму, яку зможе переглядати адміністратор.

- Ставити запитання та звертатись за допомогою - неавторизований користувач і авторизований користувач можуть заповнити форму запитання/довідки, яку бачитиме адміністратор.

- Перегляд події - основна вимога до програми, яка дозволяє ознайомитися з подією у форматі візуальної картки в Tinder для Авторизованого користувача та адміністратора.

- Подобається - розширена функція для перегляду події, що дозволяє показати вашу зацікавленість у події і вашу участь, і перейти до картки наступної події, протягнувши картку праворуч.

- Не подобається - розширена функція для перегляду події, що дозволяє вам показати свою незацікавленість у події та перейти до картки наступної події, протягнувши картку ліворуч.

- Перегляд інформації про подію – розширена функція для перегляду події, відображається кнопкою, що залишилася під картою та дає більше інформації про подію, залишену її власником.

- Створити звіт - дозволяє авторизованому користувачеві створити форму, доступну для перегляду адміністратором, повідомити про користувача або подію, яка не відповідає умовам використання застосунку.

- Видалити подію та користувачів — функція, яка дозволяє адміністратору видалити користувача та/або подію, що не відповідає умовам використання.

- Схвалити подію – дозволяє адміністратору підтвердити, що подію можна опублікувати в застосунок.

- Перегляд звіту - перегляд звітів складається з перегляду звітів про помилки, користувачів і подій.

2.7.3. Нефункціональні вимоги

- Кожен користувач повинен зареєструватися для реєстрації в події.

- Для перевірених подій потрібно завантажити офіційний документ або індивідуальний ідентифікатор організації або організатора.

- Пароль має містити щонайменше 8 і максимум 12 цифр. Дозволені всі символи ASCII.

- Ви можете створити лише один обліковий запис користувача за допомогою електронної пошти. Мати кілька облікових записів не дозволено.

- Відкинуті події можна переглянути в окремому просторі. Якщо ви випадково викинули щось.
- Ви можете переглядати всі події без реєстрації.
- Для перегляду подій, доступних поблизу вас, потрібен GPS.
- Щоб програми були законними, їх потрібно завантажити з магазину Google Play.
- Без підключення до Інтернету ви не можете використовувати цю програму.

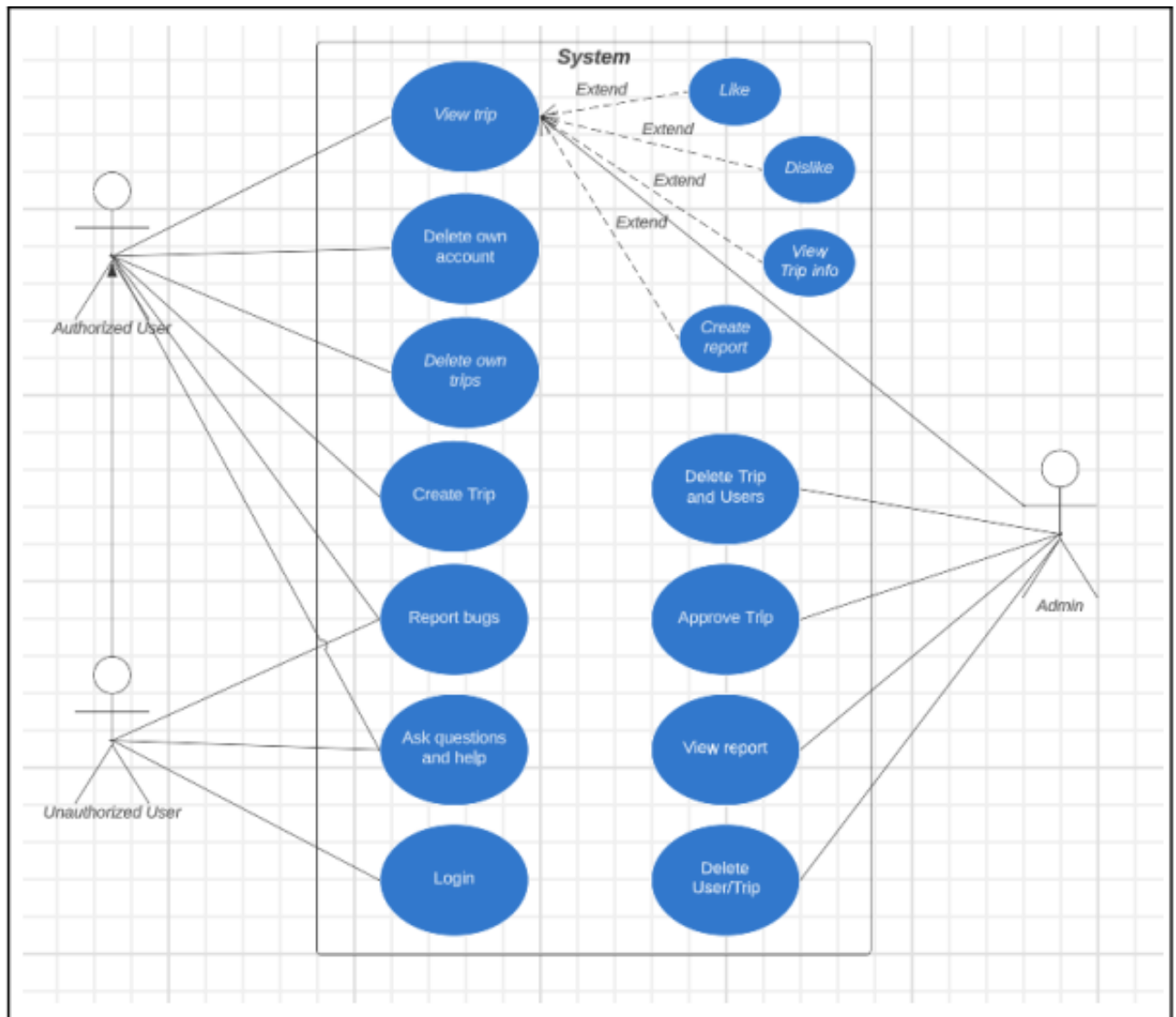


Рисунок 2.1. - Діаграма варіантів використання інтерфейсу Evento

Неавторизований користувач - користувач, який не ідентифікував себе з обмеженою функціональністю. Авторизований користувач - ідентифікований користувач із профілем, у якому представлені його особисті дані. Адміністратори - Адміністратори призначаються в системі власником системи, вони несуть відповідальність для затвердження подій, призначення санкцій і перегляду звітів про помилки та запитів на допомогу.

РОЗДІЛ 3. ТЕСТУВАННЯ РОЗРОБКИ

3.1. Архітектура застосунку

Аналіз фреймворку

Android SDK – це набір інструментів, бібліотек і API, які надає Google і який сприятиме розробці програм для Android.

Тут наведено аналіз деяких ключових аспектів Android SDK:

- надійне середовище розробки;
- широкий діапазон API;
- тестування емулятора та пристрою;
- сторонні бібліотеки та підтримка.

Чому було обрано це?

Інструменти Android SDK дуже легкі в роботі та дуже поширені.

Аналіз бази даних.

Було використано базу даних firebase, яка дуже зручна для мобільних пристроїв та додатків. Ось деякі ключові особливості firebase:

- база даних реального часу;
- автентифікація та керування користувачами;
- хмарне сховище;
- хостинг;
- аналітика та моніторинг продуктивності.

Чому Firebase?

Оскільки Firebase є зручною для розробки додатків для Android. Він має дуже хорошу гнучкість із підтримкою кількох платформ.

Навіщо людям використання цього застосунку?

Ідея Evento — застосунок для організації та реєстрації подій. Події відбуваються всюди, але не кожна подія досягне всіх. Але з Evento є можливість підключитися великій кількості людей, які матимуть змогу дуже легко дізнатися про події. Отже, з функцією перетягування карток, багатьом користувачам буде дуже легко публікувати та відвідувати події.

Вимоги до пристроїв для використання застосунку:

- ОС: Android 8.1.0 або новіша версія;
- Вільне місце на диску: 200 МБ;
- Швидкість Інтернету: від 10 до 20 Мбіт/с;
- Доступ до місцезнаходження: потрібен.

На даний момент розробляється інтерфейс користувача на java, мові сценаріїв для роботи програми на Android.

3.1.1. Вибір технологій

Програма називається Evento. Для ефективності була використана мова Java. Було багато ідей в цьому напрямку, а також деякі знання. Для створення Evento використана Android Studio. За допомогою стандартного фреймворку Android SDK, який надає необхідні бібліотеки та інструменти для розробки застосунку для Android. Використовуючи цю програму, користувачі можуть зареєструватися на подію та перевірити подробиці про неї. Також обрана база даних Firebase для підключення до бази даних як найефективніша.

3.1.2. Розробка інтерфейсу користувача

Діаграма компонентів і розгортання.

На діаграмі компонентів та схеми розгортання видно, що для використання Evento користувач повинен мати пристрій, який є або смартфоном, або

комп'ютером із програмою Evento. Коли він хоче підключитися до Evento, для підключення до сервера Evento потрібне підключення TCP/IP. Зі свого додатка користувачі можуть увійти та переглянути події, доступні через застосунок. Сервери Evento для відповіді на запит користувача проходять через з'єднання TCP/IP для підключення до зовнішньої бази даних. База даних служби Firebase.

Щоб відповісти на запит входу, програма шукатиме наявний обліковий запис у таблиці учасників. А для перегляду подій сервер використовує таблицю подій, щоб запропонувати події користувачеві.

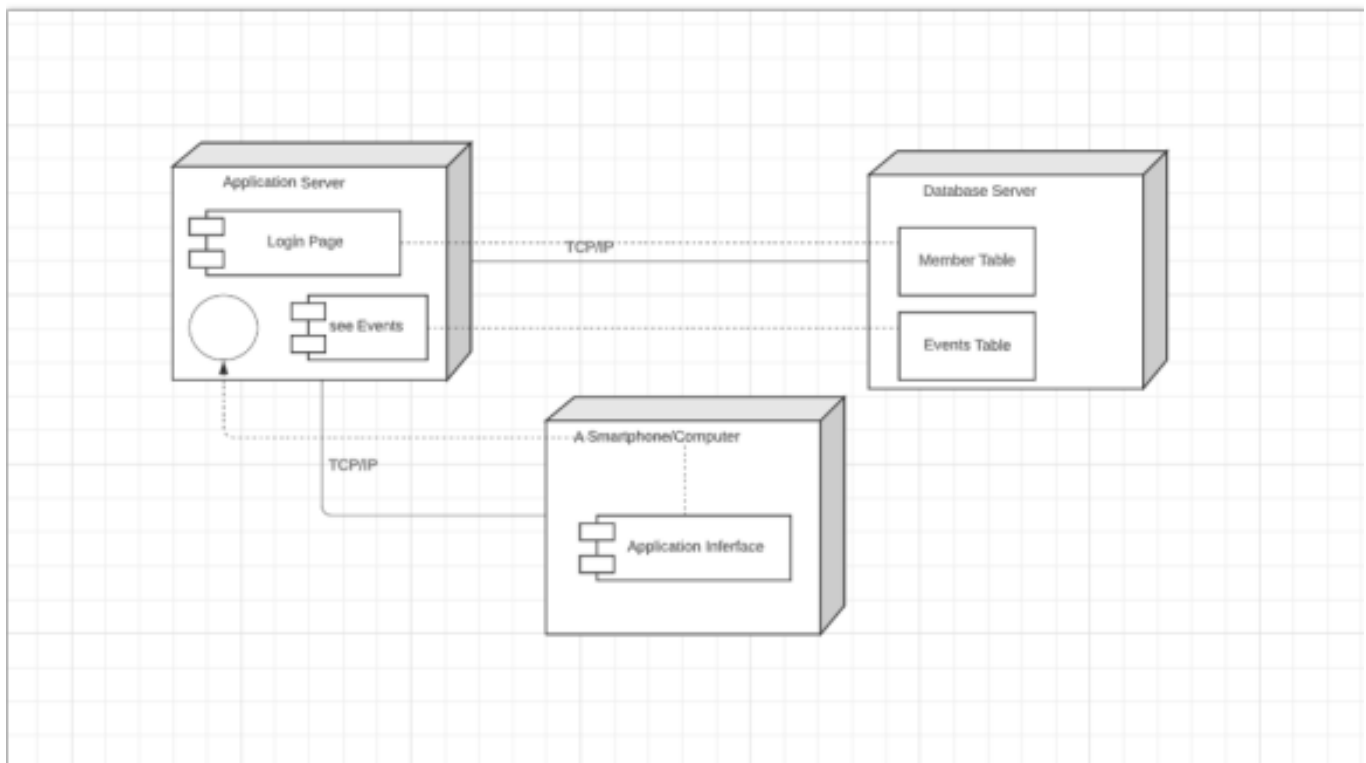


Рисунок 3.1. - Діаграма компонентів і розгортання Evento

3.1.3. Опис процесу розробки

Процес розробки веб-застосунку «Evento»:

1. Аналіз вимог.

Визначення цільової аудиторії та їх потреб. Збір і документування вимог до функціональності, користувацького інтерфейсу та нефункціональних характеристик (наприклад, продуктивність, безпека)

2. Проектування архітектури.

Вибір відповідних технологій для фронтенду, бекенду, бази даних та інфраструктури.

Розробка архітектури застосунку, включаючи розбивку на рівні та компоненти.

3. Розробка фронтенду.

Створення інтерфейсу користувача за допомогою React, Redux та матеріал-UI.

Реалізація функціональності фронтенду, такої як валідація вводу, керування станом та взаємодія з API.

4. Розробка бекенду.

Створення API бекенду за допомогою Node.js, Express.js та MongoDB.

Реалізація бізнес-логіки для керування подіями, учасниками та іншими аспектами застосунку.

5. Інтеграція фронтенду та бекенду.

Підключення фронтенду до API бекенду.

Обробка та відображення даних, отриманих від бекенду.

6. Тестування.

Проведення всебічних тестів для забезпечення функціональності, продуктивності та безпеки застосунку.

Використання як ручних, так і автоматизованих тестів.

7. Розгортання.

Розгортання застосунку в хмарній платформі або на власних серверах.

Налаштування інфраструктури для забезпечення високої доступності, масштабованості та безпеки.

8. Моніторинг та обслуговування.

Налаштування системи моніторингу для відстеження продуктивності та надійності застосунку.

Регулярне обслуговування та оновлення для забезпечення безпеки та оптимальної роботи.

3.2. Тестування та валідація

Після запуску програми користувач побачить на екрані дві кнопки:



Рисунок 3.2. - Сторінка входу/реєстрації

Кнопка «Вхід» переводить користувача до форми входу (рисунок 3.2.), а кнопка «Реєстрація» перенаправляє користувача до реєстраційної форми (рисунок 3.3).

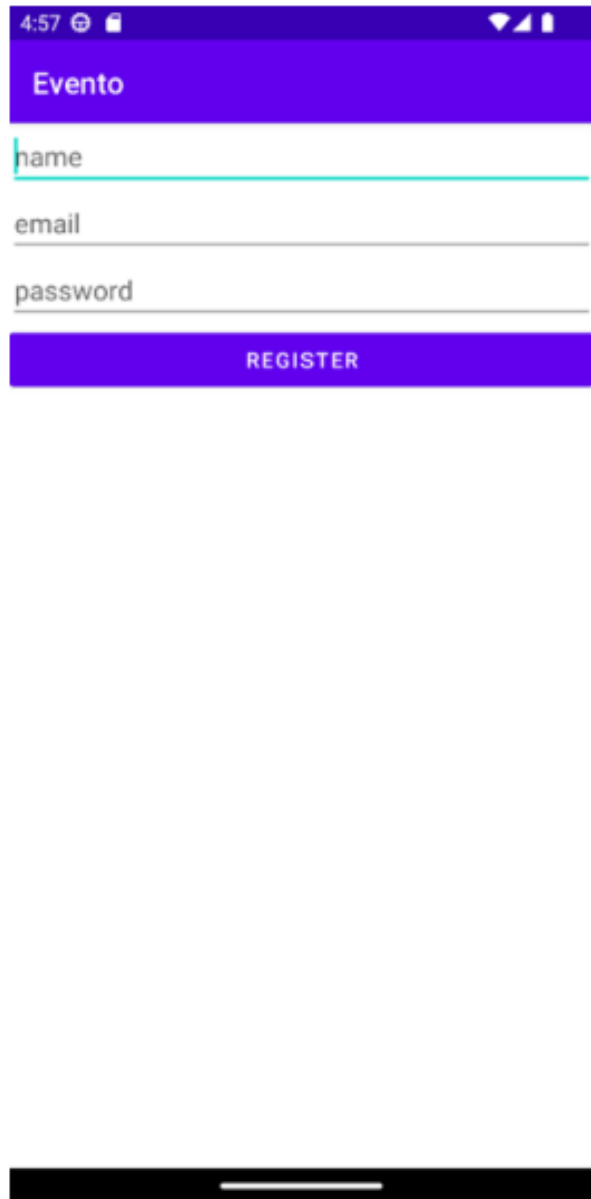
The image shows a mobile application interface for registration. At the top, there is a blue header bar with the text "Evento" in white. Below the header, there are three white input fields with blue borders, labeled "name", "email", and "password" in a light blue font. Below the input fields is a solid blue button with the word "REGISTER" in white capital letters. The entire form is set against a white background. At the very top of the screen, a status bar shows the time "4:57" and some system icons.

Рисунок 3.3. - Реєстраційна форма

Користувач повинен заповнити інформацію про ім'я, адресу електронної пошти, пароль.

Це форма входу з прикладом:

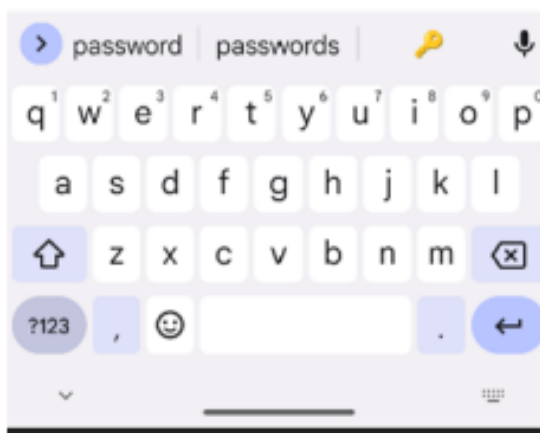
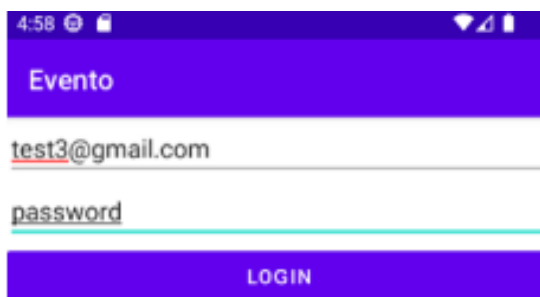


Рисунок 3.4. - Форма входу

Користувач повинен ввести інформацію про електронну адресу та пароль.
Коли користувач завершить реєстрацію/вхід, з'явиться головний екран:



Рисунок 3.5. - Сторінка події

На головному екрані користувач може пересувати картки, утримуючи їх і рухаючись праворуч або ліворуч, або користувач може торкатися карток. У верхньому лівому куті користувач може натиснути кнопку «Створити подію», яка відкриє новий екран, де користувач може створити нову подію.

У верхньому правому куті користувач може натиснути кнопку «Вийти», щоб вийти із системи, відкриється форма реєстрації (рисунок 3.3), і користувач зможе зареєструватися знову.

Торкнувшись картки, відкриється це спливаюче вікно:



Рисунок 3.6. - Докладніше про подію

Тут має бути показана інформація про подію та кнопка «ЗРОБИТИ ЩОСЬ». має закривати вікно, коли його торкаються. (Не було реалізовано).

Коли користувач натискає кнопку «СТВОРИТИ ПОДІЮ» (рисунок 3.6), з'являється наступний екран:

The image shows a mobile application interface for creating a new event. At the top, there is a status bar with the time 5:01 and various icons. Below it is a blue header bar with the word 'Evento' in white. The main content area has a white background. It contains two text input fields: the first is labeled 'name' and the second is labeled 'description'. Both fields have a light blue underline. Below the 'description' field is a solid blue button with the word 'CREATE' in white capital letters. At the very bottom of the screen is a black home indicator bar.

Рисунок 3.7. - Створення форми події

Користувач може заповнити інформацію про нову подію. Всередині області імен нова назва події буде створено та додано до карток на рисунку 3.7 на головному екрані після клікання на кнопку «СТВОРИТИ». Всередині області опису буде новий опис події, якій створюється та додається до бази даних після натискання кнопки „СТВОРИТИ”.

Користувач може заповнити інформацію про нову подію. Всередині області імен нову подію буде створено та додано до карток на рисунку 3.7 на головному екрані після клікання кнопки «СТВОРИТИ». Всередині області опису буде новий

опис події, що створюється та додається до бази даних після натискання кнопки „СТВОРИТИ”.

На цьому рисунку ми бачимо приклад переміщення карток за допомогою функції «перетягування вправо»:



Рисунок 3.8. - Приклад картки обміну

3.3. Впровадження web застосунку

1. Підготовка до впровадження

Визначення відповідних груп користувачів і отримання відгуків.

Надання навчання та документації для користувачів.

Налаштування системи підтримки для вирішення проблем.

2. Поетапне впровадження

Розгортання застосунку для обмеженої групи користувачів для отримання відгуків і виправлення помилок.

Поступове розширення доступу до більшої кількості користувачів.

Моніторинг відгуків і внесення необхідних коригувань.

3. Забезпечення прийняття користувачами.

Проведення навчальних семінарів і демонстрацій.

Надання технічної підтримки та відповідей на запитання.

Залучення користувачів до розробки майбутніх функцій.

4. Інтеграція з іншими системами.

Інтеграція з існуючими системами, такими як фінансові або системи керування відносинами з клієнтами (CRM).

Забезпечення плавного обміну даними та взаємодії між системами.

5. Безперервне вдосконалення.

Збір відгуків користувачів і аналіз даних використання.

Регулярні оновлення для додавання нових функцій, виправлення помилок і підвищення загальної продуктивності.

Залучення користувачів до процесу розробки для забезпечення відповідності їхнім мінливим потребам.

6. Управління безпекою.

Регулярні оновлення безпеки та виправлення вразливостей.

Реалізація найкращих практик безпеки, таких як шифрування даних і автентифікація користувачів.

Забезпечення відповідності нормам і стандартам безпеки галузі.

3.4. Оцінка функціональності

"Evento" - це веб-застосунок для управління подіями, призначений для спрощення організації та управління заходами будь-якого масштабу. Ця комплексна оцінка функціональності розглядає ключові аспекти "Evento", щоб визначити його можливості та обмеження.

Критерії оцінювання.

Оцінка проводиться за такими критеріями.

Управління подіями:

- створення, редагування та видалення подій;
- управління реєстрацією на події;
- відстеження відвідуваності та участі.

Реєстрація та обробка платежів:

- можливості реєстрації для користувачів;
- обробка онлайн-платежів;
- інтеграція з платіжними шлюзами.

Комунікації:

- розсилка електронних листів та повідомлень;
- відстеження взаємодії користувачів.

Аналітика та звітність:

- створення звітів про події та відвідуваність;
- відстеження ключових показників ефективності (КПЕ).

Інтерфейс користувача:

- інтуїтивно зрозумілий інтерфейс;
- зручність навігації та використання.

Інтеграції:

- інтеграція з іншими інструментами керування подіями;
- можливості налаштування та розширення.

Процес оцінювання.

Оцінка включає такі етапи:

- створення пробної облікового запису та ретельне дослідження функціональних можливостей платформи;
- перевірка функціональності відповідно до встановлених критеріїв;
- збір відгуків від користувачів та зацікавлених сторін;
- детальний аналіз звітів про аналітику та відвідуваність.

Результати.

Нижче наведено детальну оцінку функціональності "Evento" за кожним критерієм:

Управління подіями.

"Evento" надає повний спектр функцій управління подіями. Організатори можуть створювати події з налаштованими параметрами, такими як дата, час, місце проведення та реєстраційний збір.

Функції реєстрації дозволяють користувачам легко реєструватися на події та керувати своїми реєстраціями.

Модуль відстеження відвідуваності забезпечує отримання точної інформації про кількість відвідувачів, зареєстрованих і присутніх на заході.

Реєстрація та обробка платежів.

"Evento" спрощує реєстрацію на події для користувачів. Вони можуть легко зареєструватися та здійснити платіж за допомогою інтегрованої системи обробки платежів.

Платформа інтегрується з кількома платіжними шлюзами, такими як PayPal, Stripe та Authorize.Net, що забезпечує безпечну обробку платежів.

Комунікації.

"Evento" пропонує надійні функції комунікації. Організатори можуть надсилати електронні листи та повідомлення учасникам, тримаючи їх в курсі деталей заходу та оновлень.

Функції відстеження взаємодії надають інформацію про те, як учасники взаємодіють з повідомленнями, що дозволяє організаторам оптимізувати комунікаційні стратегії.

Аналітика та звітність.

"Evento" надає потужні можливості аналітики та звітності. Організатори можуть генерувати звіти про події та відвідуваність, щоб відстежувати ключові показники ефективності, такі як кількість зареєстрованих, відвідуваність та дохід.

Звіти допомагають організаторам оцінювати успіх заходів і виявляти можливості для їх покращення.

Інтерфейс користувача.

Інтерфейс користувача "Evento" винятково інтуїтивно зрозумілий і зручний у використанні. Чітка навігація та просте компонування полегшують доступ і використання функцій як для організаторів, так і для відвідувачів.

Інтеграції.

"Evento" може бути інтегрований з різноманітними іншими інструментами керування подіями, такими як Eventbrite, Salesforce та Mailchimp.

Інтеграції дозволяють користувачам розширювати функціональність "Evento" та налаштовувати його відповідно до своїх конкретних потреб.

Висновок.

"Evento" є надійним веб-застосунком для управління подіями, який пропонує широкий спектр функціональних можливостей для організаторів і відвідувачів. Його можливості управління подіями, реєстрації, обробки платежів, комунікації, аналітики та інтеграції роблять його цінним інструментом для планування та проведення успішних заходів будь-якого розміру.

Рекомендації.

Для подальшого покращення функціональності "Evento" можна розглянути наступні рекомендації:

- додати можливість створення багатоденних подій;
- інтегрувати додаткові платіжні шлюзи для більшої гнучкості;
- забезпечити додаткові можливості налаштування для підтримки різних потреб користувачів.

3.5. Порівняння з конкуруючими рішеннями

У динамічній сфері управління подіями існує безліч веб-застосунків, які пропонують різні функціональні можливості та переваги. Щоб допомогти організаторам подій зробити обґрунтований вибір, ми порівняємо "Evento" з двома його головними конкурентами: Eventbrite та Cvent.

Оцінювані критерії.

Порівняння проводитиметься за такими ключовими критеріями:

- управління подіями: функції створення, управління та просування подій;
- реєстрація та обробка платежів: можливості для реєстрації учасників та безпечної обробки платежів;
- комунікації: інструменти для ефективного спілкування з учасниками;
- аналітика та звітність: функції відстеження ефективності подій та надання цінних даних;
- інтерфейс користувача: простота використання, інтуїтивність та загальний досвід користувача;
- інтеграції: можливості інтеграції з іншими інструментами та платформами;
- ціноутворення: вартість використання платформи та структура ціноутворення.

Порівняльний аналіз.

Управління подіями.

"Evento" пропонує комплексний набір функцій управління подіями, включаючи створення подій, керування реєстрацією, відстеження відвідуваності та багато іншого.

Eventbrite відомий своїми потужними можливостями управління подіями, особливо для платних заходів. Однак його безкоштовний план має обмежену функціональність.

Cvent пропонує найширший спектр функцій управління подіями, включаючи управління місцем проведення, планування програми та мобільні додатки для учасників.

Реєстрація та обробка платежів:

"Evento" дозволяє користувачам легко реєструватися на події та здійснювати платежі через інтегровану систему обробки платежів.

Eventbrite має надійну систему реєстрації та обробки платежів. Однак він стягує комісію за продаж квитків, що може бути недоліком для деяких організаторів.

Cvent пропонує надійну систему реєстрації та обробки платежів, але вона може бути дорожчою порівняно з іншими варіантами.

Комунікації:

"Evento" надає організаторам засоби для надсилання електронних листів та повідомлень учасникам. Він відстежує взаємодію, щоб допомогти оптимізувати комунікаційні стратегії.

Eventbrite пропонує схожі функції спілкування. Однак він надає обмежені можливості для налаштування та автоматизації комунікацій.

Cvent має найпотужніші функції комунікації серед трьох платформ, включаючи сегментацію аудиторії, автоматизовані послідовності електронних листів та інтеграцію з соціальними мережами.

Аналітика та звітність:

"Evento" надає базові звіти про ефективність подій. Однак його можливості аналітики можуть бути обмеженими для великих або складних подій.

Eventbrite пропонує надійні функції аналітики та звітності. Він надає докладні дані про продаж квитків, відвідуваність та взаємодію користувачів.

Svent має найдосконаліші можливості аналітики та звітності. Він надає глибокі дані та інструменти для вимірювання успіху подій та виявлення можливостей для покращення.

Інтерфейс користувача:

"Evento" має простий у використанні та інтуїтивно зрозумілий інтерфейс користувача. Організатори та учасники можуть легко знаходити потрібну інформацію та виконувати завдання.

Інтерфейс користувача Eventbrite також простий у використанні. Однак він може бути перевантаженим для деяких користувачів, особливо для складних подій.

Svent має найскладніший інтерфейс користувача серед трьох платформ. Він може бути складним для освоєння для нових користувачів, але пропонує широкий спектр можливостей налаштування.

Інтеграції:

"Evento" інтегрується з кількома інструментами керування подіями та сторонніми платформами. Однак його можливості інтеграції можуть бути обмежені порівняно з іншими рішеннями.

Eventbrite має широкий спектр інтеграцій з іншими інструментами та платформами, включаючи маркетингові і автоматизаційні системи та CRM.

Svent має найширший спектр інтеграцій з іншими інструментами та платформами, які охоплюють всі аспекти управління подіями.

Ціноутворення:

"Evento" пропонує гнучкі плани ціноутворення залежно від розміру та складності подій. Він має безкоштовний план для невеликих подій і платні плани з додатковими функціями.

Eventbrite має просту структуру ціноутворення на основі комісії за продаж квитків. Однак комісії можуть накопичуватися для великих або складних подій.

Cvent має більш складну структуру ціноутворення, яка базується на рівні функціональних можливостей та кількості учасників. Його плани можуть бути дорожчими, але вони пропонують більш широкий спектр функцій.

Висновок.

"Evento", Eventbrite та Cvent - це провідні веб-застосунки для управління подіями, які пропонують різний набір функцій та переваг. Організаторам подій слід ретельно розглянути свої вимоги та бюджет, щоб визначити найкраще рішення для їхніх потреб.

Evento:

Підходить для малих і середніх організаторів подій, які потребують базових або середніх функціональних можливостей за доступною ціною.

Eventbrite:

Підходить для організаторів платних подій, які хочуть отримати надійні функції управління подіями та реєстрації.

Cvent:

Підходить для великих і складних організаторів подій, яким потрібні найдосконаліші функціональні можливості, аналітика та інтеграції.

Остаточний вибір залежатиме від конкретних потреб, цілей та бюджету кожної організації.

3.6. Активність програмного забезпечення

Сеанс у програмі Evento характеризується першим введенням логіна користувача. Введена інформація потім передається на сервери Evento, які, в свою чергу, обмінюються їх даними з серверами бази даних, запитуючи їх про існування цього облікового запису.

Сервер відповідає, чи існує обліковий запис користувача чи ні. Якщо його немає, сесія закінчується, інакше - продовжується. Після цього користувач може попросити переглянути різні події, сервер Evento відповідно до інформації профілю користувача запропонує кілька подій запитуючи базу даних про них.

Для кожної події користувач може поставити лайк або не лайк. Якщо подія йому подобається, вона буде збережена у профілі користувача, і тоді в обох випадках йому буде запропоновано нову подію. Користувач може повторити лайки на стільки подій які йому подобаються, скільки він бажає, перш ніж завершити свій сеанс, вимкнувши програму або вийти із системи.

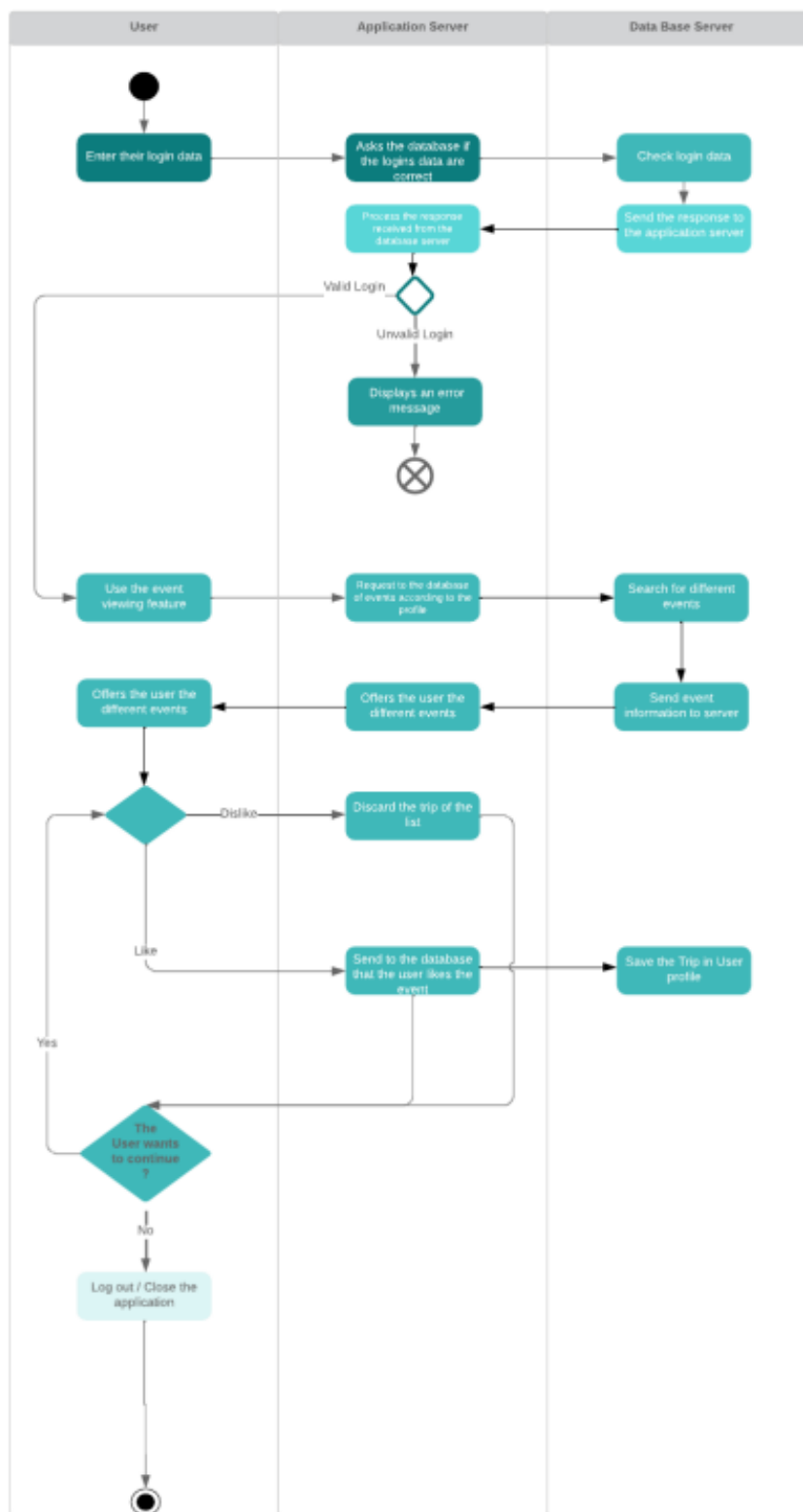


Рисунок 3.9. – Діаграма активності сеансу користувача на Evento

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності працівників у процесі трудової діяльності.

Умови праці – сукупність факторів виробничого середовища і трудового процесу, які впливають на здоров'я і працездатність людини в процесі її професійної діяльності

Основні завдання охорони праці:

- створення безпечних і здорових умов праці для працівників;
- попередження травм, професійних захворювань і нещасних випадків на виробництві;
- забезпечення санітарно-гігієнічних норм на робочих місцях;
- контроль за дотриманням вимог охорони праці роботодавцями та працівниками;
- компенсація працівникам шкоди, заподіяної внаслідок нещасних випадків на виробництві та професійних захворювань;

Основні напрямки охорони праці:

- законодавче та нормативне забезпечення;
- санітарно-гігієнічні та лікувально-профілактичні заходи;
- технічні та організаційні заходи;
- забезпечення засобами індивідуального та колективного захисту;
- навчання та інструктаж з охорони праці;
- контроль за дотриманням вимог охорони праці;

- розслідування та облік нещасних випадків та професійних захворювань;

Принципи охорони праці:

- * пріоритет життя і здоров'я працівників над результатами економічної діяльності;
- * комплексний підхід до забезпечення охорони праці;
- * профілактичний характер заходів з охорони праці;
- * соціальне партнерство у сфері охорони праці;
- * відповідальність роботодавців і працівників за стан охорони праці;

Охорона праці відіграє важливу роль, як суспільний чинник, оскільки, якими б вагомими не були трудові здобутки, вони не можуть компенсувати людині втраченого здоров'я, а тим більше життя – те і інше дається лише один раз. Тому цілком зрозуміло, що вивченню питань охорони праці приділяється серйозна увага. Представники багатьох наук вивчають проблеми створення безпечних та нешкідливих умов та засобів праці. Адже саме за таких обставин людина здатна працювати високопродуктивно, створювати необхідний матеріальний потенціал суспільства, добробут усіх громадян.

Основоположним законодавчим документом в галузі основи праці є Закон України «Про охорону праці» для якого поширюється на всі підприємства, установи і організації незалежно від форми власності та видів діяльності, на усіх громадян які працюють, а також залучені до праці на цих підприємствах. Все це затверджено у Законі України про «Охорону праці», який був прийнятий Верховною Радою України 14 жовтня 1992 року. Цей Закон визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини

між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні.

Мобільний застосунок "Evento" призначений для планування, координації та проведення подій. Він часто використовується для керування великими заходами, що вимагають участі численної кількості учасників, співробітників та підрядників. Для забезпечення безпеки та здоров'я всіх осіб, залучених до використання застосунку "Evento", необхідно дотримуватися наступних правил охорони праці.

1. Загальні вимоги

До застосунку "Evento" мають доступ лише авторизовані користувачі.

Усі користувачі повинні пройти належне навчання та інструктаж щодо безпечного та належного використання застосунку.

Забороняється використовувати застосунок "Evento" для будь-яких цілей, які не пов'язані з плануванням або координацією подій.

Користувачі повинні захищати свої облікові записи та паролі від несанкціонованого доступу.

Застосунок "Evento" не слід використовувати під час керування транспортними засобами або виконання інших завдань, які вимагають повної уваги.

Користувачі повинні негайно повідомляти про будь-які небезпечні ситуації або несправності застосунку відповідному керівнику.

2. Вимоги до обладнання

Забороняється використовувати застосунок "Evento" на пристроях, які не відповідають технічним вимогам, встановленим розробником.

Пристрої, що використовуються для доступу до застосунку "Evento", повинні мати надійне підключення до Інтернету.

Акумулятори пристроїв повинні бути повністю заряджені перед використанням застосунку "Evento".

По можливості слід використовувати тримачі або підставки для пристроїв, щоб звільнити руки користувача.

3. Вимоги до робочого місця

Застосунок "Evento" слід використовувати в добре освітлених та вентильованих місцях.

Робоче місце має бути вільним від перешкод та небезпек.

Користувачі повинні підтримувати правильну поставу та уникати тривалих періодів сидіння чи стояння.

Необхідно робити регулярні перерви, щоб уникнути втоми та м'язового напруження.

4. Техніка безпеки

Забороняється використовувати застосунок "Evento" у небезпечних зонах, таких як ділянки з електричними небезпеками або рухомим обладнанням.

Користувачі повинні бути обережними, коли використовують застосунок "Evento" в місцях із великим скупченням людей.

Особиста інформація, що зберігається в застосунку "Evento", повинна бути захищена від несанкціонованого доступу.

Користувачі повинні бути обережними при передачі файлів або зображень через застосунок "Evento".

Необхідно дотримуватися всіх місцевих та державних законів та правил щодо використання мобільних пристроїв.

5. Навчання та кваліфікація

Усі користувачі застосунку "Evento" повинні пройти початкове навчання та інструктаж, а також регулярні оновлення щодо безпечного та належного використання застосунку.

Навчання має охоплювати наступні теми:

Загальні вимоги охорони праці

Вимоги до обладнання та робочого місця

Техніка безпеки

Конфіденційність та безпека даних

Етика використання соціальних мереж

Користувачі повинні успішно скласти навчальний тест, щоб підтвердити своє розуміння та дотримання правил охорони праці.

6. Регулярний перегляд та оновлення

Інструкція з охорони праці має регулярно переглядатися та оновлюватися в міру оновлення застосунку "Evento" або змін у законодавстві та нормативних актах.

Відповідальний керівник повинен переглядати інструкцію з охорони праці щонайменше раз на рік і вносити необхідні зміни.

Користувачам має бути повідомлено про будь-які зміни в інструкції з охорони праці.

7. Повідомлення про порушення та відповідальність

Порушення правил охорони праці під час використання застосунку "Evento" може призвести до дисциплінарних стягнень або інших санкцій.

Відповідальний керівник несе відповідальність за забезпечення дотримання користувачами правил охорони праці.

Користувачі мають право відмовитися від використання застосунку "Evento", якщо вони вважають, що це становить небезпеку для їхнього здоров'я та безпеки.

Додаткові ресурси

Наступні ресурси можуть бути корисними для отримання додаткової інформації щодо охорони праці під час використання застосунку "Evento":

- вебсайт розробника застосунку "Evento";
- місцеві та державні органи охорони праці;
- торгові асоціації та професійні організації;
- інтернет-форуми та групи користувачів, присвячені застосунку "Evento";

Дотримання цих правил охорони праці є обов'язковим для всіх користувачів застосунку "Evento". Керівники несуть відповідальність за забезпечення розуміння та дотримання цих правил усіма працівниками та підрядниками, які використовують цей застосунок.

ВИСНОВКИ

Застосунок "Evento" було розроблено як мобільна платформа для надання користувачам персоналізованої інформації про майбутні події у їхньому регіоні. Щоб забезпечити відповідність додатка потребам цільової аудиторії, перед розробкою було проведено ретельне дослідження. У цьому звіті викладено ключові результати цього дослідження.

Дослідження складалося з наступних етапів:

1) огляд літератури

вивчення існуючих досліджень та тенденцій у галузі подій та мобільних додатків;

2) інтерв'ю з користувачами

проведення глибинних інтерв'ю з потенційними користувачами, щоб зрозуміти їхні потреби, бажання та очікування щодо додатків для подій;

3) аналіз конкурентів

аналіз функціональних можливостей, сильних і слабких сторін конкурентних додатків для подій.

Ключові результати:

1. Цільова аудиторія

Дослідження виявило, що цільова аудиторія для Evento складається з осіб віком від 18 до 45 років, які:

зацікавлені у відвідуванні різноманітних подій, включаючи концерти, фестивалі, конференції та спортивні заходи;

цінують зручність і простоту у пошуку та придбанні квитків на події;

віддають перевагу персоналізованим рекомендаціям, які відповідають їхнім інтересам.

2. Потреби та очікування

Дослідження визначило такі основні потреби та очікування цільової аудиторії від застосунок для подій:

персоналізація - застосунок повинен надавати персоналізовані рекомендації на основі історії пошуку та вподобань користувача;

зручність використання - застосунок повинен мати інтуїтивно зрозумілий інтерфейс та швидкий час відгуку;

комплексна інформація про події - застосунок повинен надавати детальну інформацію про події, включно з датою, часом, місцем розташування та вартістю квитків;

легке придбання квитків - застосунок повинен дозволяти користувачам купувати квитки на події швидко та безпечно;

спільне використання подій - застосунок повинен дозволяти користувачам легко ділитися подіями з друзями та родиною.

3. Конкурентний аналіз

Аналіз конкурентів виявив, що на ринку існує декілька додатків для подій, але жоден з них не пропонує всебічного набору функцій, що відповідають потребам цільової аудиторії Evento. Основні відмінності між Evento та його конкурентами включають:

персоналізація: Evento використовує передові алгоритми для надання високоперсоналізованих рекомендацій;

інтуїтивний інтерфейс: Evento розроблено з інтуїтивно зрозумілим інтерфейсом, який простий у навігації;

інтеграція з соціальними мережами: Evento дозволяє користувачам легко ділитися подіями з друзями та родиною через соціальні мережі.

Висновки:

Дослідження виявило чітку потребу в застосунку для подій, такому як Evento, який надає персоналізовану інформацію, зручність використання та всебічні функції. Результати дослідження слугуватимуть основою для розробки та вдосконалення додатка Evento, який відповідатиме потребам і перевищуватиме очікування цільової аудиторії.

За результатами досліджень можливо надати рекомендації щодо подальших кроків у розвитку web застосунку «Evento».

"Evento" - це перспективний веб-застосунок, який має потенціал стати провідною платформою для управління подіями. Щоб досягти цієї мети, необхідно постійно вдосконалювати та розширювати функціональні можливості програми. Ці рекомендації визначають конкретні кроки для просування Evento на наступний етап розвитку.

1. Оптимізація користувацького досвіду (UX):

спрощення інтерфейсу: видалити непотрібні елементи та оптимізувати навігацію, щоб забезпечити безперешкодне та інтуїтивно зрозуміле користування;

індивідуальні налаштування: дозволити користувачам налаштовувати свій інтерфейс відповідно до своїх уподобань, включно з темами, макетами та фільтрами;

мобільна оптимізація: оптимізувати програму для мобільних пристроїв, забезпечуючи бездоганну роботу на різних платформах та розмірах екрана.

2. Розширення функціональності керування подіями:

календар з можливістю синхронізації: інтегрувати календар, який дозволить користувачам переглядати та управляти подіями з інших календарів та пристроїв;

реєстрація для кількох користувачів: додати підтримку реєстрації для кількох користувачів, що дозволить організаторам запрошувати та керувати кількома учасниками для кожної події;

автоматичне складання розкладів: впровадити інструмент для автоматичного складання розкладів, що допоможе організаторам оптимізувати час подій та уникнути конфліктів.

3. Інтеграція з іншими інструментами:

постачальники платіжних систем: інтеграція з постачальниками платіжних систем, такими як PayPal і Stripe, дозволить користувачам легко здійснювати платежі за реєстрацію та інші послуги;

системи CRM: інтегрувати Evento з системами управління взаємовідносинами з клієнтами (CRM), щоб організатори могли зберігати та керувати контактною інформацією учасників;

інструменти маркетингових кампаній: додати підтримку інструментів маркетингових кампаній, таких як MailChimp і Google AdWords, щоб організатори могли просувати свої події ефективніше.

4. Аналітика та звітування:

аналітика відвідуваності: впровадити інструменти аналітики, щоб відстежувати кількість відвідувань, поведінку користувачів та ефективність сторінок;

звіти про події: передбачити можливість надавати користувачам детальні звіти про події, включаючи кількість учасників, дані про відвідуваність та відгуки;

панель керування організатора: створити спеціальну панель керування для організаторів, що забезпечить їм швидкий доступ до всіх важливих показників і дасть можливість управляти подіями з одного екрана.

5. Додаткові функції:

функціональність віртуальних подій: розширити Evento, додаючи підтримку віртуальних подій, дозволяючи організаторам проводити онлайн-заходи з трансляціями в прямому ефірі та інтерактивними елементами;

багатомовна підтримка: перекласти інтерфейс Evento на кілька мов, щоб охопити ширшу аудиторію;

форум спільноти: створити форум спільноти, де користувачі можуть взаємодіяти, ділитися досвідом і отримувати підтримку.

Висновок:

Evento створено на мові Java з використанням бази даних firebase та android studio навколишнє середовище. Кожна технологія потребувала часу, щоб її освоїти та застосувати.

Реалізація цих рекомендацій допоможе Evento стати ще більш потужним та комплексним веб-застосунком для управління подіями. Постійно оптимізуючи користувацький досвід, розширюючи функціональність, інтегруючись з іншими інструментами та надаючи цінні аналітичні дані, Evento може закріпити свої позиції як надійне рішення для організаторів подій та учасників у всьому світі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Event Technology Census 2023 -
<https://www.eventmanagerblog.com/event-tech-census-2023>
2. The Ultimate Guide to Event Management Software -
<https://www.hubspot.com/blog/event-management-software>
3. 10 Best Event Management Software Programs in 2023 -
<https://www.businessnewsdaily.com/10860-event-management-software.html>
4. Eventbrite: Event Management and Ticketing -
<https://www.eventbrite.com/>
5. Cvent: The Leading Event Management Platform - <https://www.cvent.com/>
6. Whova: The Event App Platform - <https://whova.com/>
7. Accelevents: The Virtual and Hybrid Event Platform -
<https://www.accelevents.com/>
8. Splash: The Marketing Event Platform - <https://splashthat.com/>
9. National Institute of Standards and Technology -
https://csrc.nist.gov/glossary/term/system_owner
10. Non-Functional requirement ideas retrieved from
<https://studiosoftware.com/blog/non-functional-requirements-definition-and-tips-for-better-requirements-gathering/>
11. Android Studio - <https://developer.android.com/studio/intro#:~:text=Android%20Studio%20uses%20Gradle%20as,independently%20from%20the%20command%20line.>
12. Android SDK - <https://developer.android.com/studio/debug/apk-analyzer>
13. Firebase Database - <https://developer.android.com/studio/write/firebase>
14. How firebase works - https://blog.back4app.com/firebase/#What_is_Firebase_and_how_it_works

ДОДАТОК А

Технічне завдання до застосунку «Evento»

1. Вступ

"Evento" - це мобільний застосунок, розроблений для надання користувачам персоналізованої інформації про майбутні події у їхньому регіоні.

2. Мета

Метою цього технічного завдання є визначення вимог, специфікацій та функціональних можливостей застосунку "Evento".

3. Цільова аудиторія

Цільовою аудиторією є особи, які бажають бути в курсі майбутніх подій у їхньому районі та брати в них участь.

4. Функціональні вимоги

Пошук подій: користувачі повинні мати можливість шукати події за такими критеріями, як:

категорія (напр., концерти, фестивалі, конференції);

місце розташування;

дати;

ключові слова.

Персоналізація: застосунок має надавати персоналізовані рекомендації щодо подій на основі історії пошуку та вподобань користувача.

Деталі подій: користувачі повинні мати можливість переглядати детальну інформацію про події, включаючи:

назву, опис та зображення;

дати та час проведення;
розташування;
вартість квитків;
посилання на придбання квитків.

Створення подій: організатори подій повинні мати можливість створювати та публікувати події в застосунку.

Управління квитками: користувачі повинні мати можливість купувати квитки безпосередньо через застосунок з використанням різних платіжних систем.

Спільне використання подій: користувачі повинні мати можливість ділитися подіями з друзями та родиною через соціальні мережі та інші канали.

Офлайн-режим: застосунок повинен функціонувати в режимі офлайн, дозволяючи користувачам отримувати доступ до раніше шуканих подій.

5. Нефункціональні вимоги

Швидка та ефективна робота: застосунок повинен швидко вантажитися та надавати результати пошуку в режимі реального часу.

Інтуїтивно зрозумілий інтерфейс: інтерфейс користувача має бути простим у використанні та інтуїтивно зрозумілим для всіх користувачів.

Масштабованість: застосунок має бути здатний обробляти велику кількість одночасних запитів і зберігати дані про численні події.

Надійність: застосунок повинен бути надійним і доступним більшу частину часу.

Сумісність: застосунок повинен бути сумісний з пристроями Android та iOS.

6. Інтеграції

Платіжні системи: застосунок має бути інтегрований з платіжними системами для обробки транзакцій квитків.

Соціальні мережі: застосунок має бути інтегрований із соціальними мережами для обміну подіями.

Картографічні сервіси: застосунок має бути інтегрований з картографічними сервісами для відображення місць проведення подій.

7. Технічні вимоги

Застосунок повинен бути розроблений із використанням наступних технологій:

мова програмування – Java;

база даних – Firebase;

ОС - Android 8.1.0 або новіша версія;

створення інтерфейсу користувача за допомогою React, Redux та матеріал-UI.

8. Документація

Підрядник повинен надати всю необхідну документацію, включаючи:

функціональні та нефункціональні вимоги;

архітектуру системи;

посібник користувача;

посібник розробника.

9. Стадії та етапи розробки

Стадії та етапи розробки застосунку та терміни їх виконання повинні відповідати затвердженому графіку проектування кваліфікаційної роботи та зводитись до таких, що наведені в таблиці 1.1.

Таблиця 1

Стадії, етапи розробки програми, та терміни їх виконання

Стадії розробки	Етапи робіт	Зміст робіт	Терміни виконання
Технічне завдання	Розробка технічного завдання.	Обґрунтування необхідності розробки програми	05.03.24 – 25.03.24
	Затвердження технічного завдання.	Розробка пояснювальної записки. Узгодження і затвердження технічного завдання.	27.03.24 – 06.04.24
Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних, уточнення методів рішення завдань, загальний опис алгоритму.	06.04.24 – 11.04.24
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту.	11.04.24 – 13.04.24
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних, розробка алгоритму рішення задачі, розробка структури програми.	15.04.24 – 17.04.24
	Затвердження технічного проекту	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту.	18.04.24 – 21.04.24
Робочий проект	Розробка програми	Програмування та налагодження програми	21.04.24 – 28.04.24
	Випробування програми	Розробити, узгодити та затвердити програму і методику випробувань, коригування програми за результатами випробувань.	28.04.24 – 06.05.24
	Розробка програмної документації	Розробка програмної документації	06.05.24 – 11.05.24

ДОДАТОК Б

Програма і методика випробувань застосунку «Evento»

1. Мета

Мета цієї програми та методики випробувань полягає в тому, щоб визначити, чи відповідає застосунок "Evento" функціональним та нефункціональним вимогам, викладеним у технічному завданні.

2. Область застосування

Ця програма та методика охоплюють випробування всіх основних функціональних можливостей та характеристик додатка "Evento", включаючи:

- пошук подій;
- персоналізація;
- деталі подій;
- створення подій;
- управління квитками;
- спільне використання подій;
- офлайн-режим.

3. Типи випробувань

Будуть проведені наступні типи випробувань:

- функціональне тестування: перевірка того, чи застосунок функціонує відповідно до функціональних вимог;

- нефункціональне тестування: перевірка того, чи застосунок відповідає нефункціональним вимогам, таким як продуктивність, надійність та зручність використання;
- випробування користувальницького інтерфейсу (UX): Оцінка зручності використання, інтуїтивності та естетики інтерфейсу користувача.

4. Методика випробувань

Випробування будуть проводитися за допомогою наступних методів:

- тестування чорного ящика: тестування функціональності додатка без знання його внутрішньої реалізації;
- тестування білого ящика: тестування внутрішньої логіки та структури додатка;
- дослідницьке тестування: тестування додатка шляхом вільного дослідження та пошуку несподіваної поведінки;
- тестування регресії: перевірка того, що зміни в коді не порушили існуючу функціональність.

5. Тестові сценарії

Тестові сценарії будуть розроблені для охоплення всіх функціональних вимог та нефункціональних характеристик. Сценарії будуть включати:

пошук подій:

- пошук подій за різними критеріями;
- перевірка правильності та повноти результатів пошуку;

персоналізація:

- перевірка того, чи застосунок надає персоналізовані рекомендації на основі історії пошуку та вподобань користувача;

деталі подій:

- перевірка того, чи застосунок відображає точну та повну інформацію про події;
- перевірка того, чи застосунок дозволяє користувачам купувати квитки;

створення подій:

- перевірка того, чи організатори подій можуть створювати та публікувати події в застосунку;

управління квитками:

- перевірка того, чи застосунок дозволяє користувачам купувати та керувати квитками;

спільне використання подій:

- перевірка того, чи застосунок дозволяє користувачам ділитися подіями з друзями та родиною;

офлайн-режим:

- перевірка того, чи застосунок функціонує в режимі офлайн;

продуктивність:

- вимірювання часу завантаження, часу відгуку та споживання пам'яті;

надійність:

- перевірка того, чи застосунок доступний і стабільний при високих навантаженнях;

зручність використання:

- оцінка інтуїтивності, простоти використання та естетики інтерфейсу користувача.

6. Інструменти тестування

Для виконання випробувань будуть використовуватися наступні інструменти:

Appium: для автоматизованого тестування;

Selenium WebDriver: для веб-тестування;

JUnit: для блокового тестування;

Postman: для тестування API;

Firebase Test Lab: для тестування пристроїв.

7. Очікувані результати

Очікуваними результатами випробувань є:

- перелік виявлених дефектів та недоліків;
- звіт про результати випробувань, включаючи показники проходження та невдачі;
- рекомендації щодо покращення додатка.

При виявленні помилок у роботі програмного продукту розробник складає перелік цих помилок і самостійно визначає термін їхнього виправлення. Після цього розробник проводить повторне випробування у повному обсязі.

8. Графік випробувань

Випробування будуть проводитися ітераційно протягом усього циклу розробки.

ДОДАТОК В

Інструкція користувача до застосунку «Evento»

1. Вступ

"Evento" - це мобільний застосунок, розроблений для надання користувачам персоналізованої інформації про майбутні події у їхньому регіоні.

2. Завантаження та встановлення

Завантажте та встановіть застосунок "Evento" з відповідного магазину додатків для вашого пристрою.

3. Створення облікового запису

Для початку користування додатком, створіть обліковий запис за допомогою адреси електронної пошти та пароля.

4. Пошук подій

Для пошуку події у вашому районі, виконайте такі дії:

- торкніться значка пошуку на головному екрані;
- введіть критерії пошуку, такі як назва події, категорія або місце розташування;
- торкніться кнопки "Пошук".

5. Персоналізація

Застосунок "Evento" надає персоналізовані рекомендації на основі історії пошуку та вподобань користувача.

Для налаштування параметрів персоналізації, виконайте такі дії:

- торкніться значка профілю в правому нижньому куті;

- виберіть "Налаштування";
- увімкніть або вимкніть перемикачі поруч із категоріями, які вас цікавлять.

6. Деталі подій

Для перегляду детальної інформації про подію, торкніться її в результатах пошуку. Сторінка події містить таку інформацію:

назву, опис та зображення;
дати та час проведення;
розташування;
вартість квитків;
посилання на придбання квитків.

7. Створення подій

Організатори подій можуть створювати та публікувати події в застосунку "Evento".

Для створення події, виконайте такі дії:

- торкніться значка "Створити подію" на головному екрані;
- введіть деталі події, такі як назва, опис, дата, час та місце розташування;
- натисніть кнопку "Створити".

8. Управління квитками

Користувачі можуть купувати та керувати квитками безпосередньо через застосунок "Evento". Для придбання квитку, виконайте такі дії:

- торкніться кнопки "Купити квитки" на сторінці події;
- виберіть тип квитка та кількість;
- натисніть кнопку "Продовжити";
- введіть платіжні дані та натисніть кнопку "Купити".

Після придбання квитки будуть доступні у вашому обліковому записі.

9. Спільне використання подій

Користувачі можуть ділитися подіями з друзями та родиною через соціальні мережі та інші канали. Для того, щоб поділитися подією, виконайте такі дії:

- торкніться кнопки "Поділитися" на сторінці події;
- виберіть канал, яким ви хочете поділитися;
- натисніть кнопку "Надіслати".

10. Офлайн-режим

Застосунок "Evento" функціонує в режимі офлайн, щоб дозволити користувачам отримувати доступ до раніше обраних подій. Для можливості скористуватися офлайн-режимом, виконайте такі дії:

- переконайтеся, що у вас є підключення до Інтернету;
- знайдіть події, які вас цікавлять;
- синхронізуйте дані, торкнувшись кнопки "Синхронізувати" на головному екрані.

Після синхронізації даних ви можете отримувати доступ до подій офлайн.

11. Зворотній зв'язок

Якщо у вас є якісь відгуки або пропозиції щодо додатка "Evento", будь ласка, напишіть нам на адресу support@evento.com.

ДОДАТОК Г

Опис застосунку «Evento»

1 Загальні відомості

Назва застосунку: «Evento».

Область застосування: розваги.

2 Функціональне призначення

Застосунок «Evento» призначений для надання користувачам персоналізованої інформації про майбутні події у їхньому регіоні.

Основна цільова аудиторія для Evento - це особи, які бажають:
мати інформацію щодо подій, які відбуваються у їхньому районі;
знайти події, що відповідають їхнім інтересам;
придбати квитки на події;
ділитися подіями з друзями та родиною;
отримувати сповіщення про майбутні події.

Області застосування Evento:

Відкриття нових подій: користувачі можуть використовувати Evento, щоб знаходити події, які відповідають їхнім інтересам, наприклад концерти, фестивалі, конференції та спортивні заходи.

Персоналізація досвіду: Evento використовує історію пошуку та вподобання користувача, щоб надавати персоналізовані рекомендації щодо подій.

Легкий доступ до інформації про подію: користувачі можуть переглядати детальну інформацію про події, включно з датою, часом, місцем розташування, вартістю квитків та посиланням на придбання квитків.

Зручне придбання квитків: користувачі можуть купувати квитки на події безпосередньо через застосунок Evento за допомогою різних платіжних систем.

Спільне використання подій: користувачі можуть ділитися подіями з друзями та родиною через соціальні мережі та інші канали.

Офлайн-доступ: Evento функціонує в режимі офлайн, дозволяючи користувачам отримувати доступ до раніше шуканих подій та свого облікового запису навіть за відсутності підключення до Інтернету.

Управління подіями: організатори подій можуть використовувати Evento для створення та публікації подій, а також для управління квитками та відстеження статистики відвідуваності.

Загалом, Evento - це універсальний застосунок-компаньйон для будь-кого, хто хоче знати про найкращі події у своєму районі та легко отримувати до них доступ.

3. Технічні засоби

Вимоги до пристроїв:

ОС: Android 8.1.0 або новіша версія

Вільне місце на диску: 200 МБ

Швидкість Інтернету: від 10 до 20 Мбіт/с

Доступ до місцезнаходження: потрібен

ДОДАТОК Д

Текст виконуючого коду застосунку «Evento»

```
#!/usr/bin/env sh

#
# Copyright 2015 the original author or authors.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#   https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

#####
#####

##
## Gradle start up script for UN*X
##
```

```
#####  
#####
```

```
# Attempt to set APP_HOME  
# Resolve links: $0 may be a link  
PRG="$0"  
# Need this for relative symlinks.  
while [ -h "$PRG" ] ; do  
    ls=`ls -ld "$PRG"`  
    link=`expr "$ls" : '.*-> \(.*)$`  
    if expr "$link" : '/.*' > /dev/null; then  
        PRG="$link"  
    else  
        PRG=`dirname "$PRG"`"/$link"  
    fi  
done  
SAVED=""`pwd`  
cd "`dirname \"$PRG\"`/" >/dev/null  
APP_HOME=""`pwd -P`  
cd "$SAVED" >/dev/null
```

```
APP_NAME="Gradle"  
APP_BASE_NAME=`basename "$0`
```

Add default JVM options here. You can also use JAVA_OPTS and GRADLE_OPTS to pass JVM options to this script.

```
DEFAULT_JVM_OPTS="-Xmx64m" "-Xms64m"
```

```

# Use the maximum available, or set MAX_FD != -1 to use that value.
MAX_FD="maximum"

warn () {
    echo "$*"
}

die () {
    echo
    echo "$*"
    echo
    exit 1
}

# OS specific support (must be 'true' or 'false').
cygwin=false
msys=false
darwin=false
nonstop=false
case "`uname`" in
    CYGWIN* )
        cygwin=true
        ;;
    Darwin* )
        darwin=true
        ;;
    MINGW* )
        msys=true

```

```
;;
NONSTOP* )
    nonstop=true
;;
esac
```

```
CLASSPATH=$APP_HOME/gradle/wrapper/gradle-wrapper.jar
```

```
# Determine the Java command to use to start the JVM.
```

```
if [ -n "$JAVA_HOME" ] ; then
    if [ -x "$JAVA_HOME/jre/sh/java" ] ; then
        # IBM's JDK on AIX uses strange locations for the executables
        JAVACMD="$JAVA_HOME/jre/sh/java"
    else
        JAVACMD="$JAVA_HOME/bin/java"
    fi
    if [ ! -x "$JAVACMD" ] ; then
        die "ERROR: JAVA_HOME is set to an invalid directory: $JAVA_HOME
```

```
Please set the JAVA_HOME variable in your environment to match the
location of your Java installation."
```

```
fi
else
    JAVACMD="java"
    which java >/dev/null 2>&1 || die "ERROR: JAVA_HOME is not set and no
'java' command could be found in your PATH.
```

Please set the JAVA_HOME variable in your environment to match the location of your Java installation."

fi

Increase the maximum file descriptors if we can.

if ["\$cygwin" = "false" -a "\$darwin" = "false" -a "\$nonstop" = "false"]; then

MAX_FD_LIMIT=`ulimit -H -n`

if [\$? -eq 0]; then

if ["\$MAX_FD" = "maximum" -o "\$MAX_FD" = "max"]; then

MAX_FD="\$MAX_FD_LIMIT"

fi

ulimit -n \$MAX_FD

if [\$? -ne 0]; then

warn "Could not set maximum file descriptor limit: \$MAX_FD"

fi

else

warn "Could not query maximum file descriptor limit: \$MAX_FD_LIMIT"

fi

fi

For Darwin, add options to specify how the application appears in the dock

if \$darwin; then

GRADLE_OPTS="\$GRADLE_OPTS \"-Xdock:name=\$APP_NAME\" \"-
Xdock:icon=\$APP_HOME/media/gradle.icns\""

fi

For Cygwin or MSYS, switch paths to Windows format before running java

if ["\$cygwin" = "true" -o "\$msys" = "true"]; then

```

APP_HOME=`cygpath --path --mixed "$APP_HOME"`
CLASSPATH=`cygpath --path --mixed "$CLASSPATH"`

JAVACMD=`cygpath --unix "$JAVACMD"`

# build the pattern for arguments to be converted via cygpath
ROOTDIRSRAW=`find -L / -maxdepth 1 -mindepth 1 -type d 2>/dev/null`
SEP=""
for dir in $ROOTDIRSRAW ; do
    ROOTDIRS="$ROOTDIRS$SEP$dir"
    SEP="|"
done
OURCYGPATTERN="(^($ROOTDIRS))"
# Add a user-defined pattern to the cygpath arguments
if [ "$GRADLE_CYGPATTERN" != "" ] ; then

OURCYGPATTERN="$OURCYGPATTERN|($GRADLE_CYGPATTERN)"
fi
# Now convert the arguments - kludge to limit ourselves to /bin/sh
i=0
for arg in "$@" ; do
    CHECK=`echo "$arg"|egrep -c "$OURCYGPATTERN" -`
    CHECK2=`echo "$arg"|egrep -c "^-"`           ### Determine if
an option

    if [ $CHECK -ne 0 ] && [ $CHECK2 -eq 0 ] ; then        ### Added a
condition
        eval `echo args$i`=`cygpath --path --ignore --mixed "$arg"`

```



```

else
    eval `echo args$i`="\"$arg\""
fi
i=`expr $i + 1`
done
case $i in
    0) set -- ;;
    1) set -- "$args0" ;;
    2) set -- "$args0" "$args1" ;;
    3) set -- "$args0" "$args1" "$args2" ;;
    4) set -- "$args0" "$args1" "$args2" "$args3" ;;
    5) set -- "$args0" "$args1" "$args2" "$args3" "$args4" ;;
    6) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" ;;
    7) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" ;;
    8) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6"
"$args7" ;;
    9) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6"
"$args7" "$args8" ;;
esac
fi

# Escape application args
save () {
    for i do printf %s\n "$i" | sed "s/'/\\\\'/g;1s/^/';\$/s/\$/' \\\\" ; done
    echo " "
}

APP_ARGS=`save "$@"`

```

Collect all arguments for the java command, following the shell quoting and substitution rules

```
eval set -- $DEFAULT_JVM_OPTS $JAVA_OPTS $GRADLE_OPTS "\"-
Dorg.gradle.appname=$APP_BASE_NAME\"" -classpath "\"$CLASSPATH\""
org.gradle.wrapper.GradleWrapperMain "$APP_ARGS"
```

```
exec "$JAVACMD" "$@"
```