

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА

Факультет прикладної математики та інформатики

(повне найменування назва факультету)

Інформаційних систем

(повна назва кафедри)

Дипломна робота

Розробка Web - застосунку для подорожей, з застосуванням Maps API

Виконала: студентка групи _____ ПМІ - 44

спеціальності _____ 122 – комп'ютерні науки

(шифр і назва спеціальності)

_____ **Легедза Ю. В.**

(підпис)

(прізвище та ініціали)

Керівник _____ **Бернакевич І. Є.**

(підпис)

(прізвище та ініціали)

Рецензент _____

(підпис)

(прізвище та ініціали)

Львів – 2023

Факультет _____ *прикладної математики та інформатики* _____
Кафедра _____ *інформаційних систем* _____
Освітньо-кваліфікаційний рівень _____
Напрямок підготовки _____
Спеціальність _____ *122 Комп'ютерні науки* _____

«ЗАТВЕРДЖУЮ»

Зав. кафедрою проф. Шинкаренко Г.А.

« 7 » вересня 2022 р.

ЗАВДАННЯ

НА ДИПЛОМНУ (КВАЛІФІКАЦІЙНУ) РОБОТУ СТУДЕНТА

Легадза Юлія – Яна Василівна

(прізвище, ім'я, по батькові)

1. Тема роботи

Розробка Web - застосунку для подорожей, з застосуванням Maps API

керівник роботи *доц. Бернакевич І.Є.*

затверджені Вченою радою факультету від « 13 » вересня 2023 р., №15

2. Строк подання студентом роботи 12.06.2023

3. Вихідні дані до роботи _____

1. Google Maps API [Електронний ресурс]. - Режим доступу: <https://developers.google.com/maps/documentation/javascript>

2. Node.js [Електронний ресурс]. - Режим доступу: <https://nodejs.org/en/docs>

3. Axios documentation [Електронний ресурс]. - Режим доступу: <https://axios-http.com/docs/intro>

4. Google Maps Platform [Електронний ресурс]. - Режим доступу: <https://developers.google.com/maps>

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

1. Проведення аналізу існуючих рішень;

2. Формування вимог до додатку;

3. Аналіз та вибір інструментів для розробки додатку;

4. Розробка власного додатку;

5. Загальний висновок по додатку;

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 14.09.22

КАЛЕНДАРНИЙ ПЛАН

№	Найменування етапів дипломної (кваліфікаційної) роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд існуючих систем та технологій</i>	<i>вересень 2022</i>	
2.	<i>Постановка задач</i>	<i>жовтень 2022</i>	
3.	<i>Розробка схем та алгоритмів</i>	<i>листопад 2022</i>	
4.	<i>Програмна реалізація</i>	<i>грудень-березень 2023</i>	
5.	<i>Апробація</i>	<i>квітень 2023</i>	
6.	<i>Оформлення роботи</i>	<i>травень 2023</i>	

Студент Легедза Ю.Я.
підпис

Керівник роботи доц Бернакевич І.Є.
підпис

Зміст

Анотація.....	5
Вступ.....	6
Розділ 1. Завдання та вимоги для веб-застосунку.....	8
1.1 User Stories.....	8
Розділ 2. Проектування клієнтської частини.....	10
2.1 Створення прототипу.....	11
Розділ 3. Проектування серверної системи.....	16
3.1 Середовище використане для створення веб-сайту - Visual Studio Code.....	16
3.2 Вибір мови програмування.....	16
3.3 Node.js - серверна частина.....	17
3.4 Google Maps API.....	18
Розділ 4. Створення веб-сайту.....	22
4.1 Setup.....	22
4.2 Project Setup.....	22
4.3 Створення Header частини.....	23
4.4 Створення List частини.....	23
4.5 Створення Map частини.....	24
4.6 Створення PlaceDetails частини.....	25
4.7 Як відбувається підключення Google Maps API.....	26
Розділ 5. Приклади реалізації.....	28
Висновок.....	35
Список використаних джерел.....	36

Анотація

Ця дипломна робота присвячена створенню веб-застосунку для подорожей, який допомагає користувачам знайти та оцінити різні місця для відпочинку, такі як ресторани, готелі, парки, меморіали, музеї, галереї та інші. В додатку використано Google Maps API, за допомогою нього відображаємо місця на карті та створюємо можливість пошуку та фільтрації місць за типом та рейтингом. У цій роботі описано процес створення веб-застосунку, включаючи аналіз вимог, проектування та архітектуру додатку та написання коду. Крім того у роботі описано різні технології та підходи, які було використано у процесі розробки, такі як, React, Material - UI, Node.js та інші.

Результат цієї роботи є готовий функціональний застосунок, який може бути використано користувачами для пошуку та оцінки різних місць для проведення вільного часу.

Вступ

Метою написання цієї дипломної роботи є, розробка онлайн – застосунку з картою, яка містить інформацію про ресторани, кафе та інші цікаві місця для розваг у вашому місті, або місті яке ви обираєте за допомогою пошуку. У цій інформації сказано про рейтинги та відгуки відвідувачів.

Онлайн-карти з інформацією про рейтинг ресторанів та інших місць є достатньо важливим інструментом для комфортного та ефективного планування подорожей. У теперішній час люди все більше подорожують, саме тому часто виникають проблеми, такі як, незнання місцевості та місцевих закладів, атракцій для відвідування, та незнання про якість послуг та цін. Саме тому цей застосунок, який містить детальну інформацію про місце яке цікавить туриста або місцевого, який вирішив дослідити своє місто, дозволить легко знайти та оцінити найкращі місця для відвідування в новому місці, дізнатись про відгуки та рейтинги інших користувачів.

Проект буде корисним для туристів, які шукають кращі місця для відвідування та отримання максимального задоволення від своєї подорожі, також він дозволить місцевим жителям дізнатись про нові місця та розширити свій кругозір щодо місць для відвідування у своєму місті.

Звісно існують безліч схожих застосунків, отож чи буде корисним цей проект? Так, цей веб-сайт із картою та зручно поданою інформацією буде дуже корисним, для людей. Кожен застосунок має свої особливості та переваги. Ця карта буде спеціалізованою на ресторанах та інших цікавих місцях, де можна смачно перекусити та насолодитись комфортним проведенням часу з друзями або родиною. Цей застосунок зможе допомогти зекономити багато часу на пошуках найкращих місць та плануванню маршруту своєї подорожі. Крім того, цей веб застосунок зможе допомогти відвідувачам, відшукати місцеві ресторани та інші

місця з унікальною кухнею та атмосферою, що б це не було, китайська, італійська, тощо, кухні.

Отже цей проект може бути корисним для багатьох людей, як для туристів, так і для місцевих жителів, особливо для тих, хто цінує гарну їжу, затишні апартаменти для проживання, та цікаві пам'ятки для відвідування.

Розділ 1. Завдання та вимоги для веб-застосунку.

На цьому сайті буде розміщено застосунок для зручності подорожуючих, для використання він не вимагатиме попередньої реєстрації, і тим самим дозволяє його використовувати всім бажаючим не створюючи перешкод. На ньому буде карта з відображенням різних закладів, які користувач може відфільтрувати. Також застосунок міститиме детальну інформацію яка допоможе туристу або місцевому жителю легше зрозуміти чи підходить то чи інше місце для того щоб смачно поїсти, переночувати з комфортом або розважитись на свіжому повітрі.

1.1 User Stories

- Користувач може знайти ресторани та кафе у своєму місті;
- Користувач може знайти найкращі готелі у своєму місті, куди;
- Користувач може знайти туристичні визначні місця свого місті;
- Користувач може бачити розташування ресторанів, готелів та цікавих місць на карті, для того щоб легше орієнтуватись на місцевості;
- Користувач може бачити фото закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити рейтинг закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити місце в топі, закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити кількість відгуків закладів або визначних місць на інформаційних картках про заклади/цікаві місця;

- Користувач може бачити фото закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити назви закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити цінову категорію закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити адресу закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити номер телефону закладів або визначних місць на інформаційних картках про заклади/цікаві місця;
- Користувач може бачити типи їжі які пропонує ресторан, інформаційних картках про заклади/цікаві місця;
- Користувач має змогу перейти на сайт закладу, для перегляду детальної інформації;
- Користувач має змогу перейти на сайт для додаткової інформації про відгуки;

Розділ 2. Проектування клієнтської частини.

Ця частина, відповідає за відображення графічного інтерфейсу, він працює разом з серверною частиною. Головні етапи проектування клієнтської частини:

1. Розуміння головних потреб та очікувань користувачів: до цього відноситься дослідження цільової аудиторії та потреби. Вимоги до функціональності, зручності використання та інтерфейсу повинні бути з'ясовані, щоб дизайн відповідав їх потребам.

2. Розробка концепції та інформаційної архітектури: після того як зібрано відповідну інформацію, було розроблено концепцію клієнтської частини, де включено структуру інформації та інформаційну архітектуру.

3. Створення прототип: досить важливим етапом є створення прототипу інтерфейсу, щоб перевірити його функціональність та зручність використання. За допомогою прототипу, можна провести тестування з користувачами, для того щоб дізнатись наскільки задовольняє дизайн їхні потреби.

4. Дизайн інтерфейсу: за результатами тестування прототипу, розроблено дизайн інтерфейсу, та включено графічний дизайн та розміщення елементів на сторінці.

5. Розробка функціональності: після розробки дизайну інтерфейсу, було проведено роботу над розробкою каркасу програми.

6. Тестування та підтримка: коли розроблено інтерфейс та функціональність, важливо проводити тестування на те чи все відповідає вимогам, та виправлення помилок, які можуть вплинути на користувачів. Та важливо

проводити тестування на швидкість роботи та завантаження сторінок, для того щоб переконатись, що сайт працює ефективно. Але після цього всього, важливо продовжувати вдосконалювати сайт з часом, враховуючи потреби користувачів, які в перспективі могли б скористатися цим сайтом, з урахуванням розвитку технологій.

2.1 Створення прототипу.

Прототип для головної сторінки:

- Карта, на якій можна відображати місця, ресторани, готелі тощо.

(Рис. 2.1 По праву частину потенційне місце розташування карти)

- Меню для пошуку та фільтрації цих місць за різними критеріями, такими як тип, ціна, відстань, рейтинг тощо.

(Рис. 2.1 По ліву частину, зверху, потенційне місце фільтрації для ресторані, готелів, та інших місць)

- Список рекомендованих місць з фотографіями, описом та рейтингом.

(Рис. 2.1 По праву частину, знизу, потенційне карток з місцями)

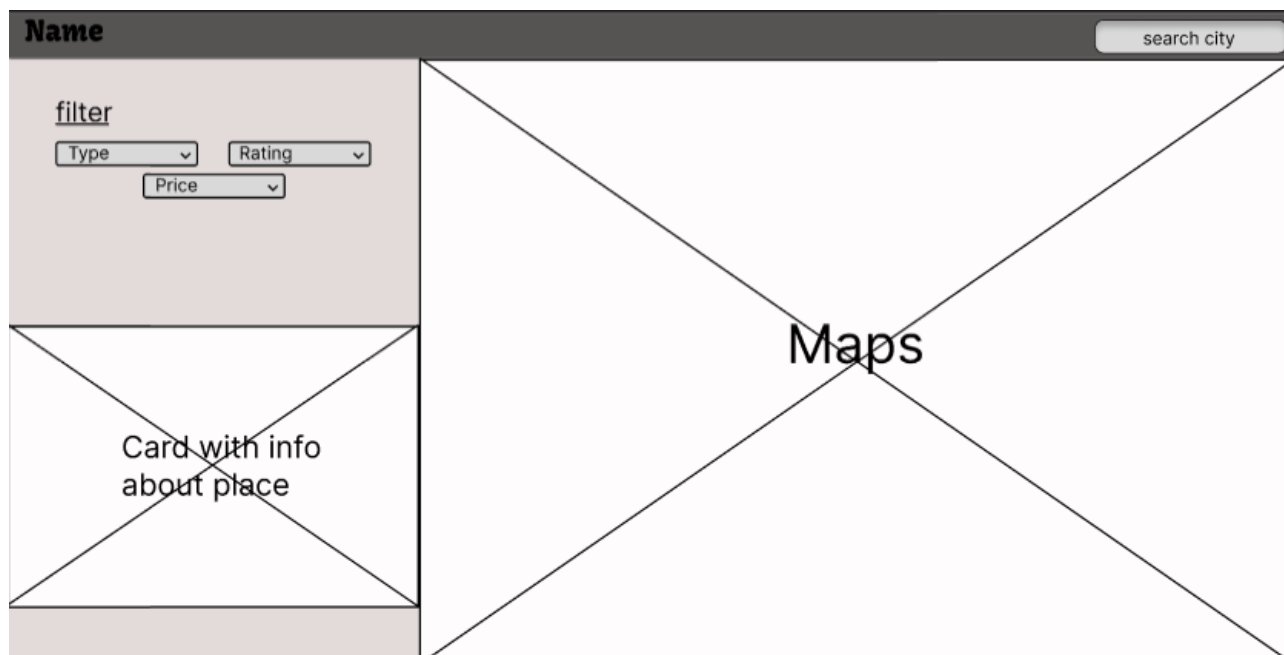


Рис. 2.1 Прототип головної сторінки.

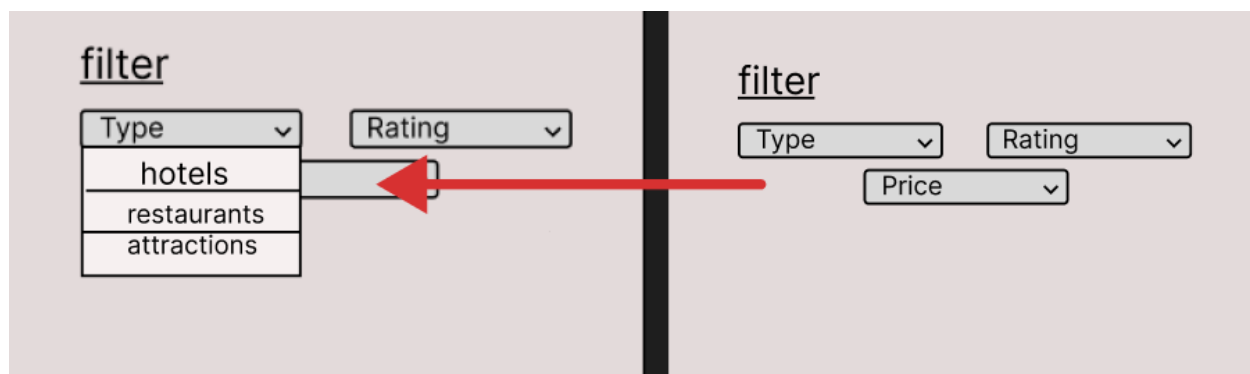


Рис. 2.2 Прототип для фільтрації.

На рис. 2.3 можна побачити, як виглядатиме розташування місць на карті, на них буде зображене фото так скільки зірочок в рейтингу має цей чи інший заклад.



Рис. 2.3 Деталізований прототип карти.

На рис. 2.3 буде знаходитись:

- фото закладу/місця;
- рейтинг;
- цінова категорія;
- адреса;
- телефон;
- додаткова інформація;
- посилання для детального ознайомлення;

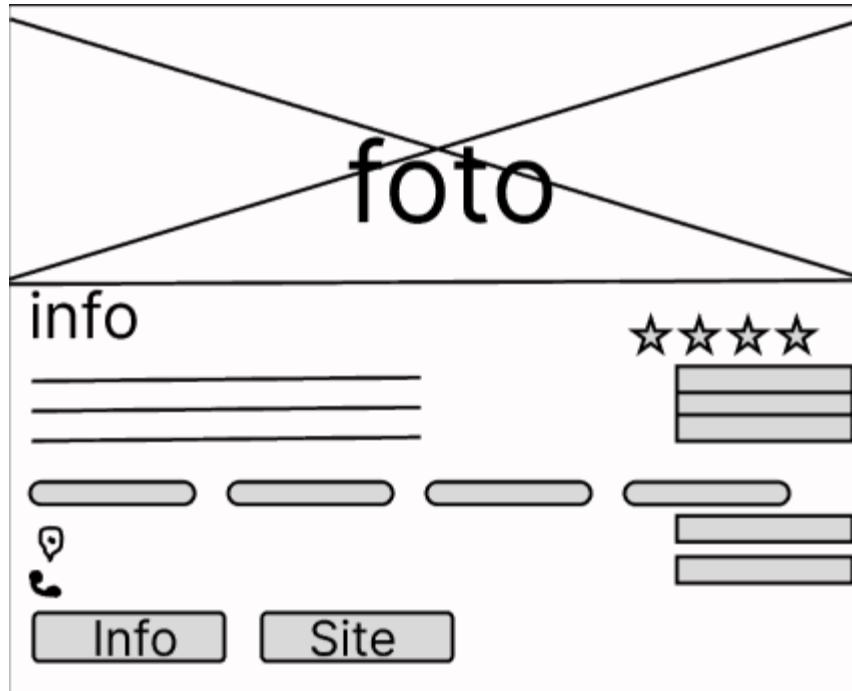


Рис. 2.4 Прототип картки для закладу/місця.

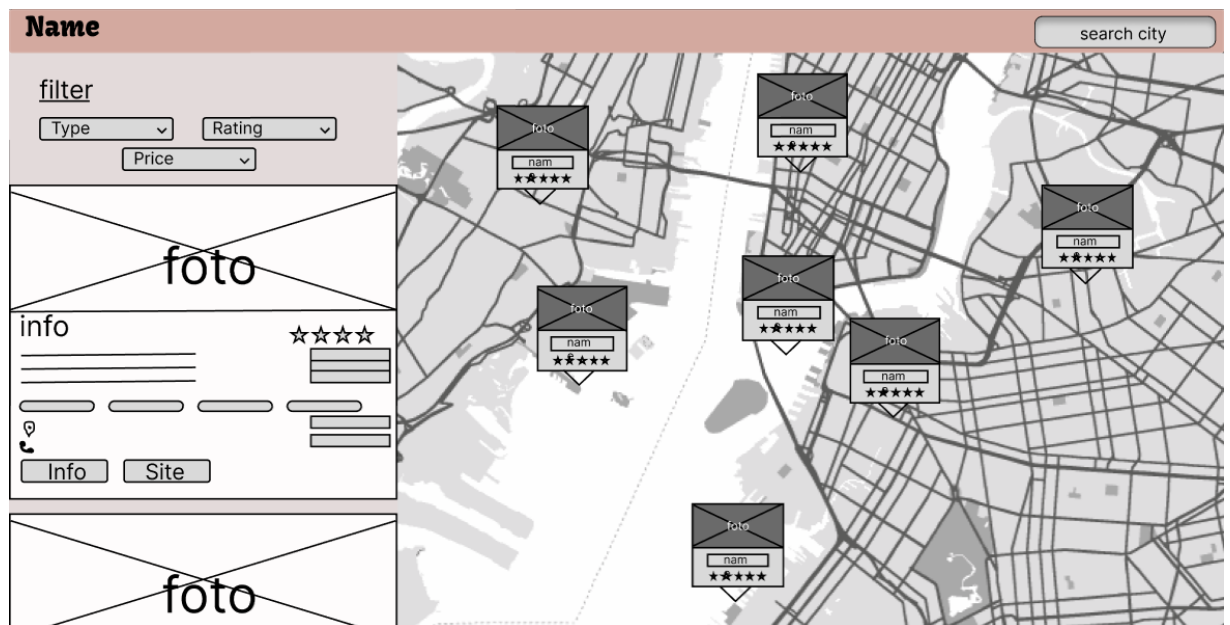


Рис. 2.5 Деталізований прототип.

Як вже було сказано, прототип є важливим кроком у створенні веб-сайту, так як він дає змогу отримати загальне уявлення про функціонал, макет та інтерфейс веб-сайту до його розробки.

Прототип дозволяє визначити потреби та очікування користувачів, виявити проблеми та допомогти їх вирішити ще на початкових етапах розробки проекту, та це зменшує ризики помилок та провалів під час наступних кроків розробки.

Без цього етапу може виникнути ситуація, коли певні функціональні можливості не покривають потреби користувача, і це може ускладнити просування сайту.

Розділ 3. Проектування серверної системи.

3.1 Середовище використане для створення веб-сайту - Visual Studio Code.

Visual Studio Code - є редактором вихідного коду, створений компанією Microsoft. Він є досить легкий та швидкий, в ньому можна знайти всі потрібні інструменти для роботи, а якщо чогось немає, то швидко скачати додаткові плагіни (розширення). Він є безкоштовним та розробляється як програмне забезпечення з відкритим вихідним кодом.

3.2 Вибір мови програмування.

Було обрано мову програмування JavaScript, тому що вона є дуже швидкою, так як запускається зразу у браузері користувача. Більшість підтримує Just In Time - компіляцію для JavaScript, тому не потрібно компілювати код перед тим як його запускати.

У теперішній час майже всі браузери підтримують цю мову програмування. Воно є безкоштовним і не потребує ніяких дій встановлення або налаштувань. Ця мова є дуже популярною для розробки клієнтської частини. JavaScript - є досить широкою для використання, а це означає, що програми на мові JavaScript можуть запускатись на різних типах обладнання, для яких вони були написані.

Варто додати, що ця мова може використовуватись і для Front-end частини, так і для Back-End частини. Для серверної частини можна використовувати розширену версію JavaScript - Node.js, вона має змогу взаємодіяти з файловими системами, базами даних та іншими ресурсами.

Щоб зрозуміти роботу обробки JavaScript код, надано схему його роботи на рис. 5.

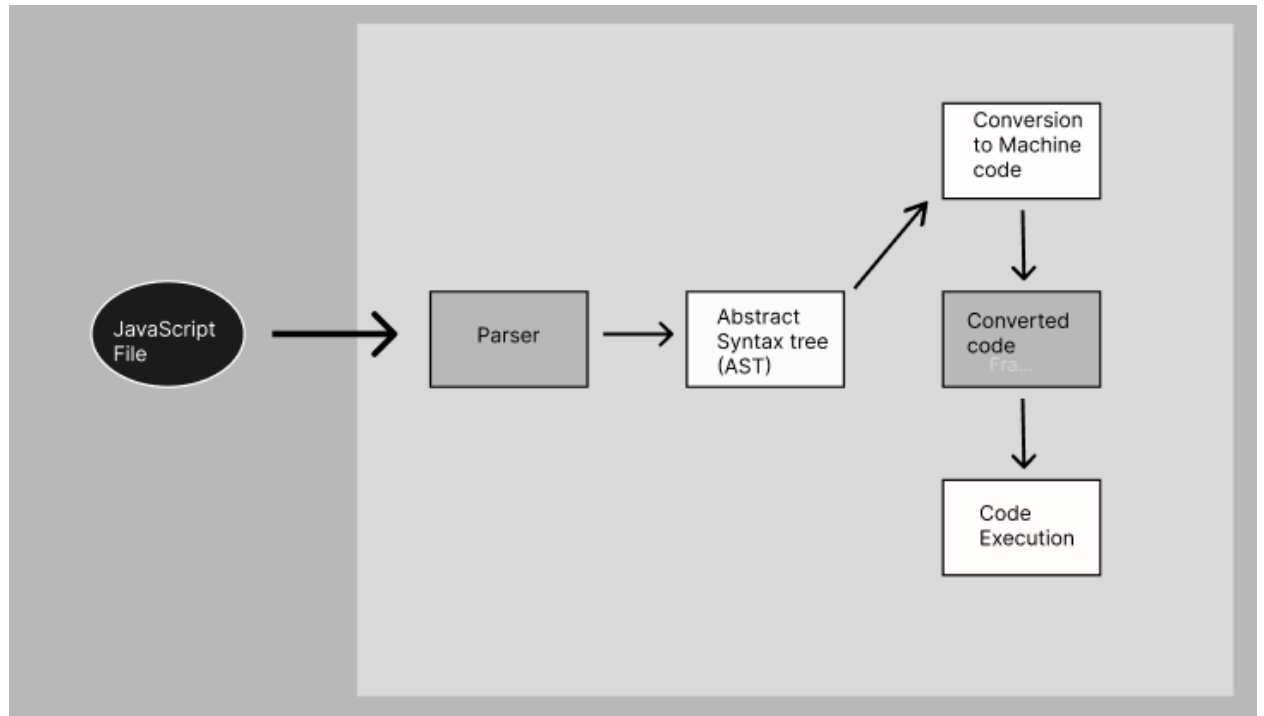


Рис. 3.5 Схема роботи для JavaScript.

3.3 Node.js - серверна частина.

Для Back-end частини використано Node.js - ця платформа є для створення серверних застосунків на JavaScript. Node.js дозволяє використовувати JS на клієнтській та серверній частині. Node.js є швидким та ефективним для розробки веб-сайтів з великим потоком даних. Використання цього фреймворку, дозволяє створювати високопродуктивні застосунки, з використанням лише однієї мови.

У Node.js є одна важлива перевага, це її асинхронність та дієво-орієнтований підхід. Можна переглянути, як це працює на рис.6.



Рис. 3.6 Схема зображення дієво-орієнтованої архітектури.

Подієво-орієнтована архітектура (англ. Event-Driven Architecture або EDA) - це певний тип архітектури програмного забезпечення, де частини програми взаємодіють один між одним через певні об'єкти (події).

3.4 Google Maps API

В загальному API -це Application Programming Interface. Google Maps API - це доступний набір інструментів, як дають змогу використати дані та функції від Google Maps, у власному проекті. За допомогою цього API можна інтегрувати карту та маршрути, також розраховувати дистанції та досить багато інших можливостей для створення власних додатків та веб-сайтів. Звісно, за використання цих послуг може стягуватись оплата, але є безкоштовний тариф на певну кількість запитів. У цьому проекті надано безкоштовні 500 запитів на місяць.

Загалом, Google Maps API є потужним інструментом, який допомагає створити великі функціональні та зручні додатки з використанням карти.

Переваги:

- Багато можливостей:

Це API тримає у собі велику кількість функцій та додаткових можливостей, які будуть корисні для створення картографічних додатків або веб-сайтів, це може бути як вище згадано, побудова маршрутів, геокодування (це давання певному місцю, універсальної географічної ідентифікації, це те саме як координати, широта, довгота) та для додатків з певним відстеженням місця знаходження.

- Надійність:

Так як це інструмент від Google, він має високий рівень захисту, це дозволяє створювати стабільні та безперервні застосунки.

- Легке підключення та інтеграція:

Google Maps API можна дуже просто інтегрувати до різних додатків, які використовуються на сайтах або мобільних пристроях.

- Постійна підтримка:

Так як, це Google, він містить в собі багато корисної документації та посилань на зрозумілі ресурси, які використовують Maps API.

Крім цього, під час розробки, можна отримати технічну підтримку від Google у разі виникнення будь-яких проблем з API.

- Безкоштовні можливості використання:

Це API може безкоштовно використовуватись для багатьох додатків, але з обмеженням по кількості запитів. Але це дозволяє економити кошти при створення власних додатків.

Не зважаючи на те, що у використанні цих інструментів є багато плюсів, мінуси там також присутні, зараз розберемо їх.

- Відкритість даних:

Загалом отримана інформація з Google Maps API може бути у відкритому доступі, це означає, наприклад інші можуть бачити адресу, або місцезнаходження, та деяку іншу особисту інформацію.

- Обмеження по запитах:

Там надається певна кількість безкоштовних запитів на місяць, після їх використання, можна оплатити додаткові запити. Але це може бути проблемою для масштабних проектів, де високі трафіки.

- Прив'язаність до стороннього сервісу:

При використанні Google Maps API, все залежить від стороннього сервісу, і якщо у виникне проблема з ним, то проект також може призупинитись або загалом не відповідати.

- Вартість:

Якщо знадобиться більше функціональних можливостей або потрібно підвищити рівень обслуговування, то потрібно буде придбати інший тариф, а це може бути досить дорого.

Для цього проекту, ці мінуси не сильно критичні, тому використано Google Maps API для того щоб підтягнути карту на сайт та знайти місця які потрібні. При цьому використовується безкоштовний тариф та документація, яку надає Google, для того щоб ознайомитись зі всім сервісами та можливостями.

Розділ 4. Створення веб-сайту

4.1 Setup.

Першим ділом було встановлено середовище розробки, як згадувалось у розділі вище, використовується Visual Studio Code та встановлюється Node.js.

4.2 Project Setup.

Після налаштувань середовища розробки, потрібно створити новий проект, для цього створено головну папку проекту. Було розгорнуто проект у Visual Studio Code та зроблено структуру папок. Цей проект можна розбити на дві папки, “src” - для всього вихідного коду та “public” - для публічних файлів, таких як зображення, та інші статичні файли, “components” - для розбиття компонентів React на менші частини та “styles” - для всього CSS коду.

Ці папки містять у собі внутрішні під папки, які відповідають за різні функціональні частини проекту. Тому у папці “components” - створено додаткові папки щоб розділити все по функціях:

- Header
- List
- Map
- PlaceDetails

Також створено папку “api” - де зберігаються всі дані, які пов’язані з підключенням та викликом запитів по API.

Така структура файлів, розділених по папках, дозволить підтримувати організований проект та допоможе легше орієнтуватись, редагувати та розширювати проект у майбутньому.

4.3 Створення Header частини.

Компонент “Header” використовуємо для того щоб відображати верхню панель навігації. Для його створення ми використали:

- “AppBar”: це елемент, використаний для фіксації верхньої панелі.
- “ToolBar”: це контейнер, він відповідає за розміщення елементів внутрішньої структури панелі.
- “Typography”: воно відповідає за відображення текстового контенту на сторінці, в цьому проекті використано його, для відображення назви додатку та заголовку.
- “Box”: цей елемент, використаний для групування елементів та розміщення їх відповідний контейнер.
- “SearchIcon”: це є елементом бібліотеки ‘@material-ui/icons’, який відображає пошукову іконку.
- “InputBase”: це є елементом бібліотеки ‘@material-ui/core’, використовується для введення адреси.

4.4 Створення List частини.

List приймає різні поля такі як список місць - “places”, тип місця - “type”, рейтинг місць - “rating”, стан вибраного дочірнього елемента та стан завантаження. Для стилізації компонента використовується хук “useStyle”, який повертає об'єкт стилів.

Компонент має заголовок “Find your favorite place around you” і складається з різних фільтрів за типом місця, рейтингом, та списком місць. Кожен елемент з цього списку містить детальну інформацію про конкретне місце, і цю інформацію відображає компонент “PlaceDetails”.

Реалізацію цього файлу, додано у додатках.

4.5 Створення Мар частини.

Цей файл відповідає за відображення карти та шпильок на ній. Він використовує бібліотеку “google-map-react” для роботи з Google Maps.

Звісно на початку імпортовано усі потрібні бібліотеки та компоненти:

- React;
- useState;
- GoogleMapReact;
- Paper;
- Typography;
- useMediaQuery;
- LocationOnOutlinedIcon;
- Rating;
- mapStyles;
- useStyles;

Компонент Мар приймає на вхід різну інформацію, таку як широта, довгота, місце, і так далше. У цьому файлі є частина коду, яка перебирає масив місць і для кожного елемента створює свій блоковий елемент `<div>`, який виступає як контейнер для мітки на карті, в ньому можна визначити вміст цієї мітки/місця, наприклад використовуючи компоненти з бібліотеки Material-UI.

Також, використовуємо такий компонент як, `<Paper>` - це компонент з бібліотеки Material-UI. Цей компонент надає естетичний вигляд блокам з текстом, зображеннями або іншими елементами, які потрібно виділити на сторінці. Він може

бути використаний для розміщення інформації, карток, панелей або будь-яких інших блоків, які потребують візуального розмежування від інших елементів сторінки. Тому, у цьому проекті він містить назву місця, зображення та рейтинг на карті. На карті воно зображено у вигляді невеликого блоку з інформацією.

Під час створення кожної позначки місця, визначаються його координати “lat” та “lng”, взятих на основі `place.latitude` та `place.longitude` з даних про місце.

Якщо коротко, то воно динамічно створює мітки з місцями на карті, що дозволяє відображати мітки для кожного місця наданих координат.

4.6 Створення PlaceDetails частини.

Основним завданням, у цьому файлі є відображення інформації про місце, таку як назва, рейтинг, фото, адреса, телефон тощо.

Основними кроками там є:

- Імпорт потрібних компонентів з бібліотеки Material-UI;
- Відображення карти, де вказано фото, назву місця, рейтинг, ціновий діапазон, тип кухні (якщо це місце відноситься до ресторанів), адреса та телефон місцевості (якщо така інформація присутня);
- Кнопки, які дають змогу переходити на веб-сайт місця або допоміжного сервісу від Rapid.API;

Цей компонент є використаний у батьківському компоненті, такому, як “List”, для того щоб відображати детальну інформацію про кожне місце зі списку. Він приймає вхідні дані про місця, та додаткову інформацію, для кожного елемента

списку. Коротко кажучи він використаний для створення картки, у якій відображено всі вказані деталі про місце.

Ці блоки з інформацією можна прокручувати, це забезпечує зручну навігацію для користувача. В цілому, цей компонент відповідає за візуальне представлення та надає користувачу достатньо зручну інтерактивну можливість дізнатись більше інформації про певне місце. Завдяки тому компоненту, можна побачити фото місця, назву, рейтинг, приблизний діапазон цін, рейтинг цього місця, тип кухні, адресу та телефон.

У загальному, компонент “PlaceDetails” створений, щоб забезпечувати зручну інформаційну платформу для користувача, і це допомагає зробити людині правильний вибір щодо конкретного місця, яке варто відвідати.

4.7 Як відбувається підключення Google Maps API.

Щоб використовувати сервіси Google, було створено проект на Google Cloud Platform. Після чого отримано ключ для API, що дозволяє використовувати карту та інші можливості від Google. Далі, було налаштовано проект так, що він використовує наданий ключ у файлі “Map”. Це дозволяє зберігати дані, виконувати серверну логіку та використовувати інші корисні сервіси цієї платформи.

Загалом, використання цієї платформи дозволяє використовувати Google Maps API для відображення карти та міток з місцями і їх відомостями. Це відкриває потужні функції від Google Maps, такі як пошук місць, отримання координат тощо.

Для легшого отримання даних було використано додатковий інструмент Rapid API, для того щоб отримати дані про ресторани, готелі та різні пам’ятки. Цей

інструмент як об'єднуючий міст між програмою та можливостями Google, адже він набагато більше полегшує нам роботу.

Основними кроками було:

- Реєстрація та отримання ключа API: відбулась реєстрація на платформі Rapid API та отримано ключ, який використано для автентифікації запитів;
- Після чого було встановлено потрібні пакети для Node.js, "axios" - щоб здійснювати HTTP-запитів;
- Створено запити для Rapid API: з допомогою пакетів "axios" створено HTTP - запити з використанням API ключа, який надано на Google Cloud Platform та відповідної URL-адреси. В цих запитах можна включати параметри пошуку, фільтрацію або сортування даних;
- Обробка даних, які вдалось отримати: витягнуто необхідні поля для цього проекту;
- Відображення даних на веб-сайті: тепер відображено списки ресторанів, готелів або інших цікавих місць, також показано деталізовану інформацію про кожне з цих місць;

Розділ 5. Приклади реалізації.

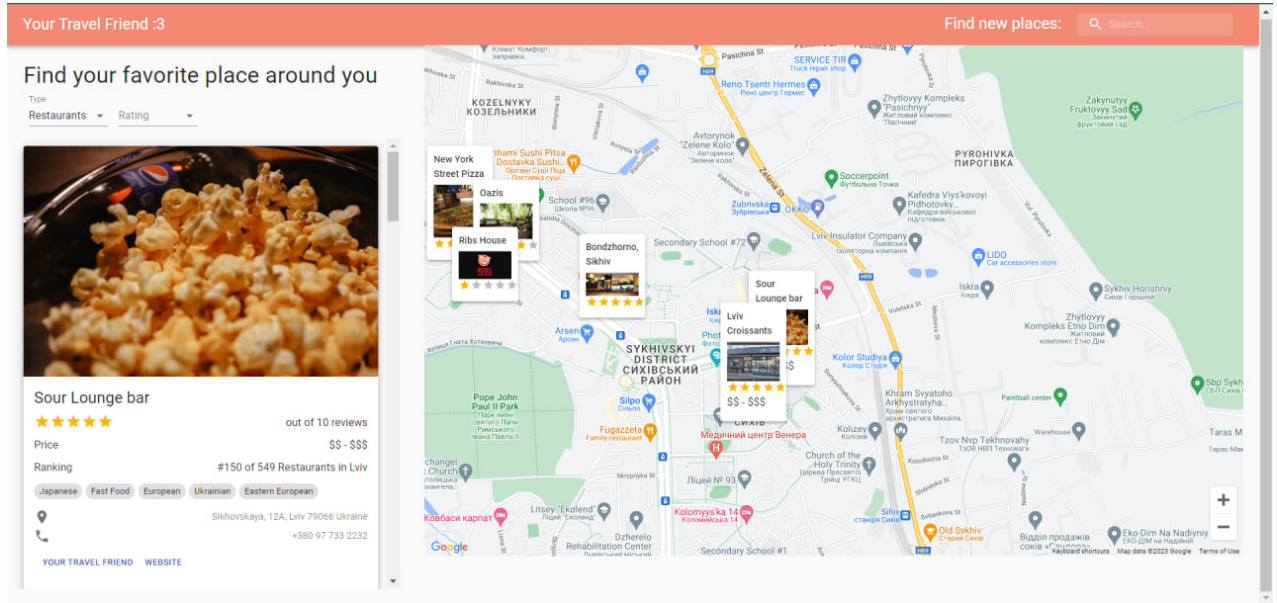


Рис. 5.1 Головний екран, після завантаження веб-сайту. За допомогою датчиків місця знаходження, на карті відображається приблизна область місця знаходження.

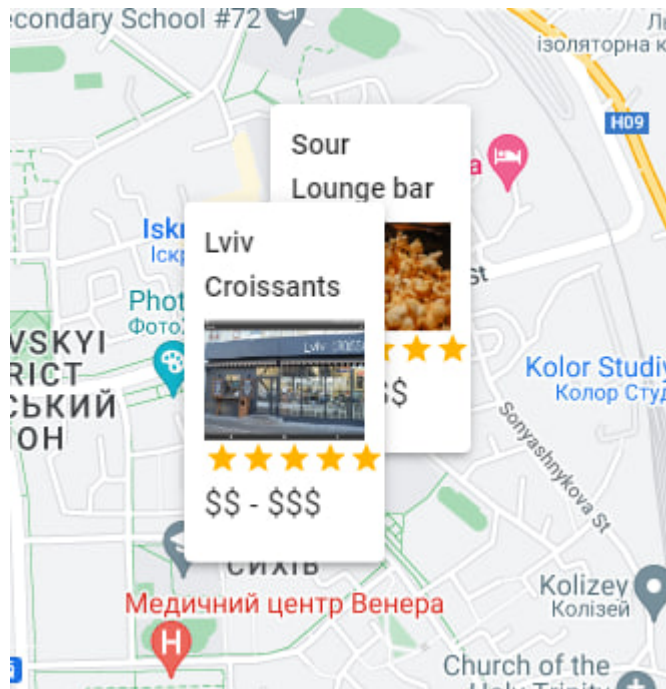


Рис. 5.2 Відображення міток з ресторанами (в цьому випадку це можуть бути і готелі, і різні цікаві місця такі як парки, ратуші, музеї)

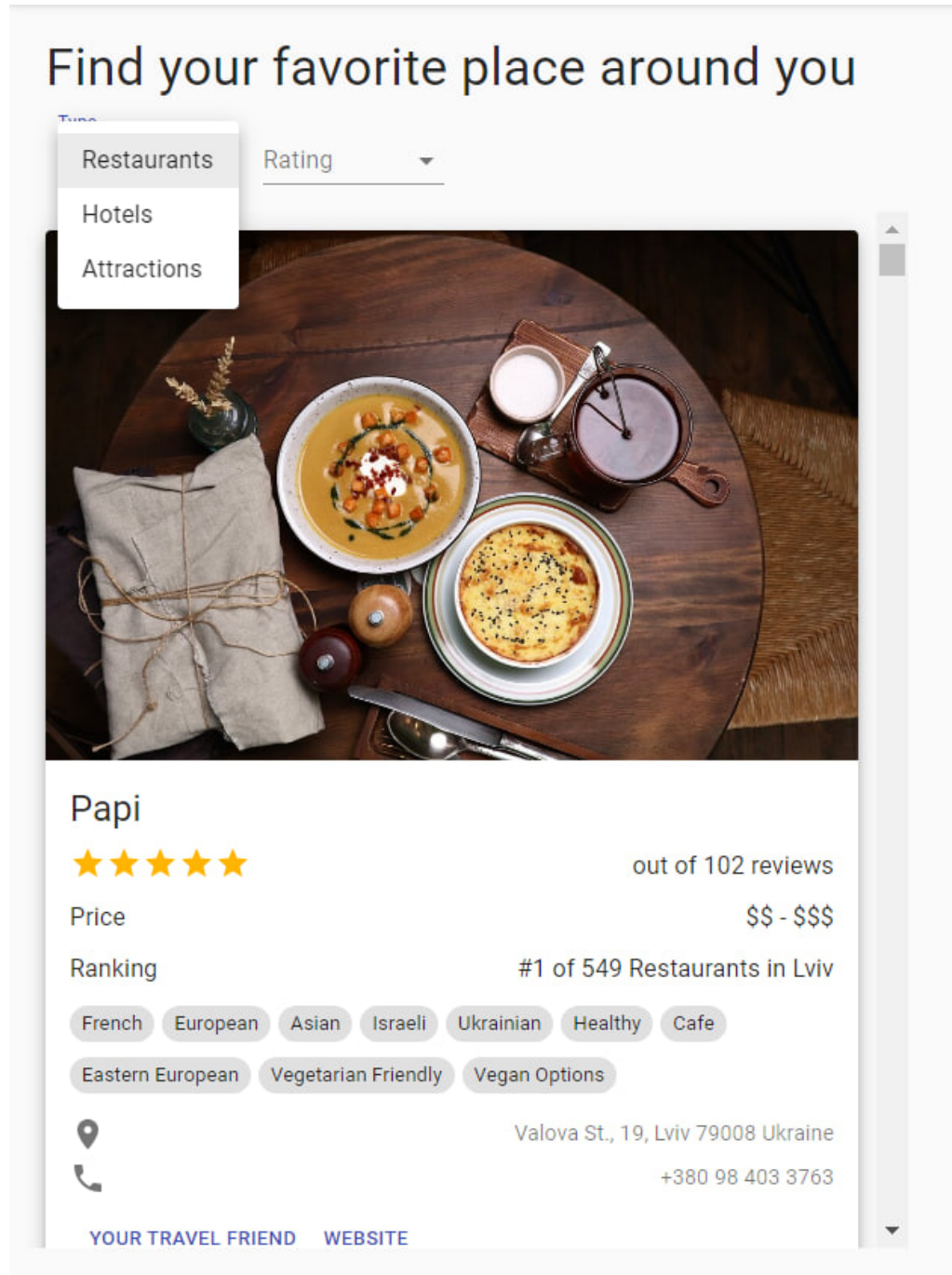


Рис. 5.3 Спадне меню для фільтрації по типу місця.

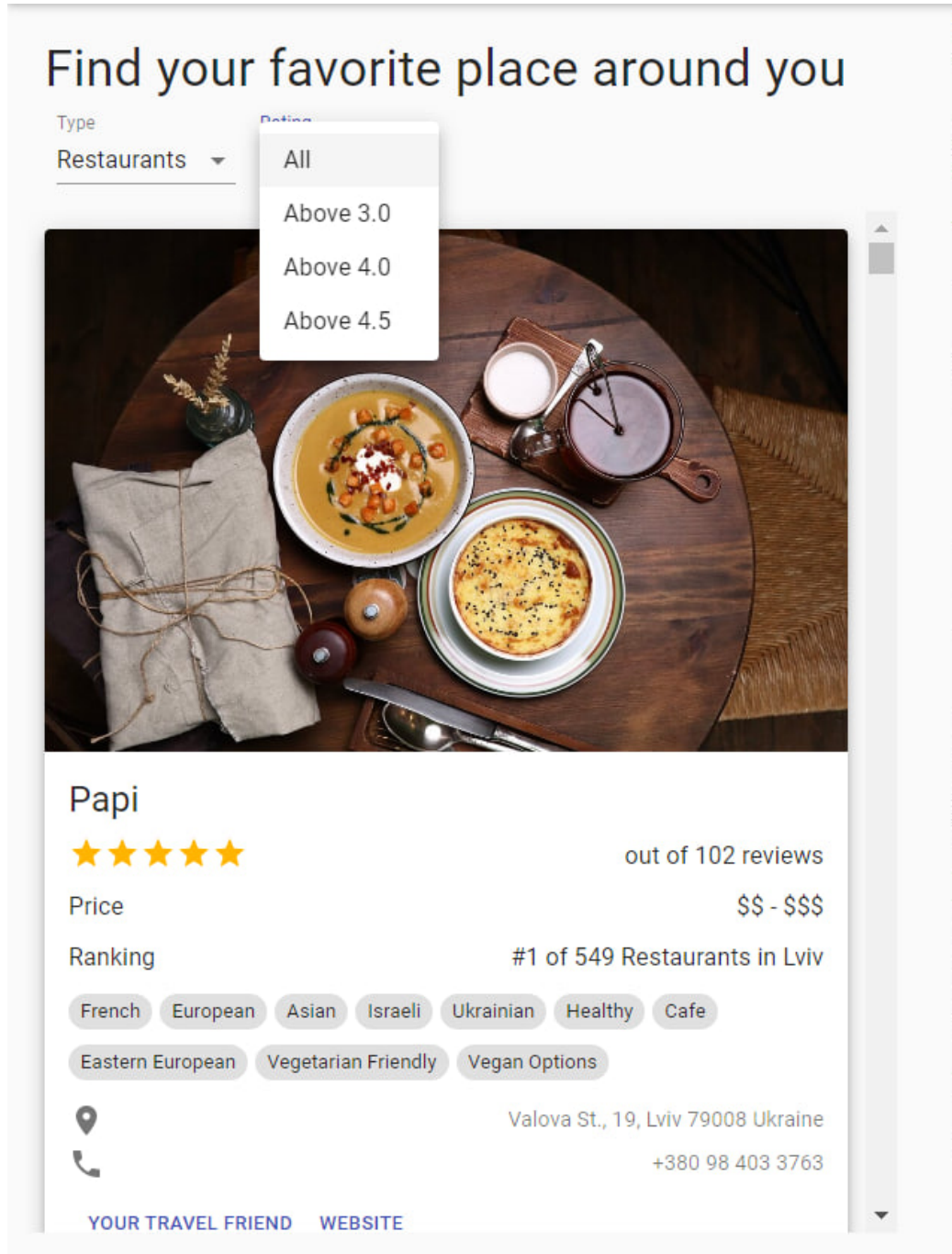


Рис. 5.4 Спадне меню для фільтрації по рейтингу місць.

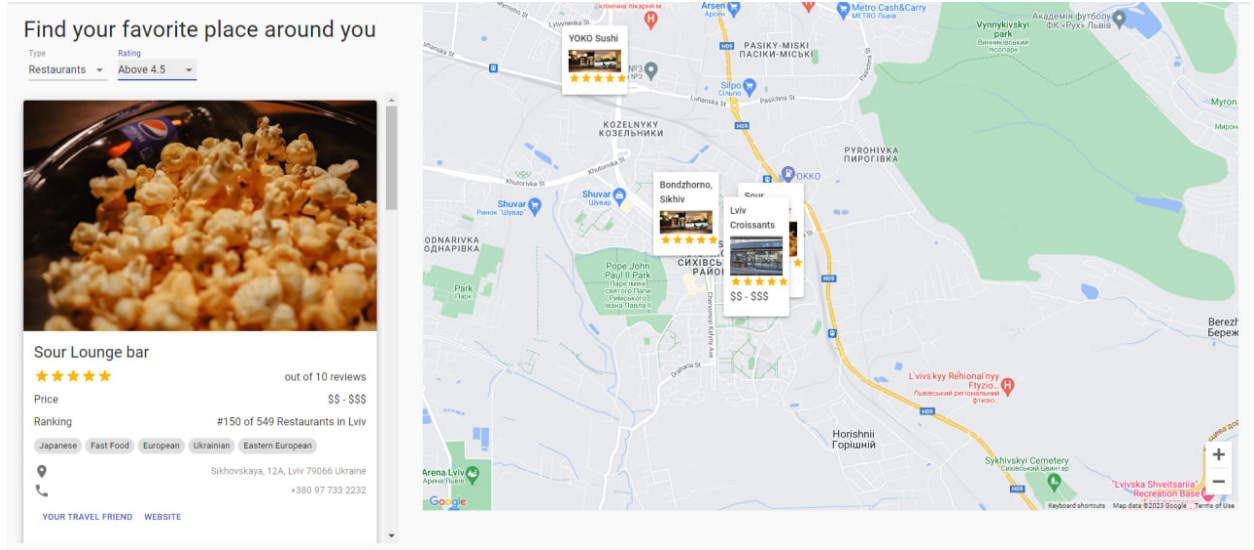


Рис. 5.5 Відфільтровані місця, по типу “Ресторани” та за рейтингом “Вище 4.5”.

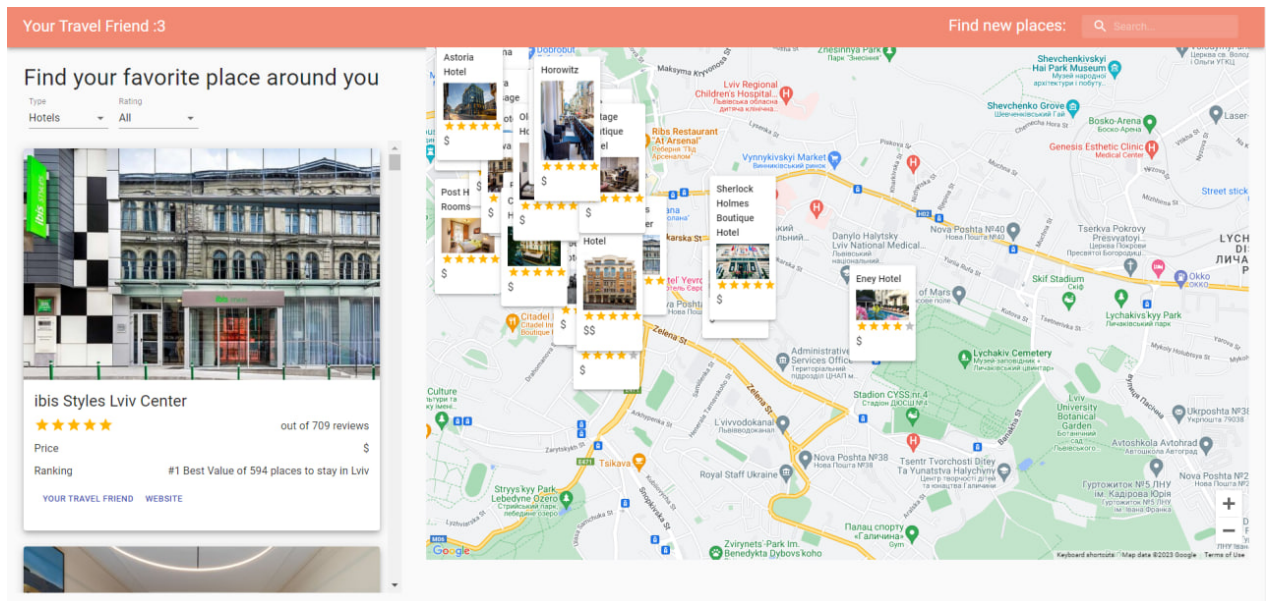


Рис. 5.6 Фільтрація по типу “Готелі” та рейтингом “Усі”.

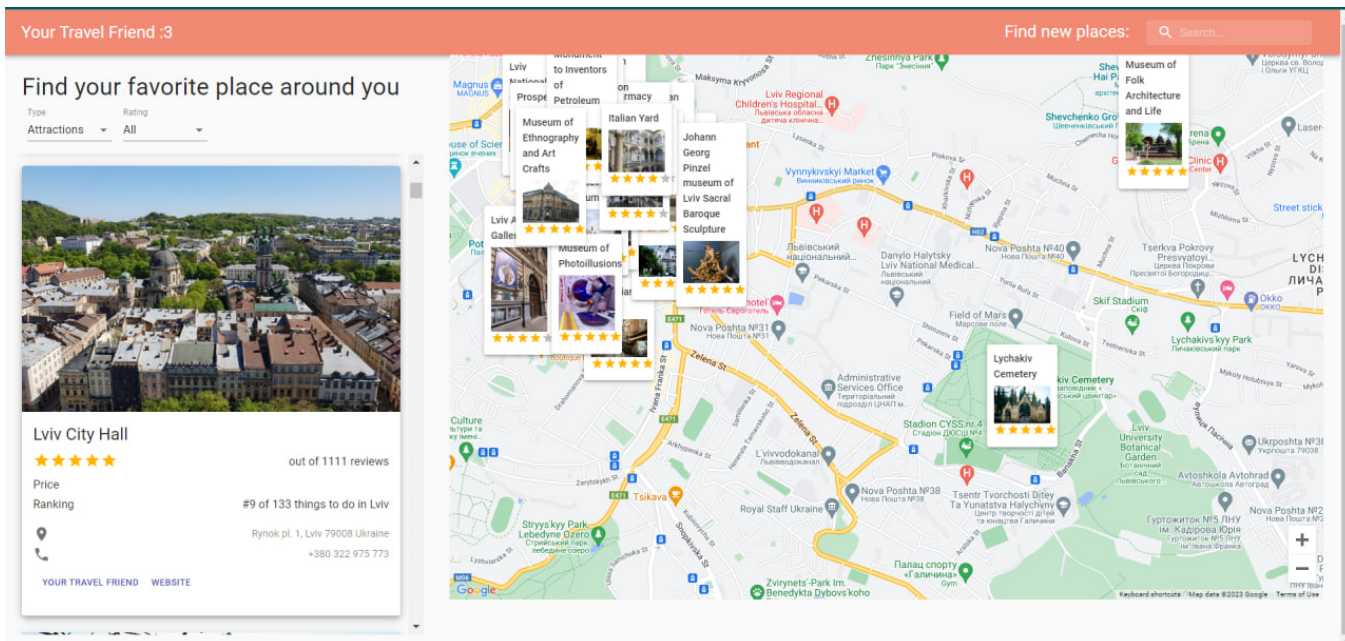


Рис. 5.7 Фільтрація по типу “Розваги” (Пам’ятні місця) та рейтингом “Усі”.

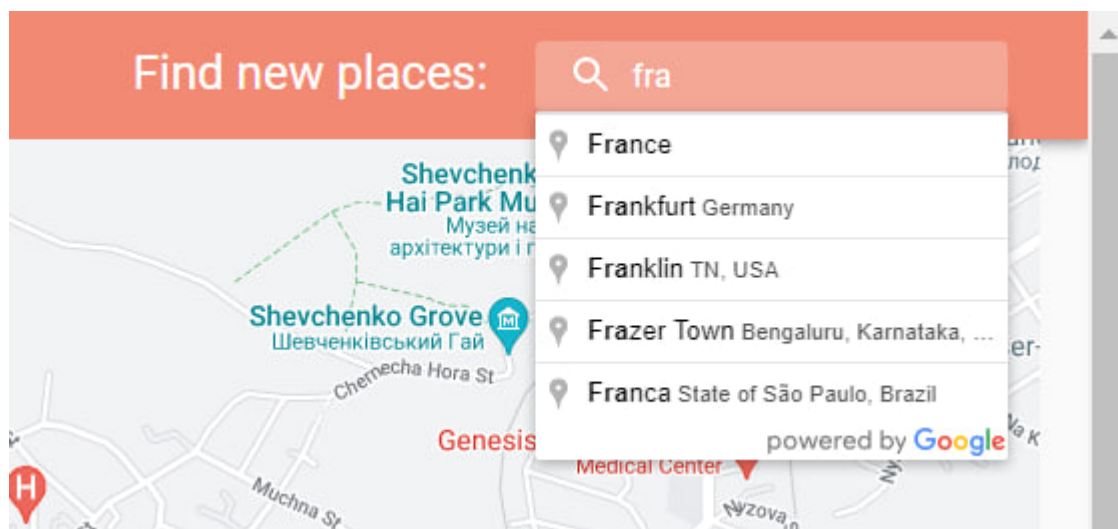


Рис. 5.8 Пошукова стрічка, для пошуку міста/країни. Вводиться в пошуку потрібне місто/країну, куди переносить та відображає потрібні місця.

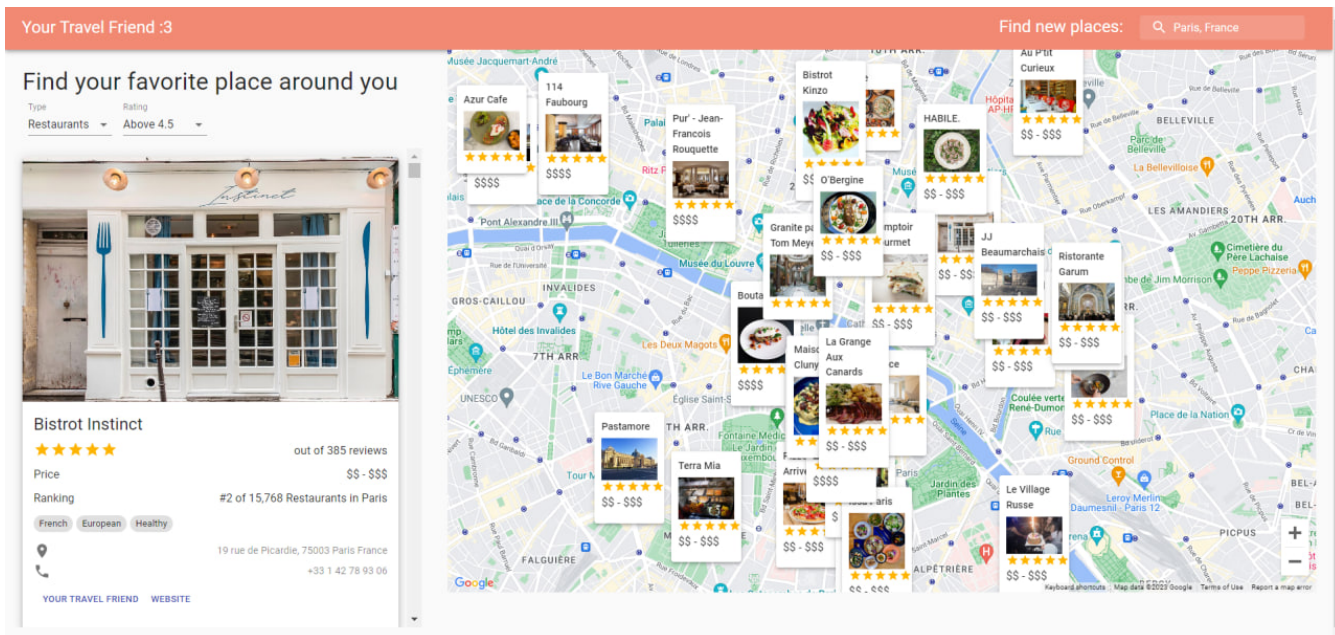


Рис. 5.9 У пошуку обрано “Париж, Франція”, з таким фільтром по типу “Ресторани” та рейтингом “Вище 4.5”

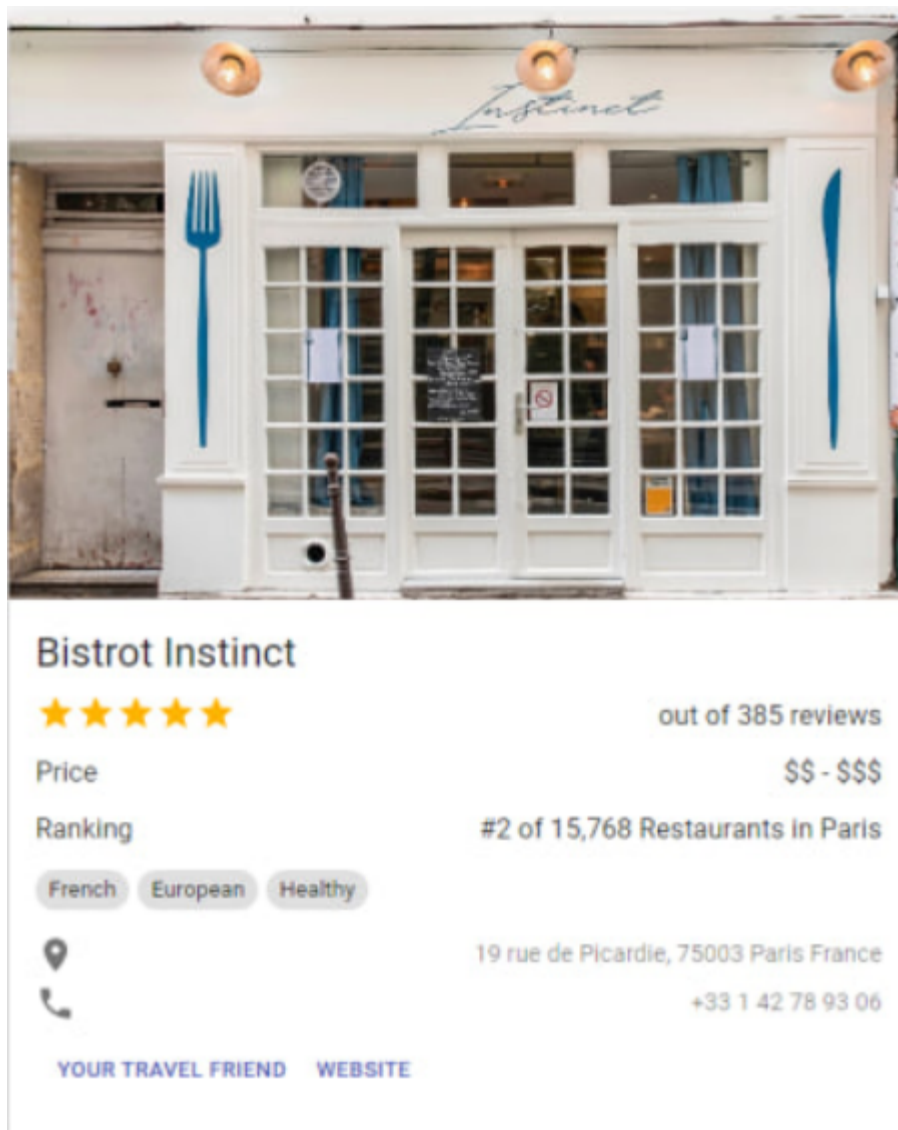


Рис. 5.10 Картка місця.

В цьому випадку це карта ресторану, на якій вказана детальна інформація, про те яку кухню вони готують, “Французька”, “Європейська”, “Здорова”, таке вказується лише для ресторанів, також тут видно різну інформацію, про те який рейтинг серед ресторанів він має (#2 з 15768 ресторанів) та загальна інформація про ціни, адреса, телефон, посилання на веб-сайт, та додатковий інформаційний веб-сайт.

Висновок

У процесі виконання цієї дипломної роботи було успішно реалізовано, функціональність за допомогою якої можна бачити різні місця, такі як ресторани, готелі та інші цікаві місця поблизу та у інших містах, країнах. Також реалізовано можливість фільтрації для цих місць за рейтингом та типом. Використовуючи такі технології як, Node.js, React та Material-UI, було створено зручний та естетичний інтерфейс користувача.

Найбільша увага була приділена інтеграції Google Maps API, що дало змогу відображати місцезнаходження потрібних місць. Крім того, використання Rapid API дало змогу отримати розширені дані про місця та допомогли збудувати функціонал пошуку та фільтрації.

Цей проект є вдалим з точки зору використання сучасних технологій, розробки функціонального та зручного інтерфейсу та інтеграції з відомими сервісами. У результаті цієї роботи є інтерактивний веб-сайт, який надає для користувача зручно подані та актуальні дані про місця, та допомагає полегшити процес пошуку.

Ця робота не тільки дозволила вдосконалити навички розробки, але й показала значимість використання API для отримання зовнішніх даних та підвищення функціональності додатків.

Список використаних джерел

1. Google Maps Platform [Електронний ресурс]. - Режим доступу: <https://developers.google.com/maps>
2. How to load Maps JavaScript API in React [Електронний ресурс]. - Режим доступу: https://www.youtube.com/watch?v=9e-5QHpadl0&ab_channel=GoogleMapsPlatform
3. Node.js [Електронний ресурс]. - Режим доступу: <https://nodejs.org/en/docs>
4. Google Cloud APIs [Електронний ресурс]. - Режим доступу: <https://cloud.google.com/apis/docs/overview>
5. JavaScript documentation [Електронний ресурс]. - Режим доступу: <https://devdocs.io/javascript/>
6. Axios documentation [Електронний ресурс]. - Режим доступу: <https://axios-http.com/docs/intro>
7. Express.js documentation [Електронний ресурс]. - Режим доступу: <https://expressjs.com/>
8. Google Maps API [Електронний ресурс]. - Режим доступу: <https://developers.google.com/maps/documentation/javascript>
9. Figma documentation for creating design [Електронний ресурс]. - Режим доступу: <https://www.figma.com/best-practices/guide-to-developer-handoff/components-styles-and-documentation/>