

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет ракетно-космічної техніки

Кафедра геоінформаційних технологій та космічного моніторингу Землі

**Пояснювальна записка**  
**до дипломного проекту (роботи)**  
(тип кваліфікаційної роботи)

магістр

(освітній ступінь)

на тему «Використання геоінформаційних технологій для розробки веб-сервісу  
вирішення задач геомаркетингу на прикладі міста Тернопіль»

ХАІ.407.462м.24О193.1904082ПЗ

Виконав: студент(ка) 2 курсу групи № 462м

Спеціальність 193 Геодезія та землеустрій

(код та найменування)

Освітня програма Геоінформаційні системи та  
технології

(найменування)

Сергеєв А.С.

(прізвище та ініціали студента (ки))

Керівник: Андрєєв С.М.

(прізвище та ініціали)

Рецензент: Полупан А.В.

(прізвище та ініціали)

Харків – 2024

**Міністерство освіти і науки України**  
**Національний аерокосмічний університет ім. М. Є. Жуковського**  
**«Харківський авіаційний інститут»**

Факультет ракетно – космічної техніки

Кафедра геоінформаційних технологій та космічного моніторингу Землі

Рівень вищої освіти магістр

Спеціальність 193 «Геодезія та землеустрій»  
(код та найменування)

Освітня програма Геоінформаційних систем та технологій  
(найменування)

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри**

Станіслав ГОРЕЛИК  
(підпис) (ініціали та прізвище)  
 «23» ЖОВТНЯ 2023 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Тема випускної роботи «Використання геоінформаційних технологій для розробки веб-сервісу вирішення задач геомаркетингу на прикладі міста Тернопіль»

керівник кваліфікаційної роботи Андреев, Сергій Михайлович, к.т.н, доц.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом Університету №2001-уч від «15» листопада 2023 року

2. Термін подання студентом кваліфікаційної роботи 11.01.2024

3. Вихідні дані до роботи

1) Векторні шари розташування закладів, об'єктів та пам'яток культури, інфраструктури та укріттів м.Тернопіль,

2) Геоінформаційне програмне забезпечення (QGIS, Blender(плагін Blender GIS),SpatialiteGui)

3) Мови програмування, фреймворкі та IDE(python, JavaScript(+HTML, CSS), SQL, VsCode, Django

4. Зміст пояснювальної записки (перелік завдань, які потрібно розв'язати)

Аналіз використання ГІС технологій для задач геомаркетингу туристичного спрямування. Аналіз туристичного потенціалу міста Тернопіль. Аналіз використання ГІС технологій для задач геомаркетингу туристичного спрямування. Практично реалізація розробки власного веб сервісу для вирішення задач геомаркетингу на прикладі міста Тернопіль.

Висновки.

5. Перелік графічного матеріалу: Структурна схема методів сегментації ринку для сфери туризму. Схема основних етапів роботи. Структурна схема основних інструментів геомаркетингу. Структурна схема отримання даних з сервісу

mygeodata. Архітектура розробленого веб-сервісу. Алгоритм взаємодії користувача з інтерфейсом. Інтерфейс користувача. Дорожня карта проєкту. Структурна схема обробки даних. Картографічна модель м. Тернопіль. Архітектура бази даних. Архітектура бази даних користувачів. Структурна схема розробки власного веб-сервісу.

#### 6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Андрєєв С.М.	23.10.2023	11.01.2024
	<i>Доцент</i>		

Нормоконтроль Красовська І.Г. « 17 » 01 2024 р.

7. Дата видачі завдання 23.10.2023

#### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз використання ГІС технологій для задач геомаркетингу	23.10.23- 29.10.23	
2	Аналіз туристичного потенціалу міста Тернопіль	29.10.23-05.11.23	
3	Аналіз використання ГІС технологій для задач Геомаркетингу	05.11.23-20.11.23	
4	Аналіз наявних методик для створення картографічних веб-сервісів	20.11.23-01.12.23	
5	Практична реалізація розробки власного веб-сервісу для вирішення задач геомаркетингу на прикладі міста Тернопіль	01.12-28.12.2023	
6	Написання пояснювальної записки	05.01.24 - 11.01.24	

Студент

\_\_\_\_\_

(підпис)

Антон СЕРГЕСЬВ

(ініціали та прізвище)

Керівник дипломної  
(кваліфікаційної) роботи

\_\_\_\_\_

(підпис)

Сергій Андрєєв

(ініціали та прізвище)

## РЕФЕРАТ

Дипломна магістерська робота містить: 94 сторінки, 55 рисунків, 5 таблиці, 20 посилань.

Об'єкт дослідження: Розробка веб-сервісу та картографічних матеріалів

Предмет дослідження: Вирішення задач геомаркетингу шляхом застосування ГІС технологій для створення веб сервісів

Мета роботи: підвищення ефективності розробки веб-сервісів для вирішення задач геомаркетингу за рахунок використання геоінформаційних технологій та геоданих з відкритих джерел, а також інформації про окремі об'єкти.

Методи дослідження: завантаження векторних та обробка векторних для створення картографічних моделей та баз даних, для подальшої розробки веб-сервісу використовуючи мови програмування JavaScript та Python.

В результаті роботи було отримано картографічну модель міста Тернопіль на основі векторних даних, а також розроблено власний картографічний веб-сервіс для вирішення задач геомаркетингу на їх основі.

Ключові слова: Геоінформаційні технології, розробка, 3D, Web, Геомаркетинг, GeoJSON, OSM, Leaflet, Гіс, веб-сервіси

## ABSRTACT

The master's thesis contains: 94 pages, 55 figures, 5 tables, 20 references.

Object of research: Development of a web service and mapping materials

Subject of research: Solving geomarketing problems by applying GIS technologies to create web services

Purpose: to increase the efficiency of web service development for solving geomarketing problems by using geoinformational technologies and geodata from open sources, as well as information about individual objects.

Research methods: downloading vector data and processing vector data to create cartographic models and databases, for further development of a web service using JavaScript and Python programming languages.

As a result of the work, a cartographic model of the city of Ternopil based on vector data was obtained, and an own cartographic web service was developed to solve geomarketing problems based on them.

Keywords: GEOINFORMATION TECHNOLOGIES, DEVELOPMENT, 3D, WEB, GEOMARKETING, GEOJSON, OSM, LEAFLET, GIS, WEB SERVICES

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ВИКОРИСТАННЯ ГІС ТЕХНОЛОГІЙ ДЛЯ ЗАДАЧ ГЕОМАРКЕТИНГУ ТУРИСТИЧНОГО СПРЯМУВАННЯ.....	8
1.1 Роль туристичної галузі в зміцненні економіки розвинутих країн світу.....	8
1.2 Геомаркетинговий підхід з використанням картографічного моделювання туристичного попиту.....	10
1.3 Постановка мети та задачі кваліфікаційної роботи.....	12
РОЗДІЛ 2 АНАЛІЗ ТУРИСТИЧНОГО ПОТЕНЦІАЛУ МІСТА ТЕРНОПІЛЬ.....	13
РОЗДІЛ 3 РОЗРОБКА НАУКОВО МЕТОДИЧНИХ ЗАСАД ДЛЯ РОЗРОБКИ ВЕБ СЕРВІСУ З ВИКОРИСТАННЯМ ГЕОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ГЕОМАРКЕТИНГУ В СФЕРІ ТУРИЗМУ.....	17
3.1 Інструменти геомаркетингу.....	17
3.2 Аналіз існуючих веб-сервісів.....	18
3.3 Аналіз платформ(джерела картографічних даних) та мов програмування.....	20
3.4 Аналіз вхідних даних та баз даних.....	23
3.5 Архітектура сервісу та інтерфейс користувача.....	27
3.6 Підсумки та потенціал для розвитку.....	29
РОЗДІЛ 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБКИ ВЛАСНОГО ВЕБ СЕРВІСУ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ГЕОМАРКЕТИНГУ НА ПРИКЛАДІ МІСТА ТЕРНОПІЛЬ.....	32
4.1 Обробка вхідних даних.....	32
4.2 Розробка баз даних.....	38
4.3 Програмування веб сервісу.....	41
ВИСНОВКИ.....	50
ПЕРЕЛІК ПОСИЛАНЬ.....	51
ДОДАТОК А Лістинг коду.....	53
ДОДАТОК Б Плакат на тему «Використання геоінформаційних технологій для розробки веб-сервісу вирішення задач геомаркетингу на прикладі міста Тернопіль».....	93
ДОДАТОК В Презентація на тему «Використання геоінформаційних технологій для розробки веб-сервісу вирішення задач геомаркетингу на прикладі міста Тернопіль».....	94

## ВСТУП

На даному етапі розвитку інформаційних систем, технології яких прогресують з кожним днем, одним із важливих аспектів, є розуміння взаємодії між географічним простором та стратегіями маркетингу, а також їх впливу на різні соціальні сфери. З цього можна визначити, що геоінформаційні технології мають великий вплив на повсякдення звичайних користувачів. Їх активне використання призвело до створення різноманітних тематичних рішень для усіх можливих галузей з використанням актуальних геопросторових даних. В даному проєкті розглядається використання геоінформаційних технологій при розробці веб-сервісу для вирішення задач геомаркетингу в сфері туризму. На даний момент спроможності подібних сервісів та додатків постійно зростають, однак вони в цілому не мають чіткого спрямування на дану сферу і скоріше вирішують задачі звичайної навігації користувачів по конкретно заданим точкам А та Б, тому розроблений веб-сервіс має охоплювати більш спеціалізовані запити на основі інструментарію, що надає геомаркетинг.

Головною метою даного проєкту є підвищення ефективності розробки веб-сервісів для вирішення задач геомаркетингу за рахунок використання геоінформаційних технологій та геоданих з відкритих джерел, а також інформації про окремі об'єкти.

## РОЗДІЛ 1 АНАЛІЗ ВИКОРИСТАННЯ ГІС ТЕХНОЛОГІЙ ДЛЯ ЗАДАЧ ГЕОМАРКЕТИНГУ ТУРИСТИЧНОГО СПРЯМУВАННЯ

### 1.1 Роль туристичної галузі в зміцненні економіки розвинутих країн світу.

Туристична галузь у світі почала поступово відновлюватися після скасування коронавірусних обмежень у країнах. За даними Всесвітньої туристичної організації ООН, минулого року міжнародні поїздки здійснили понад 900 млн осіб: це вдвічі більше, ніж у 2021 році, але становить лише 63% від рівня до пандемії. За прогнозом, в 2023-му кількість туристичних поїздок може становити вже 80-95% від доковідного рівня. Найбільше туристична активність поки що поживається на Близькому Сході та в Європі [3].

Найбільш привабливі міста приносять до бюджету країн значний капітал, підрахунки «Всесвітньої ради з туризму та подорожей» відображаються на рисунках 1.1 та 1.2 [3]. Згідно наданої нижче інфографіці, можна зробити висновок, що світовий туризм нарешті почав відновлюватися і зважаючи на поточні ситуації в світі, зможе приймати більш складні виклики.



Рисунок 1.1 – Інфографіка проекту від проекту “Слово і Діло”, витатри грошей в сфері туризму по містах.



### Top 10 Cities with the Biggest Traveller Spend 2022 and 2023



Рисунок 1.2 – Топ 10 найпривабливіших міст по світу 2022 – 2023р

Світова тенденція на відновлення туризму показує доволі позитивні результати. Однак, загальна світова тенденція не знаходиться в таких самих умовах, як Україна, де на цю галузь припадають більш складні виклики, такі як повномасштабне вторгнення російської федерації. Українські туристичні підприємства продовжують працювати та наповнювати бюджет країни. графічне відображення у відсотковому співвідношенні відповідно до виду діяльності зображено, на рисунку 1.3 [4].



Рисунок 1.3 – Податки за I півріччя від туристичної галузі

1.2 Геомаркетинговий підхід з використанням картографічного моделювання туристичного попиту.

Геомаркетинг визначається як науковий напрямок, що займається, передусім, встановленням географічних меж ринків та визначенням їхньої ємності, а також розподільчими каналами, через які товари рухаються від виробника до споживачів. В геомаркетингу можна виділити основне коло завдань дослідження: узагальнення та аналіз просторових даних про ринки збуту та збутові операції, оцінка потенціалів реальних та перспективних ринків збуту, визначення зон обслуговування та торгових зон, вибір каналів розподілу та найефективніших місць для розміщення оптової, роздрібної торгівлі та підприємств для обслуговування населення. Загальним для всіх завдань є картографування отриманих результатів і складання маркетингових карт. Роль просторових факторів у формуванні іміджу туристичного ринку виявляється в територіальному не співпадінні місць формування та реалізації туристичного попиту, його неоднорідності та залежності від доходів, структури витрат, моди, особливостей життєвого устрою споживачів, транспортної доступності туристичних напрямків, геополітичних, санітарно-епідеміологічних та інших факторів. Цінність геомаркетингу проявляється у виділенні рекреаційних територій з певною ступенем однорідності з точки зору вподобань туристів, їх поведінки, потреб, очікувань, моделей покупки та споживання туристичного продукту. Результатом є територіальна сегментація туристичного попиту згідно з комплексом кількісних та якісних характеристик туристичних територій[1].

Використання картографічного моделювання в геомаркетингових дослідженнях туристично-рекреаційної діяльності спрямоване, передусім, на виявлення географічної вибіркості рекреаційного попиту, обумовленої різноманітними умовами та факторами. При цьому потенційна та реалізована вибірквість учасників туристично-рекреаційної діяльності визначається ступенем відповідності тих чи інших природних та соціокультурних об'єктів, напрямків чи цілих районів і країн характеру склалася уявлень (включаючи стереотипи) людей про можливості, умови та результативність проведення

туристично-рекреаційних занять [9]. Базуючись на даному розділі було побудовано схему задач геомаркетингу, вона зображена на рисунку 1.4



Рисунок 1.4 - Задачі Геомаркетингу

Також важливо звернути увагу, що дуже важливий вплив на стратегію та розробку геомаркетингу мають певні методи сегментації ринку, вони наведені на рисунку 1.5, але в даній кваліфікаційній роботі розглядався саме географічний

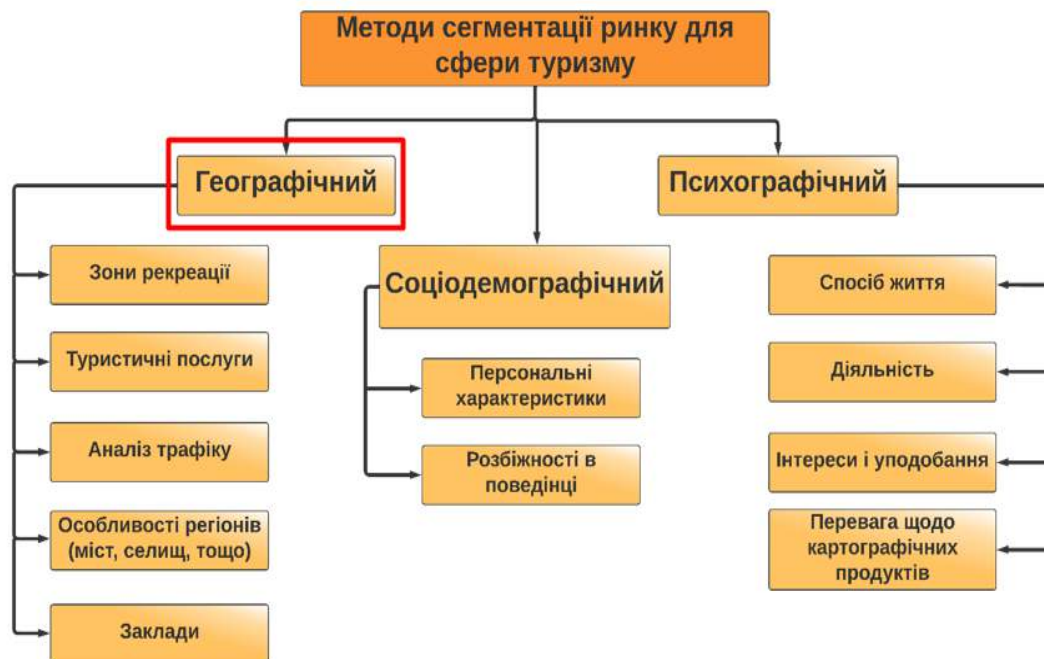


Рисунок 1.5 - Методи сегментації ринку для сфери туризму

### 1.3 Постановка мети та задачі кваліфікаційної роботи

Проаналізувавши геомаркетинг в сфері туризму та його вплив на досвід користувача, були визначені мета та задача проєкту. Метою проєкту, є підвищення ефективності розробки веб-сервісу завдяки використанню геоінформаційних технологій для вирішення задач геомаркетингу на прикладі міста Тернопіль. При розробці веб-сервісу варто звертати увагу на наявні рішення та мови програмування які будуть оптимальними для його реалізації. Також інтерфейс користувача має бути максимально доступним та зрозумілим для пересічного користувача орієнтуючись на принципи “User Friendly”.

Основними задачами є аналіз вхідних даних та їх подальша обробка, аналіз існуючих сервісів та рішень, з метою визначення необхідного функціоналу, а також на основі отриманих моделей та баз даних розробити веб-сервіс для вирішення задач геомаркетингу у сфері туризму на прикладі міста Тернопілля. Сам веб-сервіс має бути орієнтований на запити користувача, які направлені на туристичний або ж споживчий напрямок, допомагаючи знаходити цікаві або ж максимально оптимальні маршрути. Також, сервіс має надавати можливість ділитися та зберігати власні маршрути, таким чином покращуючи загальний користувацький досвід. Заклади або ж інші об’єкти, мають відповідати певному критерію, що обирає користувач (рейтинг, найближчий).

Схему основних етапів роботи зображено на рисунку 1.6.

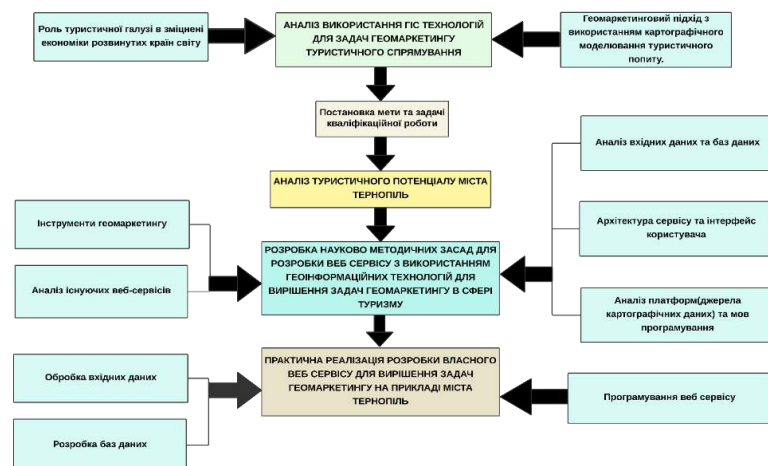


Рисунок 1.6 – Схема основних етапів роботи

## РОЗДІЛ 2 АНАЛІЗ ТУРИСТИЧНОГО ПОТЕНЦІАЛУ МІСТА ТЕРНОПІЛЬ

Тернопіль – місто в Україні (рис. 2.1), політико-адміністративний, має площу 72 км<sup>2</sup>, економіко-діловий, релігійний та культурний центр Тернопільської області. Один із трьох головних центрів історико-географічного регіону Галичина. Головне місто Галицького Поділля. Тернопіль розташований на берегах річки Серет. Місто є значним транспортним вузлом України, тут перетинаються важливі автомобільні та залізничні шляхи. Обслуговується міжнародним аеропортом «Тернопіль», що розташований за 3 км на схід від міста [2]. Згідно довоєнних даних, Тернопіль займав 8 місце, за індексом задоволеності якістю сфер та послуг, що наведено на рисунку 2.2.

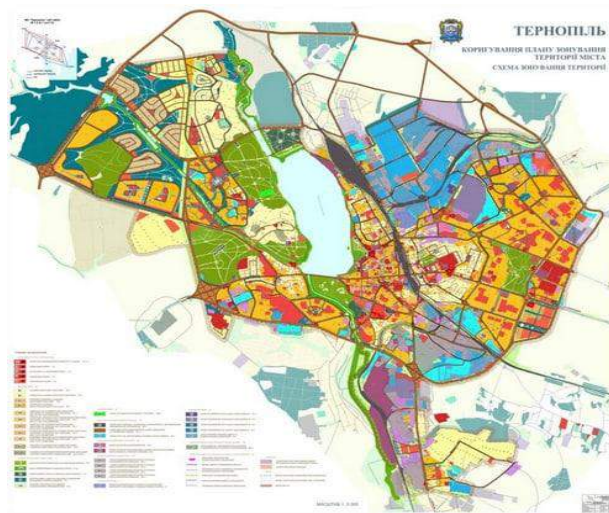


Рисунок 2.1 – Генплан м. Тернопіль



Рисунок 2.2 – Рейтинг Індексом задоволеності по містах України за 2021 рік



Місто Тернопіль має неабиякий туристичний потенціал, одразу варту звернути увагу на парк культури і відпочинку «Топільче», план-схема якого зображена на рисунку 2.2. А також на парк культури ім. Т. Шевченка, що розташований біля Тернопільського Ставу, план-схема зображена на рисунку 2.3.



Рисунок 2.2 – План-схема парку культури і відпочинку «Топільче»



Рисунок 2.3 – План-схема парку культури і відпочинку ім. Т. Шевченка

Виходячи з аналізу картографічних матеріалів міст Тернопіль, можна прийти висновку, що на жаль місто не дуже активно розвиває сферу онлайн картографії для сфери туризму, оскільки у відкритих джерелах доволі мало подібних даних, однак місто має свої додатки, де є інтерактивні карти, але вони не покривають повністю туристичну галузь, не розкриваючи цей потенціал, на зображенні 2.4 наведені актуальні інтерактивні мапи міста Тернопіль.

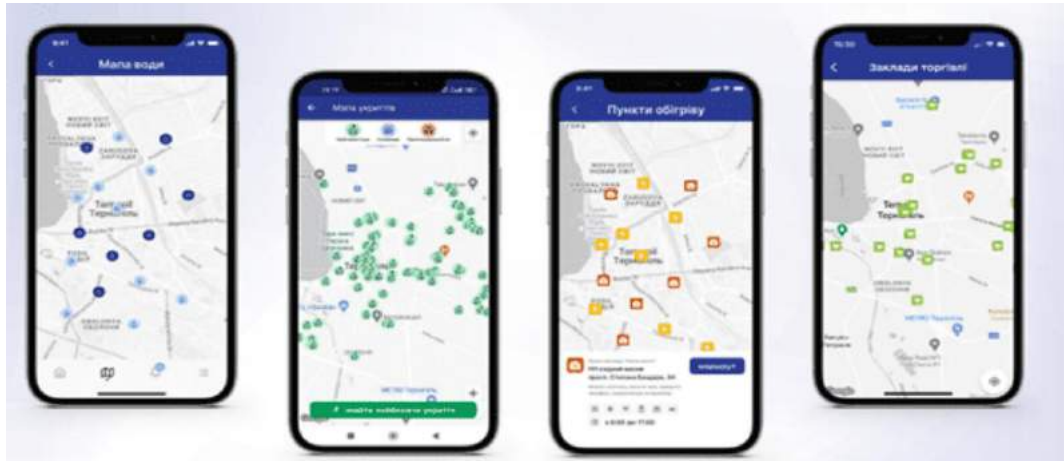


Рисунок 2.4 – Інтерактивні мапи міста Тернопіль

Одразу слід звернути увагу, що місто надає інформацію про розташування закладів торгівлі, однак в реальній ситуації не повністю відображає їх кількість, дана карта ще доробляється. Даний додаток більше орієнтований на жителів міста і дозволяє спростити деякі сфери повсякдення.

На рисунку 2.5, зображено щільність закладів в центральній частині міста.

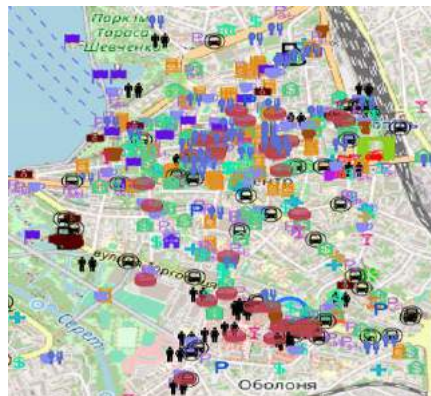


Рисунок 2.8 - Щільність закладів м. Тернопіль

Якщо підвести підсумки по місту Тернопіль, то можна одразу зазначити, що місто має не аби який потенціал в сфері туризму. Нажаль, не зважаючи на цей потенціал, місто немає допрацьованих веб-картографічних рішень для сфери туризму, існуючи орієнтуються більше на жителів міста, також наявні мапи в основному розглядають дуже маленький спектр із можливого потенціалу. Але варто зазначити, що окремі жителі міста та підприємці, а також окремі активні діячі, оновлюють інформацію про розташування різноманітних об'єктів на інших картографічних сервісах, що надає змогу об'єднати ці дані до єдиної бази даних та створювати безліч рішень під різноманітні сфери діяльності.

Використовуючи геоінформаційні технології для вирішень задач геомаркетингу в сфері туризму для міста Тернопіль, є необхідним, а також може сприяти більш ефективній та персоналізованій навігації гостей міста, завдяки веб-сервісу.



## РОЗДІЛ 3 РОЗРОБКА НАУКОВО МЕТОДИЧНИХ ЗАСАД ДЛЯ РОЗРОБКИ ВЕБ СЕРВІСУ З ВИКОРИСТАННЯМ ГЕОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ГЕОМАРКЕТИНГУ В СФЕРІ ТУРИЗМУ

### 3.1 Інструменти геомаркетингу

Геомаркетинг використовує різноманітні інструменти для аналізу географічних даних та вивчення взаємозв'язків між місцезнаходженням і бізнес-процесам[8]. Тобто завдяки геомаркетингу можна визначити, а після проаналізувати окремі зони інтересу для сфери туризму, на рисунку 3.1 наведені основні інструменти геомаркетингу.

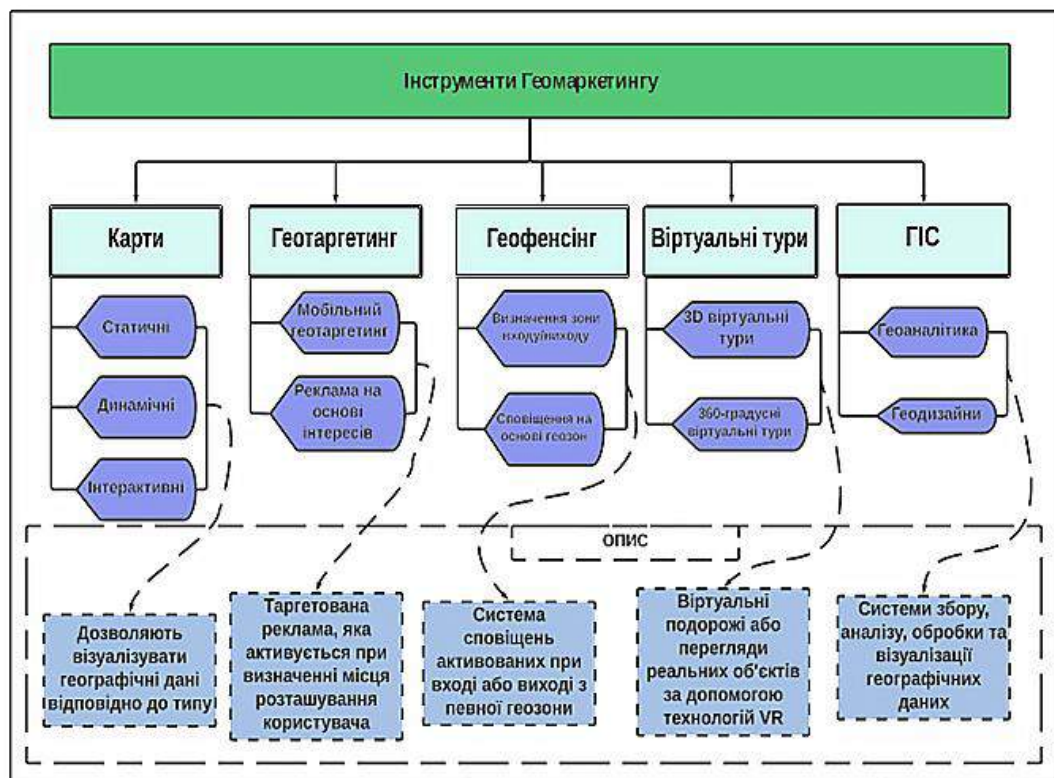


Рисунок 3.1 – Розробка структурної схеми основних інструментів геомаркетингу

Серед інструментів геомаркетингових досліджень туристичного попиту на основі синтезу геоінформаційних, маркетингових та цифрових технологій слід виділити:

- базові, які використовують традиційний просторовий аналіз на основі абсолютних географічних характеристик території[8];
- розширені, які використовують методи моделювання відносинних показників і їх похідних. Серед них виділяється аналіз транспортної доступності туристичного напрямку, аналіз положення відносно конкурентних ринків, місць формування туристичного попиту. Одним із найефективніших інструментів геомаркетингового дослідження туристичного попиту є метод картографічного моделювання[8].

### 3.2 Аналіз існуючих веб-сервісів.

В ході проекту був проаналізований ринок додатків та сервісів, які спрямовані на навігацію користувачів, за основу пошуку рішень, було взято Google Maps, даний додаток/сервіс виконує навігацію користувачів по конкретним запитам, тобто з пункту А до пункту Б, також надає можливість власноруч розширити свій маршрут шляхом додавання проміжних точок під час подорожі, надаючи декілька варіантів для досягнення пункту призначення. Найкоротший та найшвидший. Перший визначається на основі найменшої відстані між двома точками, як правило, він включає в себе коротші ділянки доріг або шляхів, тощо. Найшвидший базується на розрахунках часу, необхідного для подорожі від одного пункту до іншого. Він може включати швидші дороги або автостради, які дозволяють швидше переміщуватися, навіть якщо це призведе до збільшення загальної відстані[7].

Google Maps можна назвати великою навігаційною базою, яка постійно оновлюється і охоплює значну сферу впливу, однак даний підхід не підходить під мету даного проєкту, тому був проведений більш ґрунтовний аналіз платформ, який включає в себе мінімальний набір функцій для запланованого веб-сервісу. Далі на основі результатів аналізу Counterpoint Research(рис. 3.2), були визначені сервіси для оцінки їх функціоналу і визначення подальшого для власного проєкту.

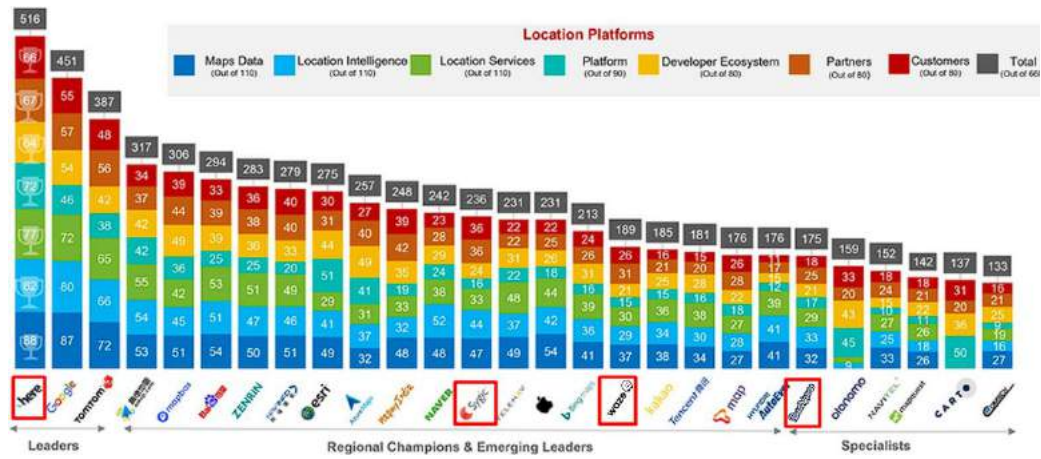


Рисунок 3.2 – Аналіз картографічних платформ/додатків по світу

В результаті аналізу картографічних платформ, було виділено 4, порівняння функцій яких, наведено у таблиці 3.1.

Таблиця 3.1 - Порівняння функцій додатків

Функції	Sygy	Waze	HERE WeGo	Roadtripper
Сповідження про трафік	Так	Так	Так	Обмежено
Варіанти маршруту	Так	Так	Так	Так
Спільнота користувачів	Ні	Так	Ні	Ні
Публічний транспорт	Ні	Ні	Так	Ні
Подорожі на велосипеді	Так	Ні	Так	Ні
Планування зупинок	Так	Так	Так	Так
Подорожі пішки	Так	Так	Так	Так
Подорожі автомобілем	Так	Так	Так	Так
Подорожі громадським транспортом	Ні	Ні	Так	Ні
Планування екскурсій	Так	Ні	Так	Так

Проаналізувавши наявні сервіси, можна прийти висновку, щодо необхідного функціоналу. Варто виділити такі як, Варіанти маршруту, публічний транспорт, подорожі різними способами пересування(транспорт, пішки, велосипед), а також спільнота користувачів. Остання функція дозволить покращувати індивідуальний досвід кожного окремого користувача на основі інших, такий висновок можна зробити на основі того як працюють соціальні мережі, де кожен може лишити свою думку у відкритій формі на загальному

бачені. Традиційні подібні сервіси працюють за принципом відгуку на окремій платформі або ж через служби підтримки.

3.3 Аналіз платформ(джерела картографічних даних) та мов програмування.

Виходячи з аналізу існуючих сервісів, можна проаналізувати платформи, які слугують в якості постачальників картографічних даних, а також мов програмування, які безпосередньо застосовувались для реалізації їхнього функціоналу.

Під час аналізу джерел картографічних даних їх було розділено на дві категорії, комерційні та відкриті. Комерційні постачальники геоданих, являють собою такі, які можуть надавати право на розповсюдження свого вмісту на певних умовах, перевищення яких веде до стягнення плати задля подальшого користування. Відкриті джерела в свою чергу, надають змогу використовувати свій вміст по ліцензії Creative Common, що дозволяє його використання будь де і будь якій формі за виконання додаткових вимог автора, тобто розміщення посилань на джерело, а також висвітлювання у публічний доступ будь яких додаткових змін до відкритого коду.

– Комерційними постачальниками геоданих, є такі як Carto, ArcGIS Online, Mango Map, MapBox. Вони надають доступ до широкого спектру географічних даних по всьому світу. Ці постачальники також пропонують власні API та інструментарій для розробки різноманітних картографічних додатків та сервісів, в тому числі і для виконання завдань геомаркетингу.

– Відкритими ж джерелами є, OpenStreetMap, Geonames та інші. Вони в свою чергу надають абсолютно безкоштовний доступ до географічних даних, включаючи інформацію про географічні об'єкти, геокодування та інші корисні дані. Ці дані можуть бути завантажені та використані для різноманітних завдань та задач, в тому числі і геомаркетингу. Недолік таких джерел, це в основному необроблені дані, оскільки вони заповнюються окремими користувачами, ступінь їх інформативності може бути різним.

В якості основного джерела вхідних даних було обрано OSM, оскільки дана платформа надає велику кількість даних по всьому світу, на жаль інші рішення не мають такого рівня покриття у безкоштовному варіанті. В якості додаткового сервісу для отримання певних даних OSM також було використано mygeodata, він надає можливість отримати тільки конкретні геодані та інформацію про них. Схема отримання даних зображена на рисунку 3.4.

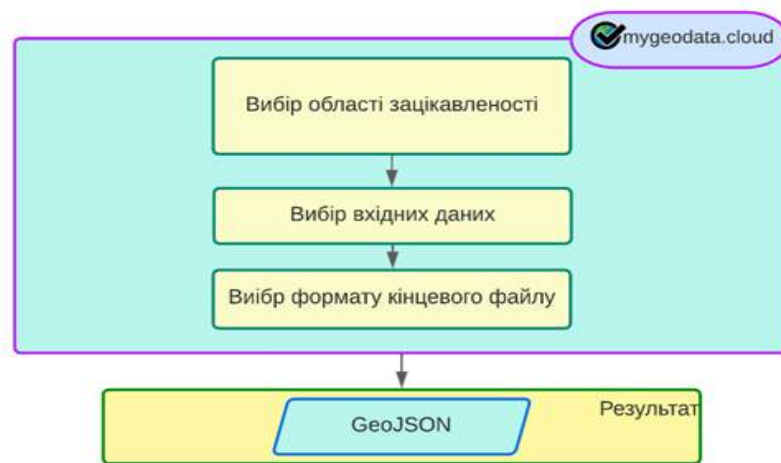


Рисунок 3.4 - Отримання даних з сервісу mygeodata

Також в якості додаткового сервісу був використаний Graphhopper, який є постачальником механізму маршрутизації для даних OSM, оскільки він працює автономно на виділеному сервері, то дозволяє зменшити навантаження на базу даних для обчислень маршрутів. Сам по собі Graphhopper, є відкритим кодом, але використання його API має як безкоштовну форму (обмежена кількість запитів), так і платну на основі підписки [10]. В проєкті даний сервіс буде використовуватись виключно паралельно з власною базою даних.

В результаті аналізу різноманітних сервісів, була виділена низка мов програмування для їх реалізації, вони наведені у таблиці 3.2, де розглянуті їх характеристики по відношенню до поставлених задач.

Таблиця 3.2 - Характеристи мов програмування для веб-розробки

Характеристики	JavaScript	Python	PHP
Сфера застосування	Веб-розробка, аналіз даних	Веб-розробка, аналіз даних, штучний інтелект	Веб-розробка, серверні додатки
Спрощення синтаксису	Синтаксис має виразність та динаміку.	Простий та читабельний синтаксис	Легкий синтаксис
Асинхронність	Велика активна спільнота, широкий вибір бібліотек для розробки.	Підтримує асинхронність, але менше ніж JavaScript	Підтримує асинхронність завдяки додатковим бібліотекам
Спільнота та бібліотеки	Велика активна спільнота, широкий вибір бібліотек для розробки.	Активна спільнота, велика кількість бібліотек під різноманітні завдання	Значна спільнота та велика кількість фреймворків
Легкість вивчення	Середня, має свої особливості.	Легка читабельність	Легка та інтуїтивна
Швидкість виконання коду	Швидка	Середня	Не сильно швидка, поступається JS та Py

Отже, виходячи з проаналізованого та поставленої задачі, можна одразу сказати, що мова програмування PHP не підходить, вона призначена біль для написання серверної(BackEnd) частини, що не дуже підходить під задачі проекту. З цього, пріоритет був наданий такій мові програмування, як JavaScript, оскільки дана мова має велику кількість бібліотек направлених саме на створення подібних сервісів, або мають суб-функціонал, що дозволяє їх використовувати в даних цілях, але також був розглянутий і python, а саме його фреймворк Django. Django є безкоштовним і відкритим фреймворком для створення веб додатків, він дозволяє оптимізувати керування окремими елементами та посиланнями для більш зручного та зрозумілого використання веб сервісу. Однак, слід зазначити, що Python має низку переваг над JavaScript в сфері ГІС, але для даного проекту був обраний саме другий, оскільки реалізація на ньому даного напрямку є більш зрозумілою і простішою, через наявну розширену документацію, навідміну від Python.

Також в проєкті був використаний додаток QGIS, який являє собою професійний ГІС-додаток, призначений для створення і використання картографії. Дана програма має широке використання по світу і є відкрити кодом. Не дивлячись на свою безкоштовність, QGIS має ряд переваг над комерційними системами, в якості основних можна виділити, його мультиплатформерність, багатий асортимент безкоштовних модулів(від користувачів), вільний доступ до вхідного коду спрощує створення нових модулів або скриптів, програма має сумісність з різними спеціалізованим обладнання.

Ще одним програмним забезпеченням в проєкті є Blender з його плагіном BlenderGIS, який є потужним інструментом, що використовує геодані в графічному редакторі. Він надає змогу створення тривимірних моделей світу з географічною прив'язкою. Що в свою чергу надає змогу подальшого використання на веб-сервісі для покращення користувацького досвіду.

#### 3.4 Аналіз вхідних даних та баз даних

На основі проведеного аналізу існуючих рішень, які використовують окремі джерела векторних даних, було визначено, що найкращим форматом для їх зберігання та подальшої обробки, краще використовувати формат, GeoJSON. Він являє собою формат кодування різноманітних структур геоданих, вигляд якої представлений на рисунку 3.5. Він підтримує наступні типи геометрії: Point, LineString, Polygon, MultiPoint, MultiLineString та MultiPolygon. Варто зазначити, що даний формат є доволі розповсюдженим і використовується багатьма сервісами та додатками для зберігання, роботи з геоданими.

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

Рисунок 3.5 - Структура формату GeoJSON



Даний формат також можна представити у вигляді окремої таблиці в базі даних, яка містить інформацію різноманітну інформацію про об'єкт. На рисунку 3.5 представлені векторні дані м. Тернопіль



Рисунок 3.5 - Векторні дані м. Тернопіль

На даній картографічній моделі представлені такі дані як, розташування об'єктів(точкові), дороги(лінійні) та будівлі(полігони). Всі ці об'єкти необхідні для подальшої обробки та зберігання.

Точкові об'єкти будуть використані для візуалізації місцезнаходження закладів у м. Тернопіль. Лінійні об'єкти необхідні для розрахунків маршрутів згідно запиту користувача, вони будуть зберігатись в окремій таблиці бази даних.

Також до точкових об'єктів необхідно додати додаткову інформацію у вигляді рейтингу, який базується на відгуках про об'єкти у різноманітних джерелах. Ці дані можна отримати з гугл відгуків, а також інших ресурсів орієнтованих на споживача. Відгуки зберігаються у xml файлі після чого експортуються у базу даних і прив'язуються до кожного точкового об'єкту.

При виборі бази даних варто звернути увагу на такі критерії, як робота з геоданими, реаліційні можливості, відкритість, швидкодія та масштабованість. Ці критерії є основою при виборі БД для проектування подібних веб сервісів або додатків. Звісно, найважливішим аспектом буде робота з геоданими та



відкритість. В таблиці 3.3 наведене порівняння БД, які є найбільш підходящими для даного проєкту.

Таблиця 3.3 - Порівняння Баз Даних

Критерій	PostgreSQL	MySQL	Oracle	Microsoft SQL server
Геодані	Підтримка географічних типів та функцій через розширення PostGIS	Підтримка географічних типів та функцій через розширення SpatialLite.	Повна географічна підтримка з Oracle Spatial	Географічна підтримка з Microsoft SQL Server Spatial Extensions.
Реляційні можливості	Повноцінна реляційна система баз даних	Реляційна система баз даних з обмеженою географічною підтримкою.	Повноцінна реляційна система баз даних.	Повноцінна реляційна система баз даних.
Відкритість	Відкрита система з активною спільнотою, розповсюдження безкоштовне	Відкрита система, з активною спільнотою, розповсюдження безкоштовне	Відкрита система з окремими комерційними опціями.	Відкрита система з комерційними опціями.
Швидкодія	Незважаючи на великі обсяги даних працює доволі швидко, залежить від запиту	Залежить від конкретної конфігурації та оптимізацій.	Швидкодія може бути високою, але залежить від конфігурації та оптимізації.	Швидкодія може бути високою, але залежить від конфігурації та оптимізації.
Масштабованість	Здатна масштабуватися на великі обсяги даних та велику кількість одночасних користувачів	Може вимагати додаткових оптимізацій для масштабування.	Здатний масштабуватися на великі обсяги даних та велику кількість одночасних користувачів.	Здатний масштабуватися на великі обсяги даних та велику кількість одночасних користувачів.

Виходячи з аналізу варто звернути увагу на три СУБД, а саме PostgreSQL, Oracle та Microsoft SQL server. Оглядаючись на критерій по роботі з геоданими усі СУБД мають майже однаковий ступінь в роботі з зазначеними даними, однак з точки зору відкритості краще розглядати саме PostgreSQL, який незважаючи на свою повну безкоштовність має доволі велику швидкодію з великими даними. Тому у якості бази даних варто розглянути, PostgreSQL та її просторове розширення PostGIS. PostgreSQL базується на мові SQL. PostgreSQL належить до наступного типу СУБД – об’єктно-реляційного (тобто поєднує в собі переваги

реляційних баз даних і об'єктно-орієнтованого програмування). PostgreSQL підтримує широкий спектр функцій, які роблять її потужною і гнучкою СУБД. Ось деякі з її ключових особливостей:

- PostgreSQL забезпечує транзакції з ACID-властивостями, що гарантує цілісність даних.
- Підтримує автоматично оновлювані подання, що забезпечує можливість користувачам переглядати дані в базі даних у режимі реального часу.
- Підтримує матеріалізовані подання, які зберігають копію даних подання в базі даних.
- Підтримує тригери, які дають змогу виконувати дії під час зміни даних у таблиці.
- Підтримує зовнішні ключі для створення зв'язку між таблицями.
- Підтримує збережені процедури для виконання складних обчислень на стороні сервера. [11]

Виходячи з вищезазначеного, PostgreSQL підходить для створення динамічних веб-сервісів та додатків. Також дана БД має високу продуктивність при обробці великих обсягів інформації, з цього можна зробити висновки, що дана СУБД ідеально підходить для аналізу даних. Також варто зазначити, що дана БД має просторове розширення PostGIS.

PostGIS - це розширення для системи керування базами даних PostgreSQL, призначене для роботи з просторовою інформацією. PostGIS дозволяє зберігати та опрацьовувати географічні об'єкти, такі як точки, лінії та полігональні області, що робить його потужним інструментом для роботи з геоданими.

Основні можливості PostGIS включають:

- Зберігання геоданих: PostGIS надає можливість зберігання географічних об'єктів в базі даних PostgreSQL, використовуючи різні типи даних, такі як POINT, LINESTRING, POLYGON тощо.
- Операції з просторовими даними: PostGIS підтримує широкий спектр операцій з геоданими, такі як перетин, об'єднання, визначення відстаней та інші.

Це дозволяє виконувати складні аналізи та операції над просторовими об'єктами безпосередньо в базі даних.

- Пошук об'єктів в межах географічних областей: PostGIS дозволяє виконувати запити, спрямовані на знаходження географічних об'єктів, які перетинають або містяться в межах певних географічних областей.

- Підтримка растрових даних: Починаючи з версії 2.0, PostGIS також підтримує растрові дані, що дозволяє зберігати та опрацьовувати зображення та інші растрові дані.

- Інтеграція з іншими інструментами геоінформаційних систем: PostGIS може взаємодіяти з іншими системами геоінформацій, підтримуючи стандарти та формати, такі як GeoJSON, KML, GML тощо. [11]

Отже, база даних PostgreSQL, підходить для збереження інформації про користувачів, а також геоданих, що є основною перевагою для даного проєкту. В якості зручного використання для даної БД, можна використовувати десктопний додаток *spatialite gui*.

### 3.5 Архітектура сервісу та інтерфейс користувача.

Під архітектурою веб-сервісу розуміють структурний та організаційний план, який визначає розподіл функціональних компонентів, взаємодію між ними, а також способи забезпечення високої ефективності. Архітектура веб-сервісу має охоплення різних рівнів, тобто включаючи, як клієнтську частину, так і сервер з базами даних. [5]

Виходячи з проаналізованої інформації про роботу подібних веб-сервісів, була розроблена архітектура власного веб рішення, яка зображена на рисунку 3.6 та відображає взаємодію між користувачем, логіку сервісу та взаємодію зі сховищем геоданих.

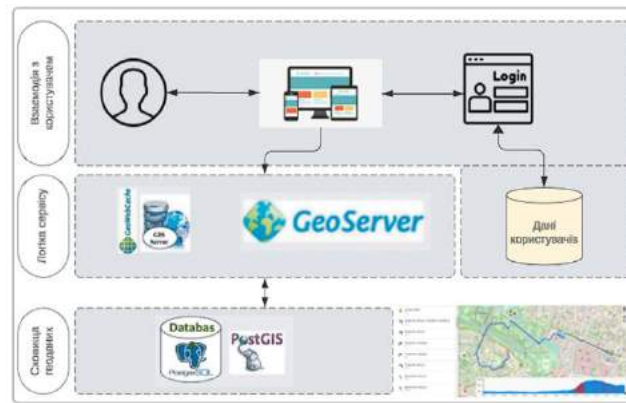


Рисунок 3.6 - Архітектура розробленого веб-сервісу.

За даною схемою користувач при використанні сервісу, має пройти процес авторизації або ж реєстрації, після чого його дані будуть збережені у окремій БД, далі йому вже буде наданий доступ до основного функціоналу ресурсу, який має взаємодіяти з базами геоданих та геосервером OSM, який надає додаткову картографічну інформацію і взаємодіє вже з наявними даними.

Одним з важливих етапів є розробка інтерфейсу користувача (Далі ІК), оскільки він визначає головний аспект веб-сервісу, який і є визначенням засобу взаємодії користувач-платформа. Однією з ключових функцій ІК, є створення графічного інтерфейсу, який відображається на екрані у користувача. ІК має бути легким і зрозумілим, щоб доступ до основних функцій веб-сервісу не був ускладнений. З цього важливо виділити такі визначення як UI та UX. UI являє собою той самий ІК і є процесом візуалізації прототипу, який розробили на основі UX. Він може містити в собі роботу над: анімацією, ілюстраціями, кнопками, меню, слайдерами, фотографіями та шрифтами.

UX являє собою користувацький досвід, інакше цей різновид дизайну відповідає за функції, адаптивність продукту і те, які емоції він викликає у користувача. Чим зрозуміліший інтерфейс, тим легше користувачеві отримати результат.

Отже підсумувавши проаналізовану інформацію був розроблений інтерфейс користувача для власного веб сервісу з урахуванням усіх потенційних функцій,

на рисунку 3.7 приведено алгоритм взаємодії користувача з інтерфейсом, а повний вигляд на рисунку 3.8.



Рисунок 3.7 – Алгоритм взаємодії користувача з інтерфейсом



Рисунок 3.8 - Інтерфейс користувача

### 3.6 Підсумки та потенціал для розвитку

На основі проаналізованих даних для проєкту по створенню веб-сервісу з використання ГІС технологій для вирішень задач геомаркетингу на прикладі міста Тернопіль, були визначені необхідні вхідні дані, а саме векторні шари у форматі GeoJSON, для їх подальшого збереження та обробки була обрана СУБД PostgreSQL з розширенням PostGIS. Для створення та тестування картографічних моделей був обраний ГІС-додаток QGIS, а також для створення 3D моделі забудови міста Тернопіль, графічний редактор Blender з його розширенням BlenderGIS. І звісно для створення та програмування функціоналу були обрані мови програмування, JavaScript та Python.

Для реалізації веб-сервісу також було розглянуто використання додаткової бібліотеки для JavaScript, Leaflet. Він являє собою бібліотеку з відкритим кодом для інтерактивних карт з використанням різних провайдерів та власних даних. Ключове в Leaflet, це його простота, продуктивність та зручність використання. Він ефективно працює на всіх пристроях, також може бути розширений завдяки багатьох плагінів. Основним компонентом бібліотеки, є Карта, яка може бути розміщена в будь якому елементі HTML і має прямокутну форму. Параметрами карти, є область показу, вид карти та відображення, завдяки функціоналу Leaflet, джерелом картографічної інформації може бути любе із актуальних в залежності від рівня доступу, окрім цього всі ці шари легко налаштовуються та можуть бути обрані як одні з базових. Також на карті можна відображати шари. Найчастіше використовується для відображення векторної точкової інформації, однак також є можливість відображення 3D моделей. Важливий момент, що кількість відображаємих об'єктів необмежена, все залежить від мети і цілей. Leaflet дозволяє завантажувати також і власні дані, джерелом їх можуть як файли з просторовим розширенням по типу GeoJSON, так і віддалені сервери. Зведена інформація, щодо використовуємого у проєкті наведена у таблиці 3.4.

Таблиця 3.4 - Використані платформи та мови програмування у проєкті

Назва	Використання в проєкті
QGIS	Використовується для тестування, створення картографічних моделей та налаштування векторних шарів для подальшої інтеграції до сервісу, а також для їхньої додаткової стилізації
Blender	Використовується для швидкого створення 3D забудови міста Тернопіль, завдяки плагіну BlenderGIS
PostGreSQL	Використовується для зберігання даних користувачів, геопросторових об'єктів та додаткової інформації про них, а також для додаткових розрахунків пов'язаних з маршрутизацією
JavaScript	Використовується для прописування основної логіки веб-сервісу, а також для взаємодії з елементами Frontend
Leaflet	Використовується для створення інтерактивної карти веб-сервісу, надаючи змогу інтегрувати отримані вхідні дані
HTML+CSS	Використовується для розробки інтерфейсу користувача та його стилізації
Python	Використовується для розробки бекенд частини, підключення баз даних, налаштування посилань, тощо

Щодо подальшого розвитку проєкту, варто звернути увагу на можливість інтеграції ГІС у більш соціальну сферу і зробити сервіс щось, що нагадує соціальну мережу, де користувачі комунікують один з одним, обмінюються цікавими маршрутами, подорожують разом з різних куточків міста, області, країни. Також це надало би змогу покращувати індивідуальний досвід кожного користувача, оскільки кожен би міг відкрито ділитись думкою щодо сервісу або геоданих. З точки зору власників закладів можливість доступу до доступного геомаркетингу. Тобто надаючи більше інформації про свій заклад і таким чином робити його більш привабливим для окремих користувачів.

Для більш ефективної навігації користувача, додатку необхідна інформація про знижки, акції та ціни. Це б допомогло створювати більше фільтрів за запитом. Також в подальшому є змога інтеграції нейромережі, яка буде оцінювати усі запити користувача створюючи йому цікаві підборки, наразі це реалізується завдяки звичайній фільтрації. Також для більшого охоплення та зручного використання розробка мобільної версії сервісу для IOS та Android. З вищенаведеного було розроблено дорожню карту, вона зображена на рисунку 3.9.

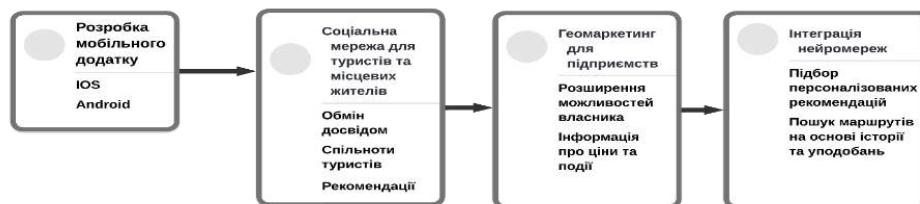


Рисунок 3.9 - Дорожня карта проєкту

## РОЗДІЛ 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБКИ ВЛАСНОГО ВЕБ СЕРВІСУ ДЛЯ ВИРІШЕННЯ ЗАДАЧ ГЕОМАРКЕТИНГУ НА ПРИКЛАДІ МІСТА ТЕРНОПІЛЬ

### 4.1 Обробка вхідних даних.

Для обробки вхідних даних було використано такі програмні забезпечення, як QGIS та Blender, основні етапи обробки зображені на рисунку 4.1.

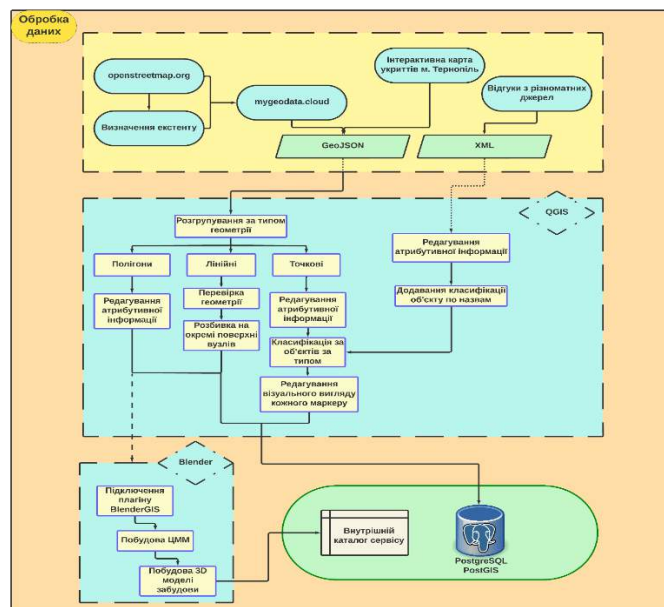


Рисунок 4.1 - Обробка даних

В першу чергу були завантажені дані GeoJSON по місту Тернопіль, куди входять всі типи геометрії, що зображено на рисунках 4.2 - 4.3.

ID	Слой	Количество объектов	Тип геометрии	Описание
1	lines	Неизвестно	LineString	
2	multilinestrings	Неизвестно	MultiLineString	
3	multipolygons	Неизвестно	MultiPolygon	
4	other_relations	Неизвестно	GeometryCollection	
0	points	Неизвестно	Point	

Вибрати все    Сняти выделение     Добавить слой в группу

OK    Отмена

Рисунок 4.2 – Завантаження шарів до проєкту





Рисунок 4.3 – Візуалізація необроблених векторних даних.

Як можна побачити на рисунку за цими даними вже можна якось ідентифікувати, що це якась частина міста, однак на даному етапі не стоїть мета досягти певного рівня візуалізації. Оскільки сервіс має показувати користувачу місце розташування певних об'єктів, необхідно прокласифікувати точкові дані.

Перед цим, до атрибутивної таблиці з точковими даними також необхідно додати ще один стовпчик, де будуть міститись відгуки(рисунки 4.4). Після цього можна класифікувати усі точкові об'єкти та надати їм візуалізацію згідно типу заклад(рис 4.5).

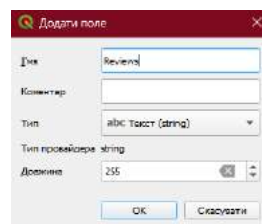


Рисунок 4.4 – Додавання нового поля

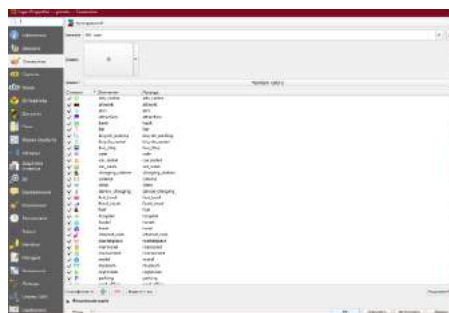


Рисунок 4.5 – Класифікація об'єктів

Тепер можна додати відгуки та рейтинг для кожного закладу, завдяки заздалегідь створеному xml файлу, для швидкої операції об'єднання використовується мова SQL, запит зображений на рисунку 4.6.

```
UPDATE points SET points.rev = RR.rev, points.raib = RR.raib FROM RR WHERE points.name = RR.name
```

Рисунок 4.6 – SQL запит

Наступним кроком була обробка усіх лінійних об'єктів, першим найважливішим етапом є перевірка геометрії, оскільки вхідні дані є необробленими, а також постійно оновлюються, можуть виникати незначні помилки, однак дуже впливові на процес побудування маршрутів, через що по таким ділянкам вони взагалі не будуть прораховані, оскільки в якості визначеної кінцевої вершини отримують значення null. Програмне забезпечення QGIS надає інструменти для аналізу та перевірки геометрії. Оскільки даний шар(лінійні об'єкти) буде використовуватись лише для покращення навігації в самому місті Тернопіль, то зайві елементи, можна обрізати по кордонам міста. Насправді побудова мережі не є обов'язковим етапом для створення подібних сервісів, оскільки таке джерело даних як OSM, надає усе необхідне, лише треба додатковий провайдер для налаштування навігації або ж геокодування, однак метою проєкту є максимально якісно покрити усі зони інтересу міста Тернопіль та сформувати підґрунтя для подальшого розвитку, а також в якості провайдера для навігації використовується API Graphhopper, він є умовно безкоштовним, але може працювати із власними даними проєктів, що надає можливість використовувати його без всяких труднощів пов'язаних із створенням нових ключів або ж внесенням оплати. На рисунку 4.7 зображено оброблений шар лінійних об'єктів.

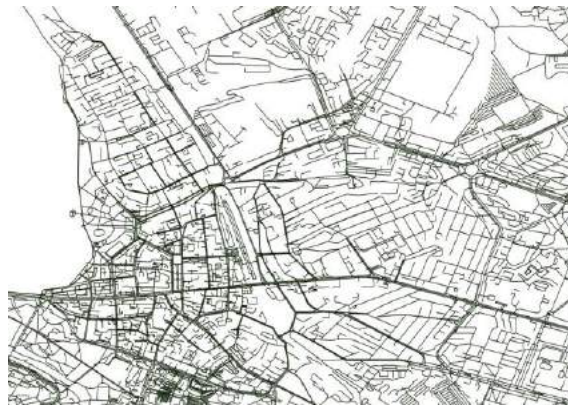


Рисунок 4.7 – Оброблений шар лінійних об'єктів

Отримані дані були додатково розбиті на окремі лінії для підвищення ефективності побудови маршрутів, після цього у форматі GeoJSON були експортовані до бази даних, атрибутивна таблиця зображена на рисунку 4.8.

id	geom_id	name	highway	waterway	oneway	barrier	man_made	r_order	other_tags
1	133173228	NULL	path	NULL	NULL	NULL	NULL		{} NULL
2	133173229	NULL	cycleway	NULL	NULL	NULL	NULL		{} "surface"="gr...
3	133173230	NULL	path	NULL	NULL	NULL	NULL		{} NULL
4	133173227	NULL	path	NULL	NULL	NULL	NULL		{} NULL
5	133173231	NULL	path	NULL	NULL	NULL	NULL		{} NULL
6	133173225	NULL	track	NULL	NULL	NULL	NULL		{} "bicycle"="ye...
7	133173219	NULL	cycleway	NULL	NULL	NULL	NULL		{} "surface"="gr...
8	133173232	NULL	path	NULL	NULL	NULL	NULL		{} NULL
9	133173954	NULL	path	NULL	NULL	NULL	NULL		{} "surface"="gr...
10	133173954	NULL	path	NULL	NULL	NULL	NULL		{} "foot_visibility"...
11	133173235	NULL	path	NULL	NULL	NULL	NULL		{} "surface"="im...
12	133173236	NULL	path	NULL	NULL	NULL	NULL		{} "foot_visibility"...
13	133173233	NULL	path	NULL	NULL	NULL	NULL		{} NULL
14	133173234	NULL	path	NULL	NULL	NULL	NULL		{} "foot_visibility"...
15	133173231	NULL	path	NULL	NULL	NULL	NULL		{} NULL
16	133173232	NULL	path	NULL	NULL	NULL	NULL		{} NULL
17	133173232	NULL	footway	NULL	NULL	NULL	NULL		{} "surface"="im...
18	133173236	NULL	path	NULL	NULL	NULL	NULL		{} "foot_visibility"...

Рисунок 4.8 – Атрибутивна таблиця лінійного шару

Для наступного етапу роботи був також відредагований шар полігональних об'єктів, оскільки 3D моделі в плагіні BlenderGIS будуються стандартно по прямокутному екстенду, то вони будуть містити зайві дані, тому для цього була визначені кордони міста Тернопіль і по ним було вибрано усі об'єкти, що входять виключно до нього, на рисунку 4.9 та 4.10 продемонстрований процес обрізки векторних шарів.



Першим кроком було розширення можливості Blender до 3D ГІС, встановивши доповнення BlenderGIS-master. Далі було створено новий проект та завантажено до нього базову карту Google(рис. ).



Рисунок 4.12 – Завантажена базова карта у Blender

Далі за допомогою пошуку було знайдено задану локацію та експортовано обраний фрагмент базової карти до робочої області проекту.

Наступним кроком було завантажено рельєф місцевості. Для цього переходимо до плагіну GIS-Web geodata-Get elevation (SRTM). Далі обираємо налаштування завантаження даних SRTM та отримуємо ЦММ відкритої ділянки базової карти(рис. )

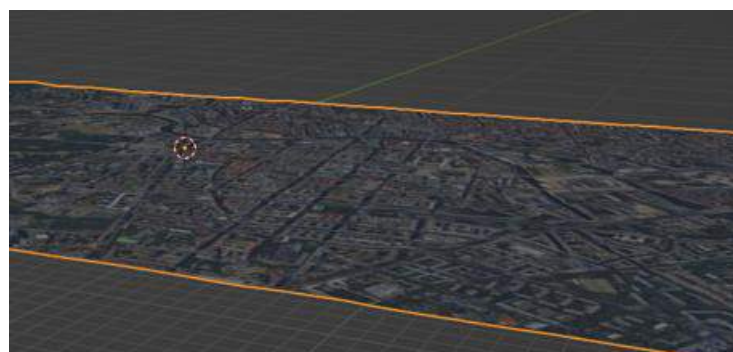


Рисунок 4.13 – Отримана ЦММ обраної ділянки базової карти

Далі було перейдено до отримання моделі забудови. Завантаження шарів відбувається автоматично з обраного ресурсу через розширення GIS- Web geodata-Get OSM. Даний модуль, використовуючи базу даних та мапу OSM,

підіймає двовимірну мапу за зазначеною висотою, отримуючи об'ємні фігури будівель зазначених на мапі. Налаштування даного модулю дозволяють отримати геодані шару будівель, рослинності, водних об'єктів, доріг та залізничних шляхів (рис. ).

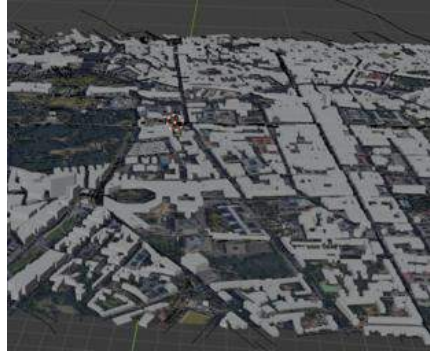


Рисунок 4.14 – Отримана модель забудови

#### 4.2 Розробка баз даних.

В якості СУБД для проєкту була обрана PostgreSQL, на її основі було побудовано дві Базы Даних, одна буде слугувати для зберігання інформації про користувачів, а інша для зберігання геоданих.

Для початку розробки бази даних було визначено, що саме буде там зберігатися. У випадку користувацької, будуть зберігатися їх nickname, пароль, історія пошуку та унікальний ідентифікатор. Для бази геоданих, координати об'єктів, векторне зображення(для точок), геометрію, опис, назву, рейтинг та відгуки. Також у випадку геоданих для подальшої побудови маршрутів в PostGIS необхідно використовувати її функцію PG Routing.

Першим кроком в створенні бази геоданих, є підключення до локального серверу з пустою БД, на рисунках 4.15 – 4.17 відображений даний процес.



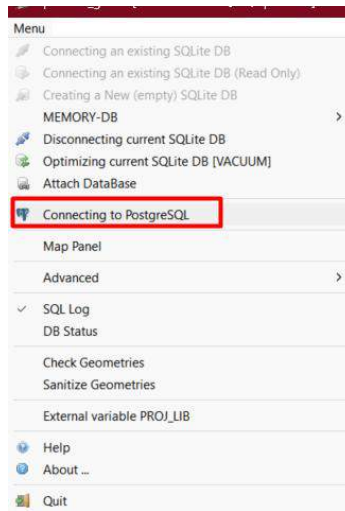


Рисунок 4.15 – Підключення до локального серверу БД

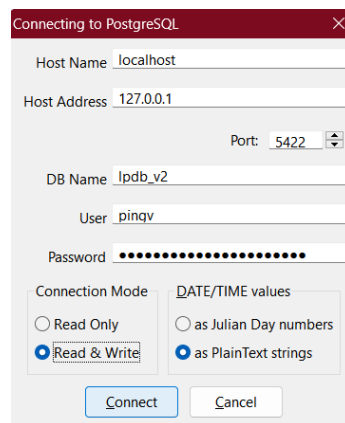


Рисунок 4.16 – Підключення до БД

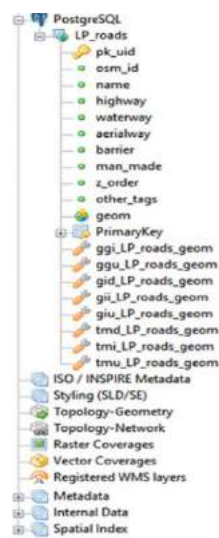


Рисунок 4.17 – Результат підключення Бази Даних

Наступним кроком було підключення `pgrouting`, для цього необхідно також підключити і модуль `postgis`, процес їх підключення зображений на рисунку 4.18.

```
psql lpdb_v2 -c "create extension postgis"
psql lpdb_v2 -c "create extension pgrouting"
```

Рисунок 4.18 – Підключення модулів `postgis` та `pgrouting`

Далі було побудовано топологію для завантажених даних. Це означає, що для будь-якого заданого ребра у ваших вуличних даних кінці цього ребра будуть з'єднані з унікальним вузлом та іншими ребрами, які також з'єднані з тим самим унікальним вузлом. Коли всі ребра з'єднано з вузлами, ми маємо граф, який можна використовувати для маршрутизації за допомогою `pgrouting`.

```
SELECT pgr_createTopology('lp_roads', 0.000001)
```

Рисунок 4.19 – Створення топології

`PgRouting` надає низку запитів за різними алгоритмами. Для побудови маршрутів згідно запитів користувачів буде використовуватись алгоритм Дейкстри, а для визначення зони інтересу використовується ізохорни. Першим кроком буде створення функції, яка буде будувати ізохорни згідно запиту з веб-сервісу. Вона буде працювати за принципом створення тимчасової таблиці, на рисунку 4.20 зображена реалізація даної функції. На рисунку 4.21 зображено архітектуру бази даних.

```
create table isochl as SELECT t1.seq, t1.id1 AS Node, t1.id2 AS Edges, t1.cost, t2.geom AS edge_geom
FROM
pgr_DrivingDistance(
  'SELECT id, source, target, length as cost FROM test1',
  (SELECT id FROM lpdb_v2 as xs WHERE "Name" = 'HILTRUP'),
  100);
-- (false, false) t1
Left join
test2
on t1.id2 = t2.id;
```

Рисунок 4.20 – Створення функції побудови ізохорн



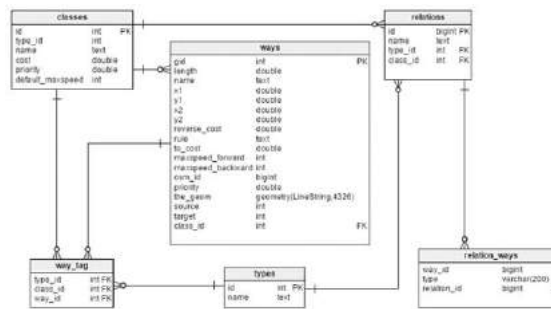


Рисунок 4.21 – Архітектура бази даних

Наступним кроком було створення бази даних для користувачів, як і попередня БД вона базується на СУБД PostgreSQL, на рисунку 4.22 наведена її архітектура.

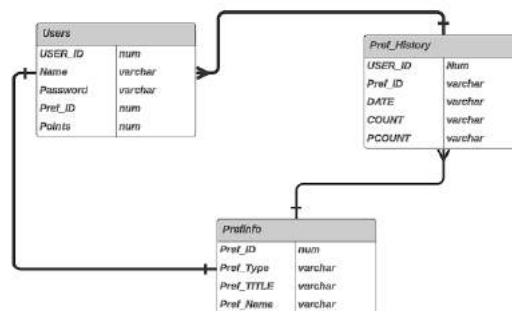


Рисунок 4.22 – Архітектура бази даних користувачів

### 4.3 Програмування веб-сервісу.

Основні етапи розробки веб-сервісу зображені на рисунку 4.23. Першочерговим було налаштування усіх стилів та структури відображення завдяки HTML та CSS, вони не є мовами програмування, а є мовами розмітки та стилів, завдяки їх використанню було визначено положення усіх елементів інтерфейсу (Наведено в першій частині ДОДАТКУ А). Повний лістинг коду знаходиться у ДОДАТКУ А. Результат створення інтерфейсу наведено на рисунку 4.24.

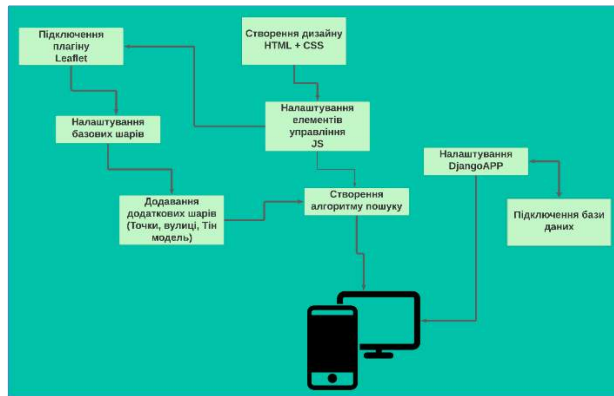


Рисунок 4.23 - Структурна схема розробки власного веб-сервісу

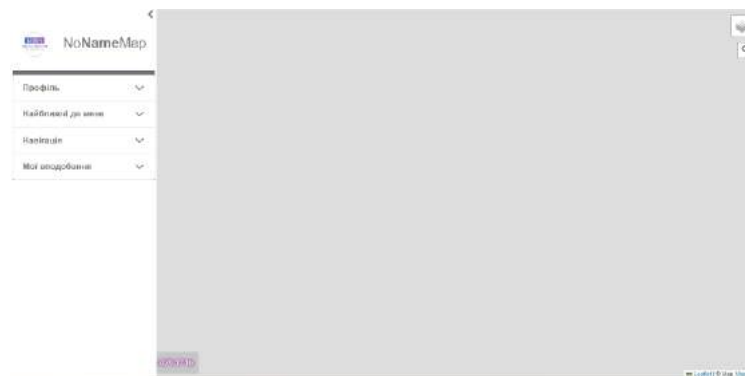


Рисунок 4.24 – Створення інтерфейсу

Далі завдяки плагіну Leaflet було підключені базові та допоміжні шари, після чого були об'єднані в елемент управління overlays. В якості основного базового шару використовується OSM Atlas та точковий шар з обробленими об'єктами, також для покращення візуалізації додано раніше створений шар 3D забудови міста Тернопіль. Код даного фрагменту зображений у лістингу 4.1, а результат наведено на рисунку 4.25 та 4.26.



Рисунок 4.25 – Відображення шарів проекту на веб-сервісі



Рисунок 4.26 – Відображення шарів проекту на веб-сервісі

#### Лістинг 4.1 – Фрагмент коду для додавання шарів до веб сервісу

```

new
L.TileLayer('https://api.mapbox.com/styles/v1/osmbuildings/cjt9gq35s09051fo7urho3m0
f/tiles/256/{z}/{x}/{y}@2x?access_token=pk.eyJ1Ijoib3NtYnVpbGRpbmdzIiwiaSI6IjNldU0t
NDAifQ.c5EU_3V8b87xO24tuWil0w', {
  attribution: '@ Map <a href="https://mapbox.com">Mapbox</a>',
  maxZoom: 18,
  maxNativeZoom: 20
}).addTo(map);

var osmb = new OSMBuildings(map).load();

var osm = L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 25,
  attribution: '&copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
});

var CartoDBLand =
L.tileLayer('https://{s}.basemaps.cartocdn.com/rastertiles/voyager_labels_under/{z}
/{x}/{y}{r}.png', {
  attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors
&copy; <a href="https://carto.com/attributions">CARTO</a>',
  subdomains: 'abcd',

  maxZoom: 20
});

var ThTransp = L.tileLayer('https://{s}.tile.thunderforest.com/transport-
dark/{z}/{x}/{y}.png?apikey=3e8c4c960faa440281865d8aba924b8e', {

```

## Продовження лістингу 4.1

```

attribution: '&copy; <a href="http://www.thunderforest.com/">Thunderforest</a>,
&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors',
  apikey: '<your apikey>',
  maxZoom: 22
});
var ThAtlas = L.tileLayer('https://{s}.tile.thunderforest.com/mobile-
atlas/{z}/{x}/{y}.png?apikey=3e8c4c960faa440281865d8aba924b8e', {
  attribution: '&copy; <a href="http://www.thunderforest.com/">Thunderforest</a>,
&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors',
  apikey: '<your apikey>',
  maxZoom: 22
}).addTo(map);

var Sat=
L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/Map
Server/tile/{z}/{y}/{x}', {
  attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-cubed, USDA, USGS, AEX,
GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community'
});

//helplayers

var streets =
L.tileLayer('https://tiles.stadiamaps.com/tiles/stamen_toner_lines/{z}/{x}/{y}{r}.{
ext}', {
  minZoom: 0,
  maxZoom: 20,
  attribution: '&copy; <a href="https://www.stadiamaps.com/"
target="_blank">Stadia Maps</a> &copy; <a href="https://www.stamen.com/"
target="_blank">Stamen Design</a> &copy; <a href="https://openmaptiles.org/"
target="_blank">OpenMapTiles</a> &copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
  ext: 'png'
});
var labels =
L.tileLayer('https://tiles.stadiamaps.com/tiles/stamen_toner_labels/{z}/{x}/{y}{r}.
{ext}', {
  minZoom: 0,
  maxZoom: 20,
  attribution: '&copy; <a href="https://www.stadiamaps.com/"
target="_blank">Stadia Maps</a> &copy; <a href="https://www.stamen.com/"
target="_blank">Stamen Design</a> &copy; <a href="https://openmaptiles.org/"

```

## Продовження лістингу 4.1

```
target="_blank">OpenMapTiles</a> &copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
  ext: 'png'
});
//BaseMap
var baseMaps = {
  "Атлас": ThAtlas,
  "Land": CartoDBLand,
  "Transport": ThTransp,

  "OSM": osm,
  "Супутник": Sat
};

var helplayers = {
};
//Markers Layer
var overlaysMaps = {
  //helplayers
  "Вулиці": streets,
  "Підписи": labels,
  "3D": osmb,
  //Points
  "Точки": layer_points_0 ,

  "Укриття на зупинках": layer__0,
  "Найпростіше укриття": layer__1,
  "Сховища": layer__2,
  "Протирадіаційні укриття": layer__3,
  "Лікувальні заклади": layer__4,

  //polygons
  //parks
  //places of interest
  //Lindes
  //bikes
};
L.control.layers(baseMaps, overlaysMaps, {collapsed: true}).addTo(map);
```

Наступним кроком було написання основних функцій проекту. Розділ профіль (рис 4.26), містить інформацію про користувача, а саме, його ім'я, скільки він пройшов, кількість запитів, найчастіший запит та кількість очок. Найближчі до мене, поки є основною функцією сервісу. Інтерфейс даної функції продемонстрований на рисунку 4.27.

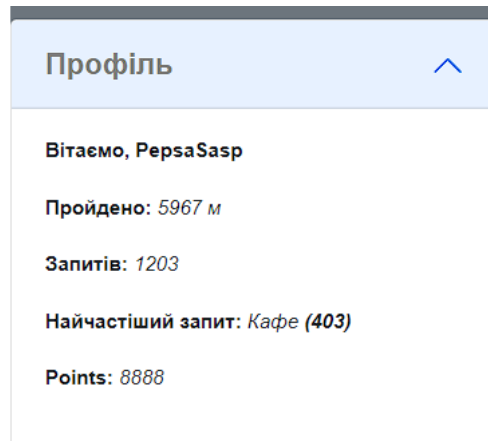


Рисунок 4. 26 – Профіль користувача

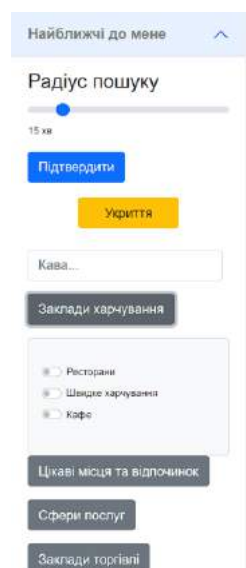


Рисунок 4.27 – Інтерфейс розділу Найближчі до мене

Дана функція спрацьовує після того, як користувач визначиться з категорію і підкатегорією, або ж скористається текстовим пошуком по ключовим словам, далі йому необхідно обрати радіус пошуку, після чого йому прийде

відповідь у вигляді результатів пошуку, які він може відсортувати за двома параметрами, «найближчі» та «за рейтингом». Результат виконання представлений на рисунках 4.28 та 4.30. Даний запит виконується базуючись на інструментах геомаркетингу, а саме аналітичному, будуючи в процесі ізохорни розділяючи 15 хвилинну зону пошуку на суб зони з досяжність в 5, 10 та 15 хвилин, при цьому орієнтуючись на тип закладу який обрав користувач, ізохорни(за 3 видами пересування) за даним прикладом наведені на рисунках 4.32 – 4.34

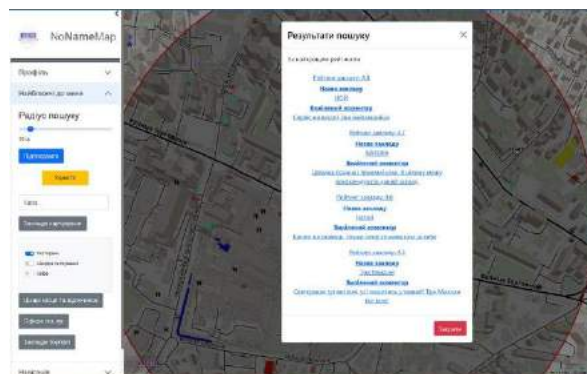


Рисунок 4.28 – Результати пошуку

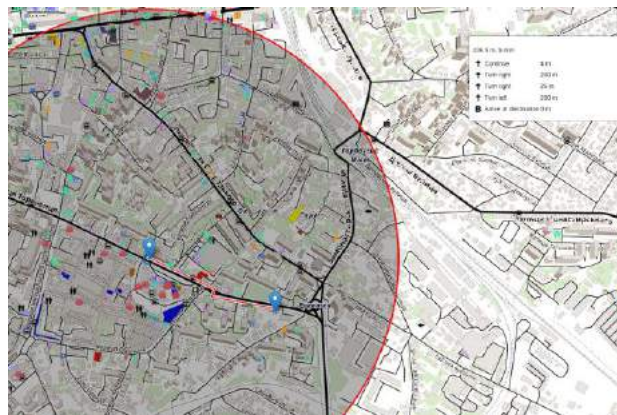


Рисунок 4.29 – Результати пошуку



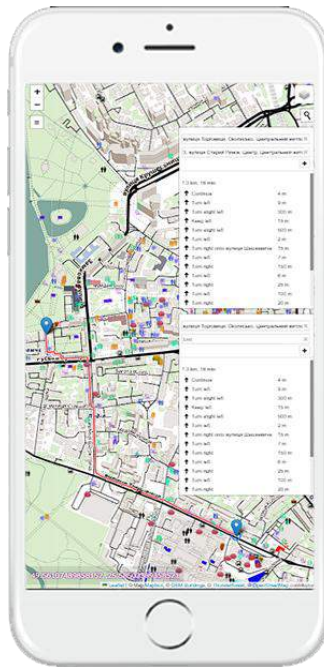


Рисунок 4.30 – Відображення на мобільному пристрої

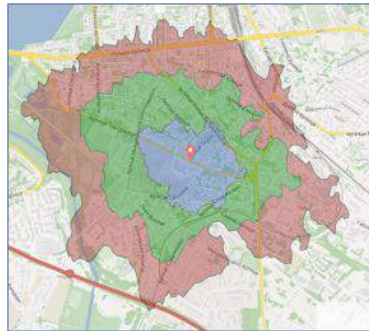


Рисунок 4.31 – Відповідь на запит найближчі до мене, пішки

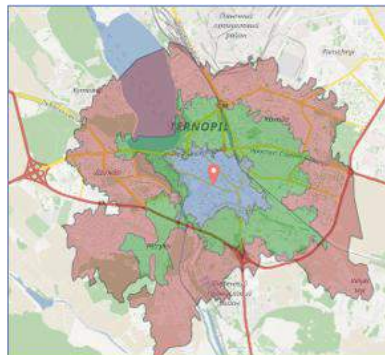


Рисунок 4.32 – Відповідь на запит найближчі до мене, велосипед



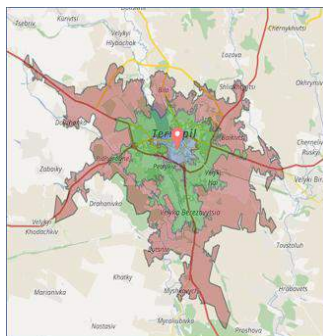


Рисунок 4.33 – Відповідь на запит найближчі до мене, авто

Також однією із ключових функцій, є «Мої вподобання», на даний момент вона працює за принципом сортування виходячи з поточної геопозиції і в пішій доступності 20хв, при виконанні функції Заклади, користувач отримує у відповідь список закладів, де першими стоять ті що він зберіг, а далі на основі його вподобань. Інтерфейс зображено на рисунку 4.33, а результат виконання на 4.34.

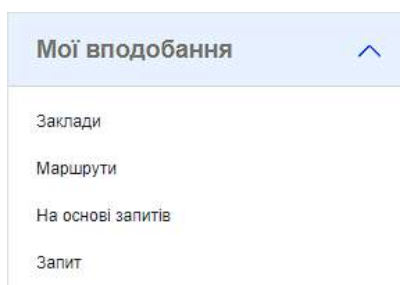


Рисунок 4.33 – Інтерфейс «Мої вподобання»

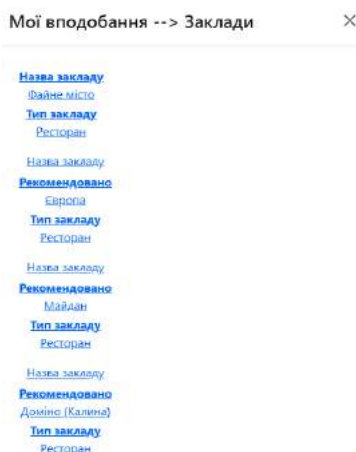


Рисунок 4.34 – Результати виконання функції «Заклади»

## ВИСНОВКИ

На основі проведеного аналізу використання геоінформаційних технологій для вирішення задач геомаркетингу в сфері туризму, було визначено, що розвиток даної галузі залежить від розробки якісних картографічних рішень для задоволення потреб користувачів.

На основі проаналізованих картографічних сервісів та платформ, були визначені основний функціонал для реалізації завдань проєкту, а також необхідні вхідні дані, вони представлені у вигляді: базових карт місцевості, супутникових знімків, а також векторні дані про об'єкти інтересу в межах тестової ділянки (м. Тернопіль) і додаткова інформація про кожний. На підставі отриманих вхідних даних для проєкту, також було обрано середовище для зберігання оброблених даних у вигляді бази даних на СУБД PostgreSQL, де окрім вищезазначеного також зберігаються дані про користувачів. В результаті обробки вхідних даних було отримано картографічну модель м. Тернопіль на основі векторних шарів, а також 3D представлення забудови міста, у вигляді окремого шару на самому веб-сервісі

На підставі аналізу сучасних платформ та мов програмування, був розроблений веб-сервіс з використання геоінформаційних технологій для вирішення задач геомаркетингу в сфері туризму.

Запропонована метод підвищення ефективності при розробці картографічних веб-сервісі покриває актуальний попит в сфері туризму з використанням елементів геомаркетингу і створює потенціал для подальшого розвитку, як галузі, так і можливих майбутніх рішень із застосування нейромереж, а також штучного інтелекту, що зможе оптимізувати усі процеси описані в самій роботі

## ПЕРЕЛІК ПОСИЛАНЬ

1. Геомаркетинг. *Urbistat - Geomarketing & Market Research*. URL: <https://www.urbistat.com/geomarketing-ukraine/#1463494027532-630255cf-4a9d> (дата звернення: 18.11.2023).
2. ТУРИЗМ | Тернопільська область. *Тернопільщина*. URL: <https://irp.te.ua/turyzm-ternopilska-oblast/> (дата звернення: 01.10.2023).
3. У яких містах світу туристи витрачають найбільше грошей. *Слово і Діло*. URL: <https://www.slovoidilo.ua/2023/05/24/infografika/svit/yakyx-mistax-svitu-turysty-vytrachayut-najbilshe-hroshej> (дата звернення: 27.12.2024).
4. Як відновлюється український туризм у 2023 році?. «Дебет-Кредит» - *Бухгалтерські новини*. URL: <https://news.dtk.ua/society/community/85439-iaak-vidnovliujetsia-ukrayinskii-turizm-u-2023-roci> (дата звернення: 03.01.2024).
5. 5 Best Web Mapping Platforms - The Battle of Web GIS - GIS Geography. *GIS Geography*. URL: <https://gisgeography.com/web-mapping/> (date of access: 01.01.2024).
6. Geomarketing 101: Tools, Examples & More | GroundTruth. *GroundTruth*. URL: <https://www.groundtruth.com/insight/geomarketing/> (date of access: 18.10.2024).
7. HERE, TomTom Lead Location Platform Effectiveness Rankings. *Counterpoint - Technology Market Research & Industry Analysis Firm*. URL: <https://www.counterpointresearch.com/insights/here-tomtom-lead-location-platform-effectiveness-rankings-2023/> (date of access: 12.10.2023).
8. What is Geomarketing?. *Techfunnel*. URL: <https://www.techfunnel.com/martech/geomarketing/> (date of access: 26.10.2024).
9. Albuquerque H., Costa C., Martins F. The use of Geographical Information Systems for Tourism Marketing purposes in Aveiro region (Portugal) // *Tourism Management Perspectives*. 2018. Vol.26. Pp. 172-178. DOI: 10.1016/j.tmp.2017.10.009.
10. Open Source - GraphHopper Directions API. GraphHopper Directions API. URL: <https://www.graphhopper.com/open-source/> (date of access: 17.01.2024).
11. pgRouting Manuals. pgRouting Manuals. URL: <https://docs.pgrouting.org/> (date of access: 17.01.2024).
12. PostGIS Manuals. pgRouting Manuals. URL: <https://postgis.net/> (date of access: 12.01.2024).
13. PostGIS Manuals. pgRouting Manuals. URL: <https://postgis.net/> (date of access: 12.01.2024).

14. WEB GIS URL: <https://www.slideshare.net/simhodi/web-gis-and-web-mappptx> (date of access: 12.01.2024).
15. RETAIL AND GEO-MARKETING URL: <https://eos.com/industries/retail-and-geo-marketing/> (date of access: 10.01.2024).
16. Web GIS using Google Map and QGIS URL: <https://www.slideshare.net/IndranilChakraborty42/web-gis-using-google-map-and-qgis> (date of access: 05.01.2024).
17. Leaflet URL: <https://leafletjs.com/> (date of access: (23.09.2023)).
18. JavaScript URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (date of access: (21.09.2023)).
19. Інтернет-тенденції 2019. Статистика та факти в усьому світі URL [uk.vpnmentor.com/](http://uk.vpnmentor.com/) (date of access: (20.09.2023)).
20. Веб-додатки електронних карт URL [https://geoknigi.com/book\\_view.php?id=1466](https://geoknigi.com/book_view.php?id=1466) (date of access: (27.09.2023)).

## ДОДАТОК А

### ЛІСТИНГ ПРОГРАМНОГО КОДУ

```

<!--html -->
<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD65VohhpuuCOmLASjC"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.8.0/dist/leaflet.css"
integrity="sha512-
hoalWLoI8r4UszCkZ5kL8vay0GVae1oxXe/2A4A06J9+580uKHD03JdHb7NzwwzK5xr/Fs0W40kiNHxM9vy
TtQ==" crossorigin="" />
  <!--<link href="https://cdn.osmbuildings.org/4.1.1/OSMBuildings.css"
rel="stylesheet">-->

  <!--<script src="https://cdn.osmbuildings.org/classic/0.2.2b/OSMBuildings-
Leaflet.js"></script> -->

  <script src="https://unpkg.com/leaflet@1.8.0/dist/leaflet.js"
integrity="sha512-
BB3hKbKW0c9Ez/TAwyWxNXeoV9c1v6FIEyIBieIWkplJauysF18Nzgr1MBNBXf8/KABdlkX68nAh1wcDFLG
PCQ=="
crossorigin=""></script>
  <link rel="stylesheet" href="https://unpkg.com/leaflet-routing-
machine@3.2.12/dist/leaflet-routing-machine.css" />
  <script src="https://unpkg.com/leaflet-routing-machine@3.2.12/dist/leaflet-
routing-machine.js"></script>
  <script src="lrm-graphhopper.js"></script>
  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet" href="styles/L.Control.SlideMenu.css">
  <script src="js/L.Control.SlideMenu.js"></script>
  <link rel="stylesheet" href="styles/style.css">
  <link rel="stylesheet" href="https://unpkg.com/leaflet-control-
geocoder/dist/Control.Geocoder.css" />
  <script src="https://unpkg.com/leaflet-control-
geocoder/dist/Control.Geocoder.js"></script>

```

```

    <link rel="stylesheet" href="https://unpkg.com/leaflet-control-geocoder/dist/Control.Geocoder.css" />
    <script src="https://unpkg.com/leaflet-control-geocoder/dist/Control.Geocoder.js"></script>
    <script src="js/Autolinker.min.js"></script>
    <script src="data/points_0.js"></script>
    <script src="data/_0.js"></script>
    <script src="data/_1.js"></script>
    <script src="data/_2.js"></script>
    <script src="data/_3.js"></script>
    <script src="data/_4.js"></script>
    <script src="data/FastFood_0.js"></script>
    <script src="data/Cafe_2.js"></script>
    <script src="https://cdn.osmbuildings.org/classic/0.2.2b/OSMBuildings-Leaflet.js"></script>

    <title>Document</title>
</head>
<body>
    <div id="map">
        <div class="leaflet-control coords"></div>
        <div class="leaflet-control search"></div>
        <div class="content bg-secondary ">
            <div class="accordion" id="accordionExample">
                <div class="accordion-item">
                    <h2 class="accordion-header" id="headingOne">
                        <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne">
                            <div class="title">Профіль</div>
                        </button>
                    </h2>
                    <div id="collapseOne" class="accordion-collapse collapse show" aria-labelledby="headingOne" data-bs-parent="#accordionExample">
                        <div class="accordion-body">
                            <p><strong> Вітаємо, PepsaSasp</strong></p>
                            <p><strong>Пройдено: </strong> <em>5967 м</em></p>
                            <p><strong>Запитів: </strong> <em>1203 </em></p>
                            <p><strong>Найчастіший запит: </strong> <em> Кафе <b>(403)</b></em></p>
                            <p><strong>Points: </strong> <em>8888</em></p>
                        </div>
                    </div>
                </div>
                <div class="accordion-item">
                    <h2 class="accordion-header" id="headingTwo">
                        <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">
                            <div class="title">Найближчі до мене</div>
                        </button>
                    </h2>
                </div>
            </div>
        </div>
    </div>

```

```

</h2>

<div id="collapseTwo" class="accordion-collapse collapse" aria-
labelledby="headingTwo" data-bs-parent="#accordionExample">
  <div class="accordion-body">

    <form oninput="output.value = Math.round(range.valueAsNumber / 1000)">
    <h4>
    Радіус пошуку
    </h4>
    <div class="range">

      <input class="form-range" name="range" type="range" min="20"
max="100000" step="5000" value ="20000">
      <output id="myrange" name="output" for="range">
      20
      </output> хв

    </div>
  </form>
  <p id="plist"></p>
  <div>
  <button onclick="searchrange()" class="btn btn-
primary">Підтвердити</button>
  </div>
  <br>

  <div class="d-grid gap-2 col-6 mx-auto">
  <button type="button" class="btn btn-warning">Укриття</button>
  <br>
  </div>

  <p><form class="d-flex">
  <input class="form-control me-2" type="search" placeholder="Кава..."
aria-label="Search">

  </form></p>
  <p>
  <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c1" aria-expanded="false" aria-controls="c1">
  Заклади харчування
  </button>

  </p>
  <div class="collapse" id="c1">
  <div class="card card-body bg-light">
  <p>

```

```

        <div class="form-check form-switch">
            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                <label class="form-check-label"
for="flexSwitchCheckDefault">Ресторани</label>
            </div>
            <div class="form-check form-switch">
                <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Швидке харчування</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                        <label class="form-check-label"
for="flexSwitchCheckDefault">Кафе</label>
                    </div>
            </p>
        </div>
    </div>
    <p>
        <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c2" aria-expanded="false" aria-controls="c2">
            Цікаві місця та відпочинок
        </button>
    </p>
    <div class="collapse" id="c2">
        <div class="card card-body bg-light">
            <p>
                <div class="form-check form-switch">
                    <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                        <label class="form-check-label"
for="flexSwitchCheckDefault">Парки</label>
                    </div>
                    <div class="form-check form-switch">
                        <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                            <label class="form-check-label"
for="flexSwitchCheckDefault">Туристичні пам'ятки</label>
                        </div>
                        <div class="form-check form-switch">
                            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                                <label class="form-check-label"
for="flexSwitchCheckDefault">Музеї</label>
                            </div>
            </p>
        </div>
    </div>

```



```

        </div>
        <p>
            <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c3" aria-expanded="false" aria-controls="c3">
                Сфери послуг
            </button>
        </p>
        <div class="collapse" id="c3">
            <div class="card card-body bg-light">
                <p>
                    <div class="form-check form-switch">
                        <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                            <label class="form-check-label"
for="flexSwitchCheckDefault">АЗС</label>
                        </div>
                        <div class="form-check form-switch">
                            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                                <label class="form-check-label"
for="flexSwitchCheckDefault">Зарядні станції</label>
                            </div>
                        <div class="form-check form-switch">
                            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                                <label class="form-check-label"
for="flexSwitchCheckDefault">Банки</label>
                            </div>
                        <div class="form-check form-switch">
                            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                                <label class="form-check-label"
for="flexSwitchCheckDefault">Ремонтні сервіси</label>
                            </div>
                        <div class="form-check form-switch">
                            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                                <label class="form-check-label"
for="flexSwitchCheckDefault">Паркування</label>
                            </div>
                        <div class="form-check form-switch">
                            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                                <label class="form-check-label"
for="flexSwitchCheckDefault">Медичні Заклади</label>
                            </div>
                    </p>
                </div>
            </div>
        </p>
    </div>

```

```

        <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c4" aria-expanded="false" aria-controls="c4">
        Заклади торгівлі
    </button>
</p>
    <div class="collapse" id="c4">
        <div class="card card-body bg-light">
            <p>
                <div class="form-check form-switch">
                    <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Торгові центри</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Продуктові магазини</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Непродовольчі товари</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Аптеки</label>
                </div>
            </p>
        </div>
    </div>

</div>
</div>
</div>
<div class="accordion-item">
    <h2 class="accordion-header" id="headingThree">
        <button class="accordion-button collapsed" type="button" data-bs-
toggle="collapse" data-bs-target="#collapseThree" aria-expanded="false" aria-
controls="collapseThree">
        <div class="title">Навігація</div>
    </button>
    </h2>
    <div id="collapseThree" class="accordion-collapse collapse" aria-
labelledby="headingThree" data-bs-parent="#accordionExample">
        <div class="accordion-body">

```

```

        <div class="btn-group" role="group" aria-label="Basic radio toggle
button group">
        <input type="radio" class="btn-check" name="btnradio" id="btnradio1"
autocomplete="off" checked>
        <label class="btn btn-outline-primary" for="btnradio1">Пішки</label>

        <input type="radio" class="btn-check" name="btnradio" id="btnradio2"
autocomplete="off">
        <label class="btn btn-outline-primary" for="btnradio2">Авто</label>

        <input type="radio" class="btn-check" name="btnradio" id="btnradio3"
autocomplete="off">
        <label class="btn btn-outline-primary"
for="btnradio3">Велосипед</label>
        <br>
        </div>
        <div>
        <br>
        <button id="Route" type="button" class="btn btn-secondary">Увімкнути
навігацію</button>
        </div>
        </div>
        </div>
        </div>
    </div>
    <div class="accordion-item">
        <h2 class="accordion-header" id="headingFour">
        <button class="accordion-button collapsed" type="button" data-bs-
toggle="collapse" data-bs-target="#collapseFour" aria-expanded="false" aria-
controls="collapseFour">
        <div class="title">Мої вподобання</div>

        </button>
        </h2>
        <div id="collapseFour" class="accordion-collapse collapse" aria-
labelledby="headingFour" data-bs-parent="#accordionExample">
        <div class="accordion-body">
            <p>Заклади</p>
            <p>Маршрути</p>
            <p>На основі запитів</p>
            <p>Запит</p>
        </div>
        </div>
    </div>

    </div>

    </div>
    <div class="modal" id="myModal" tabindex="-1">
        <div class="modal-dialog">

```

```

    <div class="modal-content">

        <!-- Заголовок модального вікна -->
        <div class="modal-header">
            <h4 class="modal-title">Результати пошуку</h4>
            <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
        </div>

        <!-- Тіло модального вікна -->
        <div class="modal-body">
            <div id="popup-list"></div>
        </div>

        <!-- Кнопка закриття модального вікна -->
        <div class="modal-footer">
            <button type="button" class="btn btn-danger"
onclick="closeModal()">Закрити</button>
        </div>

    </div>
</div>
</div>

<script src="mapapp.js"></script>
<script src="js/main.js"></script>
<script src="js/Points.js"></script>

</body>
</html>

//JS
//Options
var mapOptions = {
    center: [49.54466784075138, 25.596443337497583], //Ukraine
    [49.713220318302824, 30.631331491840474],
    zoom: 17
};

//Maps
var map = L.map('map', mapOptions);

new
L.TileLayer('https://api.mapbox.com/styles/v1/osmbuildings/cjt9gq35s09051fo7urho3m0
f/tiles/256/{z}/{x}/{y}@2x?access_token=pk.eyJ1Ijoib3NtYnVpbGRpbmdzIiwiaSI6IjNldU0t
NDAifQ.c5EU_3V8b87xO24tuWil0w', {

```

```

    attribution: '© Map <a href="https://mapbox.com">Mapbox</a>',
    maxZoom: 18,
    maxNativeZoom: 20
  }).addTo(map);

var osmb = new OSMBuildings(map).load();

var osm = L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 25,
  attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
});

var CartoDBLand =
L.tileLayer('https://{s}.basemaps.cartocdn.com/rastertiles/voyager_labels_under/{z}/{x}/{y}{r}.png', {
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors &copy; <a href="https://carto.com/attributions">CARTO</a>',
  subdomains: 'abcd',
  maxZoom: 20
});

var ThTransp = L.tileLayer('https://{s}.tile.thunderforest.com/transport-dark/{z}/{x}/{y}.png?apikey=3e8c4c960faa440281865d8aba924b8e', {
  attribution: '&copy; <a href="http://www.thunderforest.com/">Thunderforest</a>, &copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
  apikey: '<your apikey>',
  maxZoom: 22
});

var ThAtlas = L.tileLayer('https://{s}.tile.thunderforest.com/mobile-atlas/{z}/{x}/{y}.png?apikey=3e8c4c960faa440281865d8aba924b8e', {
  attribution: '&copy; <a href="http://www.thunderforest.com/">Thunderforest</a>, &copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
  apikey: '<your apikey>',
  maxZoom: 22
}).addTo(map);

var Sat=
L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}', {
  attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community'
});

```

```

//helplayers
var streets =
L.tileLayer('https://tiles.stadiamaps.com/tiles/stamen_toner_lines/{z}/{x}/{y}{r}.{
ext}', {
  minZoom: 0,
  maxZoom: 20,
  attribution: '&copy; <a href="https://www.stadiamaps.com/"
target="_blank">Stadia Maps</a> &copy; <a href="https://www.stamen.com/"
target="_blank">Stamen Design</a> &copy; <a href="https://openmaptiles.org/"
target="_blank">OpenMapTiles</a> &copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
  ext: 'png'
});
var labels =
L.tileLayer('https://tiles.stadiamaps.com/tiles/stamen_toner_labels/{z}/{x}/{y}{r}.
{ext}', {
  minZoom: 0,
  maxZoom: 20,
  attribution: '&copy; <a href="https://www.stadiamaps.com/"
target="_blank">Stadia Maps</a> &copy; <a href="https://www.stamen.com/"
target="_blank">Stamen Design</a> &copy; <a href="https://openmaptiles.org/"
target="_blank">OpenMapTiles</a> &copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors',
  ext: 'png'
});

//Markers
var marker1 = L.marker([4.55076437159158, 5.586010603696423]).addTo(map);
var autolinker = new Autolinker({truncate: {length: 30, location: 'smart'}});
function pop_points_0(feature, layer) {
  var popupContent = '<table>\
    <tr>\
      <td colspan="2"></td>\
    </tr>\
    <tr>\
      <td colspan="2">' + (feature.properties['timestamp'] !== null ?
autolinker.link(feature.properties['timestamp'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong></strong>' + (feature.properties['name']
!== null ? autolinker.link(feature.properties['name'].toLocaleString()) : '') +
'</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Тип</strong><br />' +
(feature.properties['type'] !== null ?
autolinker.link(feature.properties['type'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"></td>\

```

```

        </tr>\
    </table>';
    layer.bindPopup(popupContent, {maxHeight: 400});
}

function style_points_0_0(feature) {
switch(String(feature.properties['type'])) {
    case 'arts_centre':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
            imageUrl: 'markers/points_0_artscentre0.png',
            iconSize: [15.2, 15.2]
            })),
            interactive: true,
        }
        break;
    case 'artwork':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
            imageUrl: 'markers/points_0_artwork1.png',
            iconSize: [15.2, 15.2]
            })),
            interactive: true,
        }
        break;
    case 'atm':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
            imageUrl: 'markers/points_0_atm2.png',
            iconSize: [15.2, 15.2]
            })),
            interactive: true,
        }
        break;
    case 'attraction':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
            imageUrl: 'markers/points_0_attraction3.png',
            iconSize: [15.2, 15.2]

```

```

}),
  interactive: true,
}
    break;
    case 'bank':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_bank4.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}
    break;
    case 'bar':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_bar5.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}
    break;
    case 'bicycle_parking':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_bicycleparking6.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}
    break;
    case 'bicycle_rental':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_bicyclerental7.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}
}

```



```

        break;
    case 'bus_stop':
        return {
            pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_busstop8.png',
iconSize: [15.2, 15.2]
}),
            interactive: true,
        }
        break;
    case 'cafe':
        return {
            pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_cafe9.png',
iconSize: [15.2, 15.2]
}),
            interactive: true,
        }
        break;
    case 'car_rental':
        return {
            pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_carrental10.png',
iconSize: [15.2, 15.2]
}),
            interactive: true,
        }
        break;
    case 'car_wash':
        return {
            pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_carwash11.png',
iconSize: [26.599999999999998, 26.599999999999998]
}),
            interactive: true,
        }
        break;
    case 'charging_station':
        return {

```

```

    pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_chargingstation12.png',
iconSize: [22.79999999999997, 22.79999999999997]
}),
    interactive: true,
}

    break;
    case 'cinema':
        return {
            pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_cinema13.png',
iconSize: [15.2, 15.2]
}),
            interactive: true,
        }

        break;
        case 'clinic':
            return {
                pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_clinic14.png',
iconSize: [15.2, 15.2]
}),
                interactive: true,
            }

            break;
            case 'device_charging_':
                return {
                    pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_devicecharging15.png',
iconSize: [26.59999999999998, 26.59999999999998]
}),
                    interactive: true,
                }

                break;
                case 'fast_food':
                    return {
                        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',

```

```
icon: L.icon({
  imageUrl: 'markers/points_0_fastfood16.png',
  iconSize: [15.2, 15.2]
}),
  interactive: true,
}
    break;
  case 'food_count':
    return {
      pane: 'pane_points_0',
      rotationAngle: 0.0,
      rotationOrigin: 'center center',
      icon: L.icon({
        imageUrl: 'markers/points_0_foodcourt17.png',
        iconSize: [15.2, 15.2]
      }),
      interactive: true,
    }
    break;
  case 'fuel':
    return {
      pane: 'pane_points_0',
      rotationAngle: 0.0,
      rotationOrigin: 'center center',
      icon: L.icon({
        imageUrl: 'markers/points_0_fuel18.png',
        iconSize: [15.2, 15.2]
      }),
      interactive: true,
    }
    break;
  case 'hospital':
    return {
      pane: 'pane_points_0',
      rotationAngle: 0.0,
      rotationOrigin: 'center center',
      icon: L.icon({
        imageUrl: 'markers/points_0_hospital19.png',
        iconSize: [15.2, 15.2]
      }),
      interactive: true,
    }
    break;
  case 'hostel':
    return {
      pane: 'pane_points_0',
      rotationAngle: 0.0,
      rotationOrigin: 'center center',
      icon: L.icon({
        imageUrl: 'markers/points_0_hostel20.png',
        iconSize: [15.2, 15.2]
      })
    }
  }
}
```

```

}),
  interactive: true,
}
    break;
    case 'hotel':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_hotel121.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}
    break;
    case 'internet_cafe':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_internetcafe22.png',
iconSize: [22.799999999999997, 22.799999999999997]
}),
    interactive: true,
}
    break;
    case 'marketplace':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_marketplace23.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}
    break;
    case 'memorial':
      return {
        pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_memorial24.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}
}

```

```
        break;
    case 'monument':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
                imageUrl: 'markers/points_0_monument25.png',
                iconSize: [15.2, 15.2]
            }),
            interactive: true,
        }
        break;
    case 'motel':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
                imageUrl: 'markers/points_0_motel26.png',
                iconSize: [15.2, 15.2]
            }),
            interactive: true,
        }
        break;
    case 'museum':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
                imageUrl: 'markers/points_0_museum27.png',
                iconSize: [15.2, 15.2]
            }),
            interactive: true,
        }
        break;
    case 'nightclub':
        return {
            pane: 'pane_points_0',
            rotationAngle: 0.0,
            rotationOrigin: 'center center',
            icon: L.icon({
                imageUrl: 'markers/points_0_nightclub28.png',
                iconSize: [15.2, 15.2]
            }),
            interactive: true,
        }
        break;
    case 'parking':
        return {
```

```
    pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_parking29.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}

    break;
    case 'post_office':
    return {
    pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_postoffice30.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}

    break;
    case 'pub':
    return {
    pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_pub31.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}

    break;
    case 'rest_area':
    return {
    pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
iconUrl: 'markers/points_0_restarea32.png',
iconSize: [15.2, 15.2]
}),
    interactive: true,
}

    break;
    case 'restaurant':
    return {
    pane: 'pane_points_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
```

```

icon: L.icon({
  iconUrl: 'markers/points_0_restaurant33.png',
  iconSize: [15.2, 15.2]
}),
  interactive: true,
}
    break;
  case 'toilets':
    return {
      pane: 'pane_points_0',
      rotationAngle: 0.0,
      rotationOrigin: 'center center',
      icon: L.icon({
        iconUrl: 'markers/points_0.svg',
        iconSize: [15.2, 15.2]
      }),
      interactive: true,
    }
    break;
}
}
map.createPane('pane_points_0');
map.getPane('pane_points_0').style.zIndex = 400;
map.getPane('pane_points_0').style['mix-blend-mode'] = 'normal';
var layer_points_0 = new L.geoJson(json_points_0, {
  attribution: '',
  interactive: true,
  dataVar: 'json_points_0',
  layerName: 'layer_points_0',
  pane: 'pane_points_0',
  onEachFeature: pop_points_0,
  pointToLayer: function (feature, latlng) {
    var context = {
      feature: feature,
      variables: {}
    };
    return L.marker(latlng, style_points_0_0(feature));
  },
}).addTo(map);
//////////
function pop__0(feature, layer) {
  var popupContent = '<table>\
    <tr>\
      <td colspan="2"><strong>Name</strong><br />' +
(feature.properties['Name'] !== null ?
autolinker.link(feature.properties['Name'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Address</strong><br />' +
(feature.properties['Address'] !== null ?
autolinker.link(feature.properties['Address'].toLocaleString()) : '') + '</td>\

```

```

        </tr>\
        <tr>\
            <td colspan="2"><strong>Capacity</strong><br />' +
(feature.properties['Capacity'] !== null ?
autolinker.link(feature.properties['Capacity'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>WIFI</strong><br />' +
(feature.properties['WIFI'] !== null ?
autolinker.link(feature.properties['WIFI'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Санвузол</strong><br />' +
(feature.properties['Санвузол'] !== null ?
autolinker.link(feature.properties['Санвузол'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>OBJECTID</strong><br />' +
(feature.properties['OBJECTID'] !== null ?
autolinker.link(feature.properties['OBJECTID'].toLocaleString()) : '') + '</td>\
        </tr>\
    </table>';
    layer.bindPopup(popupContent, {maxHeight: 400});
}

function style__0_0() {
    return {
        pane: 'pane__0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
    iconUrl: 'markers/_0.svg',
    iconSize: [15.2, 15.2]
}),
        interactive: true,
    }
}
map.createPane('pane__0');
map.getPane('pane__0').style.zIndex = 400;
map.getPane('pane__0').style['mix-blend-mode'] = 'normal';
var layer__0 = new L.geoJson(json__0, {
    attribution: '',
    interactive: true,
    dataVar: 'json__0',
    layerName: 'layer__0',
    pane: 'pane__0',
    onEachFeature: pop__0,
    pointToLayer: function (feature, latlng) {
        var context = {
            feature: feature,
            variables: {}

```



```

    };
    return L.marker(latlng, style__0_0(feature));
  },
});

function pop__1(feature, layer) {
  var popupContent = '<table>\
    <tr>\
      <td colspan="2"><strong>Name</strong><br />' +
(feature.properties['Name'] !== null ?
autolinker.link(feature.properties['Name'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Address</strong><br />' +
(feature.properties['Address'] !== null ?
autolinker.link(feature.properties['Address'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Name_organization</strong><br />' +
(feature.properties['Name_organization'] !== null ?
autolinker.link(feature.properties['Name_organization'].toLocaleString()) : '') +
'</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Capacity</strong><br />' +
(feature.properties['Capacity'] !== null ?
autolinker.link(feature.properties['Capacity'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Notes</strong><br />' +
(feature.properties['Notes'] !== null ?
autolinker.link(feature.properties['Notes'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Wifi</strong><br />' +
(feature.properties['Wifi'] !== null ?
autolinker.link(feature.properties['Wifi'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Санвузол</strong><br />' +
(feature.properties['Санвузол'] !== null ?
autolinker.link(feature.properties['Санвузол'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>OBJECTID</strong><br />' +
(feature.properties['OBJECTID'] !== null ?
autolinker.link(feature.properties['OBJECTID'].toLocaleString()) : '') + '</td>\
    </tr>\
  </table>';
  layer.bindPopup(popupContent, {maxHeight: 400});
}

```

```

function style__1_0() {
  return {
    pane: 'pane__1',
  rotationAngle: 0.0,
  rotationOrigin: 'center center',
  icon: L.icon({
    iconUrl: 'markers/_1.svg',
    iconSize: [15.2, 15.2]
  }),
    interactive: true,
  }
}
map.createPane('pane__1');
map.getPane('pane__1').style.zIndex = 401;
map.getPane('pane__1').style['mix-blend-mode'] = 'normal';
var layer__1 = new L.geoJson(json__1, {
  attribution: '',
  interactive: true,
  dataVar: 'json__1',
  layerName: 'layer__1',
  pane: 'pane__1',
  onEachFeature: pop__1,
  pointToLayer: function (feature, latlng) {
    var context = {
      feature: feature,
      variables: {}
    };
    return L.marker(latlng, style__1_0(feature));
  },
});
function pop__2(feature, layer) {
  var popupContent = '<table>\
    <tr>\
      <td colspan="2"><strong>Name</strong><br />' +
(feature.properties['Name'] !== null ?
autolinker.link(feature.properties['Name'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Address</strong><br />' +
(feature.properties['Address'] !== null ?
autolinker.link(feature.properties['Address'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>Name_organization</strong><br />' +
(feature.properties['Name_organization'] !== null ?
autolinker.link(feature.properties['Name_organization'].toLocaleString()) : '') +
'</td>\
    </tr>\
  </tr>\
</table>';
}

```

```

        <td colspan="2"><strong>Capacity</strong><br />' +
(feature.properties['Capacity'] !== null ?
autolinker.link(feature.properties['Capacity'].toLocaleString()) : '') + '</td>\
</tr>\
<tr>\
        <td colspan="2"><strong>Notes</strong><br />' +
(feature.properties['Notes'] !== null ?
autolinker.link(feature.properties['Notes'].toLocaleString()) : '') + '</td>\
</tr>\
<tr>\
        <td colspan="2"><strong>WiFi</strong><br />' +
(feature.properties['WiFi'] !== null ?
autolinker.link(feature.properties['WiFi'].toLocaleString()) : '') + '</td>\
</tr>\
<tr>\
        <td colspan="2"><strong>Санвузол</strong><br />' +
(feature.properties['Санвузол'] !== null ?
autolinker.link(feature.properties['Санвузол'].toLocaleString()) : '') + '</td>\
</tr>\
<tr>\
        <td colspan="2"><strong>OBJECTID</strong><br />' +
(feature.properties['OBJECTID'] !== null ?
autolinker.link(feature.properties['OBJECTID'].toLocaleString()) : '') + '</td>\
</tr>\
</table>';
    layer.bindPopup(popupContent, {maxHeight: 400});
}

function style__2_0() {
    return {
        pane: 'pane__2',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
    iconUrl: 'markers/_2.svg',
    iconSize: [15.2, 15.2]
}),
        interactive: true,
    }
}
map.createPane('pane__2');
map.getPane('pane__2').style.zIndex = 402;
map.getPane('pane__2').style['mix-blend-mode'] = 'normal';
var layer__2 = new L.geoJson(json__2, {
    attribution: '',
    interactive: true,
    dataVar: 'json__2',
    layerName: 'layer__2',
    pane: 'pane__2',
    onEachFeature: pop__2,
    pointToLayer: function (feature, latlng) {

```

```

    var context = {
        feature: feature,
        variables: {}
    };
    return L.marker(latlng, style__2_0(feature));
},
});

function pop__3(feature, layer) {
    var popupContent = '<table>\
        <tr>\
            <td colspan="2"><strong>Name</strong><br />' +
(feature.properties['Name'] !== null ?
autolinker.link(feature.properties['Name'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Address</strong><br />' +
(feature.properties['Address'] !== null ?
autolinker.link(feature.properties['Address'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Name_organization</strong><br />' +
(feature.properties['Name_organization'] !== null ?
autolinker.link(feature.properties['Name_organization'].toLocaleString()) : '') +
'</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Capacity</strong><br />' +
(feature.properties['Capacity'] !== null ?
autolinker.link(feature.properties['Capacity'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Notes</strong><br />' +
(feature.properties['Notes'] !== null ?
autolinker.link(feature.properties['Notes'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Wifi</strong><br />' +
(feature.properties['Wifi'] !== null ?
autolinker.link(feature.properties['Wifi'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Санвузол</strong><br />' +
(feature.properties['Санвузол'] !== null ?
autolinker.link(feature.properties['Санвузол'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>OBJECTID</strong><br />' +
(feature.properties['OBJECTID'] !== null ?
autolinker.link(feature.properties['OBJECTID'].toLocaleString()) : '') + '</td>\
        </tr>\
    </table>';
}

```

```

        </table>';
    layer.bindPopup(popupContent, {maxHeight: 400});
}

function style__3_0() {
    return {
        pane: 'pane__3',
        rotationAngle: 0.0,
        rotationOrigin: 'center center',
        icon: L.icon({
            iconUrl: 'markers/_3.svg',
            iconSize: [15.2, 15.2]
        }),
        interactive: true,
    }
}

map.createPane('pane__3');
map.getPane('pane__3').style.zIndex = 403;
map.getPane('pane__3').style['mix-blend-mode'] = 'normal';
var layer__3 = new L.geoJson(json__3, {
    attribution: '',
    interactive: true,
    dataVar: 'json__3',
    layerName: 'layer__3',
    pane: 'pane__3',
    onEachFeature: pop__3,
    pointToLayer: function (feature, latlng) {
        var context = {
            feature: feature,
            variables: {}
        };
        return L.marker(latlng, style__3_0(feature));
    },
});

function pop__4(feature, layer) {
    var popupContent = '<table>\
        <tr>\
            <td colspan="2"><strong>Name</strong><br />' +
(feature.properties['Name'] !== null ?
autolinker.link(feature.properties['Name'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Address</strong><br />' +
(feature.properties['Address'] !== null ?
autolinker.link(feature.properties['Address'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>Name_organization</strong><br />' +
(feature.properties['Name_organization'] !== null ?

```

```

autolinker.link(feature.properties['Name_organization'].toLocaleString()) : '') +
'</td>\
    </tr>\
    <tr>\
        <td colspan="2"><strong>Capacity</strong><br />' +
(feature.properties['Capacity'] !== null ?
autolinker.link(feature.properties['Capacity'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
        <td colspan="2"><strong>Notes</strong><br />' +
(feature.properties['Notes'] !== null ?
autolinker.link(feature.properties['Notes'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
        <td colspan="2"><strong>WIFI</strong><br />' +
(feature.properties['WIFI'] !== null ?
autolinker.link(feature.properties['WIFI'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
        <td colspan="2"><strong>Санвузол</strong><br />' +
(feature.properties['Санвузол'] !== null ?
autolinker.link(feature.properties['Санвузол'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
        <td colspan="2"><strong>OBJECTID</strong><br />' +
(feature.properties['OBJECTID'] !== null ?
autolinker.link(feature.properties['OBJECTID'].toLocaleString()) : '') + '</td>\
    </tr>\
</table>';
    layer.bindPopup(popupContent, {maxHeight: 400});
}

function style__4_0() {
    return {
        pane: 'pane__4',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
    iconUrl: 'markers/_4.svg',
    iconSize: [15.2, 15.2]
}),
        interactive: true,
    }
}
map.createPane('pane__4');
map.getPane('pane__4').style.zIndex = 404;
map.getPane('pane__4').style['mix-blend-mode'] = 'normal';
var layer__4 = new L.geoJson(json__4, {
    attribution: '',
    interactive: true,
    dataVar: 'json__4',

```

```

layerName: 'layer__4',
pane: 'pane__4',
onEachFeature: pop__4,
pointToLayer: function (feature, latlng) {
    var context = {
        feature: feature,
        variables: {}
    };
    return L.marker(latlng, style__4_0(feature));
},
});

////Persuk
function pop_FastFood_0(feature, layer) {
    var popupContent = '<table>\
        <tr>\
            <td colspan="2">' + (feature.properties['osm_id'] !== null ?
autolinker.link(feature.properties['osm_id'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2">' + (feature.properties['timestamp'] !== null ?
autolinker.link(feature.properties['timestamp'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <th scope="row">name</th>\
            <td>' + (feature.properties['name'] !== null ?
autolinker.link(feature.properties['name'].toLocaleString()) : '') + '</td>\
        </tr>\
        <tr>\
            <td colspan="2"><strong>type</strong><br />' +
(feature.properties['type'] !== null ?
autolinker.link(feature.properties['type'].toLocaleString()) : '') + '</td>\
        </tr>\
    </table>';
    layer.bindPopup(popupContent, {maxHeight: 400});
}

function style_FastFood_0_0() {
    return {
        pane: 'pane_FastFood_0',
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
    iconUrl: 'markers/FastFood_0.svg',
    iconSize: [15.2, 15.2]
}),
        interactive: true,
    }
}
map.createPane('pane_FastFood_0');

```

```

map.getPane('pane_FastFood_0').style.zIndex = 400;
map.getPane('pane_FastFood_0').style['mix-blend-mode'] = 'normal';
var layer_FastFood_0 = new L.geoJson(json_FastFood_0, {
  attribution: '',
  interactive: true,
  dataVar: 'json_FastFood_0',
  layerName: 'layer_FastFood_0',
  pane: 'pane_FastFood_0',
  onEachFeature: pop_FastFood_0,
  pointToLayer: function (feature, latlng) {
    var context = {
      feature: feature,
      variables: {}
    };
    return L.marker(latlng, style_FastFood_0_0(feature));
  },
}).addTo(map); //layer_FastFood_0

function pop_Cafe_2(feature, layer) {
  var popupContent = '<table>\
    <tr>\
      <td colspan="2">' + (feature.properties['osm_id'] !== null ?
autolinker.link(feature.properties['osm_id'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2">' + (feature.properties['timestamp'] !== null ?
autolinker.link(feature.properties['timestamp'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>name</strong><br />' +
(feature.properties['name'] !== null ?
autolinker.link(feature.properties['name'].toLocaleString()) : '') + '</td>\
    </tr>\
    <tr>\
      <td colspan="2"><strong>type</strong><br />' +
(feature.properties['type'] !== null ?
autolinker.link(feature.properties['type'].toLocaleString()) : '') + '</td>\
    </tr>\
  </table>';
  layer.bindPopup(popupContent, {maxHeight: 400});
}

function style_Cafe_2_0() {
  return {
    pane: 'pane_Cafe_2',
    rotationAngle: 0.0,
    rotationOrigin: 'center center',
    icon: L.icon({
      iconUrl: 'markers/points_0_restaurant33.png',
      iconSize: [15.2, 15.2]
    })
  },
}

```



```

        interactive: true,
    }
}
map.createPane('pane_Cafe_2');
map.getPane('pane_Cafe_2').style.zIndex = 402;
map.getPane('pane_Cafe_2').style['mix-blend-mode'] = 'normal';
var layer_Cafe_2 = new L.geoJson(json_Cafe_2, {
    attribution: '',
    interactive: true,
    dataVar: 'json_Cafe_2',
    layerName: 'layer_Cafe_2',
    pane: 'pane_Cafe_2',
    onEachFeature: pop_Cafe_2,
    pointToLayer: function (feature, latlng) {
        var context = {
            feature: feature,
            variables: {}
        };
        return L.marker(latlng, style_Cafe_2_0(feature));
    },
}).addTo(map); //layer_Cafe_2

////

//BaseMap
var baseMaps = {
    "Атлас": ThAtlas,
    "Land": CartoDBLand,
    "Transport": ThTransp,

    "OSM": osm,
    "Спутник": Sat
};

var helplayers = {
};
//Markers Layer
var overlaysMaps = {
    //helplayers

```

```

    "Вулиці": streets,
    "Підписи": labels,
    "3D": osmb,
    //Points
    "Точки": layer_points_0 ,
    "Укриття на зупинках": layer__0,
    "Найпростіше укриття": layer__1,
    "Сховища": layer__2,
    "Протирадіаційні укриття": layer__3,
    "Лікувальні заклади": layer__4,

    //polygons
    //parks
    //places of interest
    //Lindes
    //bikes

};
L.control.layers(baseMaps, overlaysMaps, {collapsed: true}).addTo(map);

//Location
map.locate({
    setView: true,
    maxZoom: 20
})
    //foundLocation
function onLocationFound(e) {
    var radius = e.accuracy;
    L.marker(e.latlng, radius).addTo(map);
}
map.on('locationfound', onLocationFound);
    //Error
/*function onLocationError(e) {
    alert(e.message);
}
map.on('locationerror', onLocationError);*/

//SidePanel
/* contents */
const left = '<div class="header"> No<strong>Name</strong>Map</div>';
let contents = `
    <div class="content bg-secondary ">

```

```

<div class="accordion" id="accordionExample">
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingOne">
      <button class="accordion-button" type="button" data-bs-
toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true" aria-
controls="collapseOne">
        <div class="title">Профіль</div>
      </button>
    </h2>
    <div id="collapseOne" class="accordion-collapse collapse show" aria-
labelledby="headingOne" data-bs-parent="#accordionExample">
      <div class="accordion-body">
        <p><strong> Вітаємо, PepsaSasp</strong></p>
        <p><strong>Пройдено: </strong> <em>5967 м</em></p>
        <p><strong>Запитів: </strong> <em> 1203 </em></p>
        <p><strong>Найчастіший запит: </strong> <em> Кафе <b>(403)</b></em></p>
        <p><strong>Points: </strong> <em>8888</em></p>
      </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingTwo">
      <button class="accordion-button collapsed" type="button" data-bs-
toggle="collapse" data-bs-target="#collapseTwo" aria-expanded="false" aria-
controls="collapseTwo">
        <div class="title">Найближчі до мене</div>
      </button>
    </h2>
    <div id="collapseTwo" class="accordion-collapse collapse" aria-
labelledby="headingTwo" data-bs-parent="#accordionExample">
      <div class="accordion-body">
        <form oninput="output.value = Math.round(range.valueAsNumber / 1000)">
          <h4>
            Радіус пошуку
          </h4>
          <div class="range">
            <input class="form-range" name="range" type="range" min="20"
max="100000" step="5000" value ="20000">
            <output id="myrange" name="output" for="range">
              20
            </output> xв
          </div>
        </form>
        <p id="plist"></p>
      </div>
    </div>
  </div>
</div>

```

```

        <button onclick="searchrange()" class="btn btn-
primary">Підтвердити</button>
    </div>
    <br>

    <div class="d-grid gap-2 col-6 mx-auto">
    <button type="button" class="btn btn-warning">Укриття</button>
    <br>
    </div>

    <p><form class="d-flex">
    <input class="form-control me-2" type="search" placeholder="Кава..."
aria-label="Search">

    </form></p>
    <p>
        <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c1" aria-expanded="false" aria-controls="c1">
Заклади харчування
    </button>

    </p>
    <div class="collapse" id="c1">
        <div class="card card-body bg-light">
            <p>

                <div class="form-check form-switch">
                    <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Ресторани</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Швидке харчування</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Кафе</label>
                </div>

            </p>
        </div>
    </div>
    </div>
    <p>

```

```

        <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c2" aria-expanded="false" aria-controls="c2">
    Цікаві місця та відпочинок
    </button>
</p>
    <div class="collapse" id="c2">
        <div class="card card-body bg-light">
            <p>
                <div class="form-check form-switch">
                    <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                        <label class="form-check-label"
for="flexSwitchCheckDefault">Парки</label>
                    </div>
                    <div class="form-check form-switch">
                        <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                            <label class="form-check-label"
for="flexSwitchCheckDefault">Туристичні пам'ятки</label>
                        </div>
                        <div class="form-check form-switch">
                            <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                                <label class="form-check-label"
for="flexSwitchCheckDefault">Музеї</label>
                            </div>
                        </p>
                    </div>
                </div>
            <p>
                <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c3" aria-expanded="false" aria-controls="c3">
    Сфери послуг
    </button>
</p>
    <div class="collapse" id="c3">
        <div class="card card-body bg-light">
            <p>
                <div class="form-check form-switch">
                    <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                        <label class="form-check-label"
for="flexSwitchCheckDefault">АЗС</label>
                    </div>
                    <div class="form-check form-switch">
                        <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                            <label class="form-check-label"
for="flexSwitchCheckDefault">Зарядні станції</label>
                        </div>
                    <div class="form-check form-switch">

```

```

                <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                <label class="form-check-label"
for="flexSwitchCheckDefault">Банки</label>
            </div>
            <div class="form-check form-switch">
                <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                <label class="form-check-label"
for="flexSwitchCheckDefault">Ремонтні сервіси</label>
            </div>
            <div class="form-check form-switch">
                <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                <label class="form-check-label"
for="flexSwitchCheckDefault">Паркування</label>
            </div>
            <div class="form-check form-switch">
                <input class="form-check-input" type="checkbox"
id="flexSwitchCheckDefault">
                <label class="form-check-label"
for="flexSwitchCheckDefault">Медичні Заклади</label>
            </div>
        </p>
    </div>
</div>
<p>
    <button class="btn btn-secondary" type="button" data-bs-
toggle="collapse" data-bs-target="#c4" aria-expanded="false" aria-controls="c4">
        Заклади торгівлі
    </button>
</p>
    <div class="collapse" id="c4">
        <div class="card card-body bg-light">
            <p>
                <div class="form-check form-switch">
                    <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Торгові центри</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">
                    <label class="form-check-label"
for="flexSwitchCheckDefault">Продуктові магазини</label>
                </div>
                <div class="form-check form-switch">
                    <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">

```

```

                <label class="form-check-label"
for="flexSwitchCheckDefault">Непродовольчі товари</label>
            </div>
            <div class="form-check form-switch">
                <input class="form-check-input"
type="checkbox" id="flexSwitchCheckDefault">
                <label class="form-check-label"
for="flexSwitchCheckDefault">Аптеки</label>
            </div>
        </p>
    </div>
</div>

</div>
</div>
</div>
<div class="accordion-item">
    <h2 class="accordion-header" id="headingThree">
        <button class="accordion-button collapsed" type="button" data-bs-
toggle="collapse" data-bs-target="#collapseThree" aria-expanded="false" aria-
controls="collapseThree">
            <div class="title">Навігація</div>
        </button>
    </h2>
    <div id="collapseThree" class="accordion-collapse collapse" aria-
labelledby="headingThree" data-bs-parent="#accordionExample">
        <div class="accordion-body">

            <div class="btn-group" role="group" aria-label="Basic radio toggle
button group">
                <input type="radio" class="btn-check" name="btnradio" id="btnradio1"
autocomplete="off" checked>
                <label class="btn btn-outline-primary" for="btnradio1">Пішки</label>

                <input type="radio" class="btn-check" name="btnradio" id="btnradio2"
autocomplete="off">
                <label class="btn btn-outline-primary" for="btnradio2">Авто</label>

                <input type="radio" class="btn-check" name="btnradio" id="btnradio3"
autocomplete="off">
                <label class="btn btn-outline-primary"
for="btnradio3">Велосипед</label>
            <br>
            </div>
        </div>
    </div>
    <br>
    <button id="Route" type="button" class="btn btn-secondary">Увімкнути
навігацію</button>
</div>
</div>
</div>

```

```

        </div>
        <div class="accordion-item">
            <h2 class="accordion-header" id="headingFour">
                <button class="accordion-button collapsed" type="button" data-bs-
toggle="collapse" data-bs-target="#collapseFour" aria-expanded="false" aria-
controls="collapseFour">
                    <div class="title">Мої вподобання</div>

                </button>
            </h2>
            <div id="collapseFour" class="accordion-collapse collapse" aria-
labelledby="headingFour" data-bs-parent="#accordionExample">
                <div class="accordion-body">
                    <p>Заклади</p>
                    <p>Маршрути</p>
                    <p>На основі запитів</p>
                    <p>Запит</p>
                </div>
            </div>
        </div>

</div>` ;

/* left */
L.control.slideMenu(left + contents).addTo(map);

/* right */
/*const slideMenu = L.control
.slideMenu("", {
    position: "topright",
    menuposition: "topright",
    width: "30%",
    height: "200x",
    delay: "50",
    icon: "chevron-left",
})
.addTo(map);*/
//slideMenu.setContents(right + '<p>Test</p>');

//Navigation
function NavRad(waypoints) {
    var route = L.Routing.control({
        waypoints: waypoints,
        router: new L.Routing.graphHopper('66df4d39-8ec7-4c00-9cc6-8cb97e51c3c6', {
            urlParameters: {
                vehicle: 'foot'
            }
        })
    });
}

```



```

    })
  }).addTo(map);
}
////
var plist = document.getElementById("plist");
let circles;
let srange;
var stpoint = [49.54466784075138, 25.596443337497583];
var waypoints = [];

function removeCirle (){
  if(circles != undefined)
    map.removeLayer(circles);
}

function displayResultsInModal(filteredMarkers) {
  var popupList = document.getElementById('popup-list');
  popupList.innerHTML = '';

  // Очищаємо масив waypoints перед додаванням нових точок
  waypoints = [L.latLng(49.54466784075138, 25.596443337497583)];

  filteredMarkers.forEach(function (marker) {
    var listItem = document.createElement('li');
    var button = document.createElement('button');
    button.type = 'button';
    button.className = 'btn btn-link';
    button.innerHTML = marker.getPopup().getContent();
    button.onclick = function () {
      showMarkerCoords(marker.getLatLng().lat, marker.getLatLng().lng);
      // Додаємо нову точку до масиву waypoints
      waypoints.push(marker.getLatLng());
      // Викликаємо функцію для відображення маршруту
      NavRad(waypoints);
      myModal.hide(); // Ховаємо модальне вікно після натискання на будь-
який маркер
    };
    listItem.appendChild(button);
    popupList.appendChild(listItem);
  });

  var myModal = new bootstrap.Modal(document.getElementById('myModal'));
  myModal.show(); // Показати модальне вікно
}

function showMarkerCoords(lat, lng) {
  console.log('Marker Coordinates:', lat, lng);
}

function searchrange() {

```

```

removeCircle ();
srange = document.getElementById("myrange").value;
console.log(srange);

circles = L.circle(stpoint, 50 * srange, {
    color: 'red',
    fillColor: '#white',
    fillOpacity: 0.3
}).addTo(map);
var circleLatLng = circles.getLatLng();
var circleRadius = circles.getRadius();
console.log(circleRadius);
var filteredMarkers = layer_Cafe_2.getLayers().filter(function(marker) {
    var markerLatLng = marker.getLatLng();

    return markerLatLng.distanceTo(circleLatLng) <= circleRadius;
});
var valuesInCircle = filteredMarkers.map(function(marker) {

    return marker.getPopup().getContent(); // або будь-яке інше значення
маркера
});
console.log(valuesInCircle);
/*var plistres = document.getElementById("plist");
plistres.innerHTML = valuesInCircle.join("<br>");*/

waypoints = [L.latLng(49.54466784075138, 25.596443337497583)]; // Додаємо
початкову точку
waypoints = waypoints.concat(filteredMarkers.map(function (marker) {
    return marker.getLatLng();
}));
console.log(waypoints);
displayResultsInModal(filteredMarkers);
/* var popupList = document.getElementById('popup-list');
popupList.innerHTML = valuesInCircle.join("<br>");
document.getElementById('myModal').style.display = 'block';*/
}
function closeModal() {
    document.getElementById('myModal').style.display = 'none';
}
}

///  

///  

///Пошук
///d369607edf3aceb1c3f3ee54209a71bb

```

```

///

document.getElementById("Route").onclick = function(){
    NavFunc()
};
function NavFunc(){
    map.on('click', function(e){
        var endMarker = L.marker([e.latlng.lat, e.latlng.lng]).addTo(map);
        L.Routing.control({
            waypoints: [
                L.latLng(49.54466784075138, 25.596443337497583),
                L.latLng(e.latlng.lat, e.latlng.lng)
            ],
            showAlternatives: true,
            altLineOptions: {
                styles: [
                    { color: 'black', opacity: 0.15, weight: 9 },
                    { color: 'white', opacity: 0.15, weight: 9 },
                    { color: 'blue', opacity: 0.15, weight: 9 },
                ]
            },
            routeWhileDragging: true,
            geocoder: L.Control.Geocoder.nominatim(),
            router: new L.Routing.graphHopper('c4cbf9a5-0bfc-4a42-acd5-
82f3fa5bd9b7' , {
                urlParameters: {
                    vehicle: 'foot'
                }
            })
        }).addTo(map);

    });
}
L.circleMarker([49.54466784075138, 25.596443337497583]).addTo(map);

//geocode
var geocoder = L.Control.geocoder({
    defaultMarkGeocode: false
})

```

```
.on('markgeocode', function(e) {
  var bbox = e.geocode.bbox;
  var poly = L.polygon([
    bbox.getSouthEast(),
    bbox.getNorthEast(),
    bbox.getNorthWest(),
    bbox.getSouthWest()
  ]).addTo(map);
  map.fitBounds(poly.getBounds());
})
.addTo(map);

//events
map.on('mousemove', function(e){
  document.getElementsByClassName('coords')[0].innerHTML = " " + e.latlng.lat +
  " ," + e.latlng.lng;
});
```

## ДОДАТОК Б

### Використання геоінформаційних технологій для розробки веб-сервісу вирішення задач геомаркетингу на прикладі міста Тернопіль

Спеціальність: 193 – Геоземлезнавство та землеустрій

Виконавець:

Керівник:

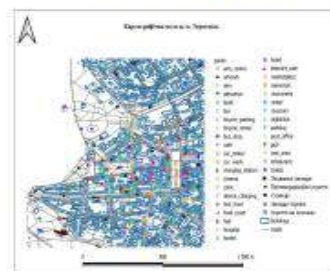
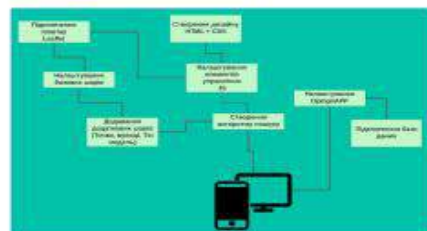
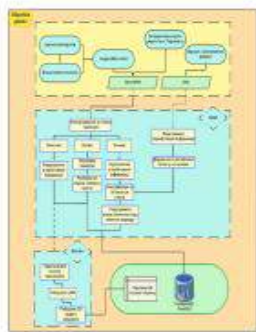
Студент гр. №462-м, Сергеев А.С. ктн., доцент Андреев С.М.

**Мета роботи** - Підвищення ефективності розробки веб-сервісів для вирішення задач геомаркетингу за рахунок використання ГІС технологій.

**Об'єкт дослідження** - Розробка веб-сервісів та картографічних матеріалів.

**Предмет дослідження** - Вирішення задач геомаркетингу шляхом застосування технологій для створення веб-сервісу.

**Задачі проєкту:** На основі аналізу існуючих сервісів визначити необхідні функції та вхідні дані  
На основі отриманих моделей та баз даних розробити веб сервіс для вирішень задач геомаркетингу



## ДОДАТОК В

ПРЕЗЕНТАЦІЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА ТЕМУ  
ВИКОРИСТАННЯ ГЕОІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ РОЗРОБКИ  
ВЕБ-СЕРВІСУ ВИРІШЕННЯ ЗАДАЧ ГЕОМАРКЕТИНГУ



Національний аерокосмічний університет ім. М.С. Жуковського  
«Харківський авіаційний інститут»  
Факультет ракетно космічної техніки  
Кафедра геоінформаційних технологій та космічного моніторингу Землі



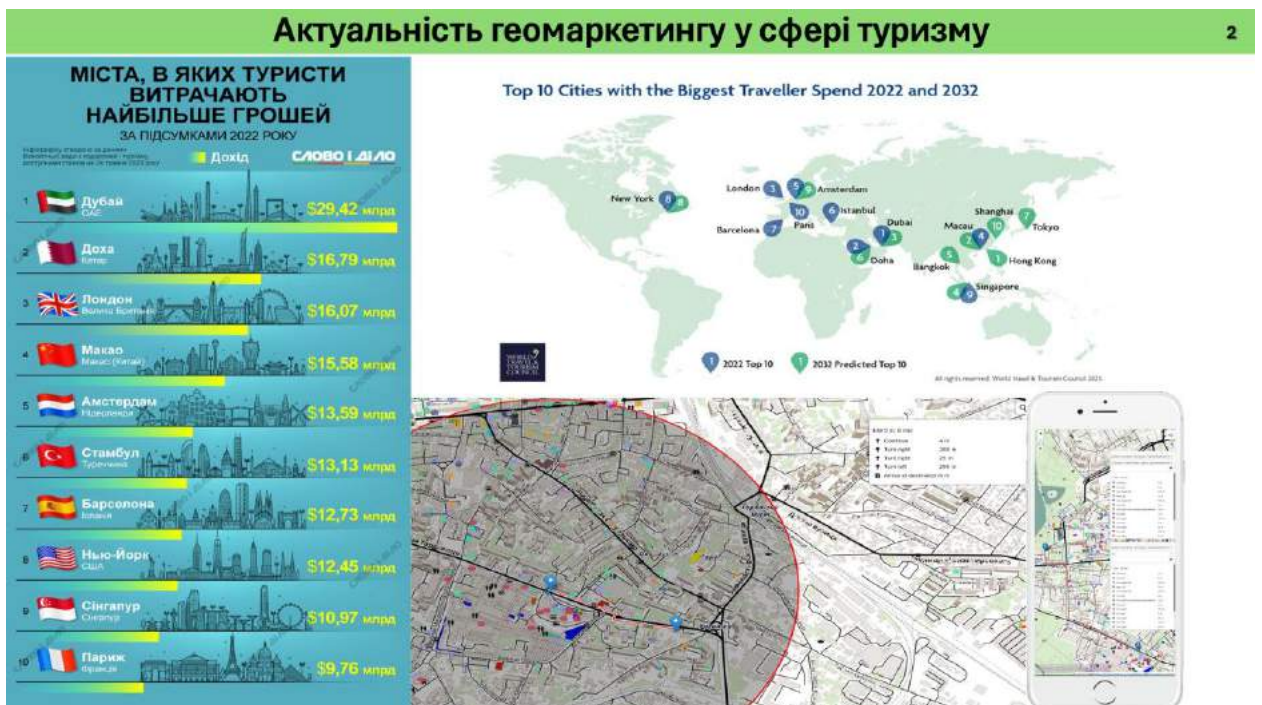
Кваліфікаційна робота магістра

Спеціальність 193 «Геодезія та землеустрій»  
Освітня програма «Геоінформаційні системи та технології»

Використання геоінформаційних технологій для розробки веб -  
сервісу вирішення задач геомаркетингу на прикладі міста  
Тернопіль

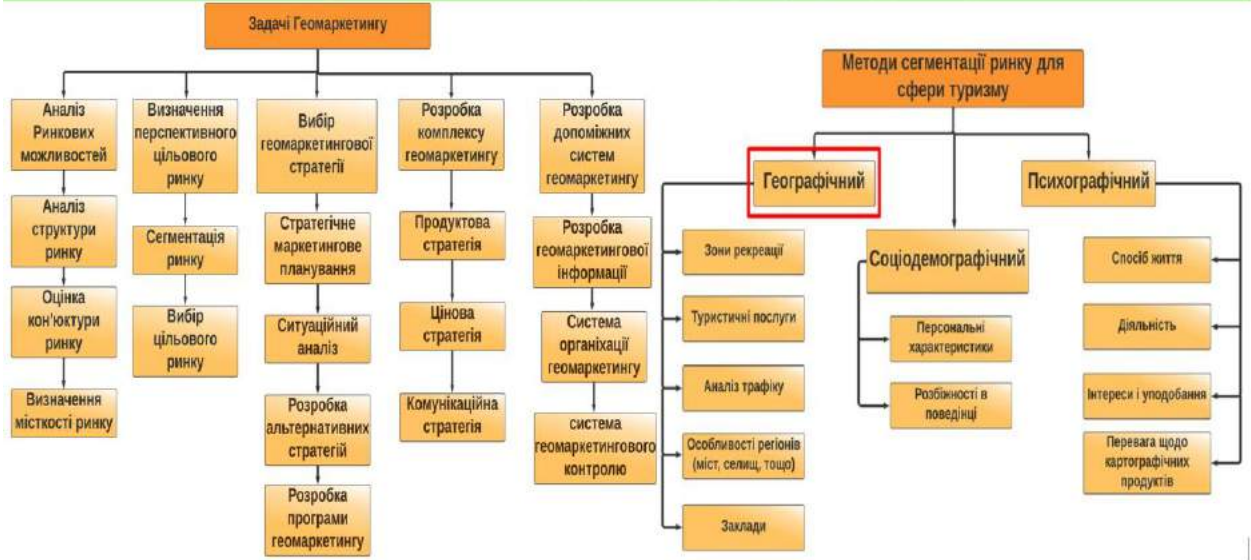
Виконав: студент групи 462-м  
Серегесв А. С.  
Керівник: к.т.н., доцент  
Андреев С. М.

Харків 2024





## Актуальність геомаркетингу у сфері туризму 3



## АНАЛІЗ ТУРИСТИЧНОГО ПОТЕНЦІАЛУ МІСТА ТЕРНОПІЛЬ 4

**Тернопіль**  
Місто туристичного потенціалу

**ПАРК ІМЕНА ГАРІСА ШЕВЧЕНКА**

**ПЛАН-СХЕМА ПАРКУ КУЛЬТУРИ І ВІДПОЧИНКУ «ТОПІАВМЕ»**

**Відносні показники міст 22 сфер та послуг**

Місто	Показник
Вінниця	1,8
Луцьк	1,3
Івано-Франківськ	1,3
Чернівці	1,2
Хмельницький	1,2
Меріуполь	1,2
Київ	1,1
<b>Тернопіль</b>	<b>1,1</b>
Черкаси	1,1
Львів	1,1
Колomyця	1,1
Сєвєродонецьк	1,0
Ужгород	1,0
Одеса	0,9
Житомир	0,8
Рівне	0,8
Суми	0,8
Бєлє	0,8
Дніпро	0,8
Закарпатськ	0,8
Львівськ	0,8
Миколаїв	0,8
Черкаси	0,8
Харків	0,7

**Динаміка ринку туризму (2020-2021)**  
відносинності міст у сфері та послуг

Міста в Україні

Туристи в Україні

Пункти та об'єкти

Детальні карти: Майдан Незалежності, Центр міста, Пункти та об'єкти, Детальні пункти.

## АНАЛІЗ ТУРИСТИЧНОГО ПОТЕНЦІАЛУ МІСТА ТЕРНОПІЛЬ 4

**Індекс надомовності міста 22 сфер 18 послуг**

Вінниця	16
Львів	13
Івано-Франківськ	13
Чернівець	13
Хмельницький	13
Миколаїв	13
Київ	13
<b>Тернопіль</b>	<b>12</b>
Чернівці	11
Відрод	11
Колomyjsький	11
Севастополь	10
Ужгород	10
Одеса	10
Житомир	10
Рівне	10
Суми	10
Берег	10
Дніпро	10
Закарпаття	10
Луганськ	10
Миколаїв	10
Черкаси	10
Хмель	10

**Динаміка індексу надомовності міста 2020-2021**

**План-схема ПАРКУ КУЛЬТУРИ І ВІДОПОЧИНКУ «ТОПІАВМЕ»**







**Аналіз джерел та сервісів для вирішення задач в сфері туризму** 8



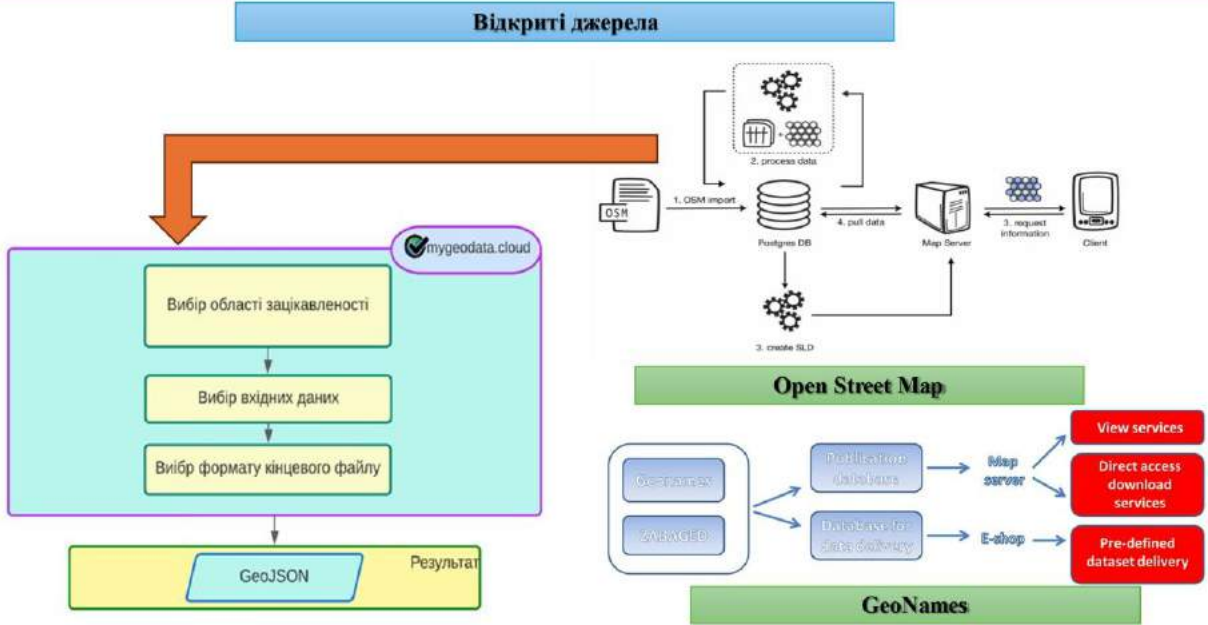
Функції	Sygic	Waze	HERE WeGo	Roadtripper
Сповідення про трафік	Так	Так	Так	Обмежено
Варіанти маршруту	Так	Так	Так	Так
Спільнота користувачів	Ні	Так	Ні	Ні
Публічний транспорт	Ні	Ні	Так	Ні
Подорожі на велосипеді	Так	Ні	Так	Ні
Планування зупинок	Так	Так	Так	Так
Подорожі пішки	Так	Так	Так	Так
Подорожі автомобілем	Так	Так	Так	Так
Подорожі громадським транспортом	Ні	Ні	Так	Ні
Планування екскурсій	Так	Ні	Так	Так

**Аналіз платформ(джерела картографічних даних) та мов програмування.** 9

**Комерційні постачальники картографічних даних**



**Аналіз платформ(джерела картографічних даних) та мов програмування.** 10



**Аналіз платформ(джерела картографічних даних) та мов програмування.** 11

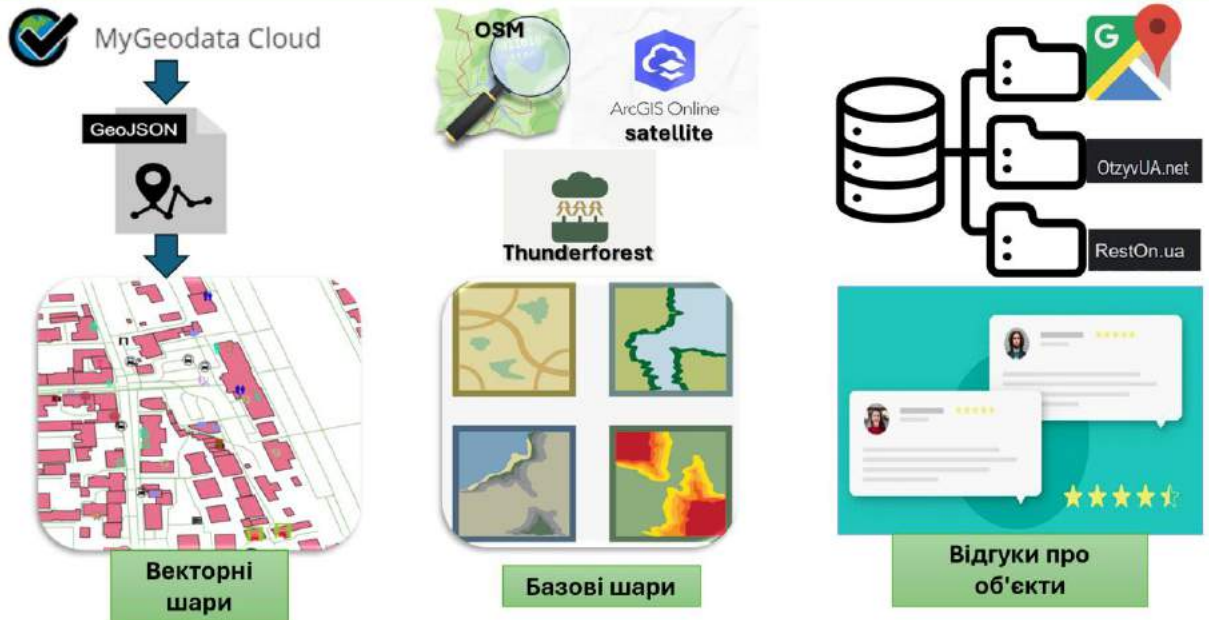
**Характеристика мов програмування**

Характеристики	JavaScript	Python	PHP
Сфера застосування	Веб-розробка, аналіз даних	Веб-розробка, аналіз даних, штучний інтелект	Веб-розробка, серверні додатки
Спрощення синтаксису	Синтаксис має виразність та динаміку.	Простий та читабельний синтаксис	Легкий синтаксис
Асинхронність	Велика активна спільнота, широкий вибір бібліотек для розробки.	Підтримує асинхронність, але менше ніж JavaScript	Підтримує асинхронність завдяки додатковим бібліотекам
Спільнота та бібліотеки	Велика активна спільнота, широкий вибір бібліотек для розробки.	Активна спільнота, велика кількість бібліотек під різноманітні завдання	Значна спільнота та велика кількість фреймворків
Легкість вивчення	Середня, має свої особливості.	Легка читабельність	Легка та інтуїтивна
Швидкість виконання коду	Швидка	Середня	Не сильно швидка, поступається JS та Py



## Вхідні дані кваліфікаційної роботи

12



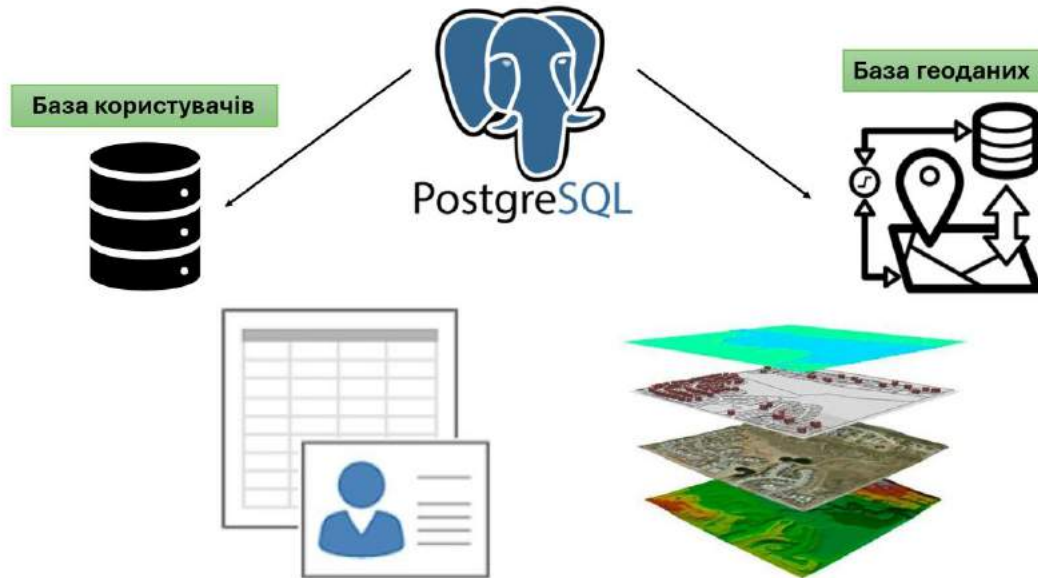
## Аналіз баз даних для проєкту

13

Критерій	PostgreSQL	MySQL	Oracle	Microsoft SQL server
Геодані	Підтримка географічних типів та функцій через розширення PostGIS	Підтримка географічних типів та функцій через розширення Spatialite.	Повна географічна підтримка з Oracle Spatial	Географічна підтримка з: Microsoft SQL Server Spatial Extensions.
Реляційні можливості	Повноцінна реляційна система баз даних	Реляційна система баз даних з обмеженою географічною підтримкою.	Повноцінна реляційна система баз даних.	Повноцінна реляційна система баз даних.
Відкритість	Відкрита система з активною спільнотою, розповсюдження безкоштовне	Відкрита система, з активною спільнотою, розповсюдження безкоштовне	Відкрита система з окремими комерційними опціями.	Відкрита система з комерційними опціями.
Швидкодія	Незважаючи на великі обсяги даних працює доволі швидко, залежить від запиту	Залежить від конкретної конфігурації та оптимізацій.	Швидкодія може бути високою, але залежить від конфігурації та оптимізацій.	Швидкодія може бути високою, але залежить від конфігурації та оптимізацій.
Масштабованість	Здатна масштабуватися на великі обсяги даних та велику кількість одночасних користувачів	Може вимагати додаткових оптимізацій для масштабування.	Здатний масштабуватися на великі обсяги даних та велику кількість одночасних користувачів.	Здатний масштабуватися на великі обсяги даних та велику кількість одночасних користувачів.

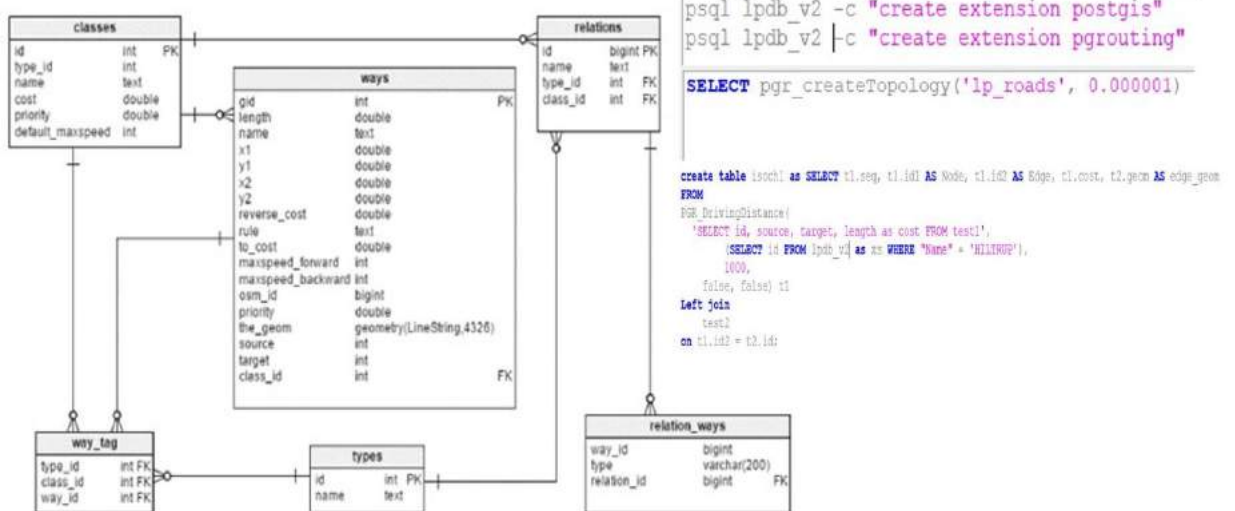
## Аналіз баз даних для проєкту

14



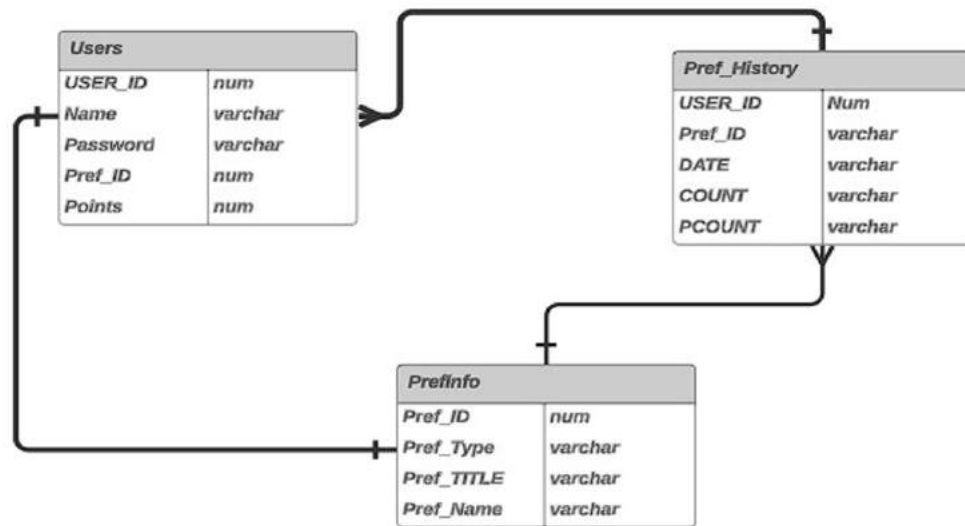
## Розробка архітектура бази геоданих

15



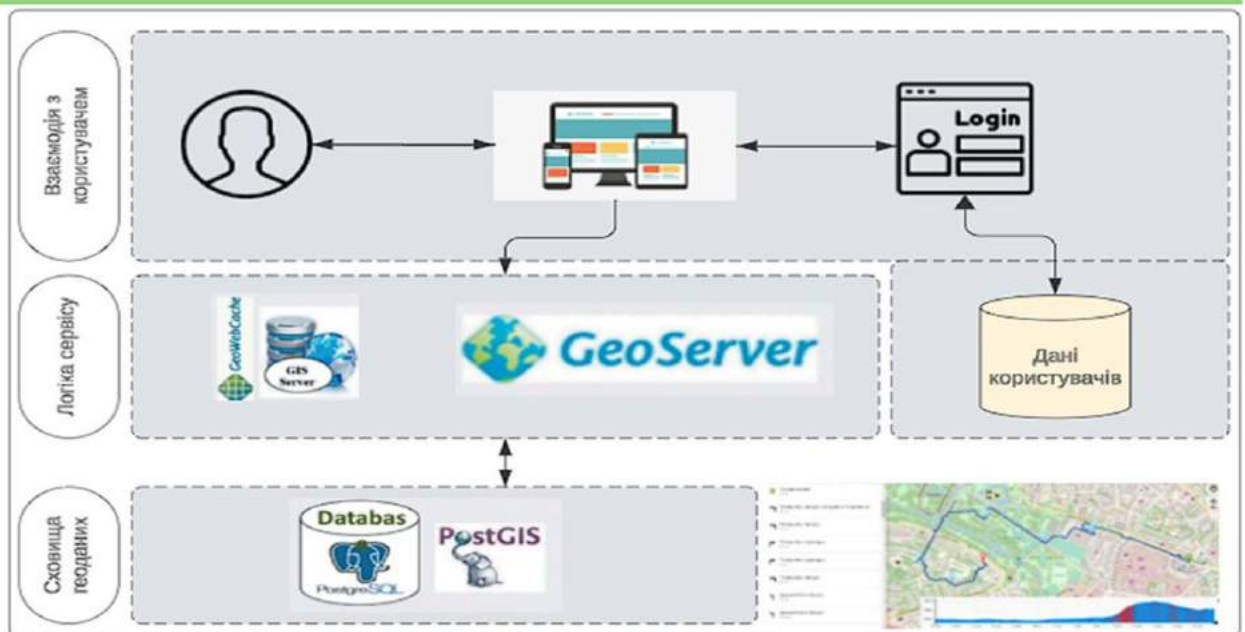
## Розробка архітектура бази геоданих

16



## Архітектура розробляемого веб-сервісу

17



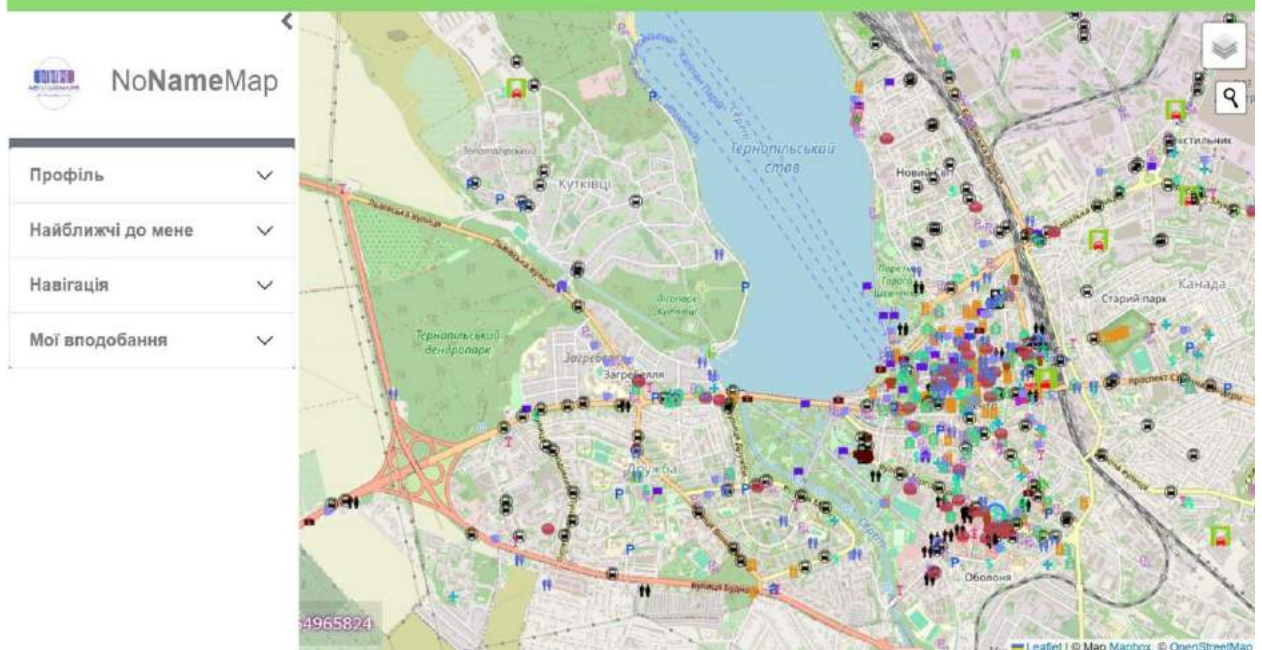
## Інтерфейс користувача розробляємого веб-сервісу

18



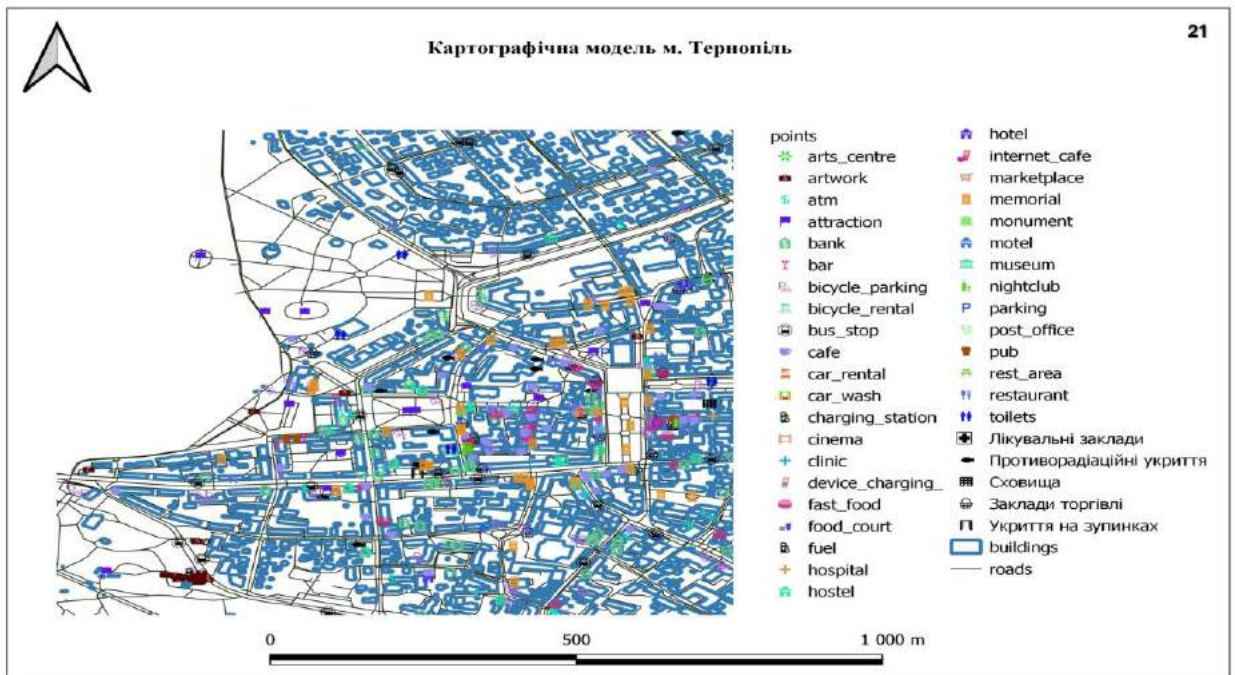
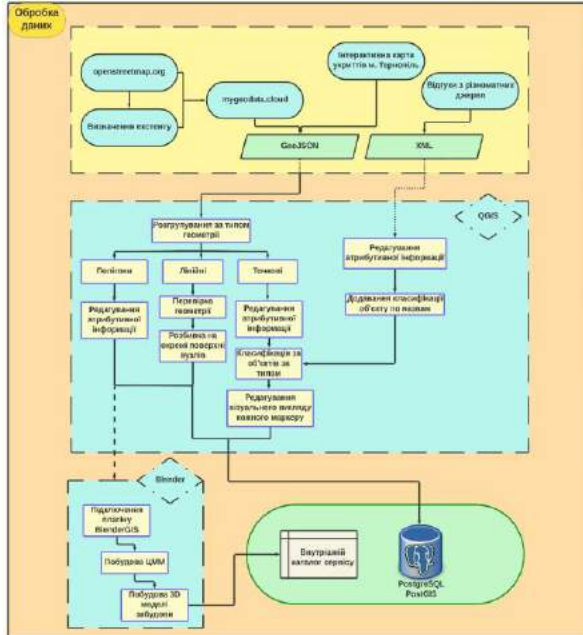
## Інтерфейс користувача розробляємого веб-сервісу

19

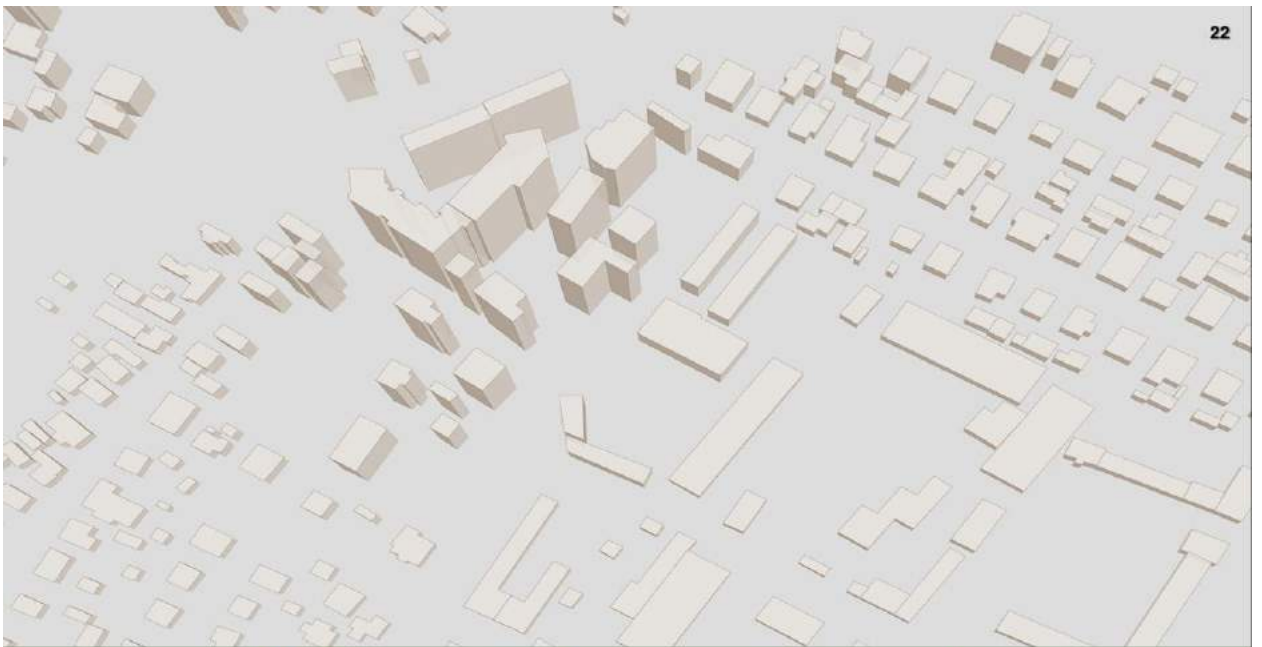




## Обробка вхідних даних





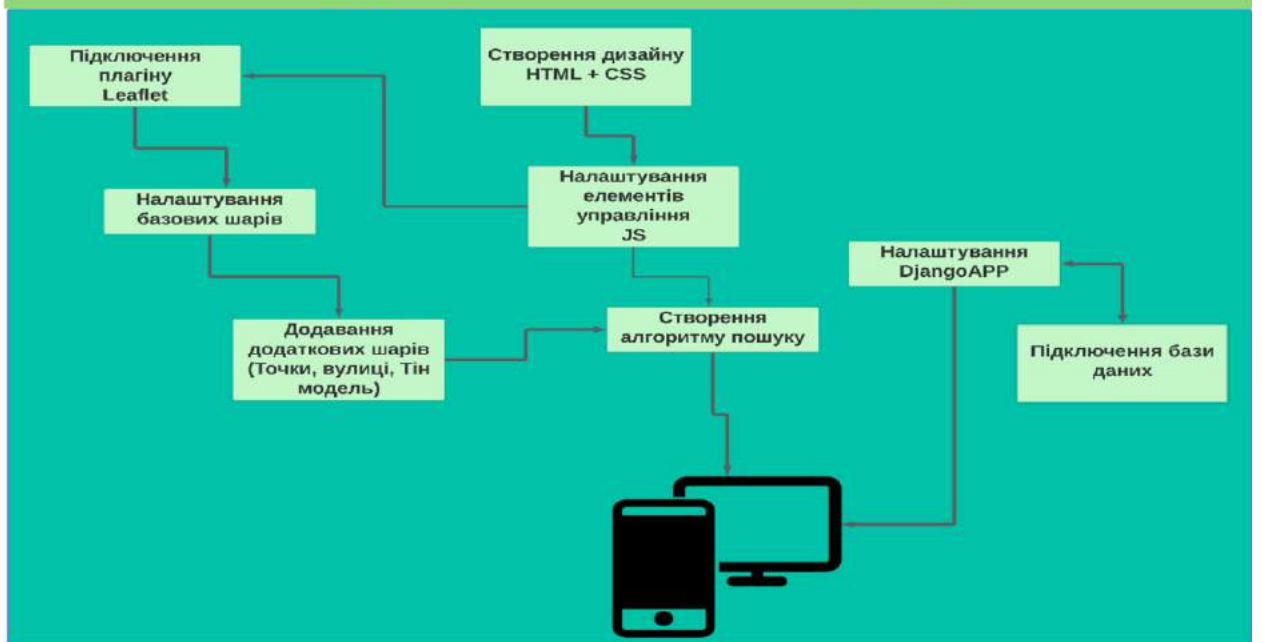


22

## 3D Модель забудови м. Тернопіль

## Програмування власного веб-сервісу

23



## Результати

24

**Результати пошуку**

За найкращим рейтингом

- Рейтинг замовки: 4.8  
Назва замовки: NINA  
Виділений коментар: Сервіс на висоті. Яка найкраща!
- Рейтинг замовки: 4.7  
Назва замовки: NATURA  
Виділений коментар: Шкоди повністю позбавилися. В наявності замки рекомендаційно дають замки.
- Рейтинг замовки: 4.6  
Назва замовки: Три Миколи  
Виділений коментар: Багато не очікуєш, тільки згоді зроби сам за себе.
- Рейтинг замовки: 4.5  
Назва замовки: Три Миколи  
Виділений коментар: Скатували тут велосипеди, усі віддалися у замки! Три Миколи the best!

**Пішки(стандартне значення)**

**Велосипед**

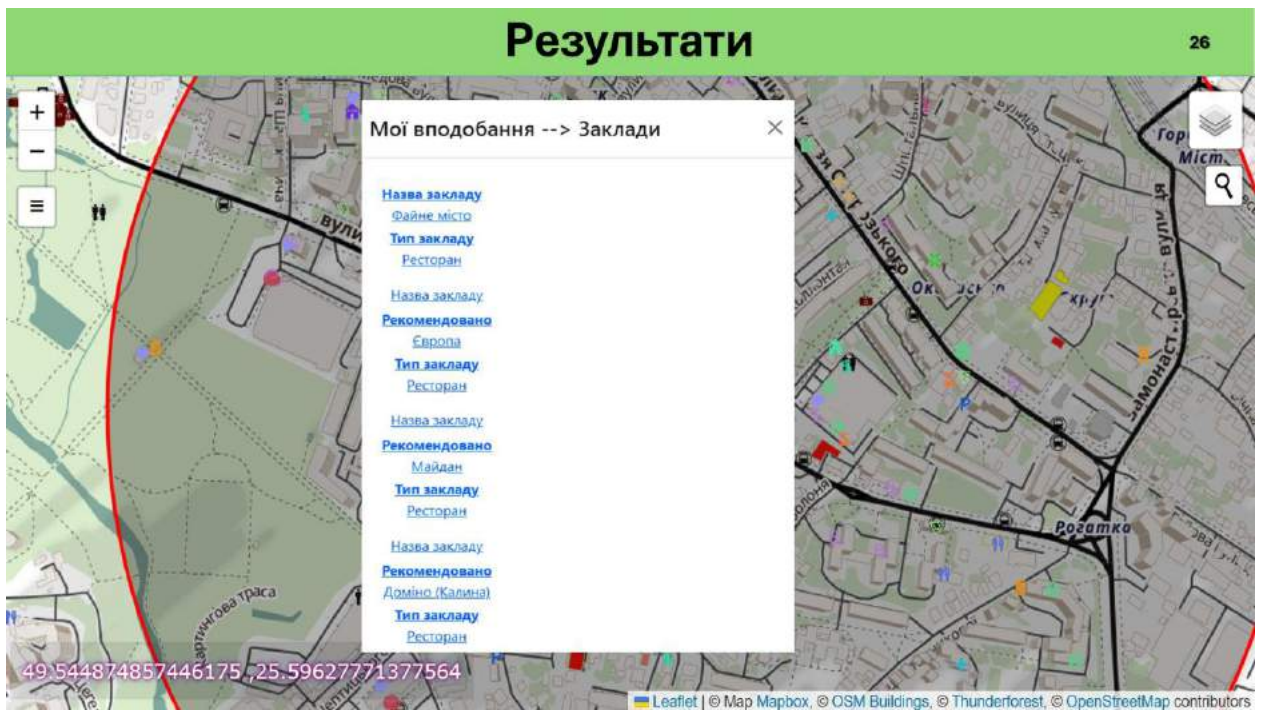
**Авто**

## Результати

25

430.5 m, 6 min  
 ↑ Continue 4 m  
 ↑ Turn right 330 m  
 ↑ Turn right 25 m  
 ↑ Turn left 230 m  
 ↓ Drive at destination 0 m





## Подальший розвиток проєкту

27

### Дорожня карта



## Висновки

28

На основі проведеного аналізу використання геоінформаційних технологій для вирішення задач геомаркетингу в сфері туризму, було визначено, що розвиток даної галузі залежить від розробки якісних картографічних рішень для задоволення потреб користувачів.

На основі проаналізованих картографічних сервісів та платформ, були визначені основний функціонал для реалізації завдань проекту, а також необхідні вхідні дані, вони представлені у вигляді: базових карт місцевості, супутникових знімків, а також векторні дані про об'єкти інтересу в межах тестової ділянки (м. Тернопіль) і додаткова інформація про кожний. На підставі отриманих вхідних даних для проекту, також було обрано середовище для зберігання оброблених даних у вигляді бази даних на СУБД PostgreSQL, де окрім вищезазначеного також зберігаються дані про користувачів. В результаті обробки вхідних даних було отримано картографічну модель м. Тернопіль на основі векторних шарів, а також 3D представлення забудови міста, у вигляді окремого шару на самому веб-сервісі.

На підставі аналізу сучасних платформ та мов програмування, був розроблений веб-сервіс з використання геоінформаційних технологій для вирішення задач геомаркетингу в сфері туризму.

Запропонована метод підвищення ефективності при розробці картографічних веб-сервісів покриває актуальний попит в сфері туризму, з використанням елементів геомаркетингу і створює потенціал для подальшого розвитку, як галузі, так і можливих майбутніх рішень із застосування нейромереж, а також штучного інтелекту, що зможе оптимізувати усі процеси описані в самій роботі.

## Апробації

29

1. Сергеев А. С. Ключников В. О. Фоміних А. В., Малихіна М. О. Гіроскоп: тези доп. II науково-практична конференція студентів технікумів та коледжів «Сучасні тенденції та перспективи розвитку природничих наук». Харків 2017

2. Сергеев А.С. Бірюков О.В. Удосконалення роботи гідрометричної вертушки за допомоги середовища "Arduino": тези доп. II науково-практична конференція студентів технікумів та коледжів Сучасні тенденції та перспективи розвитку природничих наук. Одеса 2019

3. Сергеев А.С. Калашнік А.С. Шамес Є.О. Побудова КРИВОЇ ВИТРАТ З ДАНИХ РІЧКИ ЛУГАНЬ ПОС КАЛТНІВКА ЗА 2015 РІК: тези доп. Харків II науково-практична конференція студентів технікумів та коледжів Сучасні тенденції та перспективи розвитку природничих наук. Одеса 2019

4. Сергеев А.С. Жилін В.А. ОСОБЛИВОСТІ РОЗРОБЛЕННЯ ГЕОІНФОРМАЦІЙНИХ ПРОГРАМНИХ ЗАСТОСУНКІВ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ: тези доп. XVII науково-технічна конференція Факультету ракетно-космічної техніки(2020р). Харків, 2020

5. Сергеев А.С. Жилін В.А. ОСОБЛИВОСТІ СУЧАСНІ МОВИ ПРОГРАМУВАННЯ ДЛЯ СТВОРЕННЯ МОБІЛЬНИХ ГЕОІНФОРМАЦІЙНИХ ЗАСТОСУНКІВ: тези доп. Студентська науково-технічна конференція Перспективні Мережеві і Комп'ютерні Технології(ПерСиК2020р). Харків, 2020

6. Сергеев А.С. Андреев С.М. Сучасні технології інтеграції геоданих у віртуальний простір: тези доп. Сучасні інформаційні технології управління екологічною безпекою, природокористуванням, заходами в надзвичайних ситуаціях: тези доп. Київ 2021.

7. Сергеев А.С. Андреев С.М. ВИКОРИСТАННЯ ГЕОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЗІ СТВОРЕННЯ АВТОНОМНОГО ДОДАТКУ ДЛЯ ПОБУДОВИ КАРТОГРАФІЧНИХ МОДЕЛЕЙ ТУРИСТИЧНОГО СПРЯМУВАННЯ: тези доп. I Міжнародна науково-практична конференція «ПОДОЛАННЯ ЕКОЛОГІЧНИХ РИЗИКІВ ТА ЗАГРОЗ ДЛЯ ДОВКІЛЛЯ В УМОВАХ НАДЗВИЧАЙНИХ СИТУАЦІЙ - 2022». Полтава, 2022.