

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет геоінформаційних систем та управління територіями
(факультет)

Кафедра геоінформатики і фотограмметрії
(назва кафедри)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТЬОГО СТУПЕНЯ БАКАЛАВРА**

на тему:

**Розробка інтерактивної вебкарти для надання туристичних послуг
на прикладі Івано-Франківської області**

Гаршин Максим Валерійович
(прізвище, ім'я, по батькові студента повністю)

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет геоінформаційних систем та управління територіями
(факультет)

Кафедра геоінформатики і фотограмметрії
(назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
проф., д-р. техн. наук
Юрій КАРПІНСЬКИЙ
“ _____ ” _____ 2025 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТЬОГО СТУПЕНЯ БАКАЛАВРА**

**Розробка інтерактивної вебкарти для надання туристичних послуг на
прикладі Івано-Франківської області**

Виконав: студент групи ПІСТ-21
спеціальності 193
“Геодезія та землеустрій”
спеціалізації “Геоінформаційні системи і
технології”
Гаршин Максим Валерійович
Керівник: Максимова Ю. С., к.т.н., доц.
Ідентичність підтверджую

Київ 2025 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

Інститут, факультет Геоінформаційних систем і управління територіями

Кафедра Геоінформатики і фотограмметрії

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 19 «Архітектура та будівництво»

(шифр і назва)

Спеціальність 193 “Геодезія та землеустрій”

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф., д.т.н. Юрій КАРПІНСЬКИЙ

“ ” 2025 року

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ

Гаршин Максим Валерійович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): «Розробка інтерактивної вебкарти для надання туристичних послуг на прикладі Івано-Франківської області»

керівник проекту (роботи) к.т.н., доц. Максимова Юлія Сергіївна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ ” 2025 року №

2. Строк подання студентом проекту (роботи) _____

3. Вихідні дані до проекту (роботи):

- 1) Геопросторові дані Google
- 2) Геопросторові дані OpenStreetMap

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

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ОБ'ЄКТУ ДОСЛІДЖЕННЯ

- 1.1. Туристичний потенціал Івано-Франківської області
- 1.2. Аналіз сучасного стану туристичних послуг в Україні та регіоні
- 1.3. Використання ГІС-технологій для розвитку туристичної інфраструктури
- 1.4. Нормативно-правові засади туристичної сфери в Україні

РОЗДІЛ 2. МЕТОДОЛОГІЧНІ ЗАСАДИ СТВОРЕННЯ ІНТЕРАКТИВНОЇ ВЕБ-КАРТИ

- 2.1. Концептуальна модель інтерактивної вебкарти туристичних послуг
- 2.2. Вибір інструментів та платформ для розробки вебкарти
- 2.3. Формування бази геопросторових даних туристичних об'єктів

РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІНТЕРАКТИВНОЇ ВЕБКАРТИ

- 3.1. Методика збору та обробки вихідних даних
- 3.2. Створення вебдодатка: структура, функціональні можливості
- 3.3. Інтеграція картографічних сервісів.

ВИСНОВКИ

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			
3			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	ВСТУП		
2	РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ОБ'ЄКТУ ДОСЛІДЖЕННЯ		
3	Туристичний потенціал Івано-Франківської області		
4	Аналіз сучасного стану туристичних послуг в Україні та регіоні		
5	Використання ГІС-технологій для розвитку туристичної інфраструктури		
6	Нормативно-правові засади туристичної сфери в Україні		
7	РОЗДІЛ 2. МЕТОДОЛОГІЧНІ ЗАСАДИ СТВОРЕННЯ ІНТЕРАКТИВНОЇ ВЕБКАРТИ		
8	Концептуальна модель інтерактивної вебкарти туристичних послуг		
9	Вибір інструментів та платформ для розробки вебкарти		
10	Формування бази геопросторових даних туристичних об'єктів		
11	РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІНТЕРАКТИВНОЇ ВЕБКАРТИ		
12	Методика збору та обробки вихідних даних		
13	Створення вебдодатка: структура, функціональні можливості		
14	Інтеграція картографічних сервісів.		

Студент _____ Гаршин М. В.
(підпис) (прізвище та ініціали)Керівник проекту (роботи) _____ Максимова Ю. С.
(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ОБ'ЄКТУ ДОСЛІДЖЕННЯ	8
1.1 Туристичний потенціал Івано-Франківської області	8
1.2 Аналіз сучасного стану туристичних послуг в Україні та регіоні	12
1.3 Використання ГІС-технологій для розвитку туристичної інфраструктури ..	16
1.4 Нормативно-правові засади туристичної сфери в Україні.....	23
РОЗДІЛ 2. МЕТОДОЛОГІЧНІ ЗАСАДИ СТВОРЕННЯ ІНТЕРАКТИВНОЇ ВЕБКАРТИ	31
2.1. Концептуальна модель інтерактивної вебкарти туристичних послуг	31
2.2. Вибір інструментів та платформ для розробки вебкарти.....	35
2.3. Формування бази геопросторових даних туристичних об'єктів.....	42
РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІНТЕРАКТИВНОЇ ВЕБКАРТИ	51
3.1. Методика збору та обробки вихідних даних	51
3.2. Створення вебдодатка: структура, функціональні можливості	60
3.3. Інтеграція картографічних сервісів	71
ВИСНОВКИ.....	75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
ДОДАТКИ	79

ВСТУП

Туризм є важливим сектором господарства в багатьох країнах світу, який не тільки створює нові робочі місця та збільшує доходи, але й сприяє розвитку культурного та соціального життя. З кожним роком туристичний ринок стає все більш конкурентним, і одним із ключових чинників його розвитку є забезпечення якісної та доступної інформації для туристів.

Сучасні технології відіграють важливу роль у покращенні туристичного обслуговування, і однією з найбільш ефективних інновацій у цій сфері є використання геоінформаційних систем (ГІС) для створення інтерактивних вебкарт. Такі карти дозволяють туристам легко знаходити інформацію про пам'ятки, готелі, ресторани, транспортні маршрути та інші важливі об'єкти, а також сприяють підвищенню рівня безпеки та комфорту під час подорожей.

Івано-Франківська область має значний туристичний потенціал завдяки унікальному поєднанню природних та історико-культурних ресурсів. Розробка інтерактивної вебкарти туристичних послуг для цього регіону дозволить систематизувати дані про туристичні об'єкти та зробити їх доступними для широкої аудиторії.

Метою дипломного проєкту є розробка інтерактивної вебкарти для надання туристичних послуг на прикладі Івано-Франківської області, що дозволить підвищити ефективність навігації та доступності туристичної інформації. Для досягнення цієї мети в роботі буде проведено аналіз предметної сфери, розглянуто методологічні засади створення геопросторової бази даних та розроблено технологічну схему реалізації вебкарти.

Об'єкт дослідження – інтерактивні вебкарти як засіб підвищення якості туристичного обслуговування.

Предмет дослідження – процеси та технології розробки інтерактивної вебкарти для туристичних потреб.

Методи дослідження включають аналіз літературних джерел, вивчення сучасних ГІС-технологій, моделювання бази геопросторових даних та практичне впровадження вебкарти.

Вихідними даними для цієї роботи є географічні та інфраструктурні дані зібрані із відкритих джерел інформації.

Для досягнення мети дипломної роботи необхідно вирішити такі завдання:

1. **Проаналізувати предметну сферу** – дослідити сучасний стан туристичних послуг в Україні та Івано-Франківській області.

2. **Визначити можливості застосування ГІС-технологій у сфері туризму** – розглянути існуючі методи та інструменти створення інтерактивних карт для туристичних потреб.

3. **Розробити концептуальну модель інтерактивної вебкарти** – визначити структуру та ключові функціональні можливості системи.

4. **Створити базу геопросторових даних туристичних об'єктів** – зібрати, систематизувати та підготувати інформацію про визначні місця, готелі, ресторани, транспортні маршрути та інші корисні об'єкти.

5. **Обрати технологічний стек для реалізації вебкарти** – визначити оптимальні програмні засоби для створення вебдодатка.

6. **Розробити інтерактивну вебкарту** – створити вебзастосунок із функціоналом пошуку, фільтрації та відображення туристичних об'єктів на карті.

7. **Інтегрувати картографічні сервіси та забезпечити зручний інтерфейс** – для відображення картографічної інформації.

Практична значущість роботи полягає у можливості застосування розробленої інтерактивної вебкарти для покращення туристичної інфраструктури та створення зручного цифрового сервісу. Така робота може стати корисною як для туристів, так і для комерційних установ, що займаються розвитком туризму. Створення продукту на основі використання культурно-історичних туристичних ресурсів території сприятиме розвитку туристичної галузі та буде виконувати навчально-пізнавальну, просвітницьку, патріотично-виховну, рекламну функцію.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ ТА ОБ'ЄКТУ ДОСЛІДЖЕННЯ

1.1 Туристичний потенціал Івано-Франківської області

Івано-Франківська область є одним із найперспективніших регіонів України для розвитку туризму завдяки унікальному поєднанню природних, історико-культурних та рекреаційних ресурсів. Цей регіон приваблює туристів з України та з-за кордону завдяки мальовничим Карпатам, багатій історичній спадщині та розвиненій туристичній інфраструктурі.

Івано-Франківська область за територією порівняно невелика, але за характером рельєфу належить до числа найбільш мальовничих в Україні. Третину території займають Карпатські гори, а дві третини - передгірська і рівнинна зони. Рівнинна зона знаходиться на північному сході й прилягає до Дністра. Тут знаходиться обласний центр - місто Івано-Франківськ (255 м над рівнем моря). Передгірську частину складають горбисті передгір'я. Гірська частина області зайнята Східними Карпатами, що поділяються на масиви Горгани, Черногора, Покутсько-Буковинські Карпати, Гриняви, Чивчини.

Кліматичні умови - м'яка зима із середньою температурою -5°C і тепле літо ($+18^{\circ}\text{C}$) - сприяють організації цілорічного відпочинку та лікування. Сніг на схилах Карпат лежить кілька місяців, що важливо для розвитку гірськолижного спорту.

Сьогодні більшість малих історичних міст області набули виразного європейського вигляду, зберігаючи при цьому власну історичну неповторність. Перелік та короткий опис окремих цінних пам'яток малих міст Івано-Франківської області подано у таблиці 1.1.

Найвизначніші історико-культурні об'єкти малих історичних міст Івано-Франківської області

н/п	Назва об'єкта	Дата заснування	Особливості	Місце розташування
1	Церква Пантелеймона	XII ст.	Романо-візантійський стиль, 1194 рік.	Галич
2	Успенський собор	1157 р.	Значна мистецька цінність інтер'єру та екстер'єру споруди, головний храм давнього Галича	Галич
3	Церква Серця Ісуса	1904 р.	Побудована на основі німецької кам'яної церкви, раніше була дерев'яна.	Долина
3	Церква Серця Ісуса	1904 р.	Побудована на основі німецької кам'яної церкви, раніше була дерев'яна.	Долина
4	Костел Непорочного Зачаття Діви Марії	1760 р.	Перша споруда архітектора Бернарда Меретина.	Городенка
5	Церква Пресвятої Трійці	2000 р.	Внутрішні художні роботи храму виконані митцями Львівської академії мистецтв.	Тлумач
6	Музей архітектури та побуту Прикарпаття	1979 р.	Увійшов до складу Національного заповідника "Давній Галич".	Галич
7	Музей дерев'яної архітектури -церква Святого Духа	1650 р.	Основою інтер'єру є ренесансний п'ятиярусний іконостас XVI ст.	Рогатин

8	Історико-етнографічний музей “Берегиня”	1987 р.	Експозиція знайомить з етнографією та культурою історико-географічного краю Опілля.	Бурштин
9	Косівський музей народного мистецтва і побуту Гуцульщини	1969 р.	Колекція музею нараховує близько п’яти тисяч експонатів.	Косів
10	Карпатський крайовий музей визвольних змагань	1963 р.	Представляє етапи історичного розвитку Яремчанщини.	Яремче

В області на державному обліку перебуває 3903 пам'ятки, з них: 1490 - археології, 833 - історії, 155 - монументального мистецтва. Переважна більшість пам'яток архітектури, містобудування - споруди дерев'яної і мурованої культової архітектури, пам'ятки житлового і громадського будівництва, зосередженого, в основному, в історичних містах . Проте лише невелика їхня частина використовується в сфері туризму. Технічний стан більшості пам'яток архітектури - задовільний. Деяко покращився в останні роки стан культових споруд після передачі їх релігійним громадам.



Рис.1.1.

Територіальна неоднорідність культурно-історичної спадщини справді є важливим фактором для розробки функціональних туристичних маршрутів. Вона визначає різноманіття культурних, історичних і природних об'єктів, що можуть стати основою для цікавих і унікальних подорожей, адаптованих до особливостей кожного регіону.

В Івано-Франківській області, наприклад, різні райони мають свою унікальну культурно-історичну спадщину, яку можна використовувати для створення різних типів туристичних маршрутів.

1.2 Аналіз сучасного стану туристичних послуг в Україні та регіоні

Туризм є важливою частиною економіки України, і країна має великий потенціал для розвитку цієї галузі завдяки природним, культурним і історичним ресурсам. Однак для того, щоб туризм став одним з основних джерел доходу та сприяв покращенню економічної ситуації, необхідно вирішити низку проблем, що стримують розвиток цієї сфери.

Туристичний ринок – це система взаємозв'язку, вона об'єднує велику кількість виробників туристичних продуктів з потенційними та наявними покупцями цього продукту. Також він характеризується масштабними рівнями відносин: туристичні фірми, туроператори, постачальники туристичних послуг, що є посередниками між споживачем та виробником туристичних послуг.

До початку повно-масштабної війни в Україні частка прибутку з туристичних подорожей складала вагомую частину в бюджеті держави. Туризм як вид послуг дає можливість розвитку торгових підприємств і бізнесу. До того ж розвиток туризму сприяє створенню великої кількості робочих місць і способів заробітку. З початком російського вторгнення в Україну сфера туризму значно скоротила свою роботу. Сьогодні туризм в Україні є тією галуззю, яка постраждала чи не найбільше внаслідок бойових дій. У зв'язку із проблемами, які пов'язані з війною: соціальні, економічні, матеріальні й транспортні – унеможлиблюється звичне функціонування, розвиток і вдосконалення зовнішнього й внутрішнього туризму. Ведення бойових дій на територіях України завдає чималої шкоди для туристичних об'єктів, а значить і для цілої індустрії. Особливого значення й актуальності набувають питання напрямів і факторів, які зможуть забезпечити розвиток туризму в умовах війни для створення економічного потенціалу держави у військових умовах.

Перспективами розвитку туристичної галузі є розроблення стратегії напрямів і програм, які можуть діяти в умовах воєнного стану. Такі дії держави можливі наразі лише в регіонах, які постраждали від війни менше. Якщо врахувати усі показники й стан певних туристичних інфраструктур, які знаходяться в безпечних місцевостях зараз, то є можливість залучати

туристів. Інвестиції повинні бути спрямовані із заходу на південь України. Відповідно до цього, на 2023–2025 роки є розроблені пріоритетні стратегії функціонування туристичної індустрії:

- 1) створення напрямів для розвитку національного туризму, враховуючи міжнародну практику та євроінтеграцію;
- 2) функціонування результативної розгалуженої системи управління туристичної сфери на державному й регіональному рівнях;
- 3) спрямування фінансів й залучення інвестицій до туристичної індустрії завдяки туристичним фондам;
- 4) створення туристичних об'єктів на місцевому рівні, які приносять туристичні податки державі;
- 5) створення маршрутів, які зможуть представити обсяги й масштаби військових дій;
- 6) розробка концепцій і маршрутів до курортних зон на півдні України;
- 7) створення і впровадження єдиного туристичного реєстру відповідно до змін Закону України «Про туризм»;
- 10) просування туристичних програм;
- 11) створення регіональних програм на місцевих рівнях туризму;
- 12) створення внеску туризму до ВВП країни;
- 13) застосування європейських стандартів до розвитку якості туризму;
- 14) розробка освітніх програм у сфері туристичної сфери;
- 15) розробка конкретних кроків для відновлення туристичної інфраструктури в Україні у післявоєнний час;
- 16) створення системи співпраці держави, органів місцевого самоврядування й бізнесу;
- 17) запуск національної програми підтримки продажу й просування національного туристичного продукту;

За даними Державного агентства розвитку туризму : зросла сума податків від суб'єктів туристичної діяльності в Івано-Франківській області, що дозволило Франківщині увійти до списку регіонів-лідерів. Торік тут сплатили 248 млн 469

тис. грн проти 138 млн 842 тис. грн у позаминулому році. У 2021 році регіон сплатив у казну держави 131 млн 955 тис. грн податків (Рис.1.2.1.)

АНАЛІЗ ДИНАМІКИ ПОДАТКОВИХ НАДХОДЖЕНЬ

ДАРТ

ВІД ТУРГАЛУЗИ ПО ОБЛАСТЯХ ЗА РІК (2021-2024 рр.)

ПОДАТКИ, млн грн:	2021	2022	2023	2024
Вінницька	27,4	23,9	29,9	41,3
Волинська	13,3	11,8	15,9	24,3
Дніпропетровська	98,4	69,5	98,1	128,8
Донецька	59,6	18,8	8,8	14,5
Житомирська	14,1	9,0	14,5	20,8
Закарпатська	59,5	72,4	94,2	130,4
Запорізька	98,6	26,2	15,2	25,8
Івано-Франківська	132,0	159,7	138,8	248,5
Київська	125,4	163,9	239,0	284,1
Кіровоградська	15,2	10,9	15,8	22,4
Луганська	9,2	1,9	0,5	0,2
Львівська	215,4	273,8	345,5	440,0
Миколаївська	45,9	14,3	21,7	25,6
Одеська	210,8	87,6	98,8	160,5
Полтавська	43,5	34,6	45,4	65,1
Рівненська	16,2	14,1	20,1	31,5
Сумська	11,4	7,3	9,8	14,1
Тернопільська	17,3	15,1	19,6	27,5
Харківська	104,8	38,9	49,6	47,7
Херсонська	58,7	10,4	1,7	3,0
Хмельницька	21,4	22,6	31,2	41,2
Черкаська	23,0	17,4	20,7	34,2
Чернівецька	18,2	20,7	28,8	39,1
Чернігівська	18,6	11,3	10,8	16,2
м. Київ	773,9	415,2	499,3	982,0

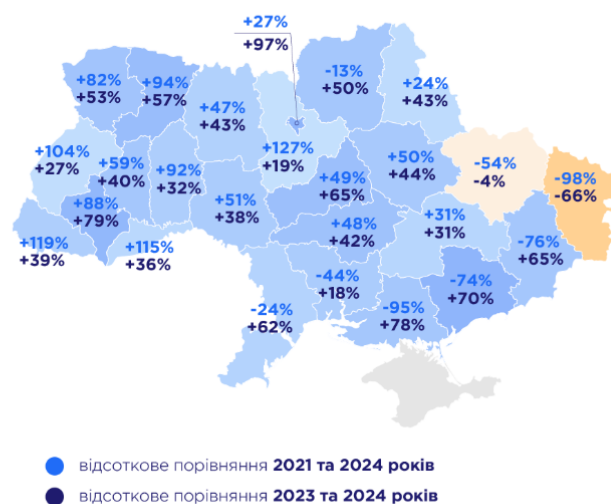


Рис.1.2.1.

Популярні туристичні напрямки в Україні:

- Закарпаття, Івано-Франківська та Львівська області є головними туристичними напрямками України, що приваблюють відпочивальників як з усієї країни, так і з-за кордону. Ці області мають багаті природні ресурси, курорти, а також культурні пам'ятки, що робить їх ідеальними для розвитку різних видів туризму.

- Гірськолижні курорти: Одним із найбільш популярних місць для зимових подорожей є Буковель -сучасний курорт з добре розвинутою інфраструктурою та лижними трасами. Крім нього, інші гірськолижні курорти, такі як Славське, також користуються великою популярністю.

- Санаторії та оздоровчі курорти: Місця для відпочинку та оздоровлення, такі як Моршин і Трускавець, приваблюють туристів, які прагнуть лікування та відновлення здоров'я.

- Зелений туризм: Останнім часом популярність набирає зелений туризм - особливо на Закарпатті, де туристи можуть зупинятися в сільських будиночках, знайомитись з місцевою культурою та традиціями.

Туризм в Україні має величезний потенціал, зокрема в таких регіонах, як Закарпаття, Івано-Франківська та Львівська області. Однак для того, щоб туризм став ефективним економічним сектором, потрібно вирішити ряд інфраструктурних та соціальних проблем, розвивати інвестиційну привабливість і підвищувати рівень обслуговування. Це дозволить не лише збільшити кількість туристів, але й сприятиме покращенню іміджу України на світовій арені.

1.3 Використання ГІС-технологій для розвитку туристичної інфраструктури

Сучасні геоінформаційні системи (ГІС) відіграють важливу роль у розвитку туристичної інфраструктури, забезпечуючи аналіз, планування та управління туристичними об'єктами й маршрутами. Використання ГІС-технологій дозволяє підвищити якість туристичних послуг, покращити доступність інформації про туристичні об'єкти та оптимізувати туристичні потоки.

Основні можливості ГІС у туризмі:

- **Картографування туристичних ресурсів**
ГІС допомагають створювати інтерактивні карти туристичних об'єктів, включаючи музеї, готелі, ресторани, пам'ятки архітектури, природні парки тощо. Наприклад, туристи можуть отримати доступ до карт з інформацією про розташування об'єктів, їхні характеристики та доступні послуги.
- **Розробка та оптимізація туристичних маршрутів**
Використання ГІС дозволяє аналізувати транспортні маршрути, розраховувати найзручніші шляхи до туристичних об'єктів, враховуючи відстань, рельєф, доступність транспорту та завантаженість доріг. Це особливо актуально для пішохідних та веломаршрутів, а також гірського туризму.
- **Моніторинг туристичних потоків**
Завдяки аналізу даних про відвідуваність можна визначати найбільш популярні місця та відповідним чином планувати інфраструктуру (парковки, транспорт, місця відпочинку тощо).
- **Забезпечення безпеки туристів**
ГІС-системи дозволяють прогнозувати небезпечні ділянки для туристів (лавини, повені, лісові пожежі), а також розробляти карти безпечних маршрутів. Це особливо важливо для гірського туризму в Карпатах.

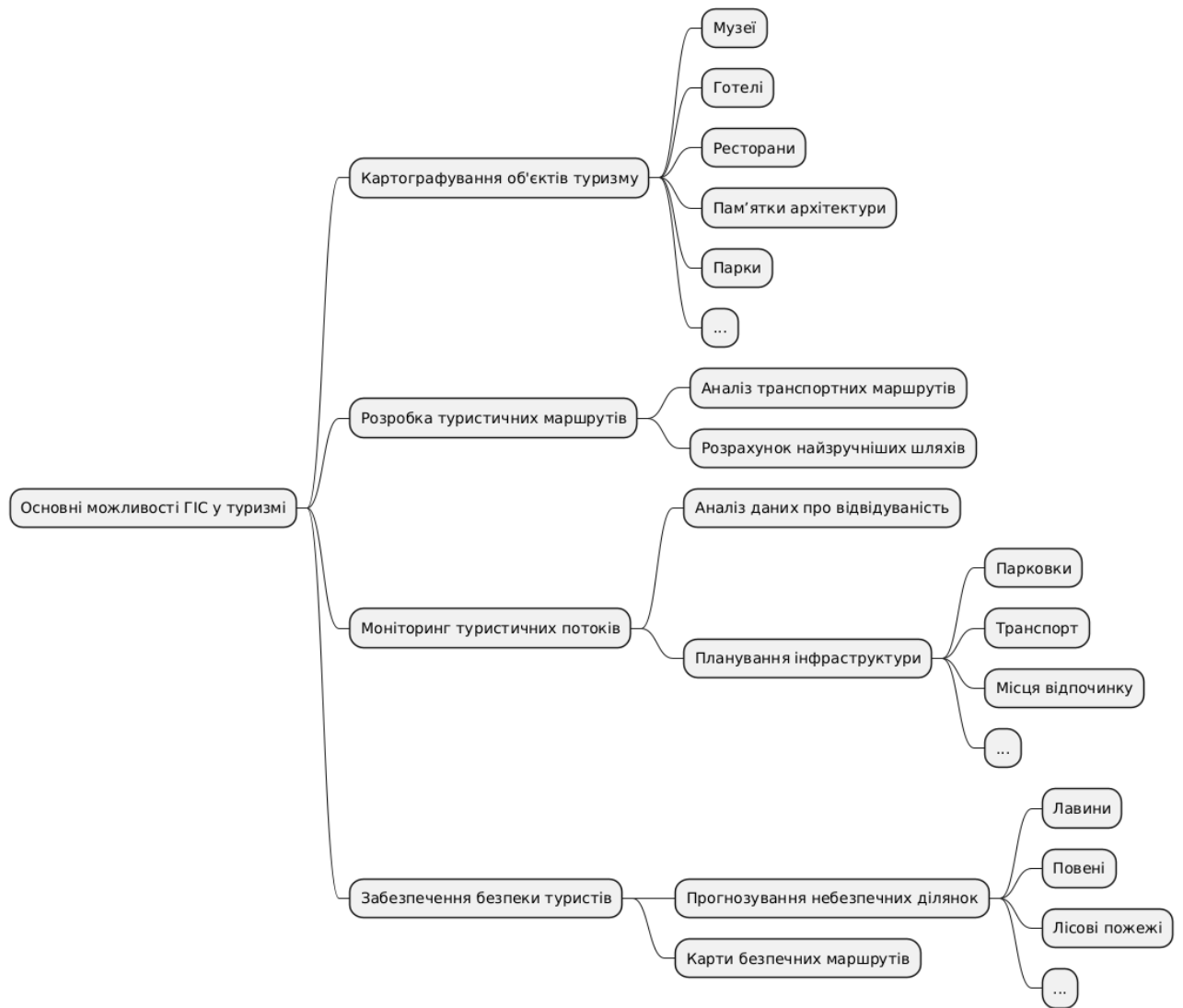


Рис.1.3.1. Схема використання ГІС у туризмі

База геопросторових даних - це система, яка зберігає, керує та надає доступ до географічної інформації, яка включає географічні об'єкти (наприклад, точки, лінії, полігональні області) та їх атрибутивні дані (наприклад, назви, характеристики). Ці дані можуть використовуватися для аналізу, візуалізації, планування та прийняття рішень у різних галузях, включаючи туризм. Проте якщо нам потрібно накопичення і збереження інформації про об'єкти та суб'єкти туризму варто застосовувати бази та банки даних, а при визначенні їх розташування доцільно використовувати просторовий аналіз і тематичне ГІС-моделювання.

Геопортал – це мережева форма реалізації ГІС. Під порталом (з англійської

portal – головний вхід, ворота), інформаційним порталом, інтернет порталом або вебпорталом розуміють вихідну точку тематичного пошуку в мережі, сервер, що надає прямий доступ користувачам до певної сукупності серверів, зокрема і встановлених на них інформаційних ресурсів, вебдодатків, які відповідають призначенню порталу.



Рис.1.3.2. Клієнт-серверна архітектура

Вебкарта - це інтерактивне відображення географічної інформації у вигляді вебсторінки (див. рис.1.3.2).

Інтерактивні карти є одними із найдинамічніших видів карт, що нині розвиваються. Класифікація та розподіл вебкарт зумовлюються сферами застосування, якісними їх характеристиками та доступністю. Інтерактивні вебкарти являють собою системи зв'язків між користувачем та сервером за допомогою мережі Інтернет, результатом якої є відображення карти або інформації у вікні браузера.

Теоретичні основи інтерактивних вебкарт у туризмі

Інтерактивні вебкарти є ключовим елементом сучасних геопорталів туризму. Вони дозволяють користувачам отримувати детальну інформацію про туристичні об'єкти, маршрути, інфраструктуру та природні пам'ятки в режимі реального часу.

Основні функції інтерактивних туристичних карт:

- **Візуалізація туристичних об'єктів** (готелі, ресторани, визначні місця, природні парки);
- **Побудова маршрутів** (автомобільних, пішохідних, вело- та гірських);
- **Пошук за категоріями** (музеї, історичні пам'ятки, екостежки тощо);
- **Додавання додаткової інформації** (фото, відео, відгуки, погодні умови);

• **Збір та аналіз даних** про потоки туристів, завантаженість об'єктів, екологічний стан територій.

Вигляд системи об'єктів туристичної інфраструктури у геоінформаційному середовищі (див. нижче).

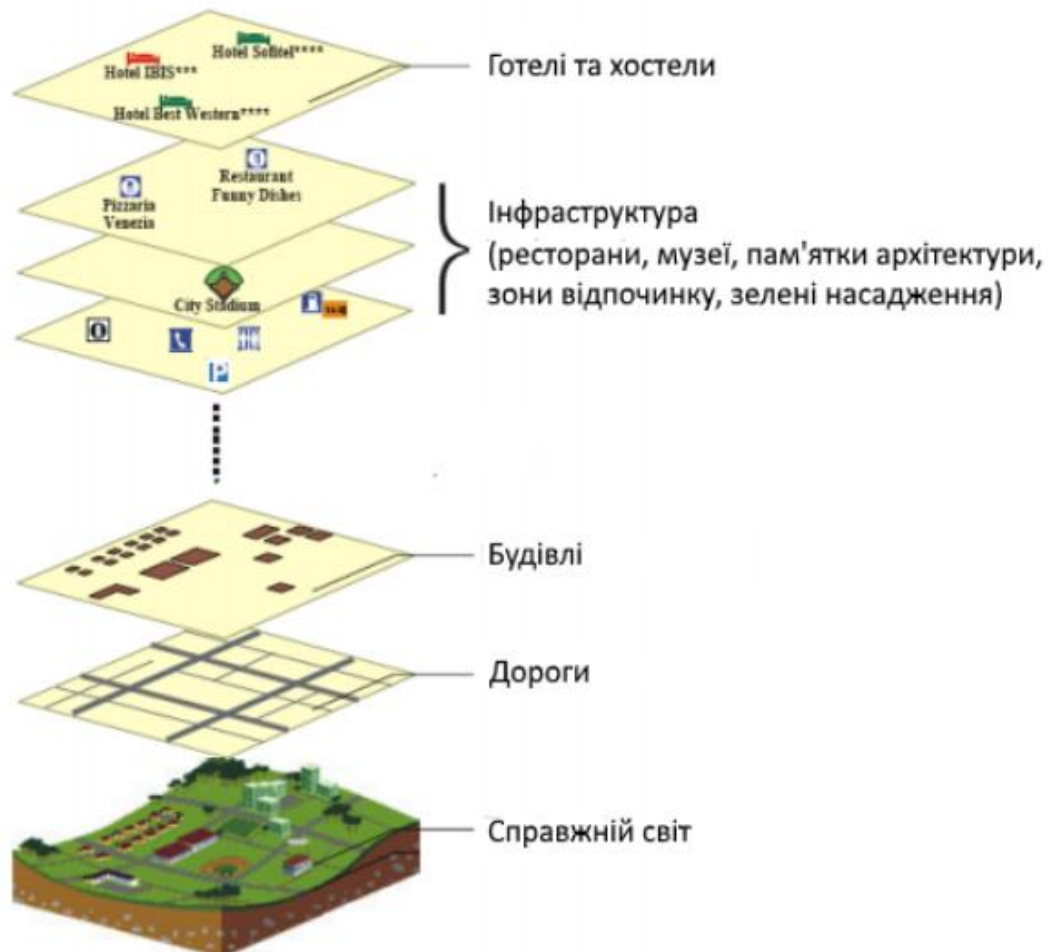


Рис.1.3.3.

Приклади інтерактивних туристичних вебкарт:

1) Сайт discover.ua використовує Google Maps у туристичних вебдодатках. Використовує Google Maps для позначення популярних туристичних локацій. Функціонал: інтеграція з маршрутами, пошук об'єктів, фільтрація за категоріями (див. нижче).

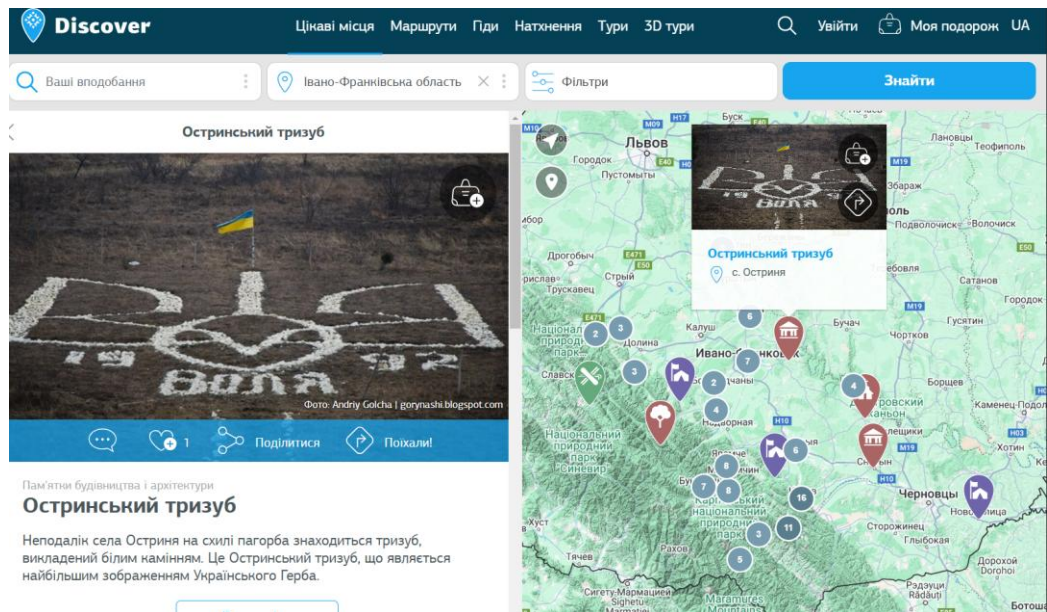


Рис.1.3.4. Сайт discover.ua

2) Геопортал "Україна Інкогніта" (OpenStreetMap + Leaflet.js)
 ukrainaincognita.com – інтерактивна карта маловідомих туристичних місць України. Функціонал: інтеграція з туристичними статтями, можливість користувацького внеску (див. нижче).

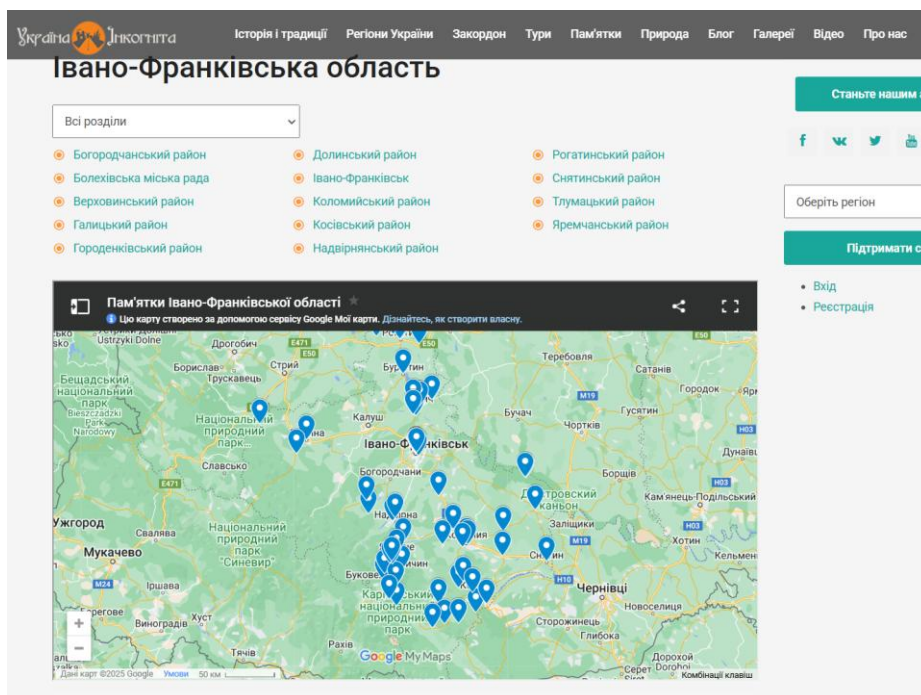


Рис.1.3.5 Геопортал "Україна Інкогніта"

3) Інтерактивна карта "Дерев'яні храми Тернопільщини" (ArcGIS Online)

Функціонал: тематичні карти, аналіз потоків туристів, управління туристичними об'єктами (див. нижче).

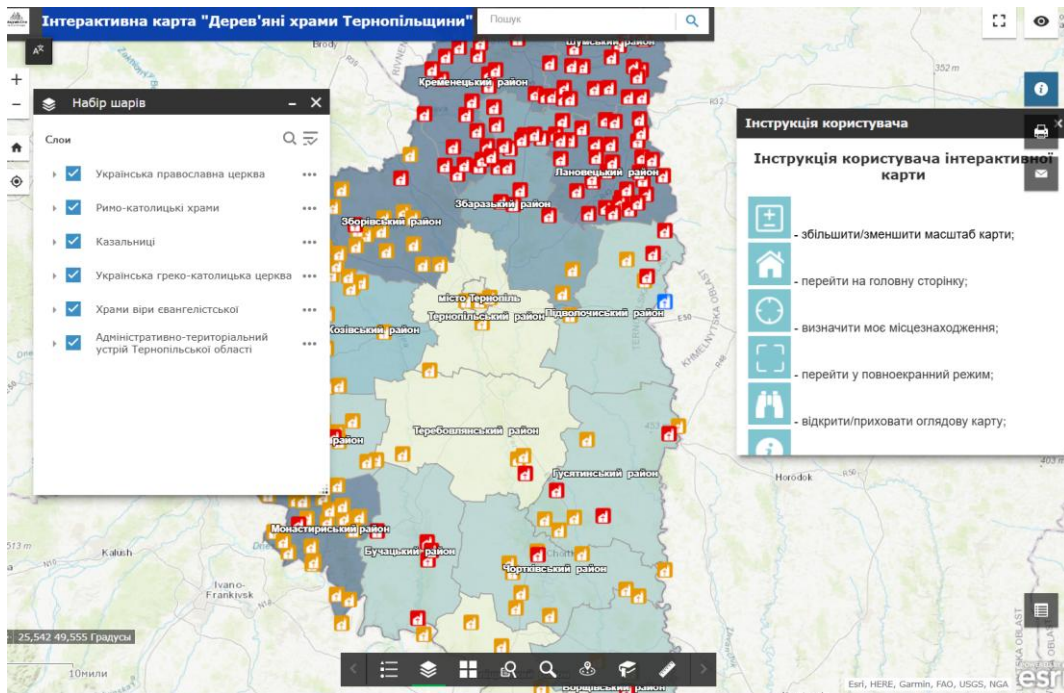


Рис.1.3.6 Інтерактивна карта "Дерев'яні храми Тернопільщини"

Принципова відмінність туристичної ГІС:

По-перше, переборюються основні недоліки звичайних туристичних паперових карт – їх статичність і обмежена ємність. Останніми роками, паперові туристичні карти через їх перезавантаженість інформацією стають нечитабельними;

По-друге, туристичні ГІС як носії інформації дозволяють формулювати не тільки прямі запити: „від властивостей у таблиці – до об'єктів на туристичній карті”, але й зворотні: „від об'єктів на туристичній карті – до властивостей об'єктів у таблиці”;

По-третє, туристичні ГІС забезпечують управління візуалізацією інформації, надаючи можливість виведення (на екран, тверду копію) тільки тих об'єктів або їх сукупностей, котрі цікавлять користувача на даний момент. У дійсності, виконується перехід від складних комплексних туристичних картографічних технологій до серії взаємопов'язаних окремих туристичних карт,

який забезпечує кращу структурованість інформації, а отже, підвищується ефективність її обробки й аналізу;

По-четверте, туристична ГІС не тільки відображає об'єкти на туристичній карті, але й, на відміну від картографічної системи, дозволяє виділяти й аналізувати об'єкти та їх взаємні зв'язки на карті, тобто здійснювати просторовий аналіз.

1.4 Нормативно-правові засади туристичної сфери в Україні

Правове регулювання туристичної діяльності в Україні здійснюють такі органи державної влади: Верховна Рада України; Кабінет Міністрів України; центральний орган виконавчої влади у сфері туризму; Верховна Рада Автономної Республіки Крим і Рада міністрів Автономної Республіки Крим; місцеві державні адміністрації; органи місцевого самоврядування; інші органи в межах їх компетенції. В умовах воєнного стану на рівні областей правове регулювання туристичної діяльності здійснюють обласні військові адміністрації, на рівні районів – районні військові адміністрації, на рівні населених пунктів – міські військові адміністрації.

Відповідно до Закону України «Про туризм» основними способами реалізації державної політики у сфері туризму, державного регулювання та управління розвитком туристичного бізнесу є: ліцензування, стандартизація та сертифікація туристичних послуг, організація та здійснення державного контролю за дотриманням законодавства.

Туризм - тимчасовий виїзд особи з місця проживання в оздоровчих, пізнавальних, професійно-ділових чи інших цілях без здійснення оплачуваної діяльності в місці, куди особа від'їжджає.

Турист - особа, яка здійснює подорож по Україні або до іншої країни з не забороненою законом країни перебування метою на термін від 24 годин до одного року без здійснення будь-якої оплачуваної діяльності та із зобов'язанням залишити країну або місце перебування в зазначений термін.

Туристичний продукт - попередньо розроблений комплекс туристичних послуг, який поєднує не менше ніж дві такі послуги, що реалізується або пропонується для реалізації за визначеною ціною, до складу якого входять послуги перевезення, послуги розміщення та інші туристичні послуги, не пов'язані з перевезенням і розміщенням (послуги з організації відвідувань об'єктів культури, відпочинку та розваг, реалізації сувенірної продукції тощо).

Туристичними ресурсами України є пропоновані або такі, що можуть пропонуватися, туристичні пропозиції на основі та з використанням об'єктів

державної, комунальної чи приватної власності. Класифікація та оцінка туристичних ресурсів України, режим їх охорони, порядок використання з обліком гранично припустимих навантажень на об'єкти культурної спадщини та довкілля, порядок збереження цілісності туристичних ресурсів України, заходи для їх відновлення визначаються відповідно до закону. Унікальні туристичні ресурси можуть знаходитися на особливому режимі охорони, що обмежує доступ до них. Обмеження доступу до туристичних ресурсів визначається їх реальною пропускною спроможністю, рівнем припустимого антропогенного навантаження, сезонними та іншими умовами.

Суб'єктами, що здійснюють та/або забезпечують туристичну діяльність (далі - суб'єкти туристичної діяльності), є:

- туристичні оператори (далі - туроператори) - юридичні особи, створені згідно із законодавством України, для яких виключною діяльністю є організація та забезпечення створення туристичного продукту, реалізація та надання туристичних послуг, а також посередницька діяльність із надання характерних та супутніх послуг і які в установленому порядку отримали ліцензію на туроператорську діяльність;

- туристичні агенти (далі - турагенти) - юридичні особи, створені згідно із законодавством України, а також фізичні особи - суб'єкти підприємницької діяльності, які здійснюють посередницьку діяльність з реалізації туристичного продукту туроператорів та туристичних послуг інших суб'єктів туристичної діяльності, а також посередницьку діяльність щодо реалізації характерних та супутніх послуг;

- інші суб'єкти господарювання, що надають послуги з тимчасового розміщення (проживання), харчування, екскурсійних, розважальних та інших туристичних послуг;

- гіді-перекладачі, екскурсоводи, спортивні інструктори, провідники та інші фахівці туристичного супроводу - фізичні особи, які проводять діяльність, пов'язану з туристичним супроводом, крім осіб, які працюють на відповідних

посадах підприємств, установ, організацій, яким належать чи які обслуговують об'єкти відвідування;

- фізичні особи, які не є суб'єктами підприємницької діяльності та надають послуги з проживання, харчування тощо.

Розглядаючи туризм як складову національної інфраструктури геопросторових даних, зазначимо, що відомості про туристичні об'єкти та об'єкти туристичної інфраструктури належать до профільних геопросторових даних (до них належать усі види галузевих і тематичних геопросторових даних, які створюються на основі базових геопросторових даних).

Законодавчою базою для цього є Закон України «Про національну інфраструктуру геопросторових даних». Цей закон визначає правові та організаційні засади створення, функціонування та розвитку національної інфраструктури геопросторових даних, спрямованої на забезпечення ефективного прийняття органами державної влади й органами місцевого самоврядування управлінських рішень, задоволення потреб суспільства в усіх видах географічної інформації, інтегрування в глобальну та європейську інфраструктуру геопросторових даних.

Національний геопортал – офіційний геопортал національної інфраструктури геопросторових даних, що забезпечує оприлюднення та доступ до геопросторових даних та метаданих

Національна інфраструктура геопросторових даних включає мережу геопорталів, причому характерною особливістю такої мережі є інформаційна взаємодія на основі спеціальних геоінформаційних сервісів, які підтримують федерацію систем баз геопросторових даних, що можуть бути географічно децентралізовані, але об'єднані в єдину мережу.

Базові геопросторові дані визначаються як загальнодоступні геопросторові дані, що складають уніфіковану цифрову координатно-просторову основу для виробництва, інтеграції та провадження іншої діяльності з різними геопросторовими даними. Набір базових геопросторових даних утворює ядро геоінформаційних ресурсів інфраструктури, завдяки якому просторово і

тематично об'єднуються всі інші геопросторові та негеопросторові: атрибутивні, профільні, тематичні дані, що спільно виробляються та використовуються в інтегрованому геоінформаційному середовищі інфраструктури. Базові геопросторові дані формуються на загальнодержавному, регіональному та місцевому рівнях. До базового набору включаються геопросторові дані, які відповідають як мінімум одному з таких критеріїв:

- придатні для використання в процесі інтеграції інформаційних ресурсів;
- забезпечують точну просторову та/або атрибутивну прив'язку тематичних даних або інших просторових об'єктів;
- мають підвищену стійкість до змін в просторі та часі;
- забезпечують зменшення обсягів атрибутивних даних постійного зберігання та скорочують витрати на їхнє введення і актуалізацію.

До складу базових геопросторових даних віднесено:

- Системи відліку координат і висот
- Державний кордон України
- Адміністративно-територіальні одиниці
- Гідрографічні об'єкти, гідротехнічні споруди
- Населені пункти, їх вулично-дорожню мережу
- Будівлі та споруди • Автомобільні дороги
- Залізниця
- Інженерні комунікації
- Аеропорти, річкові та морські порти
- Рослинний покрив та ґрунти
- Земельні ділянки
- Реєстри вулиць та адреси об'єктів
- Географічні назви
- Цифрова модель рельєфу
- Ортофотоплани

До тематичних наборів геопросторових даних належать всі види геопросторових даних, що створюються на основі базових геопросторових даних або як самостійні набори даних і відповідають вимогам стандартів на географічну інформацію та метадані, розміщені в інформаційному середовищі інфраструктури з дотриманням принципів і правил доступу та використання геоінформаційних ресурсів. Такі набори можуть створюватися органами державної влади та місцевого самоврядування, підприємствами та громадянами. Черговість створення наборів тематичних геопросторових даних визначається з урахуванням першочергових потреб суспільства, органів державної та місцевого самоврядування для забезпечення сталого розвитку, раціонального використання природних ресурсів та охорони навколишнього природного середовища. Склад наборів геопросторових даних гармонізований з вимогами Європейської інфраструктури геопросторових даних (INSPIRE)

Наказ Міністерства аграрної політики та продовольства України від 10 листопада 2021 року № 347 зареєстрований в Міністерстві юстиції України 12 січня 2022 р. за № 21/37357 “Про затвердження технічних вимог до геопросторових даних, метаданих і геоінформаційних сервісів національної інфраструктури геопросторових даних, яким затверджені:

- Технічні вимоги до метаданих для наборів геопросторових даних і геоінформаційних сервісів національної інфраструктури геопросторових даних;
- Технічні вимоги до специфікацій геопросторових даних національної інфраструктури геопросторових даних;
- Технічні вимоги до геоінформаційних сервісів геопорталів національної інфраструктури геопросторових даних;
- Технічні вимоги та методи забезпечення інтероперабельності і сумісності наборів геопросторових даних та геоінформаційних сервісів.

Елементи якості геопросторових даних Згідно з міжнародним стандартом якість геопросторових даних подається із використанням кількісних та описових елементів якості (див. нижче).



Рис.1.4.1. Елементи якості геопросторових даних

Описові елементи якості даних надають загальну не кількісну інформацію про якість. Вони дозволяють додатково оцінити придатність набору даних для конкретного застосування. Елементи "Призначення" та "Використання" описують цільову сферу застосування набору даних.

Використання набору даних визначається його розроблювачем або держателем даних. Вочевидь, що ми можемо мати уяву про різницю в очікуваній просторовій точності об'єктів в наборі даних схеми туристичних маршрутів та наборі даних навігаційних комплексів для автомобіля, або в тематичних даних земельного покриття та кадастрових даних про земельні ділянки. Елемент "Походження" описує історію формування набору даних і певною мірою його життєвий цикл, починаючи з процесів збирання, наступного кодування й перетворення у поточний формат даних.

Наприклад, дані на основі космічних знімків з просторовим розрізненням 1 м на піксель чи на основі топографічних планів масштабу 1:2000. Кількісні елементи якості геопросторових даних дають змогу оцінити, наскільки той чи інший набір даних відповідає критеріям, вказаним у специфікації дата-продукту. До кількісних елементів належать: повнота даних, логічна узгодженість, позиційна точність, тематична точність, часова точність і застосовність даних. Кожний з цих елементів якості має свої піделементи, в яких уточнюються певні аспекти елементів якості даних.

Геопросторові дані як сучасні складні інформаційні продукти з моделями реального світу потребують створення адекватних автоматизованих систем їх виробництва, управління, контролю та забезпечення якості геоінформаційної продукції та послуг.

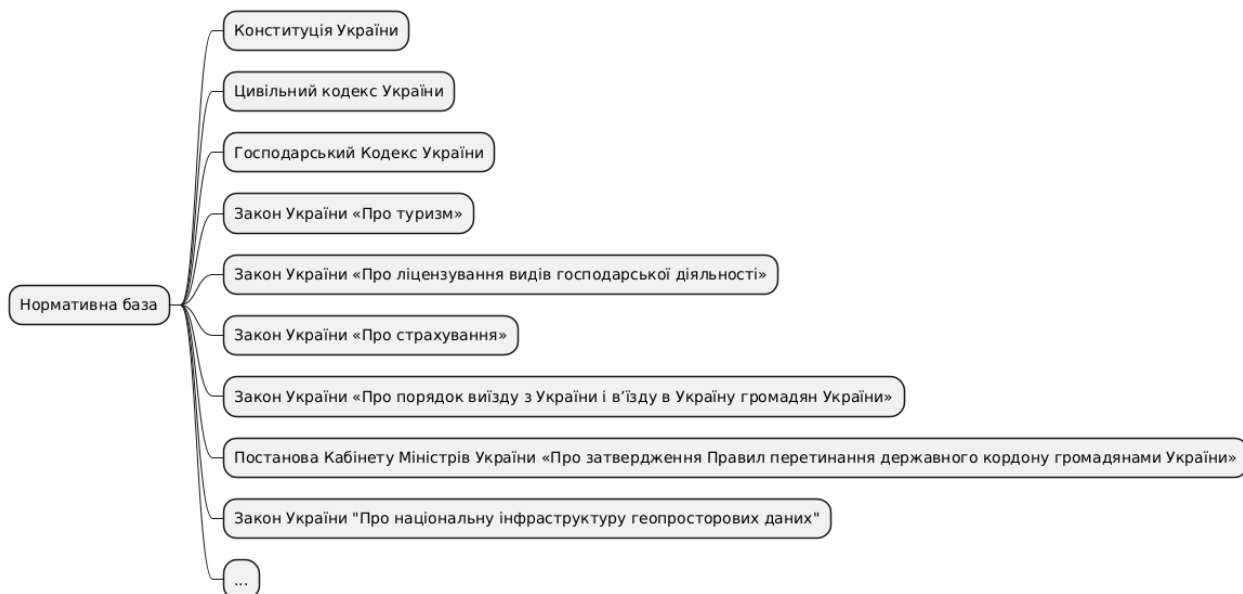


Рис.1.4.2. Нормативно-правові акти у сфері туризму

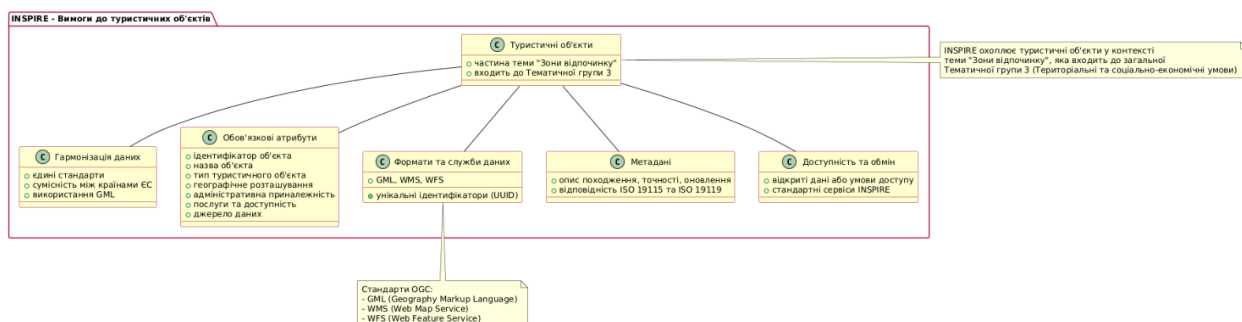


Рис. 1.4.3. Вимоги INSPIRE до туристичних об'єктів

Висновки до розділу 1

Виконано огляд туристичного потенціалу Івано-Франківської області, розглянуто популярні туристичні напрямки.

Розглянуто перспективами розвитку туристичної галузі в регіоні.

Виконано аналіз нормативно-правової бази та узгальнено у вигляді схеми (див. рис. 1.4.2).

Досліджено шляхи використання ГІС-технологій для розвитку туристичної інфраструктури та узгальнено у вигляді схеми. Розглянуто наявні туристичні вебкарти їх функціонал та реалізацію.

РОЗДІЛ 2. МЕТОДОЛОГІЧНІ ЗАСАДИ СТВОРЕННЯ ІНТЕРАКТИВНОЇ ВЕБКАРТИ

2.1. Концептуальна модель інтерактивної вебкарти туристичних послуг

Інтерактивна туристична карта – це частина геоінформаційної системи, яка створена для зберігання та відображення даних про туристичні об'єкти та інфраструктуру певного регіону або громади. Вона дозволяє переглядати маршрути, визначні місця, транспортні зв'язки та інші важливі локації для мандрівників.

До кожного туристичного об'єкта можна додати фото, відео, аудіогід, опис, паспортні дані та точні координати. Завдяки цьому карта слугує не лише навігатором, а й цифровим гідом, допомагаючи туристам швидко знаходити цікаві місця, будувати маршрути та ефективно планувати подорож, щоб відвідати якомога більше об'єктів.

Можливості туристичної вебкарти:

- відображення туристичних об'єктів, маршрутів, інфраструктури;
- переміщення по карті, зміна масштабу відображення;
- підключення та відключення додаткових даних до відображення;
- доступ до туристичної інформації з браузера;
- візуалізація даних та детальної довідкової інформації по кожному окремому об'єкту;
- виконання запитів, можливість пошуку, редагування даних.



Рис.2.1.1. Схема функцій туристичної вебкарти

Основна мета такої системи - об'єднати геопросторові дані з туристичною інформацією в єдиному зрозумілому інтерфейсі вебсторінки, що дозволяє користувачам не тільки орієнтуватися в просторі, але й отримувати глибокі знання про об'єкти, їхню історію, культурне значення та практичні аспекти відвідування. Завдяки інтерактивності, карта перетворюється на динамічний інструмент, який постійно оновлюється, враховуючи внесок користувачів і нові дані від операторів.

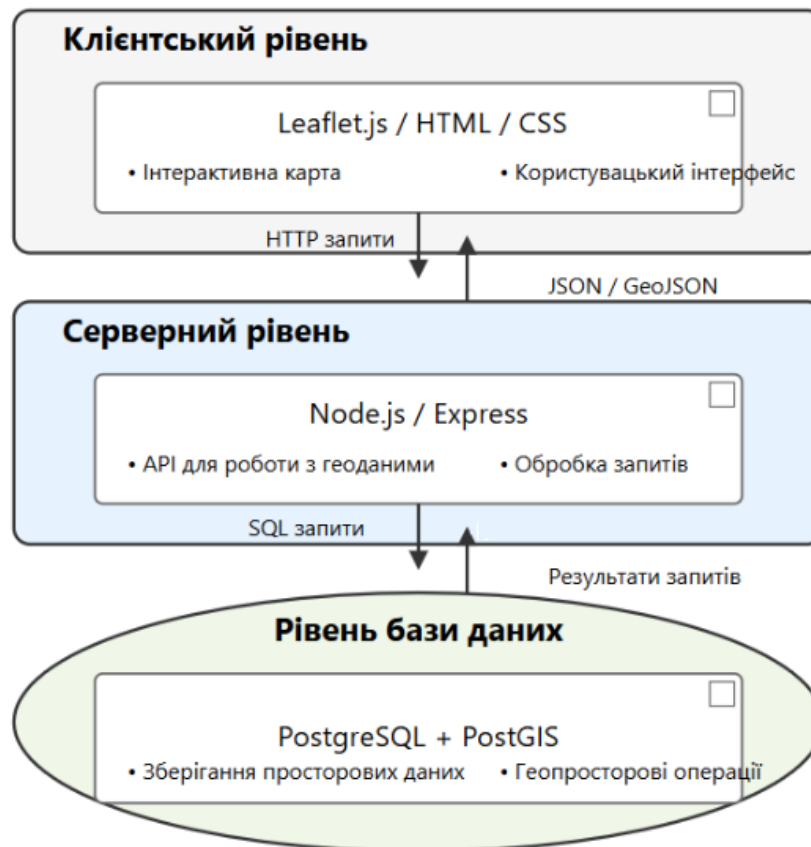


Рис. 2.1.2. Трирівнева архітектура інтерактивної вебкарти

1. Клієнтський рівень (Frontend)

Технології: Leaflet.js, HTML, CSS

Цей рівень відповідає за інтерфейс користувача, через який відбувається взаємодія з картою.

Основні функції:

Leaflet.js візуалізує карту з шарами, маркерами, маршрутами, попапами, легендою.

HTML / CSS формують структуру і зовнішній вигляд сторінки: кнопки, бокові панелі, заголовки тощо.

Користувач надсилає HTTP-запити (наприклад, на `/api/tourist_routes`) для отримання даних.

Отримані у відповідь GeoJSON / JSON-дані додаються на карту як об'єкти.

2. Серверний рівень (Backend)

Технології: Node.js, Express

Цей рівень обробляє запити з фронтенду та формує відповіді, взаємодіючи з базою даних.

Основні функції:

Express.js створює REST API, наприклад:

- /api/tourist_routes — повертає маршрути;
- /api/tourist_objects — повертає об'єкти;

Обробляє запити, наприклад:

- Фільтрація, пошук, сортування за категорією, складністю тощо;
- Формує SQL-запити до бази PostgreSQL;
- Повертає відповіді в форматі GeoJSON, зручному для карти.

3. Рівень бази даних (Database)

Технології: PostgreSQL + розширення PostGIS

Це джерело збереження всіх просторових даних про туристичні об'єкти, маршрути, зони тощо.

Основні функції:

PostGIS розширює PostgreSQL географічними можливостями:

- типи GEOMETRY, POINT, LINESTRING, POLYGON;
- функції ST_Within, ST_Intersects, ST_AsGeoJSON, ST_Distance тощо;

Зберігає інформацію про:

- назву об'єкта;
- опис, зображення, посилання;
- координати у форматі WGS84 (EPSG:4326);

Може виконувати геопросторові запити, наприклад:

- знайти всі об'єкти в межах певного радіусу;
- побудувати маршрут між точками;
- обчислити довжину маршруту.

У випадках, коли потрібна додаткова інформація або динамічне оновлення карти, API може звертатися до зовнішніх відкритих джерел. Це дозволяє:

- отримувати POI з OpenStreetMap;
- оновлювати дані без ручного введення;
- збагачувати інформацію про об'єкти.

2.2. Вибір інструментів та платформ для розробки вебкарти

Інтерактивна картографічна розробка – це картографічний твір (карта, серія карт, багатофункціональна пошарова карта, атлас) будь-якого виду (загальногеографічного, тематичного, спеціального; для атласів враховуються їхні ознаки класифікації), типу (відповідно до існуючих ознак класифікації географічних карт за типами) або призначення, що містить інтерактивні функції.

Науково-обґрунтований процес реалізації методичної схеми (алгоритму) одержання картографічних творів з інтерактивними функціями називається створенням інтерактивних картографічних розробок.

Практична реалізація вказаного процесу передбачає використання спеціалізованого програмного забезпечення, яке має функції зв'язку з картографічними серверами. При цьому важливим моментом є відмінність змісту даного поняття від визначення “інтерактивного картографування”, що асоціюється із власне процесом створення будь-якого електронного картографічного твору у діалоговому режимі з програмно-технічним забезпеченням.

Ступінь інтерактивності картографічної розробки – це реалізована на карті інтенсивність її інтерактивності.

Предметом – теоретико-методологічні та методичні основи створення інтерактивної вебкарти для надання туристичних послуг на прикладі Івано-Франківської області.

Суб'єктом створення інтерактивних карт є картограф або ІТ-фахівець, який володіє необхідними для цього компетентностями.

Реалізація інтерактивної туристичної вебкарти для області з використанням PostGIS для бази даних, Node.js для бекенду сайту та Leaflet для відображення карт -це чудовий вибір технологій, який поєднує потужність геопросторових обчислень, зручність веброзробки та легкість інтерактивної візуалізації. Розроблений план реалізації проєкту, враховуючи використання геоданих із OpenStreetMap (OSM) та державних, приватних геопорталів.

У рамках реалізації інтерактивної туристичної вебкарти області було здійснено аналіз і обґрунтований вибір відповідних інструментів та технологій.

Основна мета – створити багатофункціональний, інтерактивний вебзастосунок, здатний обробляти просторові дані, візуалізувати їх на карті, забезпечити зручну взаємодію з користувачем, а також можливість масштабування в майбутньому.

Оскільки розробка здійснюється в середовищі **Microsoft Windows**, підбір інструментів здійснювався із врахуванням сумісності з цією операційною системою.

1. Система управління базами даних: **PostgreSQL + PostGIS**

PostgreSQL -потужна об'єктно-реляційна система керування базами даних із відкритим вихідним кодом. У нашому проєкті вона обрана як основа для зберігання атрибутивних та просторових геоданих.

PostGIS -це розширення до PostgreSQL, що додає підтримку просторових типів даних (точок, ліній, полігонів) і операцій над ними, таких як фільтрація за координатами, побудова буферів, перевірка перетинів, обчислення площ тощо.

Переваги:

- Підтримка стандарту OGC (Open Geospatial Consortium)
- Висока продуктивність для запитів із геоданими
- Працює під Windows

Для встановлення використовувався дистрибутив PostgreSQL із вбудованим менеджером **StackBuilder**, який дозволяє одразу додати розширення PostGIS.

2. Серверна частина: **Node.js + Express + PostgreSQL (PostGIS)**

Node.js — це асинхронне середовище виконання JavaScript на сервері, яке ідеально підходить для побудови швидких, масштабованих вебзастосунків. У даному проєкті використовується зв'язка Node.js + Express для побудови серверної частини, а PostgreSQL з PostGIS — для зберігання геоданих.

Express.js — легкий і гнучкий фреймворк для Node.js, який дозволяє швидко реалізувати REST API, обробку запитів та інтеграцію з базами даних.

Для зберігання просторових об'єктів (точок, маршрутів) використовується PostGIS — розширення до PostgreSQL для роботи з географічними типами даних.

Переваги Node.js + Express:

- Асинхронність — висока продуктивність у роботі з API
- Простота розгортання REST-сервісів
- Відкрита екосистема npm-пакетів
- Гнучкість у структуруванні проєкту

У проєкті реалізовано API `/api/tourist-objects`, `/api/routes`, які повертають дані у форматі GeoJSON для інтеграції з фронтендом.

3. Фронтенд: **Leaflet.js**

Leaflet — популярна JavaScript-бібліотека для створення інтерактивних вебкарт. Вона дозволяє швидко і гнучко виводити геооб'єкти на карту, використовуючи як базові тайли (OpenStreetMap), так і власні шари у форматі GeoJSON.

Переваги Leaflet:

- Простота вбудовування в HTML/JavaScript
- Вбудована підтримка GeoJSON
- Підтримка кластеризації, маршрутизації, heatmap, фільтрації
- Велика кількість плагінів та прикладів

Leaflet отримує дані з REST API на Node.js (у форматі GeoJSON) і виводить їх на карту в браузері. Також реалізовано інтерактивну легенду, перемикання шарів, спливаючі вікна з фото, посиланням на Вікіпедію тощо.

4. QGIS - інструмент для обробки геоданих

QGIS - це потужна десктопна геоінформаційна система з відкритим кодом.

У проєкті вона використовується для попередньої підготовки даних:

- перегляд shapefile/GeoJSON
- очищення, класифікація об'єктів
- збереження даних у форматі, придатному для імпорту в PostGIS

Задачі:

- Валідація геоданих
- Редагування геометрій перед імпортом
- Конвертація офіційних джерел (shapefile → GeoJSON/PostGIS)

5. Імпорт та збагачення геоданих через JavaScript (Node.js)

За допомогою **Node.js-скриптів**. Це дозволяє автоматизувати процес отримання координат та зображень для об'єктів, використовуючи сучасні вебсервіси.

Імпорт координат (геокодування)

Скрипт **geocode.js** виконує геокодування назв та адрес туристичних об'єктів через **OpenStreetMap Nominatim API**. Отримані координати конвертуються в SQL-інструкції для подальшого імпорту в базу PostGIS.

Основні етапи:

- Надсилання запиту до Nominatim з адресою
- Отримання широти/довготи
- Генерація SQL-інструкції з функцією ST_SetSRID(ST_MakePoint(...), 4326)
- Автоматичне створення файлу places.sql

Переваги підходу:

- Повна автоматизація
- Легке налаштування під власні дані

- Не потребує встановлення додаткових ГІС-інструментів

Збагачення об'єктів зображеннями з Wikidata

Скрипт **update_images_by_coords.js** підключається до PostgreSQL (через pg) та:

- 1) Отримує координати об'єктів
- 2) Знаходить найближчу статтю у Вікіпедії через **Wikipedia Geosearch API**
- 3) Отримує **Wikidata QID**
- 4) Через **Wikidata API** знаходить зображення (P18)
- 5) Формує URL зображення з Wikimedia Commons і записує його у поле `gx_media_links`

Підключені сервіси:

- Wikipedia Geosearch API
- Wikipedia PageProps

Переваги:

- Динамічне збагачення об'єктів мультимедіа
- Відсутність ручної обробки
- Висока актуальність та достовірність джерел

Такий підхід дозволяє реалізувати повноцінну систему обробки геоданих виключно на JavaScript/Node.js, не використовуючи сторонні інструменти або мови. Це ідеальне рішення для веборієнтованих проєктів на сучасному стеку.

Отримання геоданих через Overpass API

Для інтеграції публічних геооб'єктів (кафе, парки, лікарні, пам'ятки тощо) у проєкті використовується **Overpass API** — спеціальний сервіс для доступу до даних OpenStreetMap за допомогою запитів на мові Overpass QL.

Запити Overpass API дозволяють:

- Отримувати **точки інтересу** (POI) за фільтрами (тип об'єкта, тег, географічні координати)
- Працювати з великим масивом відкритих даних без необхідності локального зберігання
- Накладати отримані шари на карту через Leaflet

У реалізації:

- Запити формуються динамічно через JavaScript
- Відповідь конвертується з OSM-JSON у GeoJSON за допомогою бібліотеки osmtogeojson
- Кожен об'єкт виводиться на карту з відповідною іконкою, попопом, і за потреби - зображенням з Wikimedia

Переваги Overpass API:

- Доступ до актуальних даних OpenStreetMap
- Не потребує локальної БД
- Гнучкий синтаксис для пошуку за тегами ("amenity"="cafe", "leisure"="park" тощо)
- Можливість поєднувати з власними шарами (API або PostGIS)

6. Інтегроване середовище розробки: PyCharm

PyCharm - професійне середовище розробки для Python, зручне для роботи з node.js, шаблонами, віртуальними середовищами та контролем версій (Git).

Причини вибору:

- Інтеграція з терміналом, інтерпретатором, БД
- Робота з HTML/CSS/JS в одному середовищі

Обрані технології забезпечують гнучку, масштабовану та сучасну архітектуру вебдодатку (див. рис. 2.2.1). Всі інструменти є кросплатформенними, але у розробці проєкту основною платформою виступає Windows 11, що є зручною для локальної розробки, тестування та інтеграції з програмним забезпеченням типу QGIS.

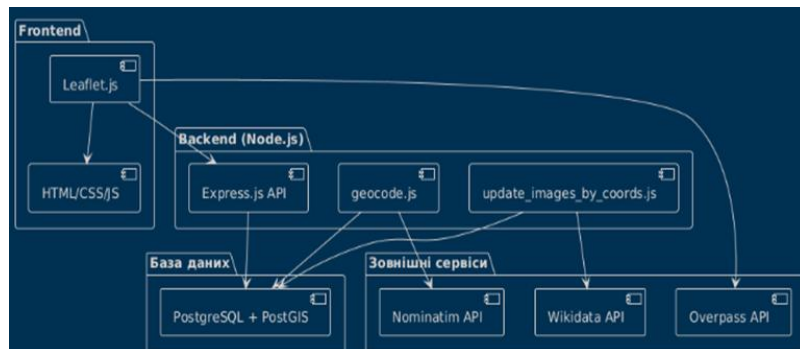


Рис.2.2.1. UML діаграма пакетів програмної реалізації вебкарти

Таблиця 2.2.1. Вибрані інструментів для розробки

Назва програми	Призначення	Переваги
PostgreSQL	Реляційна СУБД для зберігання атрибутивних даних	Надійність, відкритий код, підтримка транзакцій
PostGIS	Розширення до PostgreSQL для роботи з геопросторовими типами даних	OGC-сумісність, високопродуктивні просторові запити
Node.js	Серверне середовище виконання JavaScript для побудови бекенду	Асинхронність, масштабованість, підтримка npm
Express.js	Фреймворк для Node.js для побудови REST API	Швидкий запуск API, простота структури
Leaflet.js	JS-бібліотека для створення інтерактивних вебкарт	Легкість у використанні, підтримка GeoJSON, розширюваність
QGIS	Десктопна ГІС для перегляду, редагування та підготовки геоданих	Візуалізація, редагування, перевірка геометрій
geocode.js	Node.js-скрипт для геокодування адрес через Nominatim API	Автоматизація координат, безкоштовний API
update_images_by_coords.js	Скрипт для збагачення об'єктів зображеннями через Wikipedia/Wikidata API	Автоматизація, достовірність джерел
Overpass turbo	Інтерфейс до OpenStreetMap для запити POI (кав'ярні, парки тощо)	Не потребує локального зберігання, фільтрація за тегами
PyCharm	Інтегроване середовище розробки для роботи з JS, HTML, Python та Node.js	Інтеграція з Git, терміналом, підтримка проєктів
Windows 11	Операційна система, на якій розгорнуто розробку	Сумісність з QGIS, PostgreSQL, Node.js, простота налаштувань
HTML/CSS	HTML та CSS дозволяють швидко зібрати функціональний інтерфейс	Легкість інтеграції з Leaflet.js

2.3. Формування бази геопросторових даних туристичних об'єктів

Предметною сферою для розробки бази даних у контексті концептуальної моделі інтерактивної вебкарти туристичних послуг є туристична інфраструктура та ресурси певної області чи регіону. Це охоплює туристичні об'єкти (історичні пам'ятки, природні ландшафти, музеї, готелі, ресторани тощо), маршрути, транспортні зв'язки та пов'язані з ними послуги, які мають значення для розвитку туризму. Особливу увагу приділено інтеграції геопросторових даних із туристичною інформацією, щоб забезпечити зручний доступ до об'єктів, їхньої характеристики та можливостей для мандрівників. Така база даних слугує основою для створення інтерактивної вебкарти, яка поєднує навігаційні функції з цифровим гідом для туристів.

У контексті інтерактивної вебкарти туристичних послуг база даних виступає основою для реалізації всіх ключових функцій: відображення об'єктів на карті, пошуку за категоріями, побудови маршрутів та інтеграції мультимедійного контенту. Вона забезпечує зв'язок між геоданими (координатами об'єктів) і туристичною інформацією (описи, фото, відгуки), створюючи цілісну систему, яка підтримує як туристів, так і стейкхолдерів туристичної галузі. Такий підхід дозволяє не лише зберігати дані, але й активно використовувати їх для підвищення привабливості регіону як туристичного напрямку.

Чому важливо зберігати інформацію про туристичні послуги в базі даних:

1. Систематизація та облік

База даних дозволяє систематизувати інформацію про туристичні об'єкти, маршрути та інфраструктуру, створюючи єдиний структурований реєстр. Це допомагає уникнути дублювання даних, спрощує їх оновлення та забезпечує точний облік туристичних ресурсів регіону.

2. Зручний доступ до інформації

Завдяки базі даних користувачі (туристи, оператори туризму, місцева влада) можуть швидко отримувати актуальну інформацію про об'єкти, їхнє

розташування, характеристики та доступні послуги через інтерактивну вебкарту, що підвищує зручність планування подорожей.

3. Оперативність в управлінні та моніторингу

Центральне зберігання даних у базі забезпечує оперативний доступ до інформації для моніторингу стану туристичних об'єктів, аналізу відвідуваності та виявлення проблемних зон.

4. Наукова та громадська цінність

Інформація, зібрана в базі даних, може бути використана для наукових досліджень (наприклад, аналізу туристичних трендів чи впливу туризму на економіку регіону), а також для залучення громадськості до популяризації місцевих пам'яток через відгуки, рейтинги та обмін досвідом на платформі вебкарти.

У контексті інтерактивної вебкарти туристичних послуг база даних виступає основою для реалізації всіх ключових функцій: відображення об'єктів на карті, пошуку за категоріями, побудови маршрутів та інтеграції мультимедійного контенту. Вона забезпечує зв'язок між геоданими (координатами об'єктів) і туристичною інформацією (описи, фото, відгуки), створюючи цілісну систему, яка підтримує як туристів, так і стейкхолдерів туристичної галузі (див. рис. 2.2.2).

Такий підхід дозволяє не лише гармонізовано зберігати дані, але й активно використовувати їх для підвищення привабливості регіону як туристичного напрямку.

Нижче наведено перелік таблиць та їх атрибутів для бази даних інтерактивної вебкарти туристичних послуг, з акцентом на туристичні маршрути та пов'язані об'єкти. Для кожної сутності вказано назву атрибутів, типи даних та коментарі українською мовою, які пояснюють їхнє призначення. Сутності адаптовано до теми туристичних маршрутів і враховують геопросторові дані для інтеграції з PostGIS.

Таблиця 1: Туристичні об'єкти (TouristObjects)

Таблиця зберігає інформацію про окремі туристичні об'єкти, які можуть бути частиною маршрутів.

Атрибут	Тип даних	Коментар українською мовою
id	SERIAL	Унікальний ідентифікатор об'єкта
name	VARCHAR(255)	Назва туристичного об'єкта
category	VARCHAR(100)	Категорія об'єкта (музей, парк, готель тощо)
description	TEXT	Опис об'єкта (історія, особливості)
geom	GEOMETRY(Point, 4326)	Географічні координати об'єкта (точка)
address	VARCHAR(255)	Адреса об'єкта (якщо є)
working_hours	VARCHAR(100)	Графік роботи об'єкта
entry_fee	DECIMAL(10, 2)	Вартість входу (у гривнях, якщо застосовується)
created_at	TIMESTAMP	Дата створення запису

Таблиця 2: Туристичні маршрути (TouristRoutes)

Ця таблиця описує маршрути, які з'єднують туристичні об'єкти.

Атрибут	Тип даних	Коментар українською мовою
id	SERIAL	Унікальний ідентифікатор маршруту
name	VARCHAR(255)	Назва маршруту

description	TEXT	Опис маршруту (особливості, рекомендації)
route_geom	GEOMETRY(LineString, 4326)	Геометрія маршруту (лінія з координатами)
length	DECIMAL(10, 2)	Довжина маршруту (у кілометрах)
duration	INTERVAL	Тривалість маршруту (у годинах і хвилинах)
difficulty	VARCHAR(50)	Рівень складності (легкий, середній, важкий)
category	VARCHAR(100)	Тип маршруту (пішохідний, велосипедний, авто)
created_at	TIMESTAMP	Дата створення запису

Таблиця 3: Зв'язок маршрутів і об'єктів (RouteObjects)

Ця таблиця пов'язує маршрути з туристичними об'єктами, вказуючи, які об'єкти входять до маршруту та їхню послідовність.

Атрибут	Тип даних	Коментар українською мовою
id	SERIAL	Унікальний ідентифікатор зв'язку
route_id	INTEGER	Ідентифікатор маршруту (зовнішній ключ)
object_id	INTEGER	Ідентифікатор туристичного об'єкта (зовнішній ключ)

sequence	INTEGER	Порядок об'єкта в маршруті
distance_to_next	DECIMAL(10, 2)	Відстань до наступного об'єкта (у кілометрах)

Таблиця 4: Користувачі (Users)

Ця таблиця зберігає дані про користувачів, які можуть додавати маршрути чи залишати відгуки.

Атрибут	Тип даних	Коментар українською мовою
id	SERIAL	Унікальний ідентифікатор користувача
username	VARCHAR(100)	Ім'я користувача
email	VARCHAR(255)	Електронна пошта
password_hash	VARCHAR(255)	Хеш пароля для авторизації
role	VARCHAR(50)	Роль (турист, адміністратор, оператор)
created_at	TIMESTAMP	Дата створення запису

Таблиця 5: Відгуки (Reviews)

Таблиця дозволяє зберігати відгуки користувачів про маршрути або об'єкти.

Атрибут	Тип даних	Коментар українською мовою
id	SERIAL	Унікальний ідентифікатор відгуку
user_id	INTEGER	Ідентифікатор користувача (зовнішній ключ)

route_id	INTEGER	Ідентифікатор маршруту (зовнішній ключ, опціонально)
object_id	INTEGER	Ідентифікатор об'єкта (зовнішній ключ, опціонально)
rating	INTEGER	Оцінка (від 1 до 5)
comment	TEXT	Текст відгуку
created_at	TIMESTAMP	Дата створення відгуку

Таблиця 6: Мультимедіа (Multimedia)

Зберігає фото, відео чи аудіо, пов'язані з об'єктами або маршрутами.

Атрибут	Тип даних	Коментар українською мовою
id	SERIAL	Унікальний ідентифікатор мультимедіа
route_id	INTEGER	Ідентифікатор маршруту (зовнішній ключ, опціонально)
object_id	INTEGER	Ідентифікатор об'єкта (зовнішній ключ, опціонально)
file_url	VARCHAR(255)	Посилання на файл (URL)
file_type	VARCHAR(50)	Тип файлу (photo, video, audio)
description	TEXT	Опис мультимедійного файлу
uploaded_at	TIMESTAMP	Дата завантаження

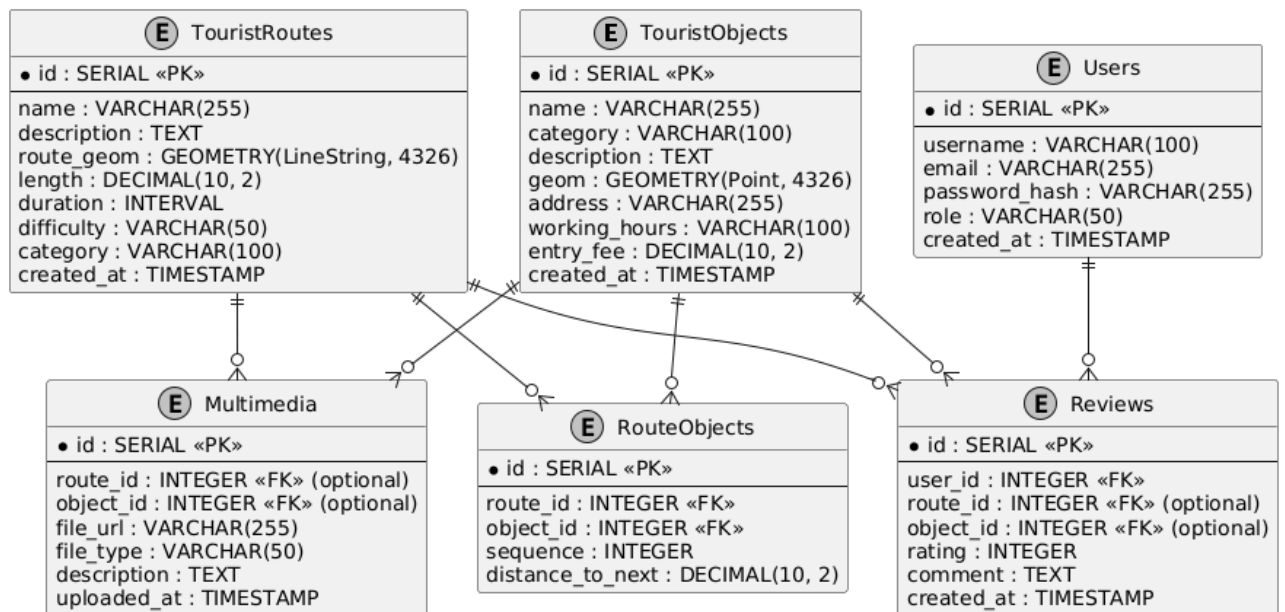


Рис.2.2.2. UML Діаграма класів Бази геопросторових даних туристичного сайту

Створення бази даних

- База даних називається `tourist_db`.
- Використовується команда `\c tourist_db` для підключення до бази (у `psql`).

Розширення PostGIS:

- Встановлюється розширення `postgis` для підтримки типів `GEOMETRY` (наприклад, `Point` і `LineString` з `SRID 4326 - WGS 84`).

Таблиці:

- Кожна таблиця відповідає сутності з логічної моделі.
- Первинні ключі (`id`) створені з типом `SERIAL` для автоматичного інкременту.
- Зовнішні ключі пов'язують таблиці з відповідними обмеженнями (`ON DELETE CASCADE` або `ON DELETE SET NULL` залежно від логіки).

Обмеження:

- У таблиці `users` додано унікальність для `username` і `email`.
- У таблиці `reviews` і `multimedia` додано обмеження `CHECK`, щоб відгук або мультимедіа стосувалися лише одного об'єкта або маршруту (або жодного).
- Для `rating` у `reviews` встановлено діапазон 1–5.

- Для `role` і `file_type` використано CHECK для обмеження допустимих значень.

Індекси:

- Додано індекси для геометричних полів (`geom`, `route_geom`) з використанням GIST для швидкого просторового пошуку.

- Індекси для зовнішніх ключів (`user_id`, `route_id`, `object_id`) для оптимізації запитів.

Представлено ER-модель бази даних (створену за допомогою **pgAdmin**), яка описує структуру взаємозв'язаних сутностей: туристичних об'єктів, маршрутів, користувачів тощо (див. рис. 2.2.3).

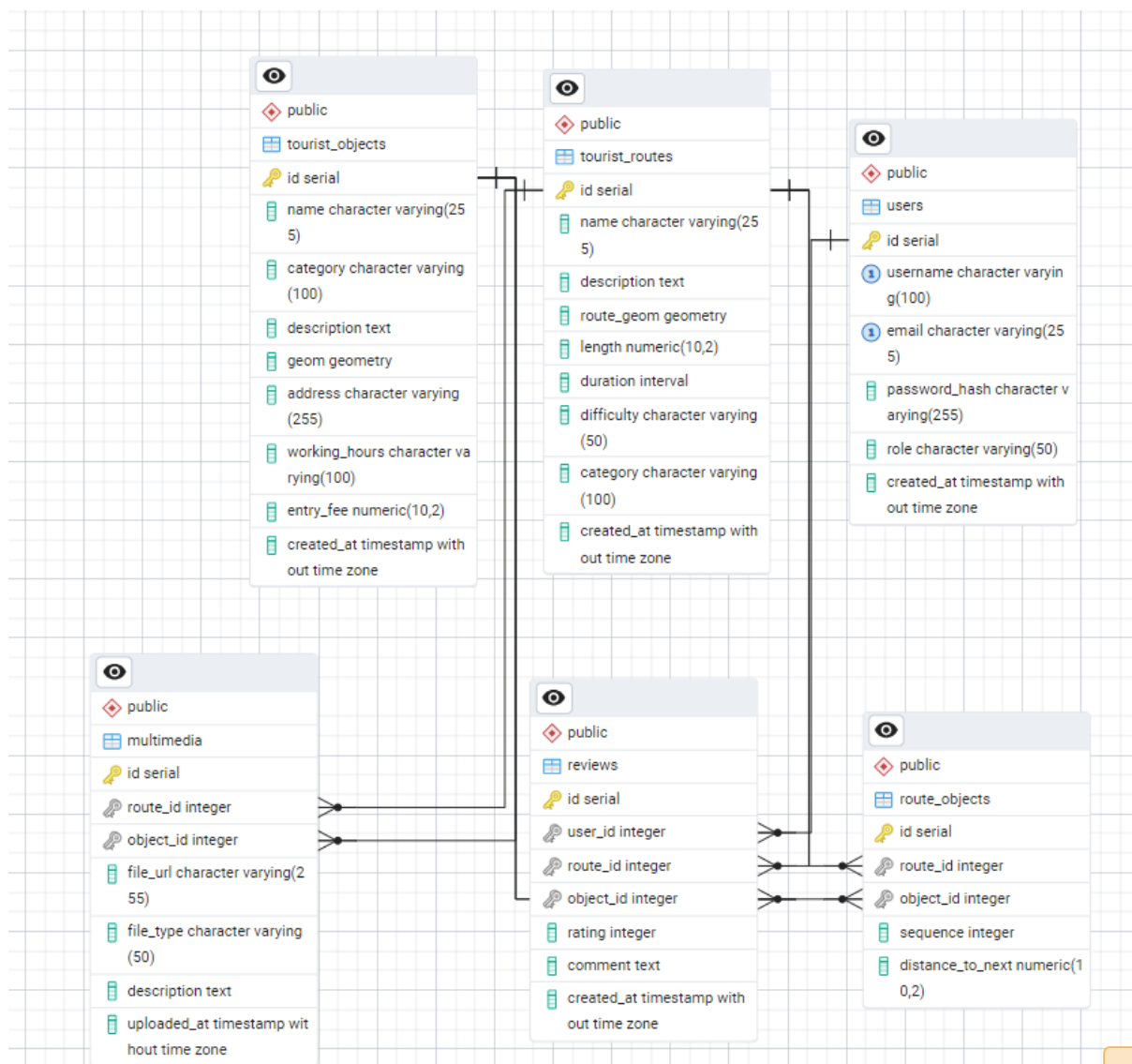


Рис.2.2.3. Візуалізація створеної БГД вебкарти в середовищі pgadmin

Висновки до розділу 2

Інтерактивна вебкарта реалізована як клієнт-серверна система з візуалізацією, пошуком, фільтрацією та динамічним оновленням даних через API. Метою було забезпечити зручний доступ до туристичних ресурсів у просторовому форматі. Узагальнено схемою (рис. 2.1.2).

Обрано технології: PostgreSQL + PostGIS (база даних), Node.js + Express (бекенд), Leaflet.js (фронтенд). Допоміжні засоби - QGIS, Overpass API, PyCharm. Обраний стек забезпечує гнучкість, масштабованість, інтерактивність і підтримку просторових операцій, що ідеально підходить для вебкартографії. Створено UML діаграму пакетів програмної реалізації вебсайту(Рис.2.2.1).

Створена база даних містить туристичні об'єкти, маршрути, мультимедіа, відгуки, користувачів. Вона забезпечує основу для візуалізації, пошуку та аналітики. Базу даних показано UML Діаграмою класів(Рис.2.2.2).

РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІНТЕРАКТИВНОЇ ВЕБКАРТИ

3.1. Методика збору та обробки вихідних даних

Створення інтерактивної туристичної вебкарти вимагає збору, обробки та інтеграції великого обсягу географічних даних. На цьому етапі важливою складовою є використання різноманітних джерел геопросторових даних, що включають як відкриті джерела, так і спеціалізовані дані від державних геопорталів. Оскільки основна мета полягає в тому, щоб забезпечити користувачів точними і актуальними відомостями про туристичні об'єкти, маршрути та інфраструктуру, методика збору та обробки даних передбачає кілька етапів.

Збір вихідних даних

Основними джерелами геопросторових даних для створення картографічної інформації є:

- 1) **OpenStreetMap (OSM)** – це відкрите джерело географічних даних, яке містить точні дані про географічні об'єкти, такі як туристичні пам'ятки, маршрути, дороги, інфраструктура (готелі, ресторани, туалети тощо). Дані OSM є безкоштовними та можуть бути використані для комерційних і некомерційних цілей. Для збору даних використовуються сервіси **Overpass API**, **Wikidata API** та **Overpass Turbo**, що дозволяють створювати запити для отримання точок, ліній та полігонів, які позначають туристичні об'єкти і маршрути.
- 2) **Державні геопортали** – офіційні геоінформаційні портали, які надають доступ до даних про об'єкти інфраструктури, природоохоронні зони, охоронні території, дороги, ландшафти, а також деталі про туристичні маршрути та пам'ятки. Для України це портали на кшталт **Geoportal.gov.ua** або **OpenData.gov.ua**. Дані з цих джерел можуть бути завантажені у форматах **shapefile** або **GeoJSON**.
- 3) **Інші відкриті дані** – спеціалізовані платформи, що публікують картографічні дані для визначених територій, наприклад, дані про гірські

маршрути, велосипедні доріжки тощо. Ці дані можуть бути використані для покращення карти, зокрема для додавання менш популярних об'єктів або для оновлення актуальної інформації.

Обробка вихідних даних

Обробка вихідних даних включає кілька етапів, що забезпечують коректне інтегрування даних у систему:

1. **Конвертація форматів даних:** Геопросторові дані можуть бути представлені в різних форматах, таких як **shapefile**, **GeoJSON**, **KML** тощо. Для того, щоб імпортувати ці дані в базу даних **PostGIS**, їх необхідно перетворити в формат, сумісний з PostgreSQL. Для цього використовується бібліотека **GDAL/OGR**, яка підтримує багато форматів і дозволяє імпортувати дані до бази даних за допомогою командного рядка або Python-скриптів.
2. **Попередня обробка даних:** Після конвертації даних важливо перевірити їх на наявність помилок, таких як непотрібні дублікатні записи, неповні атрибути чи неправильні географічні координати. Для цього використовуються геоінформаційні системи, такі як **QGIS**, які дозволяють візуалізувати дані на карті і здійснити їх перевірку. У разі потреби можуть бути застосовані фільтри для видалення або коригування некоректних об'єктів.
3. **Очищення даних:** У процесі обробки даних може знадобитися їх очищення від зайвих атрибутів або нормалізація даних для уніфікації формату. Наприклад, дані про один об'єкт можуть бути представлені в різних форматах в різних джерелах, тому їх потрібно привести до одного стандарту.
4. **Інтеграція з базою даних:** Після того, як дані очищено, вони завантажуються до бази даних **PostGIS**. Кожен об'єкт (пам'ятка, маршрут, інфраструктурний об'єкт) отримує унікальний ідентифікатор, географічні координати, а також атрибути (назва, опис, категорія тощо). За допомогою SQL-запитів можна створювати індекси для швидкого пошуку об'єктів на

карті, а також виконувати просторові операції, наприклад, пошук найближчих об'єктів або перевірку перетину маршрутів.

Інструменти для збору та обробки даних

Для ефективної роботи з геопросторовими даними в межах проєкту використано низку інструментів, які забезпечують повний цикл: від підготовки та конвертації до зберігання, оновлення і візуалізації даних.

1. QGIS – це вільно поширювана геоінформаційна система, що використовується для:

- візуалізації, аналізу та редагування геоданих;
- перевірки даних на коректність;
- конвертації форматів (наприклад, shapefile у GeoJSON);
- прямого підключення до бази PostGIS для редагування просторових об'єктів.

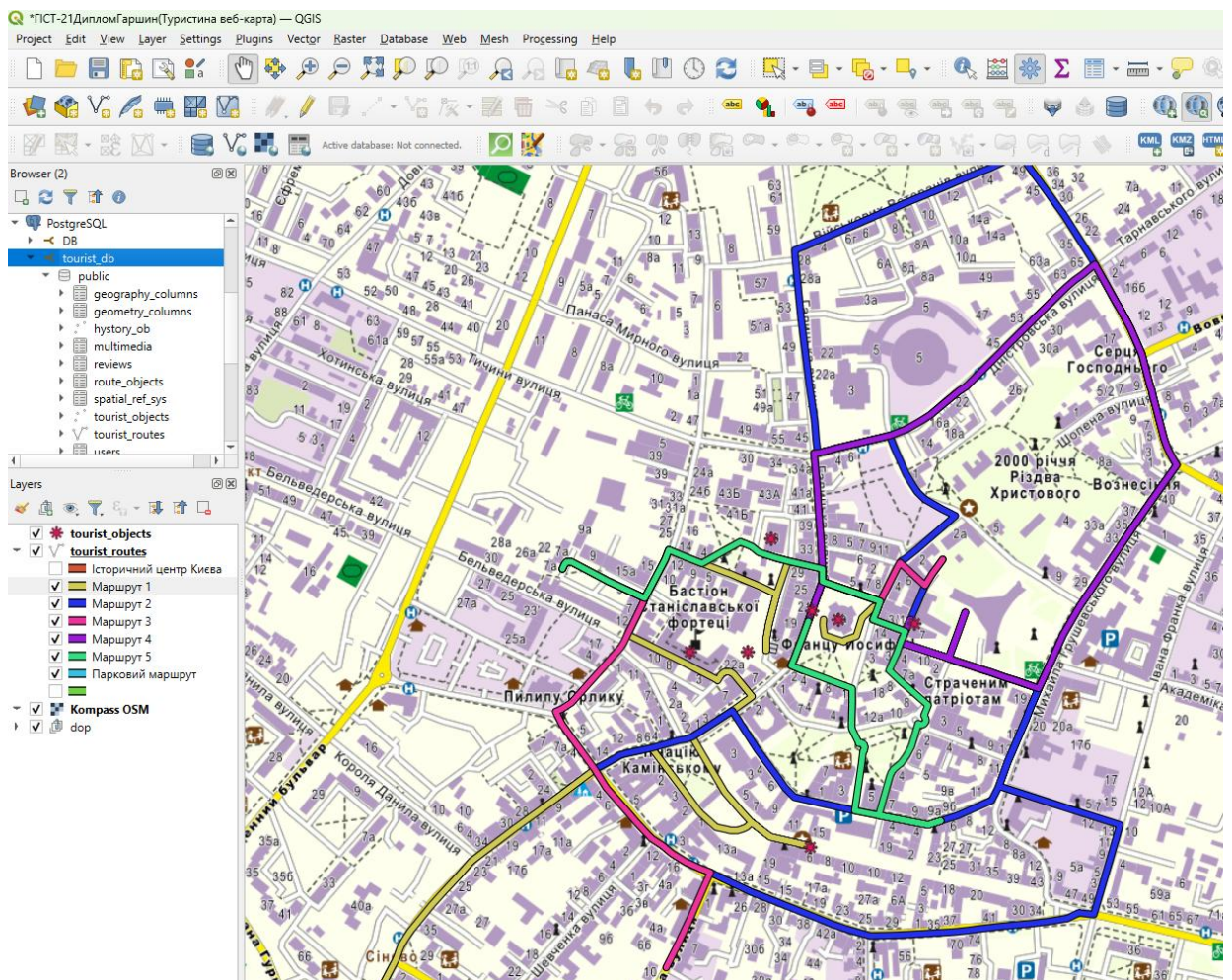


Рис. 3.1.1 QGIS обробка даних з підключеною ГБД postgres

2. GDAL/OGR

GDAL/OGR – набір консольних утиліт, призначених для:

- конвертації геопросторових файлів у різні формати;
- масового імпорту даних (зокрема GeoJSON або shapefile) у базу даних PostGIS;
- підтримки великої кількості просторових форматів (включно з GML, KML, GeoPackage, тощо).

3. PostGIS

PostGIS – розширення до реляційної бази даних PostgreSQL, яке забезпечує:

- зберігання просторових об'єктів (точок, ліній, полігонів);
- виконання просторових запитів, обчислення буферів, площ, перетинів;
- зручну інтеграцію з клієнтськими інструментами (QGIS, GeoServer, Leaflet).

4. Overpass API

Overpass API – програмний інтерфейс доступу до даних (див. рис. 3.1.2)

OpenStreetMap, що дозволяє:

- виконувати фільтрацію за тегами (наприклад, "tourism"="attraction");
- отримувати вибірки об'єктів у вигляді JSON чи GeoJSON;
- застосовувати результати до візуалізації в клієнтській частині.

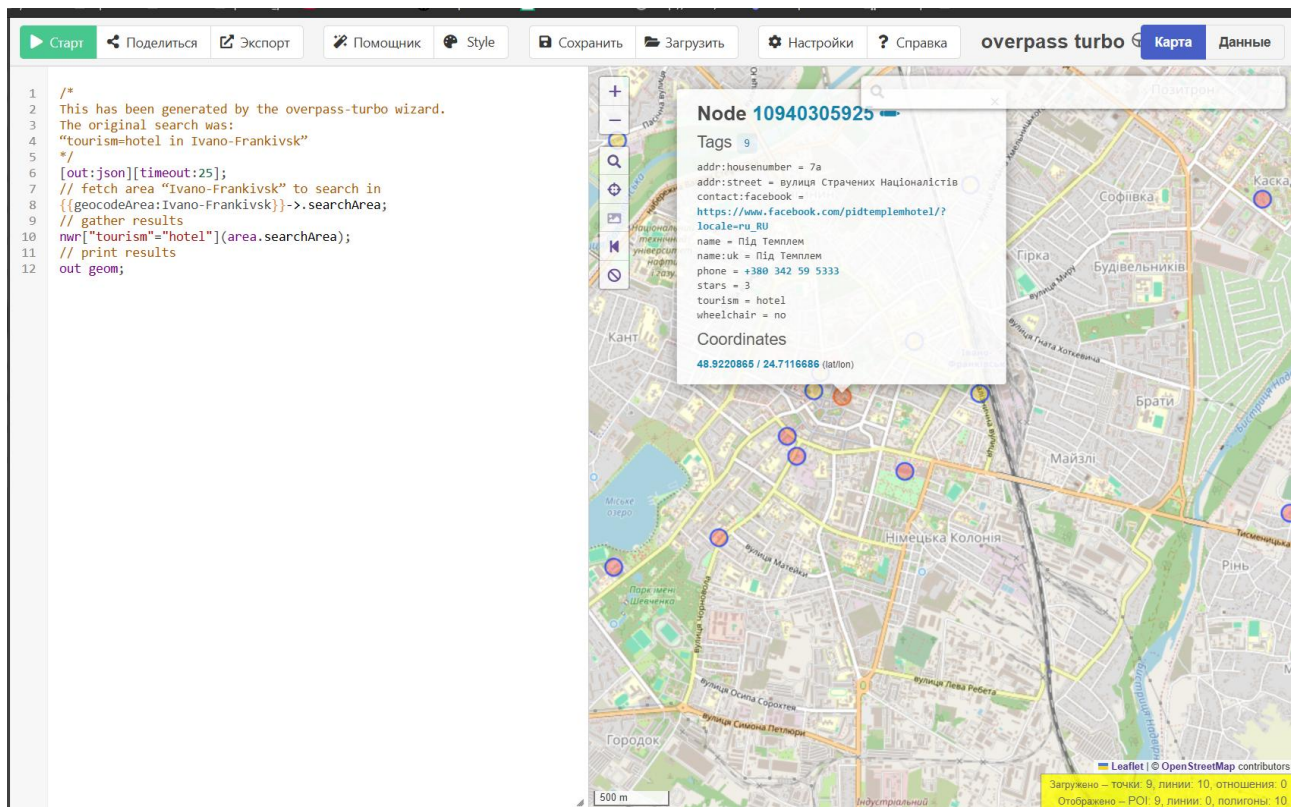


Рис.3.1.2 Інтерфейс Overpass Turbo

5. JavaScript-скрипти для автоматизованої обробки

У межах проєкту реалізовано два окремі скрипти на JavaScript, які автоматизують підготовку просторових даних та мультимедійне збагачення об'єктів.

Цей підхід забезпечує швидке наповнення бази геопросторових даних без ручного введення координат за адресами назвами. Код скрипта **geocode.js** можна переглянути в додатку Є.



Рис.3.1.3 UML Діаграма станів Логіка роботи geocode.js

```

PS C:\Users\msMax\Desktop\DIPL0M\!Диплом_ГІСТ21_Гаршин_Додаток1_ДатаПроектів\leaflet-postgres-map> node geocode.js
✓ Local History Museum → 48.9227606, 24.7103868
▲ Не знайдено координати для: Cathedral of the Holy Resurrection
▲ Не знайдено координати для: Carpathian Art Museum

✓Файл places.sql збережено!
PS C:\Users\msMax\Desktop\DIPL0M\!Диплом_ГІСТ21_Гаршин_Додаток1_ДатаПроектів\leaflet-postgres-map> #
  
```

Рис. 3.1.4 Виконання скрипта geocode.js

Скрипт виконує автоматизоване геокодування туристичних об'єктів. Його функції:

- отримання координат об'єктів з використанням OpenStreetMap Nominatim API;
- формування SQL-запитів на основі координат з геометричними полями;
- створення файлу places.sql для подальшого імпорту в PostGIS.

Також у роботі було використано JavaScript-скрипт `update_images_by_coords.js`

```
const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'tourist_db',
  password: 'postgres',
  port: 5432,
});

async function getArticleByCoords(Lat, Lon) {
  const url = `https://en.wikipedia.org/w/api.php?action=query&format=json&origin=*&list=geosearch&gscoord=${Lat}/${Lon}&gsradius=10000&gslimit=1`;
  const res = await fetch(url);
  const data = await res.json();
  const result = data.query.geosearch[0];
  return result?.title || null;
}

async function getWikidataId(title) {
  const url = `https://en.wikipedia.org/w/api.php?action=query&format=json&origin=*&prop=pageprops&titles=${encodeURIComponent(title)}`;
  const res = await fetch(url);
  const data = await res.json();
  const page = Object.values(data.query.pages)[0];
  return page?.pageprops?.wikibase_item || null;
}

async function getImageFromWikidata(qid) {
  const url = `https://www.wikidata.org/wiki/Special:EntityData/${qid}.json`;
  const res = await fetch(url);
  const data = await res.json();
  const entity = data.entities[qid];
  const imageName = entity?.claims?.P18?.[0]?.mainsnak?.datavalue?.value;
  if (!imageName) return null;

  // Створюємо URL зображення
  const fileName = encodeURIComponent(imageName.replace(/ /g, '_'));
  const md5 = require('crypto').createHash('md5').update(fileName).digest('hex');
  const urlPath = `${md5[0]}/${md5.substring(0, 2)}/${fileName}`;
  return `https://upload.wikimedia.org/wikipedia/commons/${urlPath}`;
}

async function updateImagesByWikidata() {
  const { rows } = await pool.query('SELECT id, ST_Y(geom) AS Lat, ST_X(geom) AS Lon FROM hystory_ob');

  for (const row of rows) {
    try {
      const title = await getArticleByCoords(row.Lat, row.Lon);
      if (!title) {
        console.log(` ⚠ No article near ID ${row.id}`);
        continue;
      }

      const qid = await getWikidataId(title);
      if (!qid) {
        console.log(` ⚠ No Wikidata ID for: ${title}`);
        continue;
      }

      const imageUrl = await getImageFromWikidata(qid);
      if (imageUrl) {
        await pool.query(
          'UPDATE hystory_ob SET gx_media_links = $1 WHERE id = $2',
          [imageUrl, row.id]
        );
        console.log(` ✅ [${title}] → Image from Wikidata`);
      } else {
        console.log(` ❌ No image on Wikidata: ${qid}`);
      }
    } catch (err) {
      console.error(` ❌ Error for ID ${row.id}:`, err.message);
    }

    await new Promise(res => setTimeout(res, 1000));
  }

  console.log(' 🎉 Успіх! Фото з Wikidata записано в базу!');
  await pool.end();
}
```

Рис.3.1.5 Фрагмент коду `update_images_by_coords.js`

Скрипт збагачує записи у базі даних зображеннями з відкритих джерел.

Основні етапи:

- пошук найближчої статті у Вікіпедії за координатами об'єкта;
- отримання унікального ідентифікатора об'єкта (QID) через Wikidata API;

- доступ до властивості P18 (зображення) у відповідному об'єкті Wikidata;
- формування URL-адреси до файлу на Wikimedia Commons та запис у поле `gx_media_links`.



Рис. 3.1.6. UML Діаграма станів Загальна логіка скрипта

Таким чином, до кожного об'єкта автоматично додається зображення, що візуально збагачує інтерфейс користувача.

Обидва скрипти реалізують автоматизований підхід до заповнення просторової бази даних та дозволяють уникнути помилок, пов'язаних із ручною обробкою великої кількості об'єктів.

```

Terminal Local x + v
No image on Wikidata: Q3188550
[Krylos] → Image from Wikidata
[Krylos] → Image from Wikidata
[Kolomyia] → Image from Wikidata
No image on Wikidata: Q746631
[Burshtyn] → Image from Wikidata
[Rohatyn] → Image from Wikidata
[Pukiv] → Image from Wikidata
[Lopushnia] → Image from Wikidata
[Dolyna] → Image from Wikidata
node update_images_by_coords.js
  
```

Рис. 3.1.7 Виконання скрипта `update_images_by_coords.js`

6. **My Google Maps** – онлайн-інструмент для створення та візуалізації маршрутів (див. рис.3.1.8). За допомогою цього сервісу створювалися пішохідні маршрути між туристичними об'єктами. Кожен маршрут містив позначки з назвами та описами об'єктів. Створені шари експортувалися у форматі **KML**, який надалі імпортувався у QGIS або безпосередньо в PostgreSQL/PostGIS через відповідну обробку.

CSV-файли – використовувалися як проміжний формат для імпорту точкових даних з координатами та описами у My Maps.

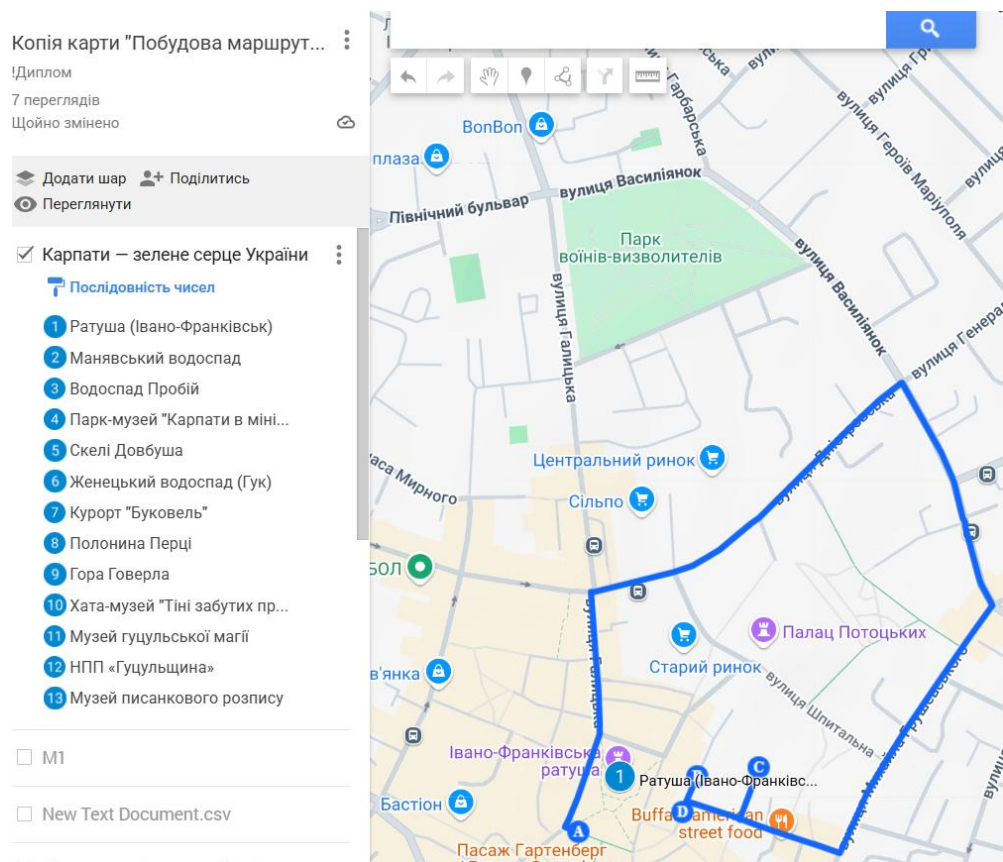


Рис.3.1.8. Інтерфейс My Google Maps

Збір та обробка вихідних геоданих

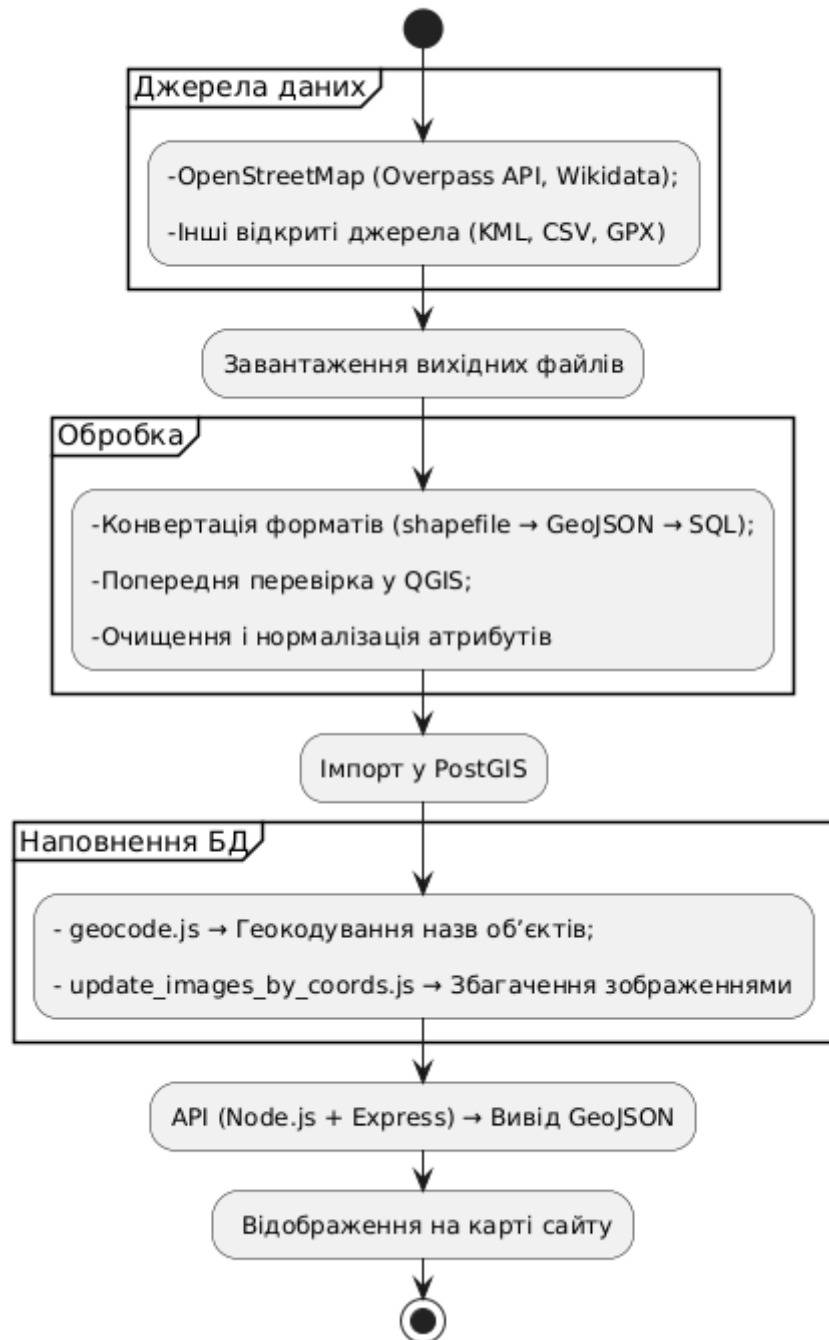


Рис.3.1.8 UML Діаграма станів збору та обробки вихідних геоданих

3.2. Створення вебдодатка: структура, функціональні можливості

Вебзастосунок реалізовано за принципом класичної серверної архітектури, де основний обмін даними відбувається через передачу інформації у вигляді контексту до HTML-шаблонів. Завдяки цьому, більшість даних передаються з сервера до браузера, або в окремих API-запитах.

Архітектура вебдодатка

Розроблений вебзастосунок побудовано за архітектурою клієнт-сервер з чітким розділенням на frontend та backend частини.

Frontend частина відповідає за інтерфейс користувача та взаємодію з картографічними даними, в той час як backend відповідає за обробку запитів, бізнес-логіку та взаємодію з базою даних.

Клієнтська частина вебдодатку реалізована з використанням сучасних технологій:

- HTML/CSS для розмітки та стилізації сторінок
- JavaScript для клієнтської логіки
- Leaflet.js для роботи з інтерактивними картами
- Основні компоненти клієнтської частини:
- Інтерфейс користувача з можливістю перегляду та вибору туристичних маршрутів
- Система фільтрації маршрутів за різними критеріями
- Інтерактивна карта з відображенням туристичних маршрутів
- Інформаційні панелі з описом маршрутів та пам'яток

Структура серверної частини (Backend)

Серверна частина реалізована на Node.js з використанням фреймворку Express.js. Основні компоненти backend:

- Express.js Controllers для обробки HTTP-запитів
- REST API для взаємодії з клієнтською частиною
- Система керування базою даних PostgreSQL з розширенням PostGIS.

Проект має наступну **структуру файлів** node.js (див. нижче):

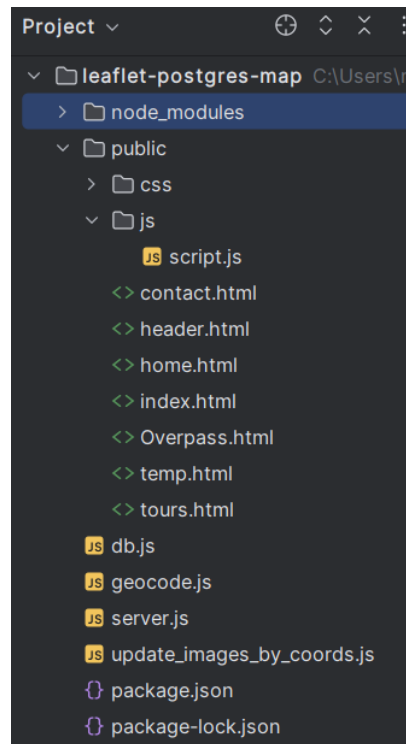


Рис. 3.2.1. Скіншот проекту в Pycharm

Підключення до бази даних

Для зберігання географічних даних використовується PostgreSQL з розширенням PostGIS.

Основні типи даних у базі:

- Туристичні маршрути (геометрія ліній)
- Точки інтересу (геометрія точок)
- Інформація про маршрути (назва, опис, дата створення, тощо)

API додатку забезпечує взаємодію з базою даних за допомогою SQL-запитів, що дозволяє ефективно отримувати і фільтрувати дані за різними критеріями.

Взаємодія з базою даних PostgreSQL здійснюється через модуль pg для Node.js.

Підключення конфігурується в файлі db.js (див. рис.3.2.2):



```
db.js x <> header.html
const { Pool } = require('pg');

const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'tourist_db',
  password: 'postgres',
  port: 5432,
});

module.exports = pool;
```

Рис. 3.2.2. Скіншот кода db.js

Серверна частина реалізована з використанням Express.js та запускається через файл server.js (див. додаток Ж). Основні функціональні можливості:

- Налаштування та запуск HTTP-сервера
- Обробка статичних файлів
- Маршрутизація запитів до відповідних контролерів
- Обробка REST API запитів для отримання, додавання та редагування туристичних маршрутів
- Обробка GeoJSON даних для відображення на карті

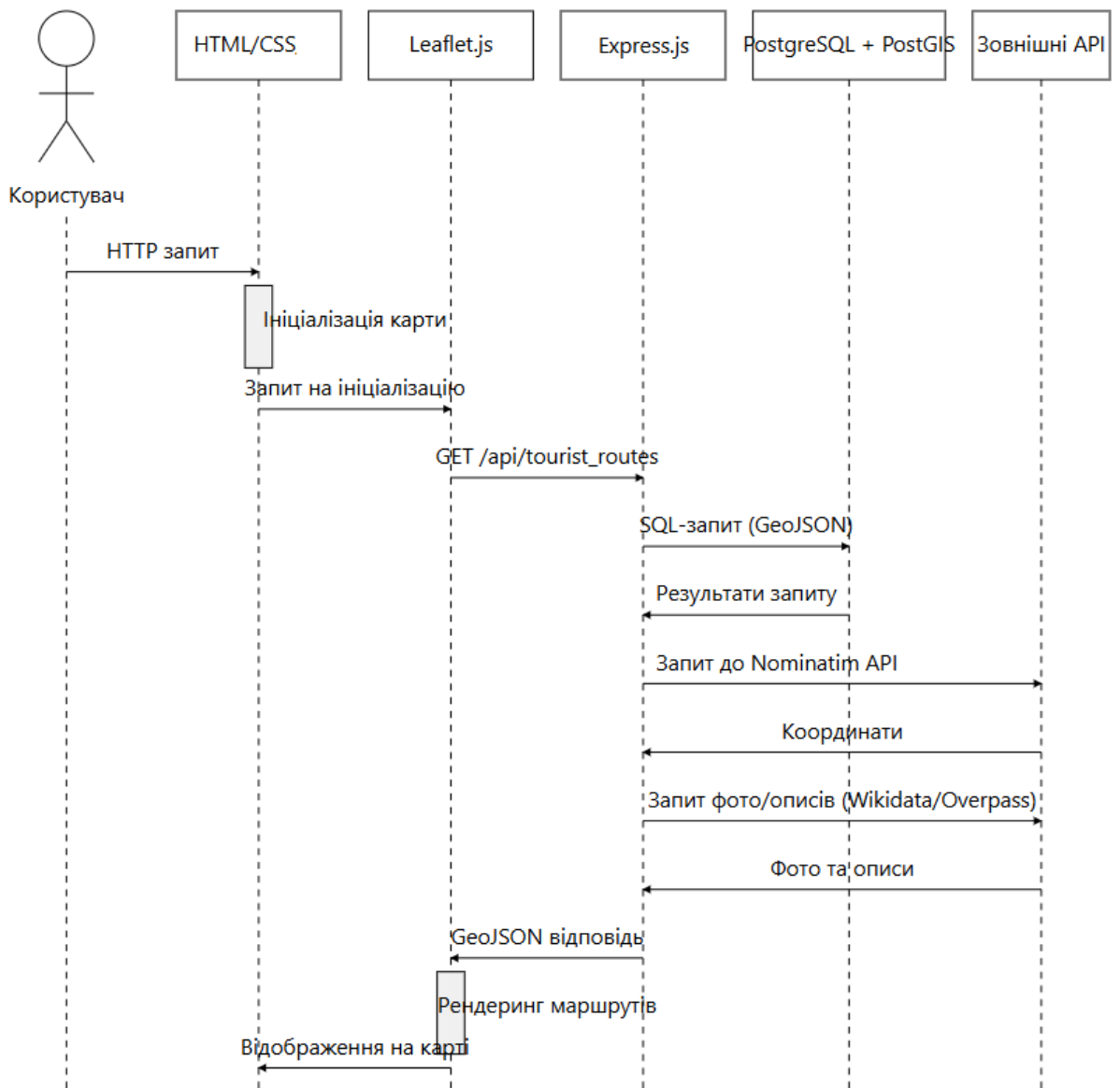


Рис. 3.2.3. UML Діаграма послідовності взаємодії між компонентами системи

Клієнтська частина реалізована через систему HTML шаблонів з використанням `iframe` для організації структури сторінки:

```
<> temp.html x db.js <> header.html <> tours.html style1.css <> contact.html
1 <!DOCTYPE html>
2 <html lang="uk">
3 <head>
4 <meta charset="utf-8">
5 <title>Диалом ГІСТ-21 Гаршин Максим</title>
6 <style...>
35 </head>
36 <body>
37 <!-- Верхній хедер -->
38 <iframe id="header" src="header.html" name="header" scrolling="no"></iframe>
39
40 <!-- Меню ліворуч -->
41 <iframe id="menu" src="tours.html" name="iframe_left"></iframe>
42
43 <!-- Основний контент праворуч -->
44 <iframe id="content" src="index.html" name="iframe_right"></iframe>
45 </body>
46 </html>
```

Рис. 3.2.4. Код головної HTML сторінки

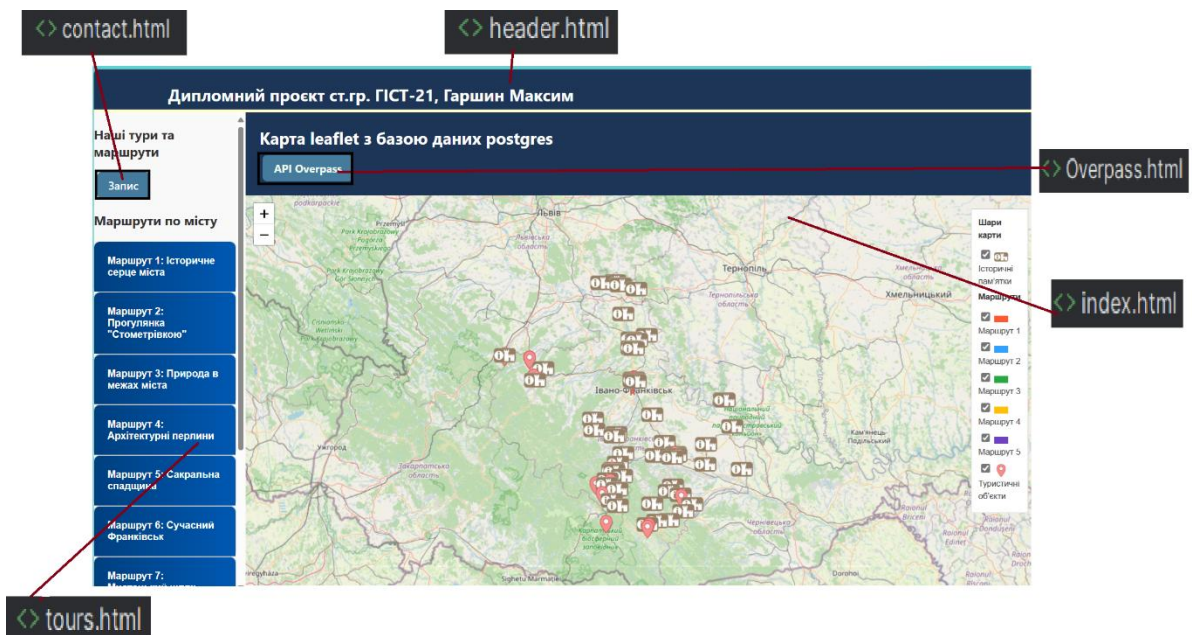


Рис. 3.2.5. Структура фрейма з набором HTML-сторінок

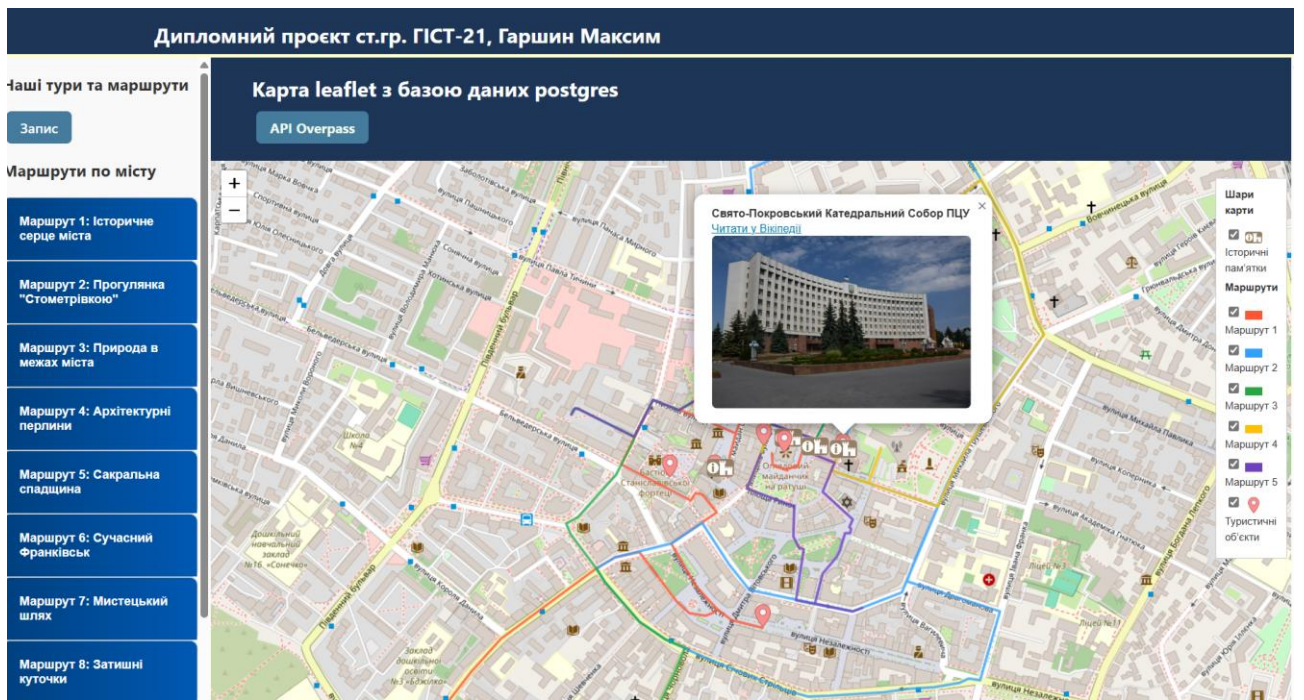


Рис. 3.2.6. Вигляд в браузері

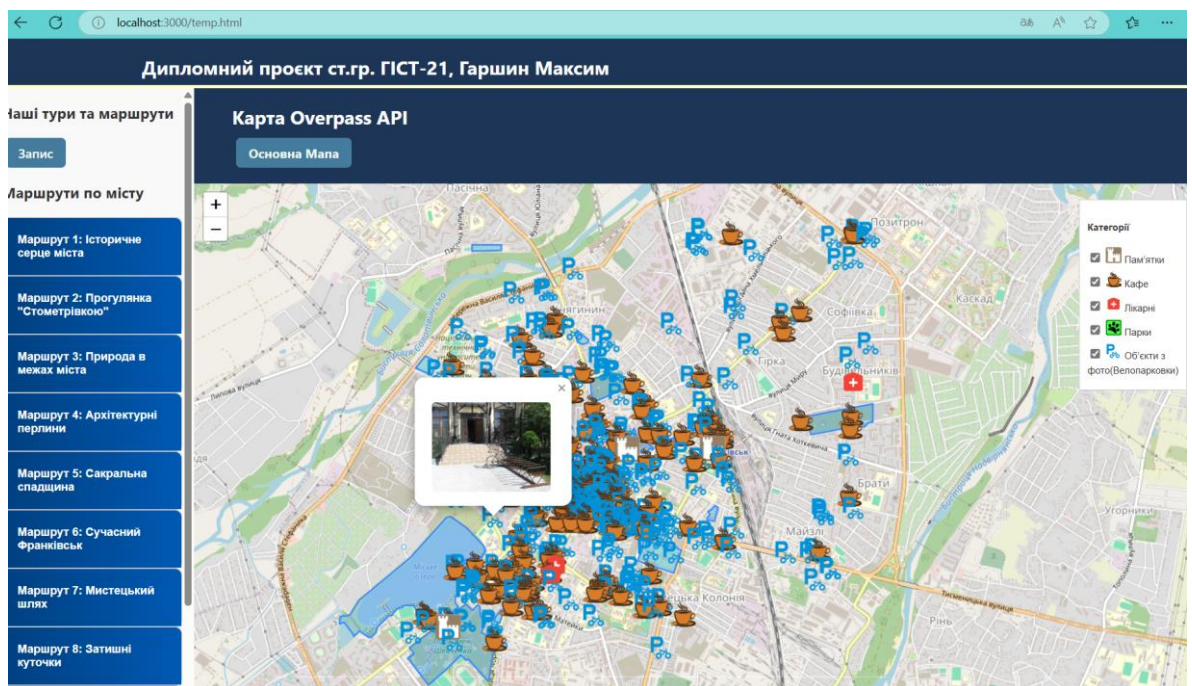


Рис. 3.2.7. Вигляд в браузері Карта Overpass API

Карта реалізована за допомогою бібліотеки Leaflet.js з використанням OpenStreetMap як базового шару та власних шарів для відображення туристичних маршрутів.

Особливості реалізації інтерфейсу

Інтерфейс додатку розроблено з урахуванням сучасних принципів UX/UI дизайну:

- Інтуїтивно зрозуміла навігація
- Адаптивний дизайн для різних розмірів екранів
- Візуальне розділення інформації на логічні блоки
- Інтерактивні елементи управління картою та фільтрами

Користувачський інтерфейс включає:

- Бічну панель з переліком маршрутів
- Основну область з інтерактивною картою
- Панель фільтрів та керування шарами
- Інформаційні картки маршрутів з описом та зображеннями

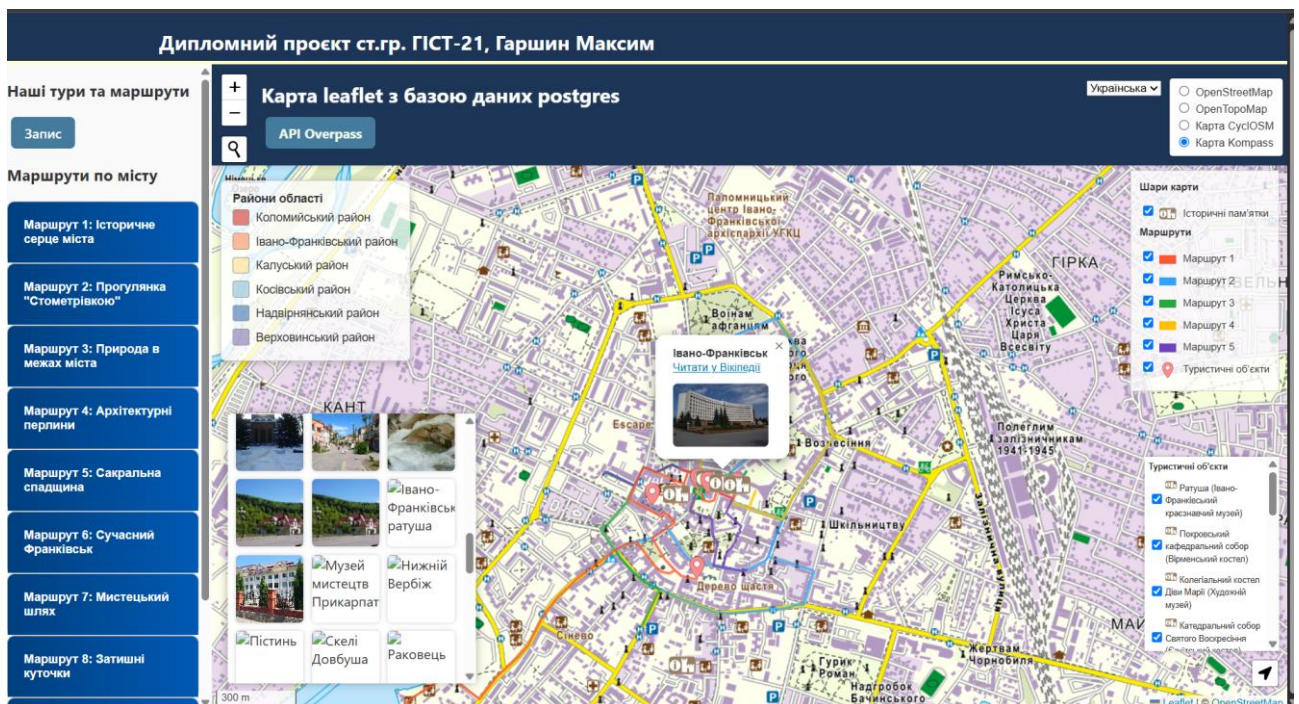
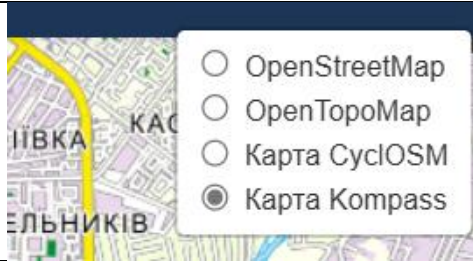
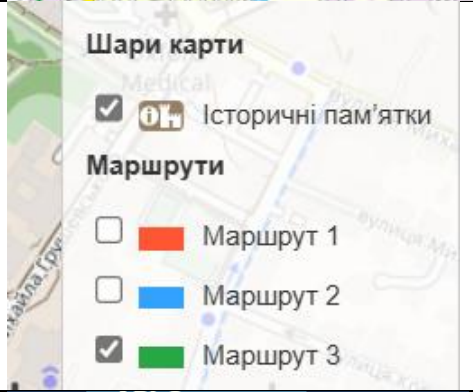
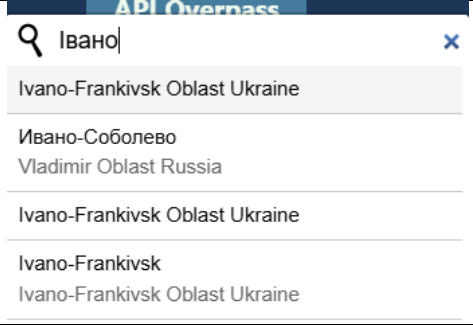
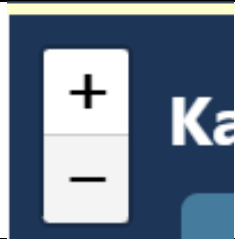
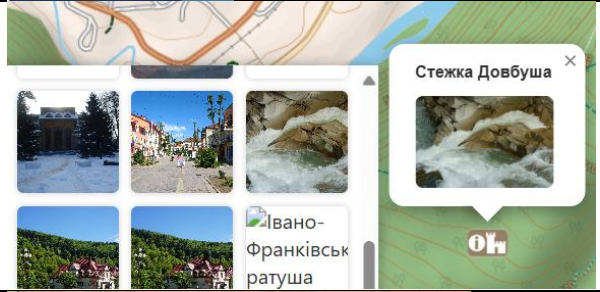
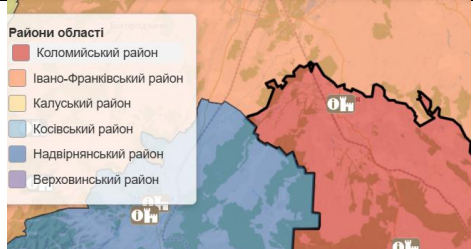
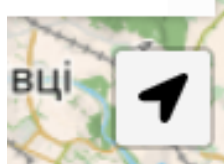
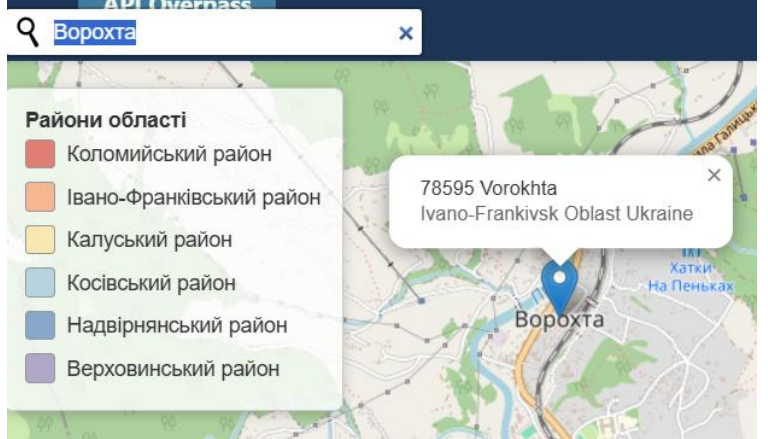
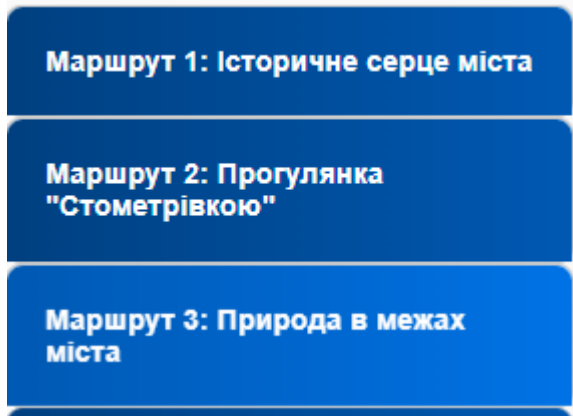
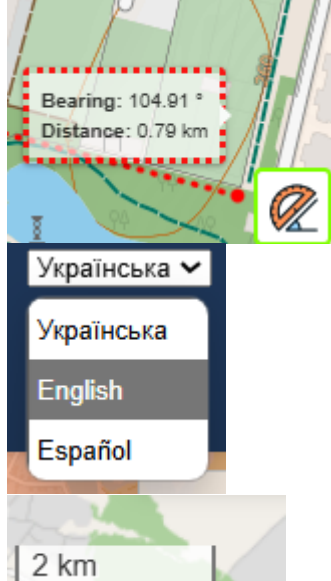


Рис. 3.2.8 Скріншот створеного інтерфейсу вебдодатку

Таблиця 3.2.1. Користувачький інтерфейс інтерактивної карти

Назва	Фото
Перегляд базових карт	
Перемикання шарів	
Пошук об'єкта	
Зміна масштабу	
Центрування карти по об'єкту	
Підсвічування об'єктів на карті через легенду	

<p>Геолокація</p>	
<p>Пошук та додавання точки</p>	
<p>Опис маршрутів</p>	
<p>Вимірювання відстані</p> <p>Зміна мови інтерфейсу</p> <p>Масштабна лінійка</p>	

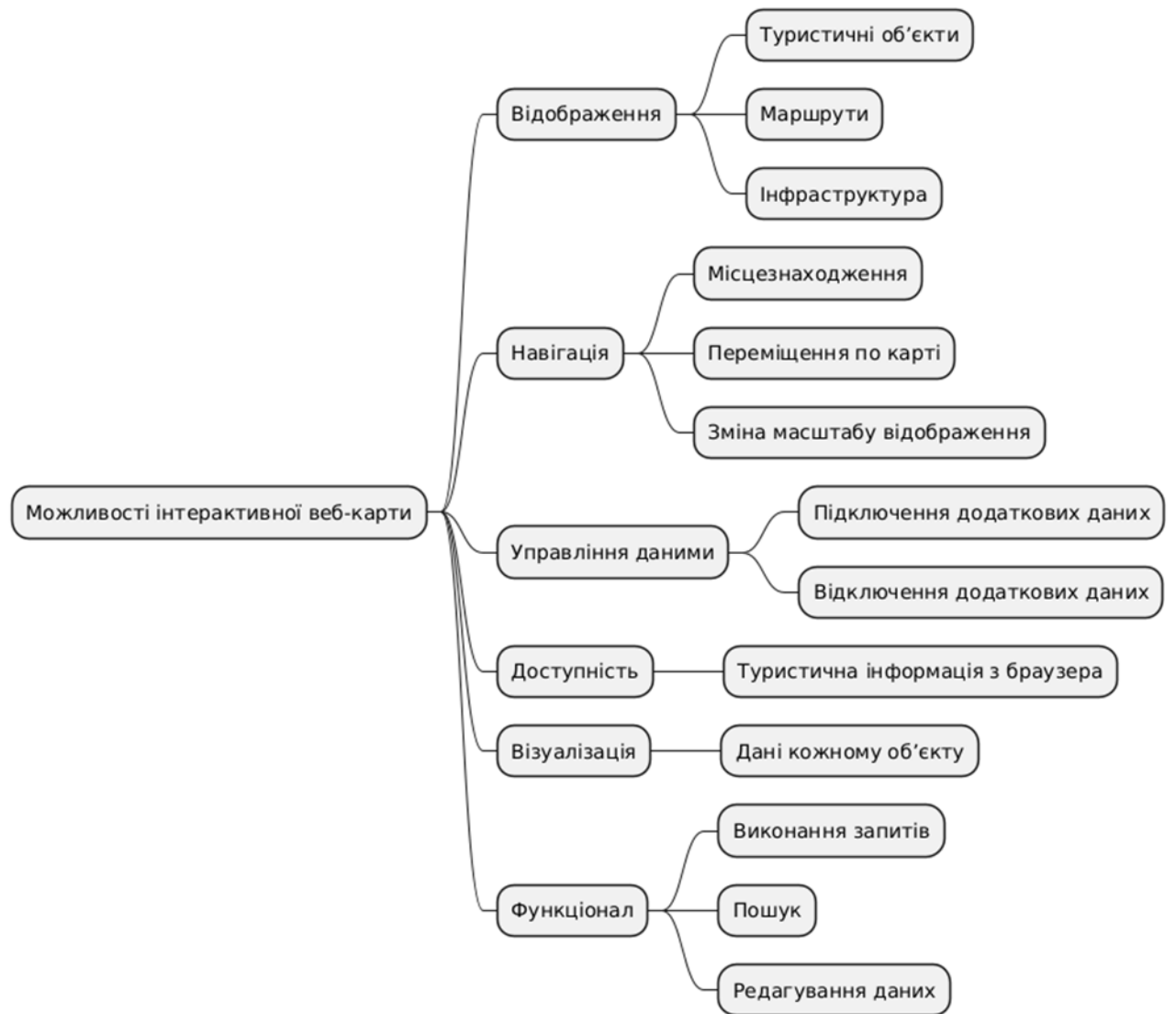


Рис. 3.2.9 Можливості створеної інтерактивної вебкарти

3.3. Інтеграція картографічних сервісів

Інтеграція картографічних сервісів є ключовим етапом у створенні інтерактивної вебкарти, що забезпечує відображення, навігацію та взаємодію з просторовими даними. У межах реалізації туристичного вебкарти для Івано-Франківської області було використано низку сучасних інструментів і сервісів, що дозволяють ефективно обробляти, зберігати та візуалізувати геодані.

Основним засобом клієнтської візуалізації виступає бібліотека Leaflet.js, яка забезпечує роботу з інтерактивними картами у браузері. Leaflet дозволяє реалізувати такі можливості:

- завантаження базових карт (OpenStreetMap, OpenTopoMap, CyclOSM),
- відображення геооб'єктів у форматі GeoJSON,
- інтерактивні шари з маршрутом, точками інтересу та легендами,
- пошуку точки за запитом.

Підключення базової карти здійснюється через тайлові сервіси:

```
// Додавання тайлів OpenStreetMap
const baseLayers = {
  "OpenStreetMap": L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '© <a href="https://openstreetmap.org">OpenStreetMap</a>'
  }),
  "OpenTopoMap": L.tileLayer('https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png', {
    attribution: '© <a href="https://opentopomap.org">OpenTopoMap</a>'
  }),
  "Карта CyclOSM": L.tileLayer('https://{s}.tile-cyclosm.openstreetmap.fr/cyclosm/{z}/{x}/{y}.png', {
    attribution: '© <a href="https://openstreetmap.org">OpenStreetMap</a>'
  }),
  "Карта Kompass": L.tileLayer('https://map.tourinfra.com/tiles/kompass_osm/{z}/{x}/{y}.png', {
    attribution: '© <a href="https://openstreetmap.org">OpenStreetMap</a>'
  }),
};
baseLayers["OpenStreetMap"].addTo(map);
```

Рис. 3.3.1 Додавання додаткових шарів OSM

Зв'язок із сервером через REST API

Інформація про туристичні маршрути, об'єкти, а також координати зберігається у базі даних PostgreSQL з розширенням PostGIS. Взаємодія з цими даними відбувається через REST API, реалізований на сервері за допомогою Node.js і Express.js (див. рис. 3.3.2).

```

app.get('/api/tourist_routes', async (req, res) => {
  try {
    const result = await pool.query(`
      SELECT id, name, length, category, ST_AsGeoJSON(route_geom)::json AS geometry
      FROM tourist_routes;
    `);
  }
}

```

Рис. 3.3.2. Фрагмент коду файла server.js

Потім відповідає даними, які потім відображаються на карті за допомогою Leaflet.js (див. рис.3.3.3).

```

fetch('/api/tourist_routes')
.then(res => res.json())
.then(data => {

```

Рис. 3.3.3. Фрагмент коду файла temp.html

Отримання зображень та назв з зовнішніх сервісів

Для збагачення інформації про об'єкти використовувалися такі зовнішні сервіси:

- Wikidata API – для автоматичного отримання зображень та описів за назвою об'єкта;
- Overpass API – для завантаження об'єктів інфраструктури (кафе, зупинки, джерела);
- Nominatim – для геокодування назв у координати.

Інтеграція реалізована через скрипти **update_images_by_coords.js** та **geocode.js**, які автоматизують завантаження даних перед збереженням у базу.

```

const categories = [
  {
    name: "Пам'ятки",
    query: `node["tourism"]="attraction"(48.91,24.69,48.94,24.74);`,
    icon: 'https://wiki.openstreetmap.org/w/images/4/41/Board_history_q.svg'
  },
  {
    name: "Кафе",
    query: `node["amenity"]="cafe"(48.91,24.69,48.94,24.74);`,
    icon: 'https://wiki.openstreetmap.org/w/images/0/07/Coffee2.svg'
  },
  {
    name: "Лікарні",
    query: `node["amenity"]="hospital"(48.91,24.69,48.94,24.74);`,
    icon: 'https://img.icons8.com/color/48/000000/hospital.png'
  },
  {
    name: "Парки",
    query: `way["leisure"]="park"(48.91,24.69,48.94,24.74);`,
    icon: 'https://wiki.openstreetmap.org/w/images/c/c9/Dog_park.png'
  }
];

```

Рис. 3.3.4. Додавання об'єктів інфраструктури через Overpass API

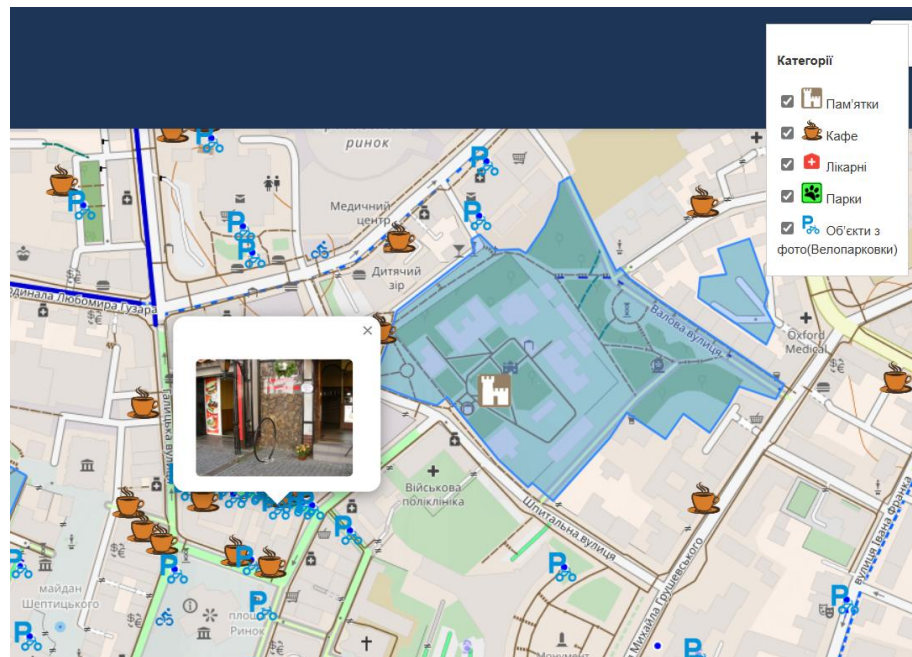


Рис. 3.3.5. Результат додавання об'єктів інфраструктури через Overpass API

Результатом інтеграції картографічних сервісів стало створення стабільного, масштабованого та динамічного вебсередовища, яке забезпечує інтуїтивну та зручну взаємодію користувачів з просторовими даними.

Реалізований функціонал дозволяє:

- переглядати туристичні маршрути та пам'ятки на різних картах;
- здійснювати пошук і фільтрацію об'єктів;
- отримувати розширену інформацію з відкритих джерел, зокрема зображення та описи з Wikidata ;
- Overpass API для автоматичного отримання об'єктів туристичної інфраструктури.

Висновки до розділу 3

У даному розділі було детально розглянуто структуру та функціональні можливості розробленого вебдодатку для туристичної компанії. Сайт реалізовано на базі клієнт-серверної архітектури.

Збір даних здійснювався з відкритих джерел. Дані проходили етапи перевірки, очищення, нормалізації, конвертації у формат GeoJSON та імпорту до PostGIS. Це забезпечило якісну основу для точного відображення інформації на карті. Побудована логічна схема методики збору даних та обробки вихідних даних (див. рис.3.1.8).

Описані в *таблиці 3.2.1* основні функції вебкарти: перегляд карти, пошук об'єктів, перемикання шарів, геолокація, побудова маршрутів, інтерактивна легенда і тд.

Додатково використано OpenStreetMap як базову карту, Overpass API для отримання геоданих, Nominatim для геокодування, Wikidata/Wikipedia для збагачення мультимедіа. Це дало змогу створити динамічну, актуальну та багатофункціональну вебкарту без необхідності локального зберігання всіх ресурсів.

ВИСНОВКИ

У процесі розробки інтерактивної вебкарти туристичних послуг було реалізовано комплекс заходів, що охоплюють збір, обробку, інтеграцію та візуалізацію геопросторових даних.

Методика збору та обробки даних передбачала використання відкритих джерел (OpenStreetMap, державні геопортали, KML/CSV-файли), конвертацію у формат GeoJSON, перевірку у QGIS та імпорт у базу даних PostGIS. Це забезпечило точність і структурованість даних.

Платформи та інструменти було обрано з урахуванням їх гнучкості, сумісності з Windows-середовищем та підтримки просторових операцій. Серед основних: PostgreSQL/PostGIS, Node.js, Express, Leaflet.js, QGIS, PyCharm.

Концептуальна модель вебдодатку ґрунтується на клієнт-серверній архітектурі. Користувач взаємодіє з інтерфейсом на Leaflet.js, який отримує дані з REST API, що працює на Node.js.

База геоданих відіграє центральну роль у зберіганні та обробці туристичної інформації. Вона містить таблиці об'єктів, маршрутів, користувачів, мультимедіа та відгуків, із просторовими індексами для ефективного запити й виводу на карту.

Інтеграція зовнішніх картографічних сервісів (Overpass API, Nominatim, Wikidata/Wikipedia) дозволила отримувати актуальні геооб'єкти, автоматично геокодувати об'єкти та збагачувати їх зображеннями. Це суттєво підвищило інформативність і динамічність карти.

Вебкарта виконує функції навігації, пошуку, перемикання шарів, перегляду об'єктів і маршрутів тощо. Загалом, реалізовано сучасну, масштабовану систему, що поєднує цифрову картографію з туристичною аналітикою, надаючи інструменти для зручного планування подорожей, навігації й популяризації туристичних ресурсів регіону.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білецька Н.В. Інвестиційний потенціал розвитку туристичної галузі України в умовах євроінтеграції : кваліфікаційна наукова праця на правах рукопису. – 2023.
2. Вуйцик О. Сучасний стан та перспективи розвитку велосипедного туризму в Надвірнянському районі Івано-Франківської області. – Івано-Франківськ : Видавництво ПНУ, 2013.
3. Гаврилук А.М. Туристичний потенціал Івано-Франківської області: сучасний стан і перспективи розвитку // Географія та туризм. – 2020. – № 45. – С. 12–20.
4. Кравців В.С. Регіональний розвиток туризму в Україні: теорія і практика. – Львів : Інститут регіональних досліджень НАН України, 2019.
5. Нодь О.Л. Формування стратегічних орієнтирів збалансованого регіонального розвитку галузі туризму та курортно-рекреаційної сфери : дис. ... канд. екон. наук. – 2022.
6. Ткаченко І.Ф. Сучасні тенденції розвитку індустрії туризму та гостинності. – Харків : ХНУ імені В.Н. Каразіна, 2021.
7. Зінько Ю.В. Гастрономічний туризм в Івано-Франківській області: сучасний стан і перспективи // Географія та туризм. – 2022. – № 48. – С. 25–33.
8. Панкова Є.В. Нормативно-правові засади розвитку туризму в Україні. – Київ : КНУ імені Тараса Шевченка, 2023.
9. Бейдик О.О. Туристичні ресурси України: оцінка та використання. – Київ : Альтерпрес, 2018.
10. Шаблій О.І. Географічні інформаційні системи в туризмі. – Львів : ЛНУ імені Івана Франка, 2020.
11. Лозинський Р.М. Використання геоінформаційних систем для аналізу туристичного потенціалу Карпатського регіону // Вісник Львівського університету. Серія географічна. – 2019. – № 53. – С. 78–85.
12. Лященко А.А. Вебкартографування : конспект лекцій. – Київ, 2023. – [Рукопис, навчальне видання].
13. Сніжко С.В. Цифрові технології розвитку міжнародного туризму. – Київ : КНТЕУ, 2022.
14. Dogman M. Вступ до вебкартографування / Майкл Дорман ; пер. з англ. за ред. Анатолія Лященко. – Бока-Ратон : CRC Press, Taylor & Francis Group, 2020. – [англ. вид.]. – 312 с.

15. PostgreSQL documentation [Електронний ресурс] // PostgreSQL Official Site. – Режим доступу: <https://www.postgresql.org/docs/>
16. PostGIS documentation [Електронний ресурс] // PostGIS.net. – Режим доступу: <https://postgis.net/documentation/>
17. Leaflet.js – JavaScript бібліотека для створення інтерактивних карт [Електронний ресурс]. – Режим доступу: <https://leafletjs.com>
18. Node.js documentation [Електронний ресурс] // Node.js Foundation. – Режим доступу: <https://nodejs.org/en/docs/>
19. Express.js documentation [Електронний ресурс] // Express.js Framework. – Режим доступу: <https://expressjs.com/>
20. QGIS Documentation [Електронний ресурс] // QGIS Project. – Режим доступу: <https://docs.qgis.org/>
21. Overpass Turbo – інтерфейс доступу до даних OpenStreetMap [Електронний ресурс]. – Режим доступу: <https://overpass-turbo.eu/>
22. Закон України «Про туризм» від 15 вересня 1995 р. № 324/95-ВР [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/324/95-вр>
23. Закон України «Про регулювання містобудівної діяльності» від 17 лютого 2011 р. № 3038-VI [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/3038-17>
24. Закон України «Про національну інфраструктуру геопросторових даних» від 13 квітня 2020 р. № 554-IX [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/554-20>
25. Закон України «Про охорону культурної спадщини» від 8 червня 2000 р. № 1805-III [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1805-14>
26. Постанова Кабінету Міністрів України «Про затвердження Порядку ведення Державного реєстру нерухомих пам'яток України» від 3 вересня 2009 р. № 928
27. ДСТУ ISO 19115-1:2017 Географічна інформація – Метадані – Частина 1: Основні положення. – [Офіційне видання].

ДОДАТКИ

```

4 <meta charset="UTF-8" />
5 <title>Історичні пам'ятки та маршрути Івано-Франківська</title>
6 <!-- TODO: Підключення основного CSS -->
7 <link rel="stylesheet" href="css/style.css" />
8 <!-- TODO: Підключення Leaflet бібліотеки -->
9 <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"/>
10 <!-- TODO: Стили для кнопки визначення місцезнаходження (LocateControl) -->
11 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet-locatecontrol/0.83.1/L.Control.Locate.min.css"/>
12
13 </head>
14 <!-- TODO: Шапка з мовним перемикачем -->
15 <header class="main-header">
16 <div id="language-switch" style="position:absolute; top:15px; right:150px; z-index:500;">
17 <!-- TODO: Додати обробку вибору мови -->
18 <select id="lang-select">
19 <option value="uk">Українська</option>
20 <option value="en">English</option>
21 <option value="esp">Español</option>
22 </select>
23 </div>
24 <div class="container">
25 <h1>Карта leaflet з базою даних postgres </h1>
26 <nav class="main-nav">
27 <!-- TODO: Додати навігацію на інші сторінки -->
28 <a href="Overpass.html">API Overpass</a>
29 </nav>
30
31 </div>
32 </header>
33 <!-- TODO: CSS для анімованого прихованого заголовку -->
34 <style>.main-header {
35 position: fixed;
36 top: -100px; /* Змістіть заголовок вгору за межі видимої області */
37 left: 0;
38 width: 100%;
39 transition: top 0.3s ease;
40 z-index: 1000;
41 }
42
43 /* Відображаємо заголовок при наведенні на верхню частину сторінки */
44 body:hover .main-header {
45 top: 0;
46 }
47 </style>
48 <link rel="stylesheet" href="https://unpkg.com/leaflet-control-search@2.9.8/dist/leaflet-search.min.css">
49 </head>
50 <body>
51 <!-- TODO: Контейнер для карти -->
52 <div id="map"></div>
53 <!-- TODO: Плагін для пошуку по об'єктах -->
54 <script src="https://unpkg.com/leaflet-control-search@2.9.8/dist/leaflet-search.min.js"></script>
55 <!-- TODO: Основна бібліотека Leaflet -->
56 <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
57 <!-- TODO: Основний JS-код карти -->
58 <script src="/js/index_map.js" defer></script>
59 <!-- TODO: Плагін геокодера (адресний пошук) -->
60 <link rel="stylesheet" href="https://unpkg.com/leaflet-control-geocoder/dist/Control.Geocoder.css" />
61 <script src="https://unpkg.com/leaflet-control-geocoder/dist/Control.Geocoder.js"></script>
62 <!-- TODO: JS-скрипт LocateControl для позиціонування -->
63 <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet-locatecontrol/0.83.1/L.Control.Locate.min.js"></script>
64 </body>
65 </html>

```

```

1  //:- Базова ініціалізація мапи
2  const map = L.map('map').setView([48.9226, 24.7103], 14);
3  //:- Додаються кілька базових шарів: звичайна OSM, OpenToroMap, Сус105M, Kompas.
4  const baseLayers = {};
27 // За замовчуванням активується OpenStreetMap.
28 baseLayers['OpenStreetMap'].addTo(map);
29 // Глобальний об'єкт для шарів
30 const layers = {};
31 //:- масштабна лінійка
32 L.control
33 .scale().
34 .addTo(map);
35 //:- перемикач шарів
36 L.control.layers(baseLayers, null, { collapsed: false }).addTo(map);
37 //:- геолокація
38 L.control
39 .locate().
40 .addTo(map);
41 //:- Додавання строки пошуку
42 L.Control.geocoder().addTo(map);
43 //:-Зберігає всі маркери
44 const objectLayers = []; // масив для зберігання всіх маркерів
45 //:-В нього додаються візуальні блоки (зображення об'єктів)
46 const legendItems = [];
47 //:- Додаємо контейнер для легенди в HTML з фото
48 const legendContainer = document.createElement('div');
49 legendContainer.id = 'legend';
50 legendContainer.style.cssText = `
51     position: absolute;
52     left: 13%;
53     top: 75%;
54     transform: translate(-50%, -50%);
55     background: white;
56     padding: 10px;
57     border-radius: 5px;
58     box-shadow: 0 0 10px rgba(0,0,0,0.3);
59     z-index: 1000;
60     max-width: 300px;
61     max-height: 300px; /* Обмеження висоти для прокрутки */
62     overflow-y: auto; /* Увімкнення вертикальної прокрутки */
63     display: grid;
64     grid-template-columns: repeat(auto-fill, minmax(80px, 1fr));
65     gap: 10px;
66 `;
67 document.body.appendChild(legendContainer);

```

```

84 //:- Завантаження історичних пам'яток
85 fetch('/api/hystory-ob')
86   .then(res => res.json())
87   .then(█)
145
146   .catch(error => console.error('Помилка завантаження пам'яток:', error));
147
148 //:- Завантаження Туристичні об'єкти
149 fetch('/api/tourist-objects')
150   .then(res => res.json())
151   .then(data => {
152     // Створення GeoJSON шару з туристичними об'єктами
153     const layer = L.geoJSON(data, {
154       // Для кожного об'єкта створити маркер з кастомним значком
155       pointToLayer: function (feature, latlng) {
156         const marker = L.marker(latlng, {
157           icon: L.icon({
158             iconUrl: 'https://img.icons8.com/office/40/marker.png',
159             iconSize: [28, 28],
160             iconAnchor: [14, 28],
161             popupAnchor: [0, -28],
162           }),
163         });
164         // :- Прив'язка назви та ID для подальшої взаємодії
165         marker.featureName = feature.properties.name;
166         marker.featureId = feature.properties.id;
167         objectLayers.push(marker); //Зберегти маркер у глобальний масив
168         return marker;
169       },
170       // :- Вивід інформації в поп-уп для кожного маркера
171       onEachFeature: function (feature, layer) {
172         const p = feature.properties;
173         const popup = `
174           <strong>${p.name}</strong><br>
175           ${p.address ? `<strong>Адреса:</strong> ${p.address}<br>` : ''}
176           ${
177             p.working_hours
178             ? `<strong>Години:</strong> ${p.working_hours}<br>`
179             : ''
180           }
181           ${p.entry_fee ? `<strong>Вхід:</strong> ${p.entry_fee} грн<br>` : ''}
182           `;
183         layer.bindPopup(popup);
184         //Генерація чекбоксу в легенду для керування видимістю об'єкта
185         const checkboxId = `cb-${p.name.replace(/\s+/g, '_')}`;
186         legendItems.push(`
187 <div class="legend-item">
188   <input type="checkbox" id="${checkboxId}" checked onchange="toggleLayer('${checkboxId}')">
189   <label for="${checkboxId}">
190     
191     ${p.name}
192   </label>
193 </div>
194 `);
195       },
196     }).addTo(map);

```

```

197 // Збереження шару для подальшого контролю
198 layers['Туристичні об'єкти'] = layer;
199
200 // -:Динамічна легенда з чекбоксами об'єктів
201 const legend = L.control({ position: 'bottomright' });
202 legend.onAdd = function () {
203   const div = L.DomUtil.create('div', 'legend-grid-wrapper');
204   div.innerHTML = `
205     <div class="legend-grid-header"><strong>Туристичні об'єкти</strong></div>
206     <div class="legend-grid-container">${legendItems.join('')}</div>
207   `;
208   return div;
209 };
210 legend.addTo(map);
211 })
212
213 .catch(error => console.error('Помилка туристичних об'єктів:', error));
214
215 // -: Функція увімкнення/вимкнення маркерів
216 function toggleLayer(checkboxId) {
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231 // -: Завантаження Туристичні маршрути
232 fetch('/api/tourist_routes')
233   .then(res => res.json())
234   .then(data => {
235     //Кольорове кодування маршрутів за назвою
236     const colorByName = {};
237     //Групування маршрутів за назвою
238     const grouped = {};
239     data.features.forEach(feature => {
240       const name = feature.properties.name;
241       if (!grouped[name]) grouped[name] = [];
242       grouped[name].push(feature);
243     });
244     // Створення окремого шару для кожного маршрута
245     Object.entries(grouped).forEach(([name, features]) => {
246       const color = colorByName[name] || '#999';
247
248       const layer = L.geoJSON(
249         { type: 'FeatureCollection', features },
250         {
251           style: {color},
252           onEachFeature: function (feature, layer) {
253             // -: Налаштувати pop-up для кожного маршруту
254             layer.bindPopup(`<strong>${name}</strong> `);
255           },
256         }
257       ).addTo(map);
258       //Зберігання шару для керування через чекбокси
259       layers[name] = layer; // Зберігаємо по назві
260     });
261   });
262   .catch(error => console.error('Помилка завантаження маршрутів:', error));

```

```

274 // -: Створення інтерактивної легенди з фільтрацією шарів
275 const legend = L.control({ position: 'topright' });
276 legend.onAdd = function () {
277     const div = L.DomUtil.create('div', 'legend');
278     // Заголовок легенди
279     div.innerHTML = '<h4>Шари карти</h4>';
280     // Чекбокс для історичних пам'яток
281     div.innerHTML += `
282         <label>
283             <input type="checkbox" checked data-layer="Історичні пам'ятки"
284                 onchange="toggleLayer('Історичні пам'ятки', this.checked)">
285             
286             Історичні пам'ятки
287         </label>`;
288     // Секція для маршрутів
289     div.innerHTML += '<h4>Маршрути</h4>';
290     // Чекбокси з індикаторами кольорів для маршрутів
291     const routeColors = {█};
292
293     Object.entries(routeColors).forEach(([name, color]) => {
294         div.innerHTML += `
295             <label>
296                 <input type="checkbox" checked onchange="toggleLayer('${name}', this.checked)">
297                 <span class="route-indicator" style="background:${color}"></span>
298                 ${name}
299             </label>`;
300     });
301     // Чекбокс для вмикання/вимикання POI (туристичних об'єктів)
302     div.innerHTML += `
303         <label>
304             <input type="checkbox" checked onchange="toggleLayer('Туристичні об'єкти', this.checked)">
305             
306             Туристичні об'єкти
307         </label>`;
308     return div;
309 };
310 legend.addTo(map);
311
312 // -: Завантаження GeoJSON Полігони районів
313 fetch('data/IvFrankDist.geojson')
314     .then(res => res.json())
315     .then(█);
316
317 // -: Легенда Райони
318 // Підсвітка при наведенні
319 function highlightLayer(layer) {█}
320 // Скидання стилю
321 function resetLayerStyle(layer, name) {█}
322 //Встановлення кольору для кожного району
323 const districtColors = {█};
324 //Ініціалізація об'єктів для шарів районів
325 const districtLayers = {}; // посилання на окремі шари
326 let districtGeoLayer = null; // загальний GeoJSON шар
327 let currentHover = null; // для підсвітки
328 const minZoomToShowDistricts = 9; // мінімальний масштаб для відображення
329

```

```

377 //Повертає колір району або дефолтний
378 function getColor(name) {
379     return districtColors[name] || 'rgba(200,200,200,0.5)';
380 }
381
382 //:-Ініціалізація легенди районів
383 function initLegend() {}
384
385
386
387
388
389
390
391
392 const maxZoomToShowDistricts = 11; // максимум, до якого райони видно
393 // -: Ввімкнення/вимкнення районів в залежності від масштабу
394 map.on('zoomend', () => {
395     if (!districtGeoLayer) return;
396
397     if (map.getZoom() > maxZoomToShowDistricts) {
398         // Занадто наближено – сховати
399         if (map.hasLayer(districtGeoLayer)) {
400             map.removeLayer(districtGeoLayer);
401         }
402     } else {
403         // Достатньо далеко – показати
404         if (!map.hasLayer(districtGeoLayer)) {
405             map.addLayer(districtGeoLayer);
406         }
407     }
408 });
409
410 // -: Перемикач видимості шарів з центруванням
411 function toggleLayer(layerName, checked) {
412     if (layers[layerName]) {
413         if (checked) {
414             map.addLayer(layers[layerName]);
415             // -: Центрування карти до шару
416             const bounds = layers[layerName].getBounds?();
417             if (bounds && bounds.isValid()) {
418                 map.fitBounds(bounds, { padding: [30, 30] });
419             }
420         } else {
421             map.removeLayer(layers[layerName]);
422         }
423     }
424 }

```

```

1 <!DOCTYPE html>
2 <html lang="uk">
3 <head>
4   <meta charset="UTF-8" />
5   <title>Мапа туристичних об'єктів Івано-Франківська</title>
6   <link rel="stylesheet" href="css/style.css" />
7   <link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css" />
8   <link
9     rel="stylesheet"
10    href="https://cdn.jsdelivr.net/gh/gokertanrisever/leaflet-ruler@master/src/leaflet-ruler.css"/>
11 <header class="main-header">
12   <div class="container">
13     <h1>Карта Overpass API </h1>
14     <nav class="main-nav">
15       <a href="index.html">Основна Мапа</a>
16     </nav>
17   </div>
18 </header>
19 <style>.main-header {
20   position: fixed;
21   top: -100px; /* Змістить заголовок вгору за межі видимої області */
22   left: 0;
23   width: 100%;
24   transition: top 0.3s ease;
25   z-index: 1000;
26 }
27
28 /* Відображаємо заголовок при наведенні на верхню частину сторінки */
29 body:hover .main-header {
30   top: 0;
31 }
32 </style>
33 <style>
34   #map { height: 100vh; width: 100vw; }
35   .popup-img { width: 150px; border-radius: 5px; }
36   .legend { position: absolute; top: 10px; right: 10px; background: white; padding: 10px; border:
37   .legend label { display: block; margin: 5px 0; }
38 </style>
39 </head>
40 <body>
41   <div id="map"></div>
42 <script
43   src="https://cdn.jsdelivr.net/gh/gokertanrisever/leaflet-ruler@master/src/leaflet-ruler.js"
44 ></script>
45   <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
46   <script src="https://unpkg.com/osmtogeojson/osmtogeojson.js"></script>
47   <script src="/js/Overpass_2map.js" defer></script>
48 <link
49   rel="stylesheet"
50   href="https://cdn.jsdelivr.net/gh/gokertanrisever/leaflet-ruler@master/src/leaflet-ruler.css"
51 />
52 <script
53   src="https://cdn.jsdelivr.net/gh/gokertanrisever/leaflet-ruler@master/src/leaflet-ruler.js"
54 ></script>
55
56 </body>
57 </html>

```

```

1 // -: Створення базових тайлових шарів (CyclOSM, OSM Standard, HOT)
2 const cyclosm = L.tileLayer('https://{s}.tile-cyclosm.openstreetmap.fr/cyclosm/{z}/{x}/{y}.png', {
6
7 const osmStandard = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
11
12 const osmHOT = L.tileLayer('https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png', {
16
17 // -: Ініціалізація мапи з центром на Івано-Франківськ і стартовим шаром CycloSM
18 const map = L.map(
23
24 // -: Об'єкт для контролю перемикання базових шарів
25 const baseLayers = {
26   "CyclOSM": cyclosm,
27   "OpenStreetMap Standard": osmStandard,
28   "OpenStreetMap HOT": osmHOT
29 };
30 // -: Додати масштабну лінійку
31 L.control.scale({ position: 'bottomleft' }).addTo(map);
32 // -: Додати контрол перемикання шарів
33 L.control.layers(baseLayers).addTo(map);
34 // -: Додати інструмент вимірювання відстані (ruler)
35 L.control.ruler({ position: 'bottomright' }).addTo(map);
36 // -: Категорії POI для завантаження з Overpass API
37 const categories = [
38   {
39     name: "Пам'ятки",
40     query: `node["tourism"]="attraction"(48.91,24.69,48.94,24.74);`,
41     icon: 'https://wiki.openstreetmap.org/w/images/4/41/Board_history_q.svg'
42   },
43   {
44     name: "Кафе",
45     query: `node["amenity"]="cafe"(48.91,24.69,48.94,24.74);`,
46     icon: 'https://wiki.openstreetmap.org/w/images/0/07/Coffee2.svg'
47   },
48   {
49     name: "Лікарні",
50     query: `node["amenity"]="hospital"(48.91,24.69,48.94,24.74);`,
51     icon: 'https://img.icons8.com/color/48/000000/hospital.png'
52   },
53   {
54     name: "Парки",
55     query: `way["leisure"]="park"(48.91,24.69,48.94,24.74);`,
56     icon: 'https://wiki.openstreetmap.org/w/images/c/c9/Dog_park.png'
57   },
58   {
59     name: "Об'єкти з фото(Велопарковки)",
60     query: `node["image"](48.91,24.69,48.94,24.74);`,
61     icon: 'https://wiki.openstreetmap.org/w/images/7/7f/Parking-bicycle-16.svg'
62   }
63 ];

```

```

65 // -: Створюємо об'єкт для збереження завантажених шарів
66 const layers = {};
67 // -: Для кожної категорії формуємо запит Overpass і додаємо відповідний шар
68 categories.forEach(cat => {
69   const fullQuery = `[out:json];${cat.query}out body;>;out skel qt;`;
70   fetch("https://overpass-api.de/api/interpreter?data=" + encodeURIComponent(fullQuery))
71     .then(res => {
72       if (!res.ok) throw new Error(`Помилка запиту для ${cat.name}`);
73       return res.json();
74     })
75     .then(data => {
76       // -: Перетворення Overpass JSON → GeoJSON
77       const geojson = osmtogeojson(data);
78       layers[cat.name] = L.geoJSON(geojson, {
79         pointToLayer: (feature, latlng) => {
80           return L.marker(latlng, {
81             icon: L.icon({ iconUrl: cat.icon, iconSize: [32, 32] })
82           });
83         },
84         onEachFeature: (feature, layer) => {
85           const tags = feature.properties || {};
86           const name = tags.name || '';
87           let popupContent = `${name}</strong>`;
88           // Додавання зображення, якщо є тег image
89           if (tags.image) {
90             popupContent += `  
`;
91           }
92           layer.bindPopup(popupContent);
93         },
94         // Стиль для полігонів (наприклад, парків)
95         style: (feature) => {
96           if (feature.geometry.type === 'Polygon' || feature.geometry.type === 'MultiPolygon') {
97             return {color: '#ccc'};
98           }
99           return {};
100         }
101       }).addTo(map);
102     })
103     .catch(error => console.error(`Помилка для категорії ${cat.name}:`, error));
104   });
105 // -: Додавання легенди
106 const legend = L.control({ position: 'topright' });
107 legend.onAdd = function () {
108   const div = L.DomUtil.create('div', 'legend');
109   div.innerHTML = `

#### Категорії</h4>`; 110 categories.forEach((cat) => { 111 const layer = layers[cat.name]; 112 if (layer) { 113 const checked = layer.isVisible(); 114 div.innerHTML += ` 115 <input type="${checked ? 'checkbox' : 'checkbox'}"/> 116 <span style="margin-left: 5px;">${cat.name}</span> 117 `; 118 } 119 }); 120 return div; 121 }; 122 legend.addTo(map); 123 // -: Глобальна функція для вмикання / вимикання POI шарів 124 window.toggleLayer = function (layerName, checked) { 125 const layer = layers[layerName]; 126 if (layer) { 127 layer.toggleLayer(checked); 128 } 129 };


```

```
1  /* TODO: Базові стилі для тіла сторінки */
2  > body {...}
7  /* TODO: Стили для заголовка (header) */
8  > .main-header {...}
14 /* TODO: Заголовок усередині шапки */
15 > .main-header h1 {...}
20 /* TODO: Основна навігація */
21 > .main-nav {...}
27 /* TODO: Посилання навігації */
28 > .main-nav a {...}
34 .main-nav a:hover {
35     color: #ffc33a;
36 }
37 /* TODO: Контейнер для контенту */
38 > .container {...}
42 /* TODO: Секція маршрутів */
43 > .routes {...}
48 /* TODO: Стиль для блоку маршруту */
49 > .route {...}
55 /* TODO: Кнопка розгортання/згорання маршруту */
56 > .route-toggle {...}
70 > .route-toggle:hover {...}
74 /* TODO: Деталі маршруту (згорнуті за замовчуванням) */
75 > .route-details {...}
80 /* TODO: Активний маршрут (деталі показуються) */
81 .route.active .route-details {
82     display: block;
83 }
84 /* TODO: Відображення фото маршрутів у flex-контейнері */
85 > .route-wrapper {...}
90
91 > .route-wrapper img {...}
96 /* TODO: Таблиця характеристик маршруту */
97 > .route-details table {...}
102
103 > .route-details th, .route-details td {...}
108
109 > .route-details th {...}
```

```

124  /* TODO: Стили для Leaflet-перемикача шарів */
125  > .leaflet-control-layers {...}
131  /* TODO: Додаткові стилі навігаційних посилань */
132  > header nav a {...}
143
144  header nav a:hover {
145      background-color: #5ca0d3;
146  }
147  /* TODO: Легенда зі скролом */
148  > .legend-scroll {...}
157  /* TODO: Сіткова легенда (багато елементів) */
158  .legend-grid-wrapper {
159      max-width: 150px;
160      padding: 5px;
161      background: white;
162      box-shadow: 0 0 4px rgba(0,0,0,0.3);
163      font-size: 10px;
164      max-height: 220px;
165      overflow-y: auto;
166
167  .legend-grid-header {
168      margin-bottom: 6px;
169  }
170  /* TODO: Зображення в попапі */
171
172  > .popup-img {...}
178  /* TODO: Легенда для районів */
179  > .info.legend {...}
190
191  /* TODO: Елемент району в легенді */
192  > .legend-item {...}
198
199  > .legend-item i {...}
207  /* TODO: Підсвітка при наведенні на елемент легенди */
208  > .legend-item:hover {...}
---
```

Додаток Е. Код файлу Update_images_by_coords.js

```
1 const { Pool } = require('pg');
2 const fetch = (...args) => import('node-fetch').then(({ default: fetch }) => fetch(...args));
3
4 const pool = new Pool({
5   user: 'postgres',
6   host: 'localhost',
7   database: 'tourist_db',
8   password: 'postgres',
9   port: 5432,
10 });
11
12 async function getArticleByCoords(lat, lon) {
13   const url = `https://en.wikipedia.org/w/api.php?action=query&format=json&origin=*&list=geosearch&gscoord
14   const res = await fetch(url);
15   const data = await res.json();
16   const result = data.query.geosearch[0];
17   return result?.title || null;
18 }
19
20 async function getWikidataId(title) {
21   const url = `https://en.wikipedia.org/w/api.php?action=query&format=json&origin=*&prop=pageprops&titles=
22   const res = await fetch(url);
23   const data = await res.json();
24   const page = Object.values(data.query.pages)[0];
25   return page?.pageprops?.wikibase_item || null;
26 }
27
28 async function getImageFromWikidata(qid) {
29   const url = `https://www.wikidata.org/wiki/Special:EntityData/${qid}.json`;
30   const res = await fetch(url);
31   const data = await res.json();
32   const entity = data.entities[qid];
33   const imageName = entity?.claims?.P18?.[0]?.mainsnak?.datavalue?.value;
34   if (!imageName) return null;
35
36   // Створюємо URL зображення
37   const fileName = encodeURIComponent(imageName.replace(/ /g, '_'));
38   const md5 = require('crypto').createHash('md5').update(fileName).digest('hex');
39   const urlPath = `${md5[0]}/${md5.substring(0, 2)}/${fileName}`;
40   return `https://upload.wikimedia.org/wikipedia/commons/${urlPath}`;
41 }
42
43 async function updateImagesByWikidata() {
44   const { rows } = await pool.query('SELECT id, ST_Y(geom) AS lat, ST_X(geom) AS lon FROM history_ob');
45
46   for (const row of rows) {
47
48     console.log('📍 Успіх! Фото з Wikidata записано в базу!');
49     await pool.end();
50   }
51 }
52 updateImagesByWikidata();
53
```

```

1  const fs = require('fs');
2  const fetch = require('node-fetch');
3
4  // * Список назв і адрес пам'яток (приклад)
5  const places = [ ];
10
11  const delay = ms => new Promise(res => setTimeout(res, ms));
12
13  async function geocode(address) {
14    const url = `https://nominatim.openstreetmap.org/search?q=${encodeURIComponent(address)}&format=json`;
15    const res = await fetch(url, {
16      headers: {
17        'User-Agent': 'leaflet-map/1.0 (you@example.com)'
18      }
19    });
20    const data = await res.json();
21    if (data && data.length > 0) {
22      return {
23        lat: parseFloat(data[0].lat),
24        lon: parseFloat(data[0].lon)
25      };
26    } else {
27      return null;
28    }
29  }
30
31  async function generateSQL() {
32    const lines = [];
33    lines.push('CREATE EXTENSION IF NOT EXISTS postgis;');
34    lines.push('CREATE TABLE IF NOT EXISTS places (id SERIAL PRIMARY KEY, name TEXT, geom GEOMETRY(Point, 4326));');
35    lines.push('\n-- Insert places:');
36
37    for (const place of places) {
38      const coords = await geocode(place.address);
39      if (coords) {
40        lines.push(
41          `INSERT INTO places (name, geom) VALUES ('${place.name.replace(/'/g, '')}', ST_SetSRID(ST_MakePoint(${coords.lat}, ${coords.lon}), 4326));`
42        );
43        console.log(`✓ ${place.name} → ${coords.lat}, ${coords.lon}`);
44      } else {
45        console.log(`⚠ Не знайдено координати для: ${place.name}`);
46      }
47      await delay(1000); // затримка для API
48    }
49
50    fs.writeFileSync('places.sql', lines.join('\n'), 'utf8');
51    console.log('\n✅ Файл places.sql збережено!');
52  }
53
54  generateSQL();

```

```

5 const pool = require('./db');
6 const app = express();
7
8 const PORT = 3000;
9
10 app.use(express.static('public'));
11 app.use(express.urlencoded({ extended: true }));
12
13 app.get('/api/hystory-ob', async (req, res) => {});
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40 app.get('/api/tourist_routes', async (req, res) => {
41   try {
42     const result = await pool.query(`
43       SELECT id, name, length, category, ST_AsGeoJSON(route_geom)::json AS geometry
44       FROM tourist_routes;
45     `);
46
47     const features = result.rows.map(row => ({
48       type: "Feature",
49       geometry: row.geometry,
50       properties: {
51         id: row.id,
52         name: row.name,
53         length: row.length,
54         category: row.category,
55       }
56     }));
57
58     res.json({
59       type: "FeatureCollection",
60       features
61     });
62   } catch (err) {
63     console.error('DB error:', err);
64     res.status(500).send("Помилка при завантаженні маршрутів");
65   }
66 });
67
68 app.get('/api/tourist-objects', async (req, res) => {
69   try {
70     const result = await pool.query(`
71       SELECT id, name, category, description, address, working_hours, entry_fee,
72       ST_AsGeoJSON(geom)::json AS geometry
73       FROM tourist_objects;
74     `);
75
76     const features = result.rows.map(row => ({
77       type: "Feature",
78       geometry: row.geometry,
79       properties: {
80         id: row.id,
81         name: row.name,
82         category: row.category,
83         description: row.description,
84         address: row.address,
85         working_hours: row.working_hours,
86         entry_fee: row.entry_fee
87       }
88     }));
89
90     res.json(features);
91   } catch (err) {
92     console.error('Помилка при завантаженні tourist_objects:', err);
93     res.status(500).send("Помилка при завантаженні туристичних об'єктів");
94   }
95 });
96
97 app.listen(PORT, () => console.log(`Map server running: http://localhost:${PORT}`));

```