2. АНАЛІЗ МЕТОДИЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБЛІКУ ТА ВЕДЕННЯ ІНФОРМАЦІЇ ПРО ВОДНІ ОБ’ЄКТИ РЕГІОНУ

Водний об'єкт - природний або створений штучно елемент [довкілля](https://uk.wikipedia.org/wiki/%D0%94%D0%BE%D0%B2%D0%BA%D1%96%D0%BB%D0%BB%D1%8F), в якому зосереджуються води [море](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D1%80%D0%B5), [річка](https://uk.wikipedia.org/wiki/%D0%A0%D1%96%D1%87%D0%BA%D0%B0), [озеро](https://uk.wikipedia.org/wiki/%D0%9E%D0%B7%D0%B5%D1%80%D0%BE), [водосховище](https://uk.wikipedia.org/wiki/%D0%92%D0%BE%D0%B4%D0%BE%D1%81%D1%85%D0%BE%D0%B2%D0%B8%D1%89%D0%B5), [ставок](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B2%D0%BE%D0%BA), [канал](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B0%D0%BB), [водоносний горизонт](https://uk.wikipedia.org/wiki/%D0%92%D0%BE%D0%B4%D0%BE%D0%BD%D0%BE%D1%81%D0%BD%D0%B8%D0%B9_%D0%B3%D0%BE%D1%80%D0%B8%D0%B7%D0%BE%D0%BD%D1%82) (Водний кодекс України, 1995).

Водні об'єкти поділяються на водотоки (В.о. з постійним або тимчасовим рухом води в руслі - потоком, наприклад, річки) і водойми (безстічні, або з уповільненим водообміном В.о., наприклад, озера).

Водні об'єкти загальнодержавного значення:

* внутрішні морські води, територіальне море,а також акваторії морських портів.
* підземні води, які є джерелом централізованого [водопостачання](https://uk.wikipedia.org/wiki/%D0%92%D0%BE%D0%B4%D0%BE%D0%BF%D0%BE%D1%81%D1%82%D0%B0%D1%87%D0%B0%D0%BD%D0%BD%D1%8F" \o "Водопостачання).
* [поверхневі води](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B2%D0%B5%D1%80%D1%85%D0%BD%D0%B5%D0%B2%D1%96_%D0%B2%D0%BE%D0%B4%D0%B8) (озера, водосховища, річки, канали), що знаходяться і використовуються на території більш як однієї області, а також їх притоки всіх порядків.
* водні об'єкти в межах територій природно-заповідного фонду загальнодержавного значення, а також віднесені до категорії лікувальних.

Водні об'єкти місцевого значення:

* поверхневі води, що знаходяться і використовуються в межах однієї області і які не віднесені до водних об'єктів загальнодержавного значення.
* підземні води, які не можуть бути джерелом централізованого водопостачання.

Водні об'єкти (води) є виключно власністю Українського народу і надаються тільки у користування.

В даній дипломній роботі буде розглянуто пошук наступних водних об’єктів – річка, ставок, водосховище.

**2.1 Типова паспортна інформація про водні об’єкти області чи басейну**

**Паспорт водойми** — це основний документ, що включає всю інформацію по тому чи іншому водному об’єкті. Він необхідний у разі передачі водойми в аренду або в інших подібних ситуаціях. Серед характеристик, які присутні в паспорті, можна виділити наступну інформацію — кількісні показники площі, обсягу, глибини, гідрологічні дані, а також відомості про користувачів водою.

Крім того, повинно бути описано повний технічний стан, призначення об’єкта, правила його використання. Якщо на даній водоймі є якісь гідротехнічні споруди, вони також повинні бути включені в паспорт. У тому випадку, коли мова йде про водний об’єкт, який використовується у рибному господарстві, то акцент потрібно зробити на вивченні можливості розлучення того чи іншого виду риб.

Для кожного водного об’єкта (річка, озеро, ставок, джерело та інші) є свої параметри, які слід дослідити і занести в **паспорт**. Так, якщо мова йде про озеро, то в його паспорті повинні бути зазначені такі відомості — його розташування, розміри, глибина, параметри берега, характер грунту, колір води, різниця в рівнях, рослинність і тваринний світ, наявний в межах водойми, відомості про господарські споруди, джерела проблеми забруднення озера.

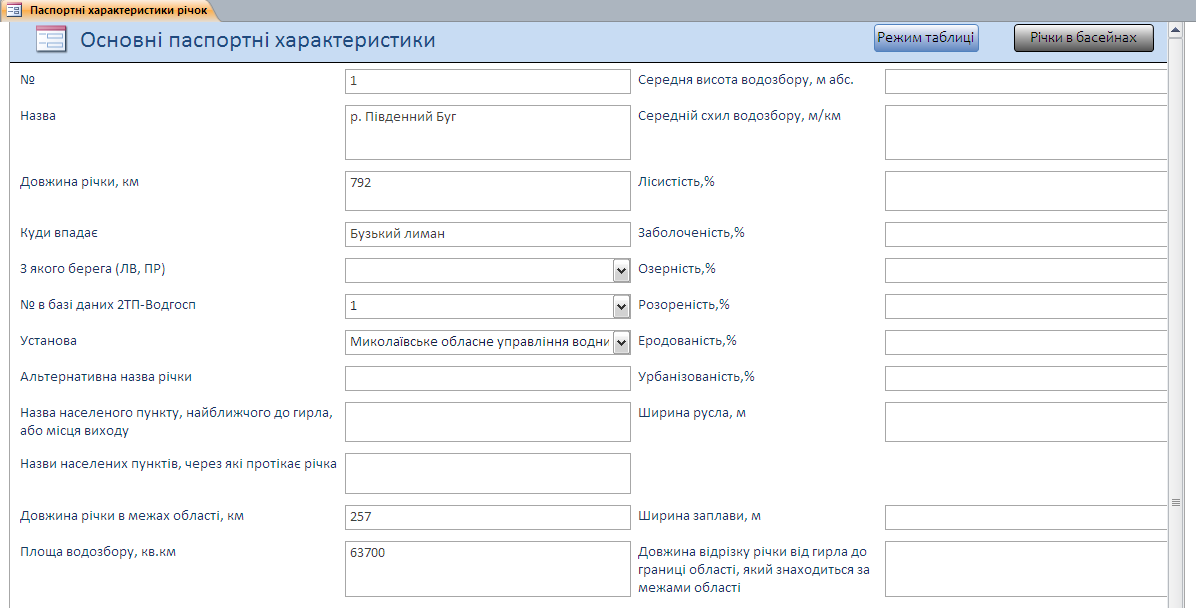
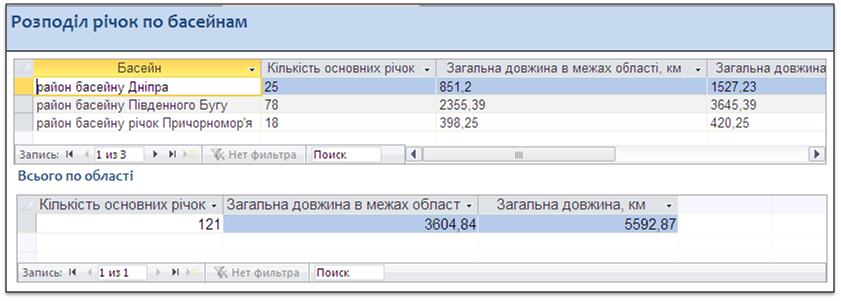


Рис. 2.1. Паспортні дані річок

Рис. 2.2. Паспортні дані річок у водних басейнах

Якщо мова йде про річку, то тут є свої особливості поряд з універсальними параметрами для будь-яких водойм. Замість стандартного розташування, слід вказати початок річки і той водний об’єкт, куди вона впадає, а також описати всі притоки. Крім того, потрібно вказати швидкість і особливості перебігу і витрат води, її колір, температурні перепади, запах і смак. **паспорт річки** мають входити і докладні відомості про рельєф, особливості берегової лінії, грунтах, руслі, а також тваринному та рослинному світі. Не потрібно упускати і момент цільового призначення водойми, — в яких цілях воно використовується і з-за чого забруднюється.

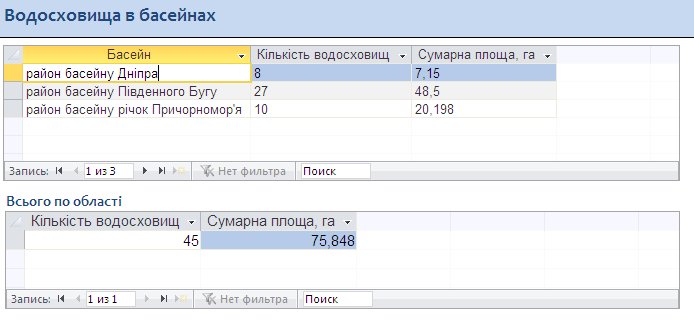


Рис. 2.3. Паспортні дані водосховища в басейнах

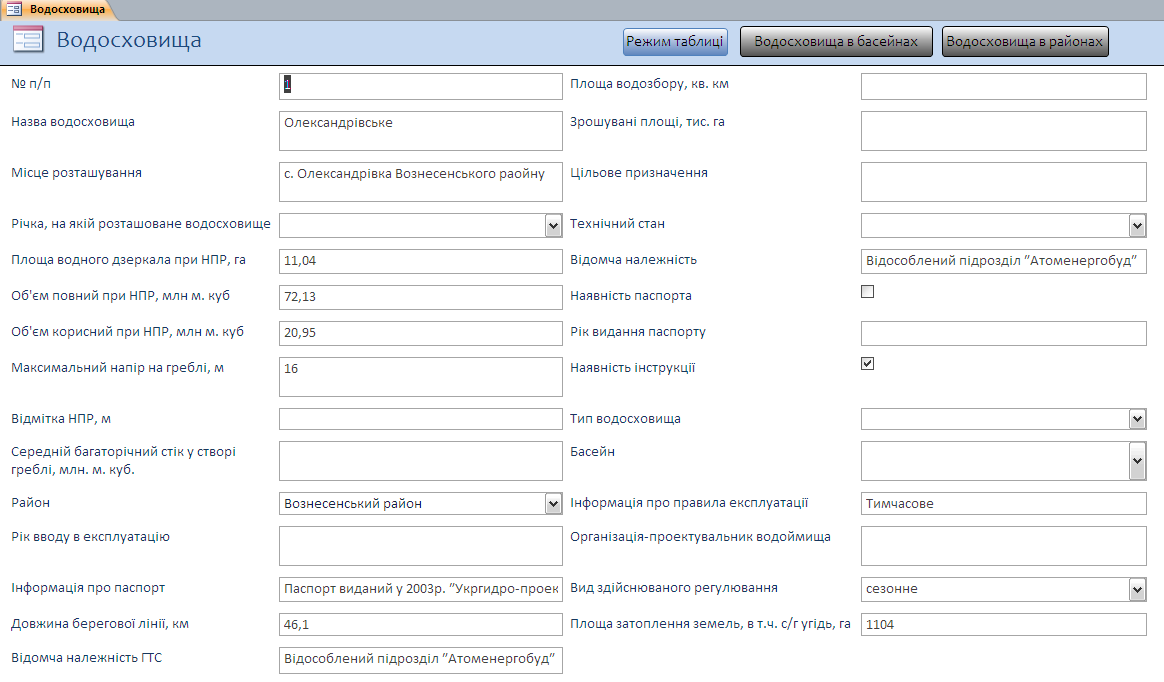


Рис. 2.4. Паспортні дані водосховища

Подібна паспортизація буде і у інших водних об’єктів. Кожен документ повинен містити обов’язкову карту-схему водойми. Як бачимо, в **паспорт** входить цілий ряд параметрів, які повинні бути вказані як можна точніше. Для цього проводиться ряд процедур геодезичного та геологічного характеру. Для того щоб дослідити склад і властивості води, необхідні гідрологічні аналізи, відомості про характер грунту будуть отримані в ході геологічних вишукувань.

А щоб досліджувати геодезичні параметри водойми та прибережної території, застосовуються різні види зйомки. За допомогою топографо-геодезичної зйомки можна отримати найбільш повні відомості про особливості дна водойми, рельєфу берегової лінії. При цьому масштаб може бути як найбільший (1:200), так і дуже дрібний — 1:10000, здатний охопити велику площу і велику кількість інформації.

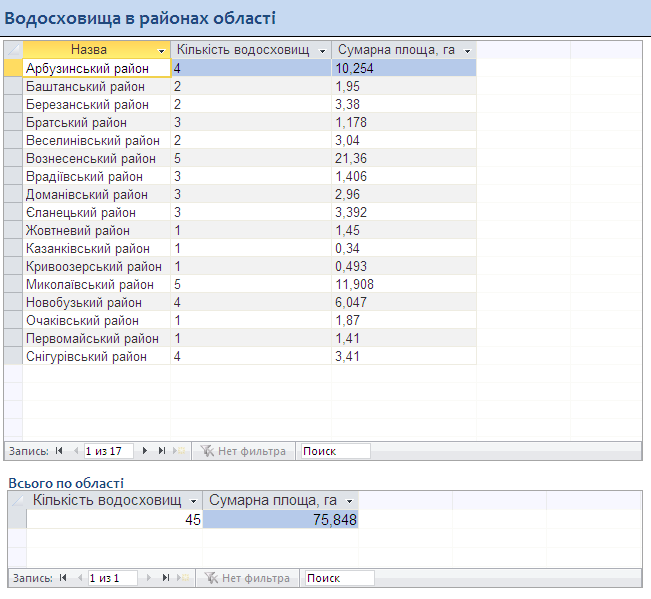


Рис. 2.5. Паспортні дані водосховища в районах області

Всі геодезичні та геологічні процедури, що передують **паспортизації**, а також безпосередньо складання самого офіційного документа якісно виконають професіонали нашої компанії.

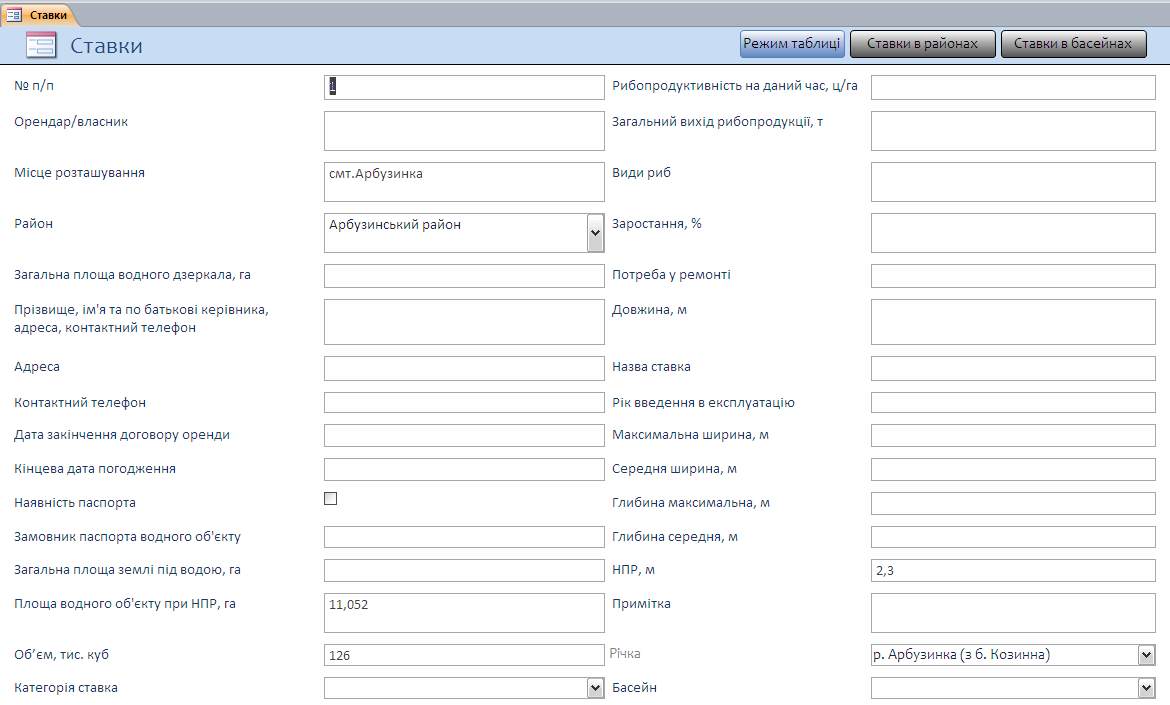


Рис. 2.6. Паспортні дані ставків що перебувають в оренді

**2.2 Аналіз підходів до обліку та пошуку інформації про водні об’єкти**

Відповідно до підписаних Україною міжнародних угод (Орхуська конвенція та ін.) та вітчизняного законодавства органи влади повинні забезпечувати доступ населення до екологічної інформації, у тому числі до актуальних даних про водні ресурси.

Проведені дослідження показали, що оптимальним за функціональністю та вартістю варіантом реалізації подібних систем є використання безкоштовних карт Google Maps, безкоштовної бази даних, наприклад MongoDB, та спеціального програмного забезпечення, яке дозволяє здійснювати зручний пошук даних. У створенні подібних систем варто відзначити провідний досвід Тернопільського обласного управління водних ресурсів зі створення інформаційно-пошукової системи по водних об’єктах області (<http://maps.vodgosp.te.ua/>) (рис. 10.1, 10.2), яку Держодагентство рекомендувало у 2013 році для адаптування в усіх областях України.

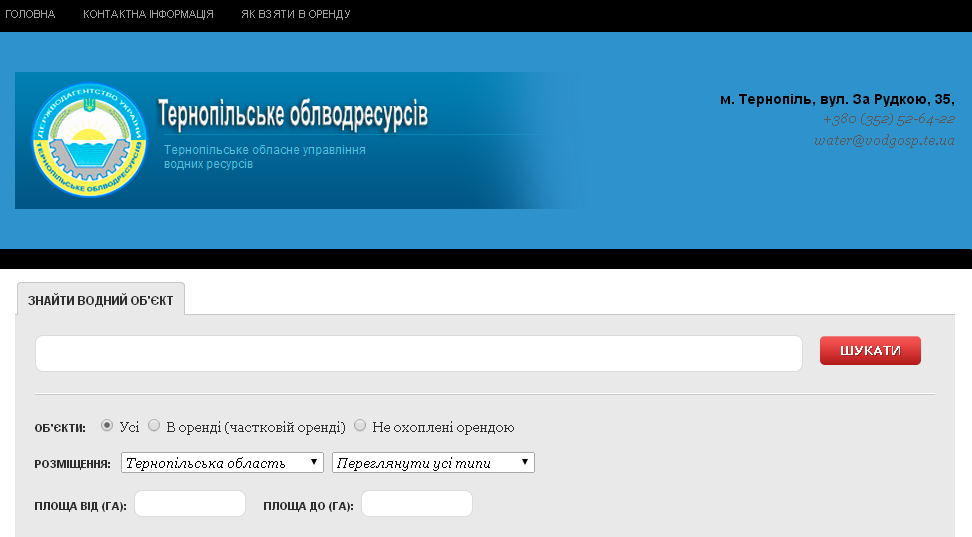


Рис. 2.7 Інтерфейс інформаційно-пошукової системи по водних об’єктах Тернопільської області Тернопільського обласного управління водних ресурсів

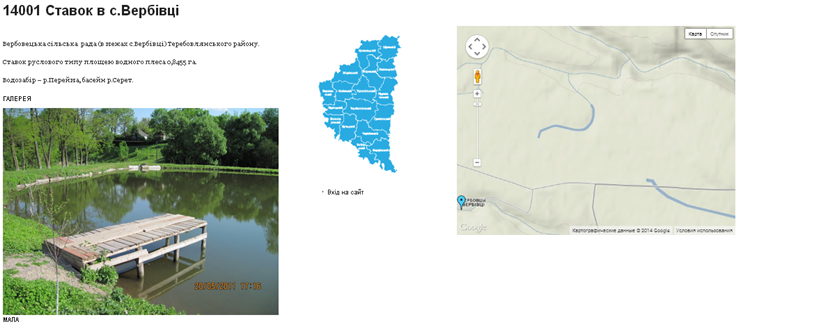


Рис. 2.8. Приклад результату пошуку ставка в інформаційно-пошуковій системі по водних об’єктах Тернопільської області Тернопільського обласного управління водних ресурсів

Однак, ця система має цілий ряд недоліків:

* не має можливості для друку результату пошуку у вигляді довідки,
* дещо обмежені можливості щодо пошуку,
* ускладнений процес внесення даних у систему, неможливість переглядати на карті одночасно багато знайдених об’єктів, позиціонування річки на карті здійснюється лише однією точкою та ін.

Ручний пошук

<http://fish.hd.kiev.ua/>

**2.3 Аналіз сучасних підходів та технологій до створення веб-систем для обліку просторових даних**

При всьому різноманітті різних веб систем для зберігання і пошуку об’єктів, типовий метод складається з трьох основних компонентів:

* розробка стратегії зберігання інформації у базі даних;
* створення серверної логіки для прийняття вихідних даних клієнту для подальшої обробки;
* створення клієнтської частини з інтерфейсом що відповідає вимогам мікроергономіки

Крім цього, побудова веб-системи спирається на апріорну інформацію про предметну область (в даному випадку – водні об’єкти фонду) і коригується вихідними даними користувача, що з'являються по мірі користування веб-системою.

**2.3.1 Клієнтська частина**

Клієнтська частина - це практичне застосування технологій HTML, CSS та JavaScript для створення сайту або веб-додатку, щоб користувач переглядав їх та взаємодіяв з ними напряму. Вимоги що відносяться до клієнстької розробки повязані з інструментами та техніками що використовуються для створення сайту постійно змінюються, і розробник повинен постійно слідкувати за розвитком цих інструментів та технологій.

Задача в розробці сайту - впевнетись що коли користувач відкриє сайт - він побачить релевантну інформацію у зручному для читання вигляді. Це ускладнюється ще й тим, що тепер користувачі використовують велику різноманітність пристроїв з різними розмірами екрану і розширенями, таким чином змушуючи верстальника враховувати ці аспекти при розробці сайту.

Розробник повинен гарантувати, що сайт коректно відображається в різних браузерах (cross-browser), різних операційні системи (cross-platform) і на різних пристроях (cross-device), що вимагає ретельного планування на стороні розробника.

Є кілька інструментів які можуть бути використані для розробки клієнтської частини веб-сайту, розуміння, які інструменти краще всього підходить для виконання конкретних завдань відзначає різницю між оптимізованим та нераціональним сайтом.

Першою з тріади технологій веб сайтів є мова розмітки HTML, це стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документа та ідеологію структурної розмітки тексту. Попри те, що HTML - штучна комп'ютерна мова, вона не є мовою програмування. HTML разом із каскадними таблицями стилів та вбудованими скриптами - це три основні технології побудови веб-сторінок.

HTML впроваджує засоби для:

* створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
* отримання інформації із Всесвітньої мережі через гіперпосилання;
* створення інтерактивних форм;
* включення зображень, звуку, відео, та інших об'єктів до тексту.

Проте для більш зручної стилізації використовуються каскадні таблиці стилів (CSS) - це спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі - сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки - розділення змісту сторінки (даних) та їхньої візуальної презентації.

Однак якщо є потреба створення динамічної та інтерактивної веб системи – потрібно застосувати JavaScript, що дозволить взаємодіяти користувачу з інтерфейсом та організувати клієнт-серверну комунікацію.

JavaScript - динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманний «у спадок» від мови С, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

**2.3.2 Серверна частина**

Веб-сервер - це [сервер](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80), що приймає [HTTP](https://uk.wikipedia.org/wiki/HTTP)-запити від [клієнтів](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), зазвичай [веб-браузерів](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), видає їм [HTTP](https://uk.wikipedia.org/wiki/HTTP)-відповіді, зазвичай разом з [HTML](https://uk.wikipedia.org/wiki/HTML)-сторінкою, зображенням, [файлом](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), медіа-потоком або іншими даними. Веб-сервер - основа [Всесвітньої павутини](https://uk.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D1%81%D0%B2%D1%96%D1%82%D0%BD%D1%8F_%D0%BF%D0%B0%D0%B2%D1%83%D1%82%D0%B8%D0%BD%D0%B0). Веб-сервером називають як [програмне забезпечення](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F), що виконує функції веб-сервера, так і комп'ютер, на якому це програмне забезпечення працює. Клієнти дістаються веб-сервера за [URL](https://uk.wikipedia.org/wiki/URL)-адресою потрібної їм [веб-сторінки](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0) або іншого ресурсу.

Додатковими функціями багатьох веб-серверів є:

- Ведення [журналу серверу](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3) про звернення користувачів до ресурсів

- [Автентифікація](https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F) користувачів

- Підтримка сторінок, що динамічно генеруються

- Підтримка [HTTPS](https://uk.wikipedia.org/wiki/HTTPS) для захищених з'єднань з клієнтами

Сервер - у комп'ютерній термінології термін може стосуватися окремого комп'ютера чи програми. Головною ознакою в обох випадках є здатність машини чи програми переважну кількість часу працювати автономно, без втручання людини, реагуючи на зовнішні події відповідно до встановленого програмного забезпечення. Втручання людини відбувається під час встановлення серверу і під час його сервісного обслуговування. Часто це роблять окремі [адміністратори](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%B8%D0%B9_%D0%B0%D0%B4%D0%BC%D1%96%D0%BD%D1%96%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80) серверів з вищою кваліфікацією.

Сервер як комп'ютер - це [комп'ютер](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80) у [локальній](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0) чи глобальній мережі, який надає користувачам свої обчислювальні і дискові ресурси, а також доступ до встановлених сервісів; найчастіше працює цілодобово, чи у час роботи групи його користувачів.

Сервер як програма - [програма](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0), що надає деякі послуги іншим програмам ([клієнтам](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82)). Зв'язок між клієнтом і сервером зазвичай здійснюється за допомогою [передачі повідомлень](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D1%96%D0%BD_%D0%BF%D0%BE%D0%B2%D1%96%D0%B4%D0%BE%D0%BC%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F%D0%BC%D0%B8), часто через мережу, і використовує певний [протокол](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) для [кодування](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) запитів клієнта і відповідей сервера. Серверні програми можуть бути встановлені як на серверному, так і на персональному комп'ютері, щоразу вони забезпечують виконання певних служб (наприклад, [сервер баз даних](https://uk.wikipedia.org/wiki/%D0%A1%D0%9A%D0%91%D0%94) чи [веб-сервер](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)).

У більшості загального користування сервер фізичного [комп'ютера](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80) (система комп'ютерної техніки) призначений запустити одну або декілька послуг (як приймаюча сторона) для задоволення потреб користувачів інших комп'ютерів в мережі. В залежності від обчислювальних послуг, які вона пропонує, це може бути сервер баз даних, файловий сервер, поштовий сервер, сервер друку, веб-сервер, ігровий сервер, або якийсь інший сервер. У контексті архітектури клієнт-сервер, сервер являє собою комп'ютерну програму, яка обслуговує запити інших програм - «клієнтів».

Таким чином, сервер виконує деякі обчислювальні завдання від імені "клієнтів". Сервери часто надають основні послуги через мережу, або в приватних користувачів - всередині великої організації або громадським користувачам - через [Інтернет](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Мережевий сервер являє собою комп'ютер, призначений для обробки запитів і передачі даних на інші (клієнт) комп'ютери по локальній мережі або через Інтернет. Мережеві сервери зазвичай конфігуруються з додатковою пам'яттю і ємністю для обробки навантаження з обслуговування клієнтів.

Вимоги до обладнання для серверів варіюються залежно від сервера додатків. Абсолютна швидкість [процесора](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) не настільки важлива для сервера, як для настільного комп'ютера. Обов'язки сервера надавати послуги багатьом користувачам по мережі призводять до різних вимог, таких як швидке підключенням до мережі та висока [пропускна спроможність](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%83%D1%81%D0%BA%D0%BD%D0%B0_%D1%81%D0%BF%D1%80%D0%BE%D0%BC%D0%BE%D0%B6%D0%BD%D1%96%D1%81%D1%82%D1%8C_%D0%BA%D0%B0%D0%BD%D0%B0%D0%BB%D1%83). Так як сервери, як правило, доступні по мережі, вони можуть працювати без монітору. Процеси, які не потрібні для функції сервера не використовуються. Багато серверів не мають графічного інтерфейсу користувача. Крім того, аудіо-та USB інтерфейси можуть бути опущені. Сервери часто працюють протягом тривалого часу без перерви, тому надійність обладнання і довговічність надзвичайно важлива.

Хоча сервери можуть бути побудовані з частин комп'ютера, критично важливі корпоративні сервери не можливі без використання спеціалізованого устаткування з низьким рівнем збою в цілях максимального часу безперебійної роботи, оскільки навіть короткострокові невдачі можуть коштувати дорожче, ніж покупка і установка системи . Наприклад, це може зайняти всього декілька хвилин часу простою на національній [фондовій біржі](https://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D0%BD%D0%B4%D0%BE%D0%B2%D0%B0_%D0%B1%D1%96%D1%80%D0%B6%D0%B0), щоб виправдати рахунок повністю замінити системи з чимось більш надійним. Сервери можуть включати в себе більшу ємність [жорстких дисків](https://uk.wikipedia.org/wiki/%D0%96%D0%BE%D1%80%D1%81%D1%82%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA), більше комп'ютерних вентиляторів або водяного охолодження, щоб допомогти усунути тепло, і джерела безперебійного живлення, які забезпечують роботу сервера в разі збою живлення. Ці компоненти забезпечують більш високу продуктивність і надійність за відповідно більш високою ціною.

**2.3.3 Аналіз існуючих баз даних**

База даних – сукупність [даних](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96_(%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%82%D0%B5%D1%85%D0%BD%D1%96%D0%BA%D0%B0)), організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти.

Дані у базі організовують відповідно до моделі організації [даних](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96). Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки.

В загальному випадку базою даних можна вважати будь-який впорядкований набір даних. Наприклад, паперову картотеку з формулярами про працівників підприємства у відділі кадрів. Але дана стаття зосереджена на використанні баз даних в [інформаційних системах](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0). На даний час [застосунки](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) для роботи з базами даних є одними з найпоширеніших прикладних програм.

У сучасних мережевих інформаційних системах для роботи із загальною базою даних використовують архітектуру [«клієнт-сервер»](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). При цьому в мережі розміщують сервер баз даних. Ним виступає [комп'ютер](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80) (або комп'ютери), який містить бази даних, [СКБД](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) та пов'язане з ними програмне забезпечення, і налаштований для надання користувачам інформаційної системи доступу до бази даних.

Клієнти, які працюють із даними (вони можуть бути розташовані на різнихкомп'ютерах мережі), надсилають відповідні запити серверу. Сервер їх отримує, опрацьовує, та надсилає відповідь клієнту.

Сучасні [СКБД](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) ([MySQL](https://uk.wikipedia.org/wiki/MySQL" \o "MySQL), [PostgreSQL](https://uk.wikipedia.org/wiki/PostgreSQL" \o "PostgreSQL), [Microsoft SQL Server](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server) та інші) працюють відповідно до цієї архітектури. Сервер баз даних, як правило, є достатньо потужною багатопроцесорною системою, яка використовує масиви дисків [RAID](https://uk.wikipedia.org/wiki/RAID) для підвищення надійності зберігання даних. Використання дискових масивів [RAID](https://uk.wikipedia.org/wiki/RAID) дозволяє відновити дані, навіть якщо один з дисків вийшов з ладу.

Первинним призначенням бази даних є зберігання масивів даних. Але їх широко використовують і для збереження адміністративної інформації та спеціалізованих даних, наприклад, для інженерних даних чи для економічних моделей. Прикладами використання баз даних можуть бути:

* автоматизовані системи обліку;
* реєстри та каталоги;
* геоінформаційні системи;

Об'єктно-орієнтована база даних - база даних, яка створюється і використовується в середовищі СУБД, яке засновано на принципах об'єктно-орієнтованого підходу і підтримує об'єктну модель даних. Об'єктна база даних являє собою сукупність взаємозв'язаних об'єктів, які відповідають певній схемі.

Використання об’єктної моделі не є доцільним для предметної області нормативних документів ВНТУ, так як переваги такої моделі будуть слабо відчуватися через малу потужність універсального відношення. Крім того виникнуть проблеми на фізичному та даталогічному рівнях проектування у зв’язку із складністю реалізації алгоритмів роботи СУБД.

Після аналізу основних моделей даних, для курсової роботи було обрано реляційну модель даних. Саме ця модель має ряд переваг, таких як: незалежність від фізичного рівня представлення, зручність і розуміння організації даних користувачами, максимальна гнучкість при обробці непередбачених запитів, можливість розширення бази приєднанням нових елементів, записів без зміни при цьому існуючих підсхем та прикладних програм.

Через такі недоліки ієрархічної моделі, як: надлишковість зберігання інформації, так як ієрархічні структури не підтримують взаємозв'язки Б:Б; строгу ієрархічну впорядкованість, яка ускладнює процедури включення та вилучення записів; вилучення вихідних вузлів призводить до вилучення відповідних їм породжених , що вимагає особливої обережності; ускладнюється доступ до даних , які лежать на більш низьких рівнях ієрархії, так як кореневий вузол завжди є головним, а доступ до любого породженого вузла може здійснюватись через вихідний, ієрархічна модель не може бути використана для даної курсової роботи.

А недоліки використання мережевої моделі даних полягають в складності по відношенню до ієрархічних структур. Прикладному програмісту часто необхідно знати логічну структуру бази даних.

**2.3.4 Аналіз існуючих геопорталів**

Розглянемо геопортали які активно використовуються для розробки інформаційно-довідкових веб-систем що працюють з відображенням геоданих в Україні та за кордоном. Варто відзначити наступні сервіси:

* Open Street Map
* Map.ua
* Публічна кадастрова карта України
* Google Maps

Розглянемо кожен з вказаних ресурсів та оцінемо можливість доступу до інформації що вони надають, цінність для візуалізації водних об’єктів та зручність у використанні.

Open Street Map – — це відкритий проект зі створення загальнодоступних мап світу силами спільноти. У квітні 2006-го OSM зареєстровано як фонд. Фонд OpenStreetMap — міжнародна некомерційна організація, створена для підтримки розвитку та розповсюдження геопросторових даних, а також надання можливості використання геопросторових даних будь-ким. Волонтери цього проекту використовують аерофотознімки, GPS-пристрої та низькотехнологічні карти земель для перевірки і валідації даних що дані OSM є точними та актуальними

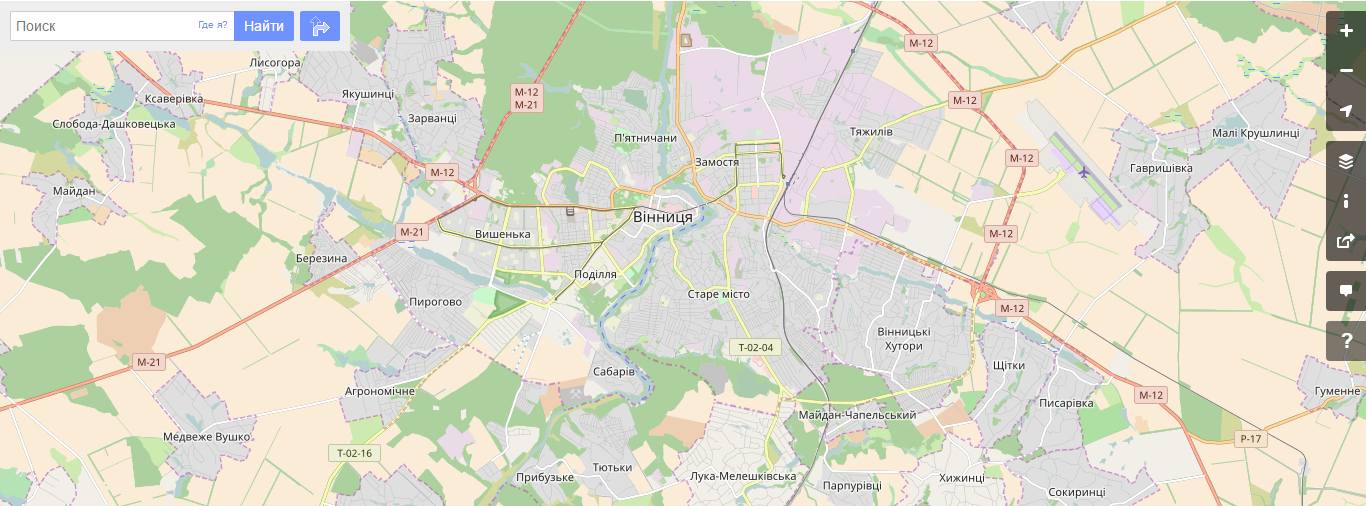


Рис. 2.9. Інтерфейс картографічного сервісу Open Street Map

У сервісу Open Street Map не надає прямого доступу до свої данних через API. Розробнику пропонується скачати з архіву Open Street Map що знаходиться за адресою <https://planet.openstreetmap.org/>. На цьому сайті адміністрація Open Street Map щотижня завантажує копію усіх геоданих у чистому вигляді що використовуються сервісом Open Street Map. Окрім з цим розробник повинен скачати серверну та клієнтську частину що знаходиться за адресою <https://github.com/openstreetmap/openstreetmap-website> встановити відповідні додатки та сконфігорувати локальний ресурс з завантаженою копією данних. Це рішення не є оптимальним, так як воно вимагає створеня додаткового ресурсу як сервера картографічних данних, або інтеграція програмного коду в інформаційно-довідкову веб-систему разом з вивчення технологій та програмної мови на яких створений додаток Open Street Map

Також варто відзначити що портал Open Street Map працює за тими ж принципами що і сайт Вікіпедія – кожен зареєстрований учасник може вносити свої зміні які з певним часом будуть перевірені такими ж учасниками як і він, або ж адмістраторами та модераторами ресурсу. Це створить певні ризики при використані Open Street Map для інформаційно-довідкового веб-сайту обліку об’єктів водного фонду Миколаївської області так як інформація та дані що використовуються повинні відображатись на картах котрі гарантують високу актуальність даних, у чому важко переконатись через вищеописані факти.

Map.ua – український картографічний сервіс що охоплює дані тільки Української території що знаходиться за адресом <http://map.ua/ukr/>. Геодані сервісу Map.ua надає Науково-Дослідний Інститут Геодезії і Картографії.

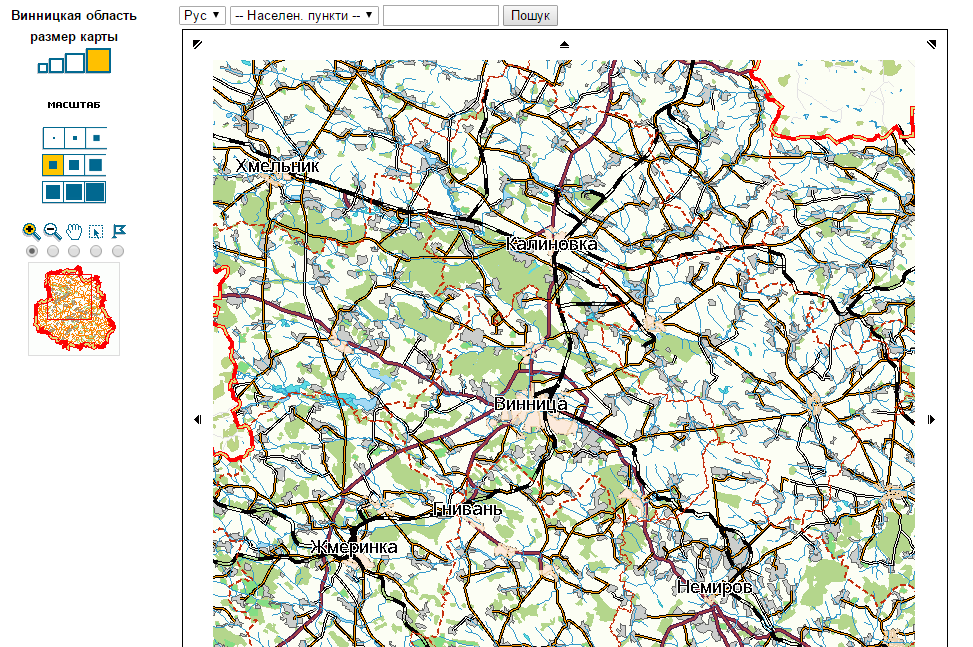


Рис. 2.10. Інтерфейс картографічного сервісу Map.ua

На сайті є посилання на документацію щодо користування API у вигляді цільного PDF файлу що є не типовим форматом для загальнодоступної документації по API. Також документація написана у довільному не технічному стилі, хоча і має зміст що дозволяє орієнтуватись в методах користування сервісом. Однак за таких умов важко виконати інтелектуальний пошук по документації.

Також даний ресурс використовує застарілу технологію відображення карти – при кожній взаємодії з картою (збільшення, зменшення, переміщення) потрібно виконувати запит на сервер, котрий згенерує зображення з результатом, це зображення потрібно буде оновити на сайті щоб користувач побачив зміну. Таки підхід є нераціональним з програмної точки зору, так як кожен запит містить в собі запит на сервер, що буде використовувати трафік користувача, та кожен запит буде займати деякий час, що в результаті виглядатиме повільним на відміну від Open Street Map.

Публічна кадастрова карта України **-** це інформаційний портал, на якому оприлюднюються відомості Державного земельного кадастру що знаходиться за адресом <http://map.land.gov.ua/>. Переглядаючи ПКК можна здійснювати навігацію між картографічними матеріалами та виводити на екран комбіновану інформацію з різних інформаційних шарів (ортофотоплани та ін.).

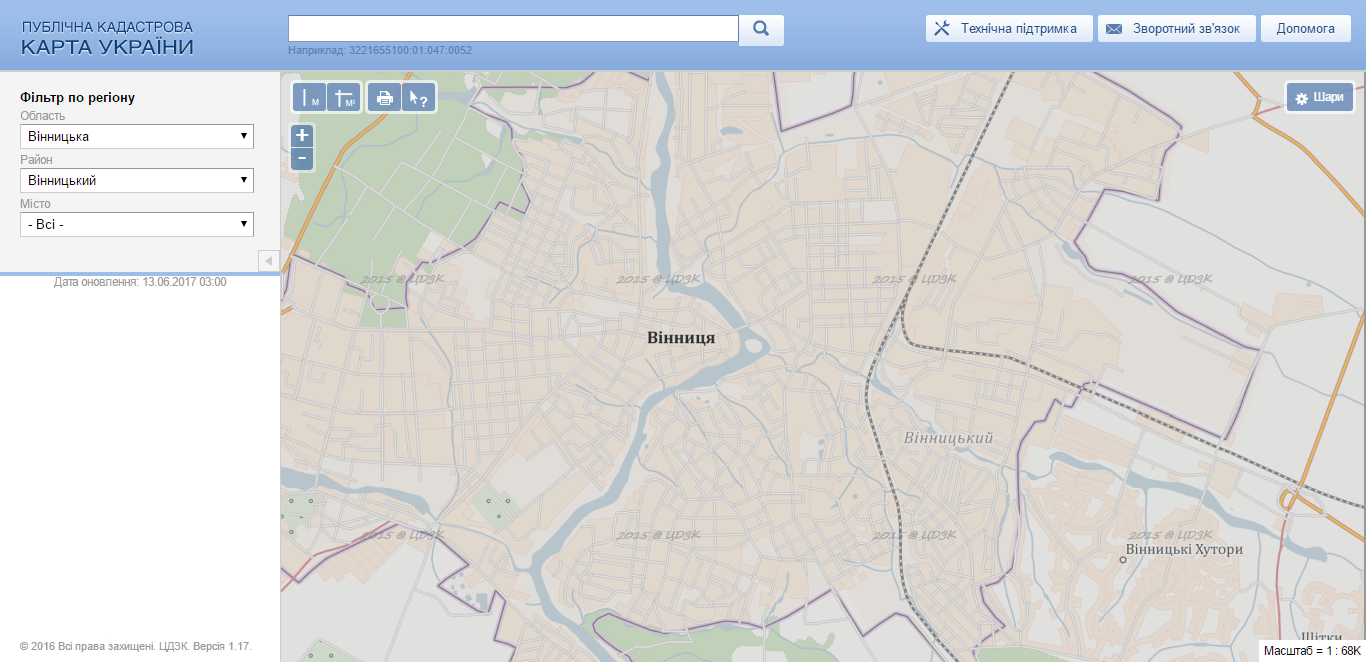


Рис. 2.11. – Інтерфейс інформаційного порталу публічної кадастрової карти України

На сайті ПКК є можливість експорту інформації про межі адміністративно-територіальних одиниць (областей, районів), кадастровий поділ та місцерозташування земельних ділянок, а також інформацію про грунти у вигляді WMS шару. Однак така можливість підходить лише для програмних продуктів QGIS та ArcMap, а загальнодоступного API публічна кадастрова карта не надає, а дані що експортуються в вищевказані програмні продукти є лише інформацією про кадастровий поділ і ніякої цінності для відображення водних об'єктів не несе. Також у публічної кадастрової карти обмежений високий масштаб – картографічних результатів вище масштабу 1:68K немає:

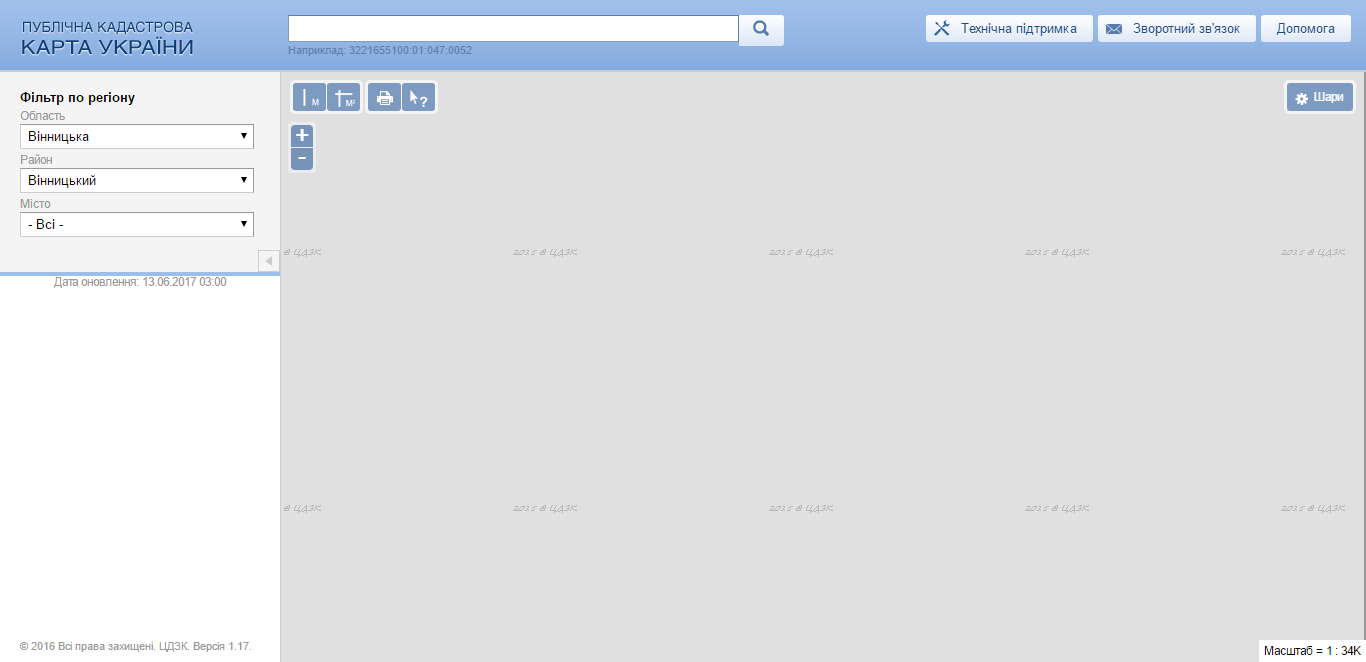


Рис. 2.12. Зображення публічної кадастрової карти при масштабі 1:34К

Варто додати що для високих масштабів існують шари ортофотопланів та карти з масштабом 1:100000К проте на кожному з цих шарів низька кількість позначок про річки, ставки та водосховища, та сама якість зображення карти низька – що робить її непридатною для використання у розробці інформаційно-довідковій веб-системі пошуку водних об’єктів Миколаєвської області.

Google Maps **-** набір додатків, побудованих на основі безкоштовного картографічного сервісу і технологій, які надає компанія Google. Існує можливість використовувати сервіс для створення своїх продуктів сторонніми компаніями. На сьогоднішній день це безкоштовна служба.

Для розробників сайтів зручно буде використати JavaScript для керування функціональністю карт, кількість запитів з одного сервера обмежена. Google Static Maps API дозволяє будувати статичні мапи. Також існують версії API під різні види мобільних пристроїв.

Також сервіс надає знімки поверхності зі супутників, перегляд вулиць у 360° панорамі, візуалізація рельєфу планети, інформацію про завантаженість доріг та ін.

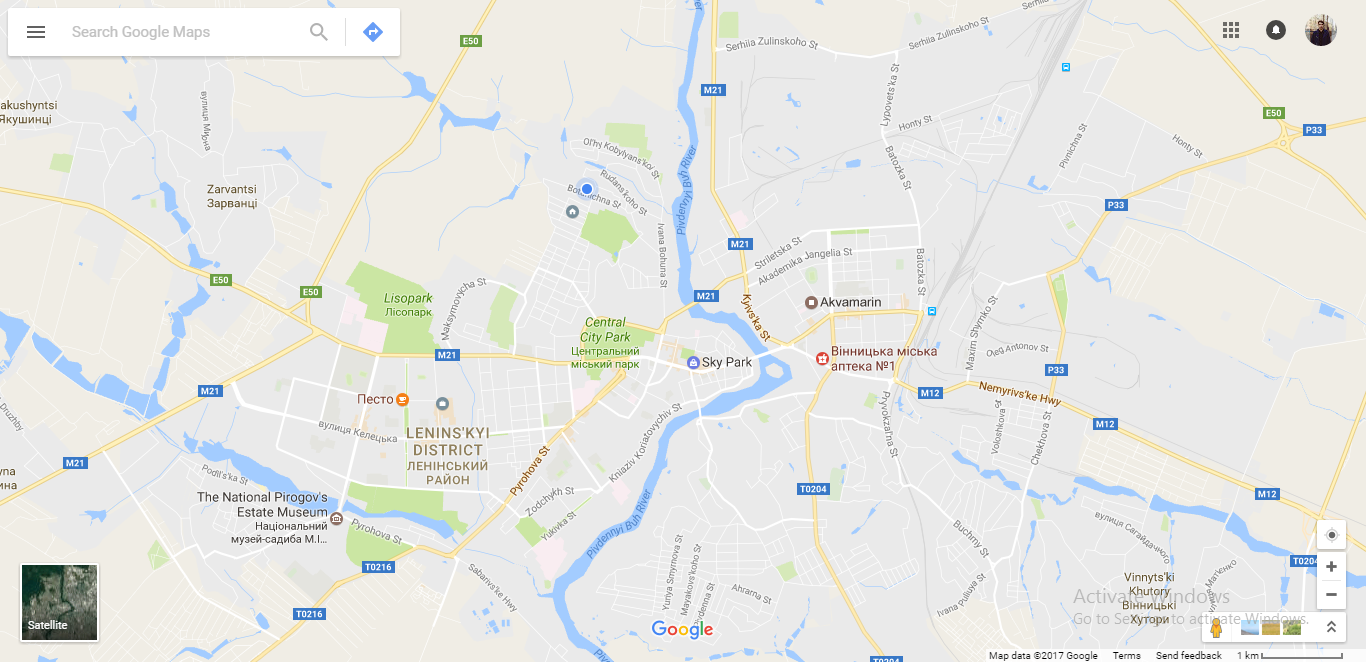


Рис. 2.13. Інтерфейс картографічного сервісу Google Maps

На відміну від вище оглянутих картографічних сервісів Google Maps – адмініструється та модерується виключно компанією Google Maps, хоча і є можливість пропонувати зміни у карті для користувачів – ці зміни будуть перевірині на достовірність та актуальність. Також користувацький інтерфейс набагато зручніше аніж на сайті Map.ua. Наявність високодеталізованої топографічної карти та зображень зі супутників у високій якості дозволяють зручно переглядати карту при високих масштабах.

На сайті сервісу є посилання що веде на сайт з документацією по загальнодоступному АРІ для Google Maps.

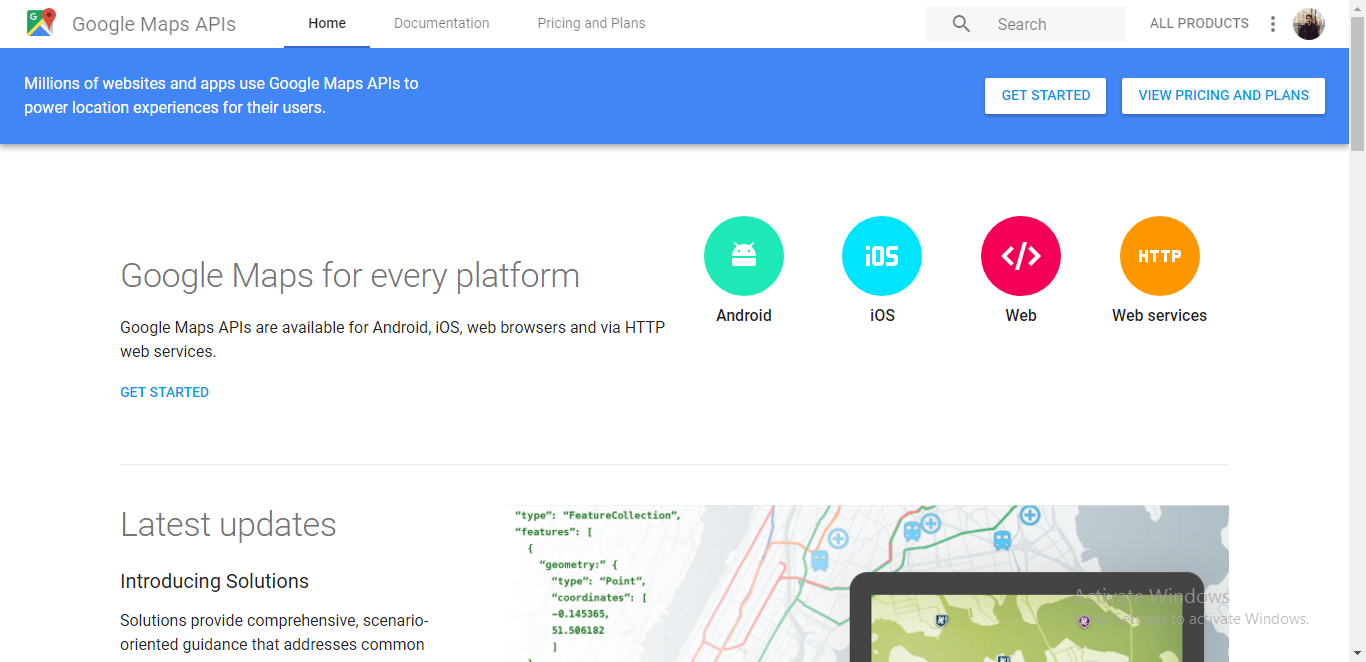


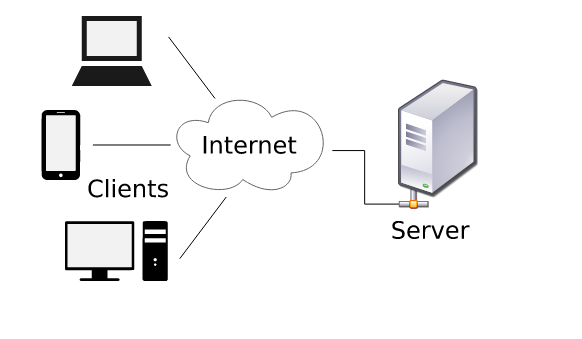
Рис. 2.14. Сайт з документацією по загальнодоступному АРІ для Google Maps.

**2.3.5 Клієнт-серверна архітектура**

Архітектура клієнт-сервер є одним із [архітектурних шаблонів програмного забезпечення](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%BD%D1%96_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) та є домінуючою концепцією у створенні розподілених мережних [застосунків](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

* набір [серверів](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80), які надають інформацію або інші послуги програмам, які звертаються до них;
* набір [клієнтів](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), які використовують сервіси, що надаються серверами;
* [мережа](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D1%83%D0%BD%D1%96%D0%BA%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.



Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

- прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідна обробка інформації;

- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів - клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка застосунку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузерами (веб-оглядачами). Найпоширеніші браузери: Google Chrome, Internet Explorer, Firefox, Safari і Opera.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме - автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

3. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

При розробці веб-системи, необхідно звернути увагу на функції, які повинна вона виконувати, а саме: зручність та швидкість пошуку.

Пошук водних об’єктів – клопіткий процес, який потребує детального аналізу. Не завжди вдається знайти той чи інший ставок, річку або водосховище у відповідних архівах. Тому використання програмного засобу для такого завдання є актуальним.

Веб-система для пошуку водних об’єктів повинен отримати вхідні критерії водного об’єкту, та виконати відповідні алгоритми на сервері для перевірки наявності таких об’єктів, знайти геодані, відповідну статистику для вказаного типу водного об’єкту (ставок, річка, водосховище). Після пошуку веб-система повинна вивести інформацію знайдені об’єкти та відобразити їх користувачу.

Це повинно полегшити процес пошуку водних об’єктів для людей, які мають справи з водними кадастрами, підприємців, особи що працюють у сфері гідрології, або просто бажаючих дізнатись більше про водні об’єкти. Веб-система є інструментом, який забезпечить зручний та ефективний пошук водних об’єктів на відміну від малофункціональних або важких для розуміння аналогів.

Проблема полягає у відсутності простого, безкоштовного, і водночас багатофункціонального та продуктивного інструменту для пошуку водних об’єктів. Дана проблема стосується осіб що працюють у сфері ботаніки, або просто бажаючих дізнатись більше інформації про рослин. Як наслідок – виникають проблеми з швидким пошуком інформації про певні водні об’єкти. Вирішення проблеми дозволить створити ефективний та зручний інструмент для пошуку водних об’єктів.

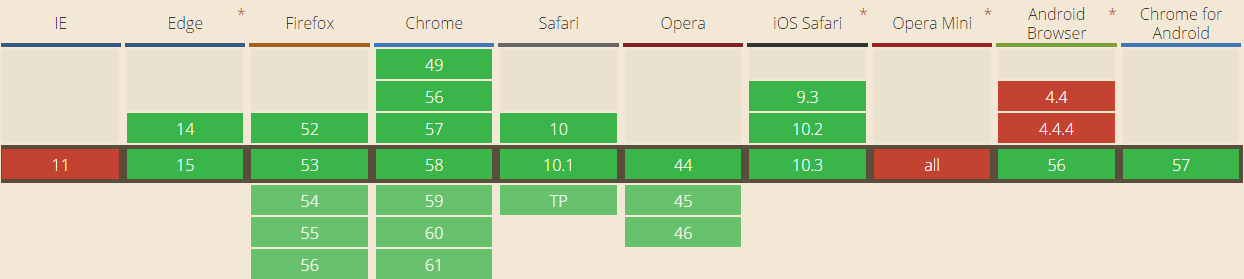
Для створення веб-системи що дозволить виконувати пошуку водних об’єктів використаємо стек MERNRW (MySQL, Express, React, NodeJS, Redux, Webpack).

Стек - набір програмних підсистем або компонентів, необхідних для створення повної платформи таким чином, що ніякого додаткового програмного забезпечення не вимагається для підтримки додатків.

Стек MERNRW складається з чотирьох основних технологій:

* MySQL - вільна система керування реляційними базами даних, використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування. Це компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання
* Express - програмний каркас розробки веб-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення під ліцензією MIT. Він спроектований для створення веб-застосунків і API
* React - відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків. React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки. React обробляє тільки користувацький інтерфейс у застосунках
* Node.js - платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Node.js характеризується такими властивостями як асинхронна однопотокова модель виконання запитів, неблокуючий ввід/вивід, система модулів CommonJS, рушій JavaScript Google V8
* Redux - JavaScript бібліотека з відкритим кодом створена для управління станом веб-додатка. В основному використовується разом з React для розробки гнучких веб інтерфейсів.
* Webpack - JavaScript модуль збірки з відкритим кодом. Webpack бере модулі веб додатку та їх залежності та генерує статичні файли що репрезентують ці модулі. Також може відобразити граф залежностей проекту, що дозволить розробнику застосувати модулярний підхід для створення веб додатку

Однією з особливостей Node.js та React є можливість використання експерементального стандарту JavaScript що має назву ECMAScript 6 або скорочено ES6. На відміну від свого попередника ECMAScript 2015 в ES6 раціонально реалізовано об’єктно-орієнтований підхід та ряд інших функцій що дозволяє розробнику створювати більш гнучкі додатки. Проте стандарт ES6 підтримується лише у 73% браузерів на всіх пристроях [15]



Однак цю ситуацію можна виправити за допомогою JavaScript компіляторів, що дозволяють виконати конвертацію коду стандарту ES6 в стандарт ECMAScript 2015, котрий підтримується набагато більшою кількістю браузерів. Для такої задачі найчастіше за все використовують Webpack. Він дозволяє налаштувати процес непреривної компіляцій вихідних файлів JavaScript ES6 у JavaScript ECMAScript 2015.

4. ПОБУДОВА ІНФОРМАЦІЙНОЇ МОДЕЛІ СИСТЕМИ