

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

“ ___ ” червня 2020 р.

Дипломний проект

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системне програмування»

зі спеціальності

123 «Комп'ютерна інженерія»

на тему: Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини

Виконав: студент IV курсу, групи КВ-61

(шифр групи)

Сметана Максим Олександрович _____

(прізвище, ім'я, по батькові)

(підпис)

Керівник доц. каф. СПіСКС, к.т.н., доцент Сапсай Т.Г. _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з нормоконтролю, доц.каф.СПіСКС, к.т.н. Клятченко Я.М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2020 року

Пояснювальна записка до дипломного проекту

на тему: Веборієнтований сервіс для сортування та пошуку місць прийому
вторинної сировини

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віталій РОМАНКЕВИЧ

(підпис) (ініціали, прізвище)

«__» _____ 20__ р.

**ЗАВДАННЯ
на дипломний проєкт студента**

Сметана Максим Олександрович

1. Тема проєкту «Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини»,

керівник проєкту доц. каф. СПіСКС к.т.н., доцент Сапсай Т.Г.

затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Термін подання студентом проєкту 22.05.2020

3. Вихідні дані до проєкту Назва. Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини.

4. Зміст пояснювальної записки:

- аналіз існуючих рішень та обґрунтування теми бакалаврського проєкту
- вибір програмних засобів для реалізації проєкту
- розробка проєкту

5. Перелік графічного матеріалу: презентація, структурна схема додатку, схема алгоритму сортування сировини, схема алгоритму фільтрації місць прийому сировини, схема алгоритму роутингу.

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
нормоконтроль	Клятченко Я.М., к.т.н., доцент каф. СПіСКС		

7. Дата видачі завдання 30.10.2019

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	При мітка
1.	Вивчення літератури за тематикою проєкту	18.11.2019	
2.	Розроблення та узгодження технічного завдання	30.11.2019	
3.	Аналіз існуючих рішень	20.01.2020	
4.	Підготовка матеріалів першого розділу проєкту	27.01.2020	
5.	Розроблення програмного забезпечення	21.02.2020	
6.	Відлагодження програмного продукту	20.03.2020	
7.	Підготовка матеріалів другого розділу проєкту	09.04.2020	
8.	Розроблення інтерфейсу програмного забезпечення	22.04.2020	
9.	Підготовка графічної частини	04.05.2020	
10	Оформлення документації дипломного проєкту	16.05.2020	

Студент _____
 Керівник проєкту _____

Максим СМЕТАНА
 Тетяна САПСАЙ

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (55 с., 33 рис., 4 додатки).

Об'єкт розробки – створення веборієнтованого сервісу для сортування та пошуку місць прийому вторинної сировини, яке спрямоване на полегшення процесу сортування сировини.

Розроблений вебдодаток дозволяє:

- отримати інформацію про сортування необхідної сировини;
- сортувати всі вибрані сутності сировини за допомогою реалізованого алгоритму;
- шукати місця прийому сировини на мапі;
- фільтрувати місця прийому на мапі.

В ході виконання дипломного проєкту:

- розроблено вебдодаток для сортування та пошуку місця прийому вторинної сировини;
- проведено аналіз існуючих рішень;
- розроблено структуру взаємодії клієнта та серверу.

Ключові слова: СОРТУВАННЯ ВТОРИННОЇ СИРОВИНИ, ПОШУК МІСЦЬ ПРИЙОМУ, ВЕБДОДАТОК, МАПА, СЕРВЕР, КЛІЄНТ.

ABSTRACT

Qualification work includes an explanatory note (55 pages, 33 pictures, 4 appendices).

The object of research is to create a web-oriented application for sorting and searching the places for the reception of recyclable materials, that is directed to simplify recyclable materials sorting process.

Developed web-application allows to:

- receive the information about necessary raw materials sorting;
- sort all the chosen recyclable materials according to the implemented algorithm;
- search for recyclable materials reception places on the map;
- to filter reception places on the map.

During the implementation of the diploma project:

- the web-application for sorting and searching the places for the reception of recyclable materials was developed;
- all existing solutions were analyzed;
- the structure of interaction between the client and server was created.

Keywords: RECYCLABLE MATERIALS SORTING, RECEPTION PLACES SEARCHING, WEB-APPLICATION, MAP, SERVER, CLIENT.

№ з/п	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	Примітки
1	A4	ІАЛЦ.045490.002 ТЗ	Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини. Технічне завдання	4	
2	A4	ІАЛЦ.045490.003 ТП	Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини. Відомість технічного проекту	1	
3	A4	ІАЛЦ.045490.004 ПЗ	Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини. Пояснювальна записка	55	
4	A4	ІАЛЦ.045490.005 Д1	Загальна структура додатку. Схема структурна.	1	

					ІАЛЦ.045490.001 ОА			
Змін.	Арк.	№ докум.	Підпис	Дата				
Розробив	Сметана М.О.				Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини. Опис альбому	Літ.	Аркуш	Аркушів
Перевірив	Сапсай Т.Г.						1	2
Консуьлт.						КПІ ім. Ігоря Сікорського, ФПМ КВ-61		
Н. контроль	Клятченко Я.М.							
Зав. каф.	Романкевич В.О							

ЗМІСТ

1.НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.	2
2.ПІДСТАВА ДЛЯ РОЗРОБКИ.	2
3.ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ.	2
4.ДЖЕРЕЛА РОЗРОБКИ.	2
5.ТЕХНІЧНІ ВИМОГИ.	2
5.1.Вимоги до програмного продукту, що розробляється.	2
5.2.Вимоги до апаратного забезпечення.	3
5.3.Вимоги до програмного та апаратного забезпечення користувача. .	3
6.ЕТАПИ РОЗРОБКИ.	4

						ІАЛЦ.045490.002 ТЗ		
Змін	Арк.	№ докум.	Підпис	Дата				
Розробив		Сметана М.О.			Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини Технічне завдання	Літ.	Аркуш	Аркушів
Перевір.		Сапсай Т.Г.					1	4
Н. контроль		Клятченко Я.М.				НТУУ «КПІ ім. Ігоря Сікорського», ФПМ КВ-61		
Затвердив		Романкевич В.О.						

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини».

Галузь застосування: розробка вебдодатку для сортування та пошуку місць прийому вторинної сировини.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення сервісу в мережі інтернет для полегшення процесу сортування та пошуку місць прийому вторинної сировини.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- простий та зручний інтерфейс;
- реалізація алгоритму сортування;
- відображення місць прийому на мапі;

					ІАЛЦ.045490.002 ТЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

- можливість фільтрації місць прийому на мапі;
- можливість пошуку інформації про сировину.

5.2 Вимоги до апаратного забезпечення

- наявність доступу до глобальної мережі Internet.

					ІАЛЦ.045490.002 ТЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		3

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Видача завдання на дипломне проєктування	30.10.2019
2.	Вивчення літератури за тематикою роботи	20.11.2019
3.	Розроблення та узгодження технічного завдання	31.11.2019
4.	Розроблення структури додатку	20.01.2020
5.	Розроблення дизайну та графічних елементів	09.02.2020
6.	Програмна реалізація додатку	21.02.2020
7.	Тестування додатку	22.03.2020
8.	Підготовка матеріалів текстової частини проєкту	12.04.2020
9.	Підготовка матеріалів графічної частини проєкту	04.05.2020
10.	Оформлення технічної документації проєкту	16.05.2020

№ З/п	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	Примітки
1	A4	ІАЛЦ.045490.004 ПЗ	Веборієнтований сервіс для сортування та пошуку місць місць прийому вторинної сировини. Пояснювальна записка	55	
2	A4	ІАЛЦ.045490.005 Д1	Загальна структура додатку. Схема структурна.	1	
3	A4	ІАЛЦ.045490.006 Д2	Алгоритм сортування сировини. Схема алгоритму.	1	
4	A4	ІАЛЦ.045490.007 Д3	Алгоритм фільтрації місць прийому сировини. Схема алгоритму.	1	
5	A4	ІАЛЦ.045490.008 Д4	Алгоритм роутингу. Схема алгоритму.	1	
6		Диск CD-ROM	Текст пояснювальної записки. Графічний матеріал	1	

					ІАЛЦ.045490.003 ТП						
Змін.	Арк.	№ докум.	Підпис	Дата							
Розробив	Сметана М.О.				Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини. Відомість технічного проекту			Літ.	Аркуш	Аркушів	
Перевірив	Сапсай Т.Г.								1	1	
Консульт.								КПІ ім. Ігоря Сікорського, ФПМ КВ-61			
Н. контроль	Клятченко Я.М.										
Зав. каф.	Романкевич В.О										

ЗМІСТ

Перелік скорочень, умовних позначень, термінів _____	3
ВСТУП _____	4
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ БАКАЛАВРСЬКОГО ПРОЄКТУ _____	6
1.1. Різновиди вебсайтів _____	6
1.2. Аналіз існуючих рішень _____	8
1.3. Обґрунтування теми бакалаврського проєкту _____	11
2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЄКТУ ____	13
2.1. Вибір програмних засобів для реалізації клієнтської частини __	13
2.1.1. Мова програмування JavaScript _____	13
2.1.2. Бібліотека React _____	15
2.1.3. Бібліотека Redux _____	18
2.2. Вибір програмних засобів для реалізації серверної частини __	19
2.6. Програмна платформа Node.js _____	19
2.5. База даних MongoDB _____	20
3. РОЗРОБКА ПРОЄКТУ _____	24
3.1. Клієнт-серверна архітектура _____	24
3.2. Розробка клієнтської частини проєкту _____	28
3.2.1. Опис розділів вебдодатку _____	29
3.2.2. Роутинг вебдодатку _____	30
3.2.3. Опис розділу сортування _____	32
3.2.4. Опис розділу мапи _____	42
3.2.5. Опис розділу пошуку _____	46
3.3. Розробка серверної частини проєкту _____	48

						ІАЛЦ.045490.004 ПЗ										
Змін.	Арк.	№ докум.	Підпис	Дата	Веборієнтований сервіс для сортування та пошуку місць прийому вторинної сировини Пояснювальна записка						Літ.	Аркуш	Аркушів			
Розробив		Сметана М.О.									1	55				
Перевірив		Сапсай Т.Г.														
Н. контроль		Клятчєнко Я.М.									КПІ ім. Ігоря Сікорського, ФПМ КВ-61					
Затвердив		Романкевич В.О.														

3.3.1. Створення серверу	48
3.3.2. Створення АРІ	48
3.3.3. Використання бази даних	49
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	54
ДОДАТКИ	

Додаток 1. Копії графічних матеріалів

- ІАЛЦ.045492.005 Д1. Загальна структура додатку. Схема структурна.
- ІАЛЦ.045492.006 Д2. Алгоритм сортування сировини. Схема Алгоритму.
- ІАЛЦ.045492.007 Д3. Алгоритм фільтрації місць прийому сировини. Схема Алгоритму.
- ІАЛЦ.045492.008 Д4. Алгоритм роутингу. Схема Алгоритму.

Додаток 2. Лістинг програми

Додаток 3. Презентація

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API	– Application Programming Interface;
BOM	– Browser Object Model;
BSON	– Binary JavaScript Object Notation;
DOM	– Document Object Model;
JSON	– JavaScript Object Notation;
HTML	– Hyper Text Markup Language;
UX	– User Interface;
URL	– Uniform Resource Locator;
UX	– User Expirence;
HTTP	– Hyper Text Transfer Protocol;
XML	– Extensible Markup Language;
БД	– база даних;
ЕОМ	– електронно-обчислювальна машина;
ПЗ	– програмне забезпечення;
СКБД	– система керування базами даних.

ВСТУП

У наш час екологічні проблеми виходять на передові позиції. В Україні можна було б значно зменшити розмір та кількість сміттєзвалищ, якби частина побутових відходів потрапляла на переробні заводи. Сьогодні в країні є підприємства, де, зокрема, переробляють метал, пластик, скло та папір. Проте, щоб вивозити побутові відходи на переробку, потрібно їх попередньо відсортувати. З 1 травня 2019 року вступили в силу нові Правила надання послуг з поводження з побутовими відходами, затверджені постановою Кабінету Міністрів України №318 від 27 березня 2019 року. Згідно до статті 35 Закону України про відходи: "Поводження з побутовими відходами здійснюється відповідно до державних норм, стандартів і правил. Власники або наймачі, користувачі, у тому числі орендарі, джерел утворення побутових відходів укладають договори з виконавцем послуг з вивезення побутових відходів, здійснюють оплату послуг з поводження з побутовими відходами та забезпечують роздільне збирання побутових відходів. Виконавець послуг з вивезення побутових відходів укладає договори про надання послуг з поводження з побутовими відходами із споживачами. Виконавець послуг з вивезення побутових відходів зобов'язаний укласти договори про надання послуг з перероблення та захоронення побутових відходів із суб'єктами господарювання, що надають такі послуги відповідно до правил благоустрою території населеного пункту, розроблених з урахуванням схеми санітарного очищення населеного пункту" [1].

Нові правила вступають в дію з 1 травня 2020. Отже, сортування відходів – тепер обов'язок кожного.

Питання пошуку повної інформації про сортування та місця здачі вторинної сировини стає досить актуальним. Для вирішення даної задачі доцільно скористатись сучасними інформаційними ресурсами. Інтернет є

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		4

найдоступнішим та найпоширенішим джерелом інформації на сьогодні. Завдання дипломного проекту направлено на розробку вебресурсу для максимального полегшення процесу сортування вторинної сировини. Це також допоможе користувачам грамотно поводитися з відходами та виконувати діюче законодавство України.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		5

1. АНАЛІЗ ІСНУЮЧИХ РЕСУРСІВ ТА ОБГРУНТУВАННЯ ТЕМИ БАКАЛАВРСЬКОГО ПРОЄКТУ

1.1. Різновиди вебсайтів

Складно переоцінити ступінь впливу глобальної мережі на повсякденне життя людини. Інтернет використовується в усіх сферах діяльності людини. Люди звикли використовувати Інтернет для вибору товарів, пошуку і бронювання авіаквитків, визначення найкоротшого маршруту до місця призначення, запису на прийом до лікаря. Інтернет-сервіси повідомляють людям про погоду, затори на дорогах, підказують, коли вийти з дому, щоб встигнути в те чи інше місце.

Вебсайт являє собою набір вебсторінок, які є доступні у мережі Інтернет, мають спільну тематику та знаходяться за однією адресою. У залежності від величини, сайт може бути розміщений як на одному, так і на кількох серверах. Під вебсайтом також розуміють деякий вузол у мережі Інтернет, який має свою унікальну IP-адресу. За допомогою системи доменних імен у відповідність до IP-адреси ставиться унікальне ім'я. Як правильно, користувачі доступуються до вебсайту через його унікальне доменне ім'я. Всі вебсайти Інтернету в сукупності входять до складу Всесвітньої павутини.

Офіційної класифікації вебсайтів немає, але є відмінні характеристики. Загалом розрізняють такі типи.

1. Landing Page – сайт, що складається лише з однієї сторінки, створений для вчинення користувачем конкретної дії (вибрати товар чи послугу, підписатися на розсилку інформації, зареєструватися). Сайт виступає у ролі інструменту для:

- "розкрутки" і реклами товарів та послуг;

- розширення бази користувачів.
2. Сайт-візитка - довідник компанії. На такому сайті розміщена тільки офіційна і конкретна інформація – контакти, адреси, засновники чи співвласники, діяльність, перелік товарів або послуг. Забезпечує:
- залучення нової аудиторії;
 - економію коштів.
3. Корпоративний сайт - формує авторитет компанії, сприяє залученню нових клієнтів, розширяє ринок збуту, допомагає швидко і зручно знайти необхідну інформацію щодо цін на товари та послуги. Забезпечує:
- репрезентацію компанії;
 - швидкий доступ до інформації;
 - підвищення прибутковості.
4. Інтернет магазин – вебресурс, де користувачі мають змогу придбати товар з урахуванням доставки. При цьому розміщення даних про товар, процес замовлення товару і складання угоди проходить через мережу Інтернет. Можливості:
- розширює ринок збуту;
 - охоплює велику аудиторію;
 - автоматизує процес торгівлі;
 - автоматизує контроль за прибутковістю.
5. Сайт-портал – такі сайти містять велику кількість інформації. Як правило це статті, новини, додатки, опитування. Існує два види таких сайтів: корпоративні портали та інформаційні портали. Перші використовуються всередині компаній та оперують з даними співробітників і партнерів, інші – всіма користувачами

мережі Інтернет. Головною метою такого сайту є поширення інформації.

6. Онлайн-сервіс - вебсайти для оптимізації роботи і економії часу, застосовуються в усіх сферах: фінансові операції, робота з документами, оброблення інформації та інші. Такі онлайн ресурси можна поділити на такі групи: пошукові, соціальні, фінансові, дорожні. При створенні онлайн-сервісу велика увага приділяється інтерфейсу користувача, зручності і легкості використання та можливості доступу як з комп'ютера так і мобільного телефону. Забезпечує:

- економію часу користувачів;
- зручність в процесі використання.

1.2. Аналіз існуючих рішень

На сьогодні в мережі Інтернет та у засобах масової інформації приділяється велика увага питанням збереження навколишнього середовища і, зокрема, сортуванню вторинної сировини. Загалом є багато невирішених питань щодо процесу сортування твердих відходів. Тому можна стверджувати, що на даний момент Інтернет ресурси, які б забезпечили вирішення нагальних питань, відсутні.

Розглянемо існуючі вебдодатки, що надають можливість знайти необхідні місця прийому вторинної сировини. Кожна фірма, яка надає послуги збору вторинної сировини, на своєму сайті, як правило, вказує дані лише про свої місця прийому. Сьогодні можна виділити чотири ресурси, на яких є можливість переглянути більшість точок прийому різних типів вторинної сировини.

Мапа esomara.gov

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		8

Перший розглянутий ресурс - офіційний мапа Міністерства екології та природних ресурсів України [2].

Переваги:

- пункти прийому по всій Україні;
- можливість фільтрації місць прийому за потрібним типом сировини;
- інформація на мапі про тип вторинної сировини, що підлягає здачі;
- можливість пошуку на сайті;
- можливість додати нові місця на карті.

Недоліки:

- незручний інтерфейс;
- відсутність точних адрес;
- відсутня можливість коментувати та оцінювати пункти прийому.

Мапа epochtimes.com.ua

Другий розглянутий ресурс - мапа надана порталом epochtimes [3].

Переваги:

- можливість фільтрації місць прийому за потрібним типом сировини;
- інформація на мапі про тип вторинної сировини, яка підлягає здачі;
- можливість пошуку місця прийому на сайті за адресою;
- наявна контактна інформація місць прийому;
- можливість додати коментар до точки.

Недоліки:

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		9

- вказані місця прийому тільки в місті Києві;
- відсутня можливість додати коментар за допомогою соціальних мереж чи сервісу gmail;

Мапи google.com та 2gis.ua

Наступні розглянуті ресурси – мапи google.com [4] та 2gis.ua [5]. На відміну від попередніх ресурсів це мапи широкого застосування, які містять всю інформацію про місцевість, а не тільки місця прийому.

Переваги:

- зручний та дружній інтерфейс;
- можливість оцінки та коментування точок;
- інформація на мапі про тип вторинної сировини, яка підлягає здачі;
- наявна контактна інформація місць прийому;
- ідентифікація особистості за допомогою сервісу gmail.

Недоліки:

- відносно мала кількість точок;
- відсутня можливість фільтрації місць прийому за потрібним типом сировини;
- відсутня інформація на мапі про тип вторинної сировини, який приймають у даному пункті.

1.3. Обґрунтування теми бакалаврського проєкту

У час коли переробка побутових відходів стає невід’ємною частиною нашого життя інформація про сортування твердих відходів є досить актуальною. Із аналізу існуючих рішення можна стверджувати, що

зараз процес пошуку інформації з питань сортування твердих побутових відходів та місць прийому вторинної сировини займає багато часу та є досить незручним.

Завданням даного бакалаврського проєкту є розроблення зручного сервісу для спрощення процедури сортування, адже зараз для задачі навіть повсякденних відходів потрібно відвідати цілу низку Інтернет сторінок та ознайомитися із достатньо великою кількістю інформації. Часто, це і є головною перепоною для пересічного громадянина на шляху до задачі вторсировини та турботи про довкілля.

Головною особливістю розробленого вебресурсу є можливість вибору через зручний інтерфейс бажаних для сортування видів відходів та за допомогою реалізованого алгоритму сортування отримати повідомлення із повною інформацією про сортування. А саме, інформацію про те як потрібно групувати відходи, що з чим можна здавати разом, а що взагалі не підлягає здачі та найближчі місця прийому саме для вибраної сировини. Така можливість не забезпечена в жодному з вище розглянутих ресурсів.

Таким чином, користувач більше не повинен досліджувати різні ресурси в пошуках інформації про сортування кожної сировини окремо. Все що потрібно – це додати бажані відходи в кошик та натиснути кнопку ‘сортувати’.

Загалом вебресурс передбачає наступні функції:

- швидкий та зручний пошук інформації про сортування;
- алгоритм сортування вибраних відходів;
- відображення місць прийому;
- пошук місць прийому вторинної сировини;

- можливість фільтрації місць прийому за потрібним типом сировини;
- наявна контактна інформація місць прийому.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		12

2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЄКТУ

2.1. Вибір програмних засобів для клієнтської частини

2.1.1 Мова програмування JavaScript

JavaScript – це динамічна мова програмування, за допомогою якої розробник може зробити вебсторінку інтерактивною. Інтерактивною є сторінка, із якою користувач може взаємодіяти. Так як JavaScript динамічна мова, це дозволяє визначати тип даних та виконувати синтаксичний аналіз і компіляцію під час виконання програми. Кожен веббраузер містить вбудований інтерпретатор, який безпосередньо виконує скрипт. Тобто, програма розроблена на мові JavaScript імпортується в HTML-документ і виконується на клієнтській стороні. Компіляція і виконання імпортованої програми відбувається без перезавантаження вебсторінки. Можна виділити такі архітектурні характеристики JavaScript:

- автоматизоване управління пам'яттю;
- функції як об'єкти першого класу;
- прототипне програмування;
- динамічна типізація.

У JavaScript реалізовано багато можливостей, які характерні для об'єктно-орієнтованої мови програмування. Власне наслідування призводить до того, що JavaScript працює з об'єктами не як звичайні об'єктно-орієнтовані мови. Завдяки таким реалізованим особливостям як замикання, функції як об'єкти першого класу, анонімні функції мова має додаткову гнучкість.

JavaScript можна розділити на три частини (рис. 2.1). Кожна із структурних одиниць виконує свою задачу і відрізняється від інших:

- ядро (ECMAScript);
- об'єктна модель браузера (Browser Object Model або BOM);
- об'єктна модель документа (Document Object Model або DOM).

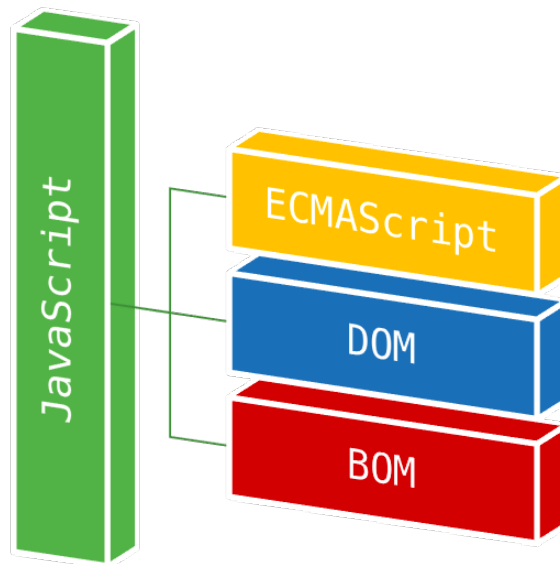


Рисунок 2.1 – Структура JavaScript

В основі JavaScript лежить ядро ECMAScript, де описані всі дані, необхідні для компіляції та виконання програми. А саме:

- ключові і зарезервовані слова;
- типи даних;
- оператори;
- регулярні вирази;
- інструкції.

ECMAScript саме виконує роль основи, яку легко можна розширити і отримати нову скриптову мову.

Об'єктна модель браузера – одна із частин мови, що реалізована в браузері. Фактично вона є посередником між ядром і об'єктною моделлю

документа. Головною задачею об'єктної моделі браузера є управління вікнами і сторінками браузера та допомога при їх взаємодії. Всі вікна та сторінки браузера описуються об'єктом ВОМ. Крім зазначених вище функції об'єктна модель браузера реалізує такі можливості:

- затримки при виконанні коду;
- системні діалоги;
- управління інформацією щодо браузера;
- управління інформацією щодо параметри монітору;
- підтримка роботи з cookie.

Об'єктна модель документа - це незалежний від платформи інтерфейс, який забезпечує взаємодію розробників із HTML і XML-документів. Фактично DOM документ являє собою дерево об'єктів, які мають ряд властивостей. Реалізований функціонал в таких об'єктах дозволяє:

- отримання вузлів;
- зміна вузлів;
- зміна зв'язків між вузлами;
- видалення вузлів [6].

Фактично JavaScript є безальтернативним та найпопулярнішим інструментом розробки клієнтської частини проєкту. За допомогою вище наведених функцій JavaScript дозволяє розробляти 'живі', динамічні вебсторінки.

2.1.2 Бібліотека React

React — це декларативна, ефективна і гнучка JavaScript бібліотека, яка використовується для розробки вебдодатків. Вона дозволяє реалізовувати складні інтерфейси з менших окремих програмних

інтерфейсів – ‘компонентів’. Компонентами називають елементи React, які створюються розробниками, носять довільні назви, мають свою специфічну структуру та виконують ряд функцій. Вони створюються за допомогою поглибленого синтаксису JavaScript - JSX. Компоненти мають життєві цикли такі як: створення, час життя, видалення. Бібліотека надає можливість визначати різні моменти життя компонентів та взаємодіяти з ними. Розрізняють різні методи роботи з компонентами при монтуванні їх в дерево, при оновленні та при розмонтуванні. При монтуванні у наведеному порядку будуть використані наступні методи (рис. 2.2):

- constructor();
- static getDerivedStateFromProps();
- render();
- componentDidMount().

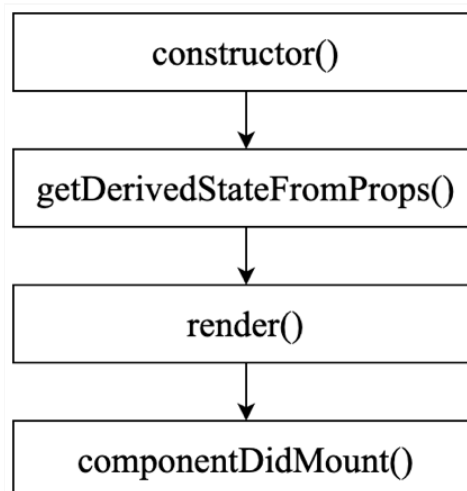


Рисунок 2.2 – Порядок викликів методів при монтуванні

При оновленні в порядку наведеному на рис. 2.3 будуть викликані методи:

- static getDerivedStateFromProps();
- shouldComponentUpdate()

- render();
- getSnapshotBeforeUpdate();
- componentDidUpdate().

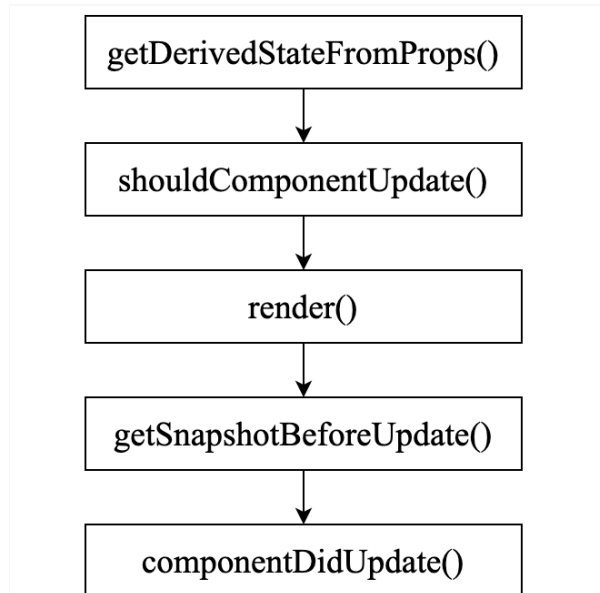


Рисунок 2.3 – Порядок викликів методів при оновленні

При розмонтуванні використовується метод:

- componentWillUnmount().

Найголовнішою перевагою, яка забезпечує високу швидкість є те, що React створює із компонентів аналог реального DOM-дерева – VirtualDOM і представляє його в браузері. Бібліотека дозволяє відстежити за змінами у віртуальному дереві та якщо такі зміни відбулися, оновлює реальне DOM-дерево. У JavaScript доступ до елементів дерева є ‘дорогим’, тобто займає багато часу. Таким чином, за рахунок маніпуляцій із віртуальним деревом забезпечується підвищення швидкодії [7].

Під час розробки складних вебдодатків із отриманням даних від серверу та великою кількістю динамічних елементів швидкість

вебсторінки зменшується, код проєкту нагромаджується і подальша розробка стає складнішою. У таких випадках використання бібліотеки React стає не просто доцільним, а необхідним. Завдяки своїм перевагам React дозволяє прискорити не тільки швидкодію проєкту, а і сам процес розробки.

2.1.3 Бібліотека Redux

Redux - це JavaScript бібліотека, яка використовується для управління єдиним станом додатку. Як правило її застосовують разом із бібліотекою React. Основним принципом Redux є збереження стану всього додатку в одному місці. Стан додатку являє собою дерево об'єктів, в яких описані особливості та характеристики компонентів.

Для рівня представлення стан додатку є доступним лише для читання. Єдиним способом змінити стан додатку є створення 'дії' (action) - об'єкту, що описує потрібні операції. Це гарантує, що ні перегляди, ні зворотні виклики мережі ніколи не змінять стан додатку. У свою чергу 'дії' лише виражають намір зробити зміни. Безпосередньо зміни відбуваються за допомогою чистих функції 'редукторів' (reducer). Чисті функції – це функції, які при однакових вхідних значення завжди повертають однакові результати. Редуктори приймають стан додатку та дію, яку потрібно виконати і повертають змінений стан додатку. В процесі розробки редуктори можуть бути розділені на дрібніші, які управляють певними частинами дерева станів. Оскільки редуктори - це лише функції, можна контролювати порядок їх надсилання, передавати додаткові дані або навіть створювати повторювані редьюсери для звичайних завдань, таких як розбиття на сторінки.

Після отримання нового стану додатку відбувається перевірка, які дані були зміннені та які компоненти, відповідно, потрібно замінити. Як результат в DOM змінюються лише ті компоненти, дані яких були зміннені в процесі 'дії' (рис. 2.4).



Рисунок 2.4 – Процес зміни стану для компонента

Таким чином, Redux дозволяє забезпечити синхронізацію всього додатку. Адже зберігаючи всю інформацію в одному місці, додаток завжди відображає актуальну інформацію, незалежно від того в якій частині додатку користувач знаходиться.

2.2 Вибір програмних засобів для серверної частини

2.2.1 Програмна платформа Node.js

Node.js призначена для виконання високопродуктивних мережевих застосунків, написаних мовою програмування JavaScript. Платформа окрім роботи із серверними скриптами для вебзапитів, також використовується для створення клієнтських та серверних програм [8].

Основне призначення Node.js – створення масштабованих розподілених мережових додатків, тобто вебсерверів.

Node.js включає в себе установник пакетів npm, за допомогою якого легко встановлювати пакети, використовуючи команду ‘npm install’. Можна використати командою ‘npm search’, щоб дізнатись, які пакунки доступні для установки і їх короткий опис. Всі команди виконуються в консолі комп'ютера. За допомогою Node.js реалізуються наступні завдання:

- HTTP-сервер;
- маршрутизація, для роботи із запитамі;
- оброблювач вхідних даних, для вхідних POST-запитів.
- логіка відображення, для відправки контенту.

Для більш зручної роботи з Node.js існує фреймворк Express.js, який широко використовується спільно з Node.js. Express.js надає великий набір функцій для мобільних і вебдодатків. За допомогою службових методів HTTP даного фреймворка і проміжних оброблювачів можна швидко і легко створити надійний API.

Node.js це не окрема мова програмування, а платформа для використання JavaScript на серверному боці проекту. Це дозволяє використовувати одну мову програмування для розробки як клієнтської, так і серверної частин. Можливість використання однієї мови програмування, а не декількох значно прискорює процес розробки, особливо в таких проектах, як даний, де більшість функціоналу реалізовано на клієнтській стороні.

2.2.2 База даних MongoDB

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		20

MongoDB – документо-орієнтована система керування базами даних (СКБД), для якою схеми таблиць не є обов'язковими.

Головною особливістю MongoDB є формат зберігання даних. Дана СКБД зберігає документи у форматі JSON. Гнучка мова MongoDB забезпечує простоту у формуванні запитів. Створення індексів для різних збережених атрибутів забезпечує зберігання великих бінарних об'єктів, підтримує реплікацію і побудову відмовостійких конфігурацій. За допомогою вбудованих засобів забезпечується шардінг (зберігання даних на різних серверах в залежності від ключа), комбінуючи який з реплікацією даних можна отримати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (працездатність БД не залежить від коректної роботи окремого вузла), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Перетворення серверу на кластр чи розширення кластера не потребує зупинки БД. MongoDB керує даними у вигляді JSON-подібних документів, що зберігаються в бінарному форматі BSON. Подібно до інших документо-орієнтованих СКБД MongoDB не є реляційною СКБД [9].

MongoDB забезпечує підхід MapReduce – це підхід паралельної обробки великих об'ємів 'сирих' даних (наприклад результат сканування сайтів). Структурно даний підхід зображений на рисунку 2.5.

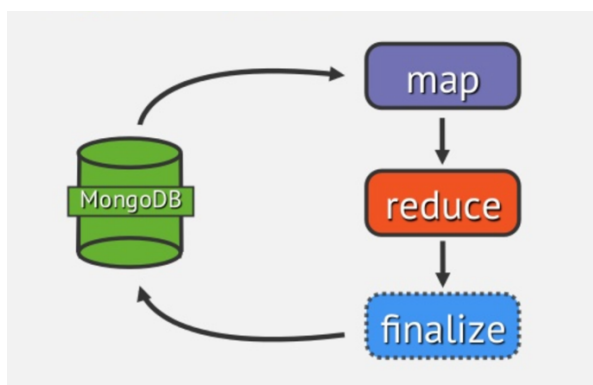


Рисунок 2.5 – Підхід MapReduce

MongoDB складається з баз даних, які зберігають в собі колекції документів. Документи в свою чергу складаються з полів. Поля є парами ключ-значення (рис. 2.6). Колекції також можуть бути індексовані, що покращує швидкість вибірки та сортування.

Основні можливості MongoDB:

- документо-орієнтоване сховище;
- гнучкість у формуванні запитів;
- динамічні запити;
- підтримка індексів;
- профілювання запитів;
- ефективне зберігання бінарних даних великих обсягів, таких як фото та відео;
- підтримка відмовостійкості і масштабованості.

Змін.	Арк.	№ докум.	Підпис	Дата

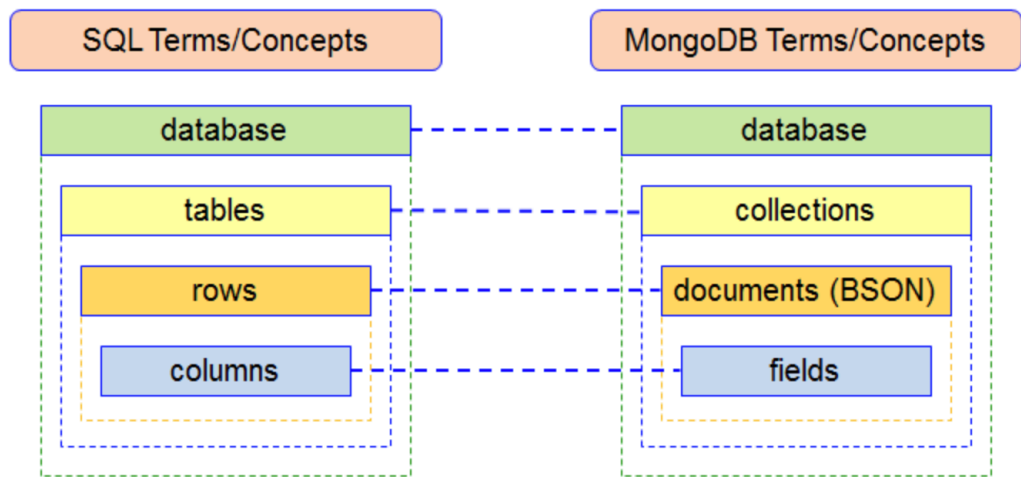


Рисунок 2.6 – Структури SQL та MongoDB БД

MongoDB забезпечує збереження файлів форматі JSON, зручне управління та доступ до даних. Великою перевагою даної СУБД є простота у використанні, особливо в порівнянні із іншими реляційними базами даних. Для проєктів, де БД зберігають невелику кількість даних та, як правило, оперують переважно із даними користувача - MongoDB є найкращим вибором.

Змін.	Арк.	№ докум.	Підпис	Дата

3. РОЗРОБКА ПРОЄКТУ

3.1. Клієнт-серверна архітектура

В онлайн-індустрії кожен день відбувається зростання впливу моделі клієнт-сервер. Клієнт-серверна архітектура – це архітектура, в якій відповідальність, права та навантаження розділені. Така архітектура представляє собою централізовану систему, де сервер зберігає всі ресурси, а клієнт доступається до них. У клієнт-серверній архітектурі виділяються такі типи компонентів:

- сервери – надають інформацію клієнтам
- клієнти – використовують інформацію надану сервером;
- мережа – забезпечує взаємодію клієнта і сервера.

Сервер – це одно- або багатопроцесорна персональна чи віртуальна ЕОМ, що включає в себе: розподілену пам'ять, оброблення даних, комунікаційні засоби та засоби управління периферійним обладнанням. Зазвичай як сервер використовують потужні ЕОМ, що мають великий дисковий простір і швидкодіючі процесори.

Клієнт – робоча станція, з якою взаємодіє користувач. Така станція здатна виконувати необхідні обчислення і забезпечує зв'язок із сервером.

Використовуючи клієнт-серверну модель структурно розрізняють три рівні операцій (рис. 3.1), де :

- рівень представлення даних - забезпечує взаємодію з користувачем;
- прикладний рівень - реалізує основну логіку додатку;
- рівень управління даними – забезпечує доступ до даних та їх зберігання.

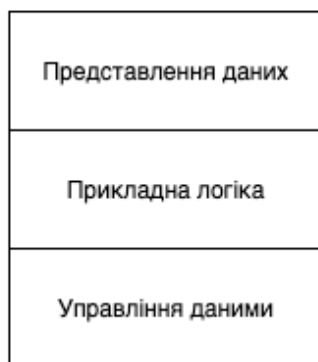


Рисунок 3.1 – Рівні клієнт-серверної моделі

Дворівнева архітектура

У кожній системі, що побудована на новітніх мережових технологіях, спостерігаються елементи клієнт-серверної взаємодії переважно на базі дворівневої архітектури. Архітектура отримала назву дворівнева, адже ми розділяємо компоненти на дві ланки (рис. 3.2) – клієнтську та серверну.

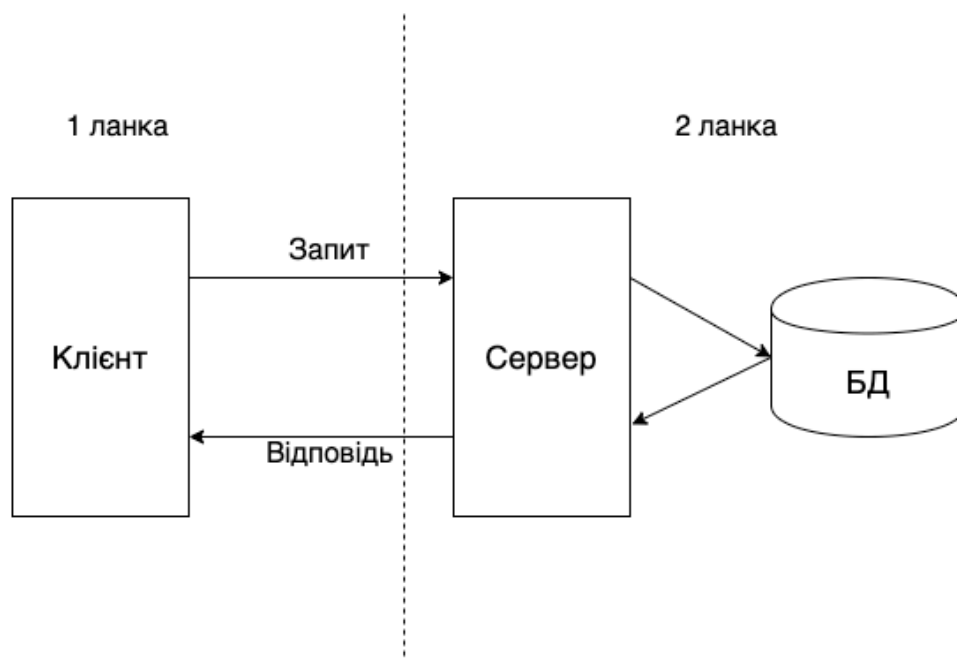


Рисунок 3.2 – Дворівнева клієнт-серверна архітектура

Змін.	Арк.	№ докум.	Підпис	Дата

Операції, які виконують клієнт і сервер, можуть розділятися по різному. Так, в залежності від того які з вище наведених функцій виконує клієнт, розрізняють дві моделі:

- модель ‘тонкого клієнта’, де вся прикладна логіка застосунку та керування ресурсами зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення даних;
- модель ‘товстого клієнта’, де сервер лише керує даними, а обробка інформації та інтерфейс користувача реалізовані на боці клієнта.

Таким чином, клієнт-серверна система є дворівневою, якщо сервер відповідає на запити клієнта напряму і в повному обсязі, не використовуючи сторонні ресурси.

Трирівнева архітектура

У наш час розподілені обчислення стають більш популярними. Вони реалізуються на базі моделі сервера додатків, в якому всі рівні операцій розділені на 2 і більше ланок. При чому кожна ланка може функціонувати на окремій ЕОМ. Ланки обмінюються повідомленнями у попередньо погодженому форматі. Так, клієнт-серверна архітектура стає трирівнева. Як правило, третім рівнем в такій архітектурі стає сервер додатків, тобто компоненти розподіляються наступним чином (рис. 3.3):

- представлення даних відбувається на боці клієнта;
- прикладний компонент реалізований на окремому сервері;
- управління даними відбувається на сервері БД.

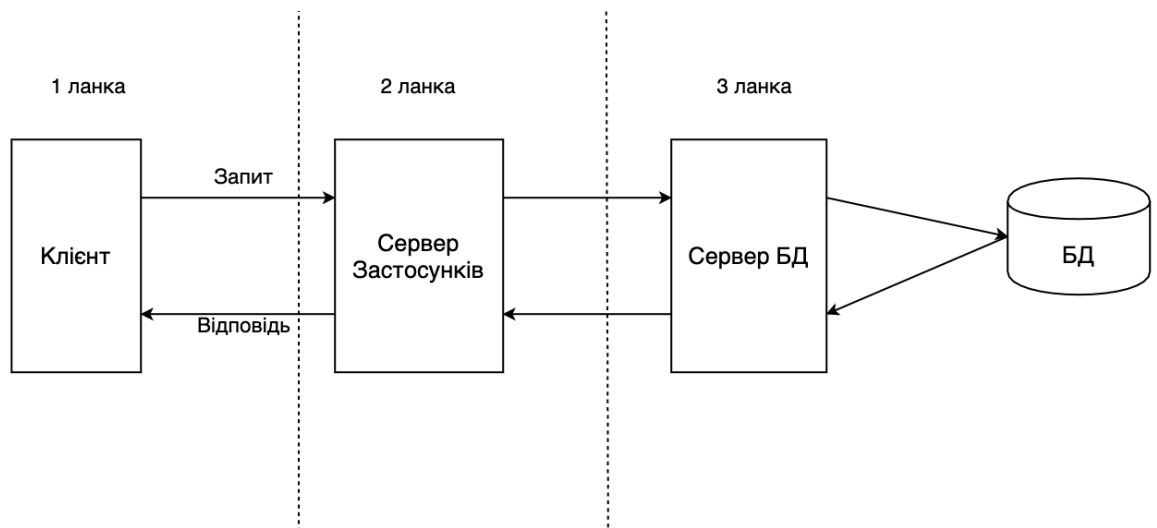


Рисунок 3.3 – Трирівнева клієнт-серверна архітектура

Багаторівнева архітектура

Якщо до трирівневої моделі додати додаткові сервери, кожен з яких реалізовуватиме власні сервіси та буде взаємодіяти з серверами інших рівнів, то дана модель буде багаторівневою (рис. 3.4).

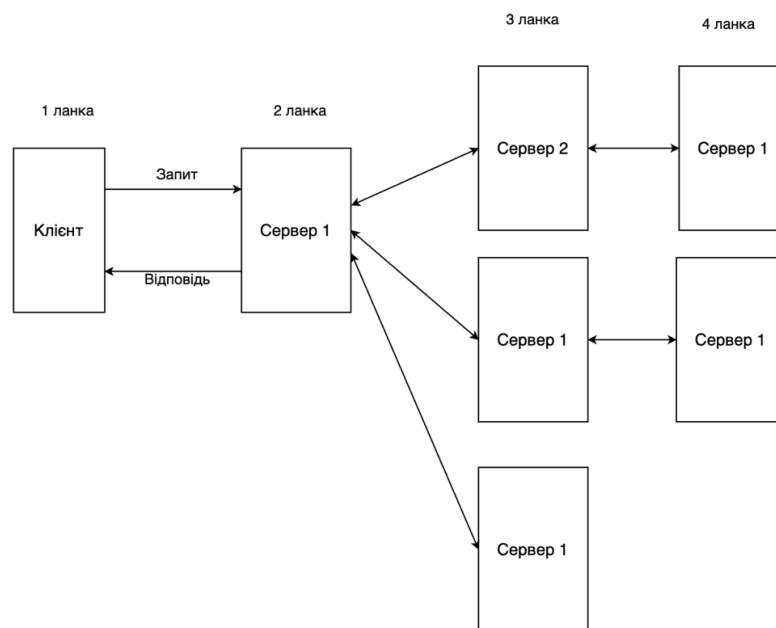


Рисунок 3.4 – Багаторівнева клієнт-серверна архітектура

Змін.	Арк.	№ докум.	Підпис	Дата

Порівняння та обґрунтування архітектури для даної роботи

Дворівнева архітектура є найпростішою, адже лише один сервер обслуговує всі запити. Вона добре підходить для невеликих проєктів, де не вимагається висока потужність.

Трирівнева архітектура більш складна, але розподілення функцій між серверами другого і третього рівня забезпечує:

- гнучкість та масштабованості;
- високу безпеку;
- високу потужність (завдання розподілені між серверами) [6,7].

Таким чином, головна особливість клієнт-серверної архітектури полягає в розділенні додатку на декілька компонентів, кожен з яких має свої права, відповідальність та реалізує деякі функції. Такі компоненти можуть виконуватися як на різних ЕОМ так і на одній.

У даному дипломному проєкті використовується дворівнева клієнт-серверна архітектура із моделлю 'товстою клієнта'. Такий вибір обумовлений тим, що серверна частина проєкту маніпулює із відносно невеликою кількістю даних та найбільший акцент зроблений на рівні представлення даних та прикладної логіки.

3.2. Розробка клієнтської частини проєкту

Інтерфейс вебдодатку являє собою візуальну частину додатку, з якою взаємодіє користувач. Інтерфейс повинен бути простим, зручним та 'дружелюбним' у використанні. Загалом візуальна прийнятність, зручність і 'дружелюбність' визначаються UI та UX характеристиками.

UI – це інтерфейс користувача, посередник між додатком та користувачем. Включає в себе як зовнішній вигляд так і інтерактивність продукту.

UX – це процес визначення проблем користувача і вирішення цих проблем. Іншими словами покращення зручності, простоти, використання додатку. Часто ці характеристики грають вирішальну роль при виборі того чи іншого додатку користувачем.

Дизайн усіх сторінок вебдодатку базується на завчасно визначеному прототипу (рис. 3.5).



Рисунок 3.5 – Прототип сторінки

3.2.1 Опис розділів вебдодатку

Додаток включає в себе наступні розділи:

- сортування;
- мапа;

Змін.	Арк.	№ докум.	Підпис	Дата

- пошук.

Кожен з розділів має свої сторінки на яких реалізований відповідний функціонал. Так процес сортування в себе включає:

- головну сторінку;
- сторінку з всіма відходами вибраного типу;
- сторінку з відсортованою вибраною вторинною сировиною.

У розділі пошуку і мапи входять по 1 відповідній вебсторінці. Загальна структура вебдодатку зображена у додатку 1.

3.2.2 Роутинг вебдодатку

Роутингом називають доступ до різних розділів додатку. Роутинг визначає набір маршрутів. Коли в додаток приходить запит або ж просто URL сторінки змінюється, система перевіряє чи знаходиться запитуваний маршрут у заданому наборі. Якщо ні – зазвичай користувач перенаправляється на сторінку, що відображає відповідне повідомлення про невдалий запит (Page not found). Якщо запитуваний маршрут є у заданому наборі, то запит обробиться компонентом, що відповідає за необхідний маршрут.

Для вибору маршруту визначений об'єкт Switch. Він вибирає перший маршрут, який задовольняє запит і використовує його для обробки запиту. Кожен маршрут представляє собою об'єкт Route. Даний об'єкт має три основних атрибути:

- path - шаблон адреси, з яким буде порівнюватися запитуваний адрес URL;
- component - компонент, що відповідає за обробку запиту по даному маршруту;

- `exact` – допускає лише точні значення маршруту із адресою запити (а не просте входження адреси в маршрут).

У даному проєкті задані наступні маршрути (рис. 3.6):

- `'/'` – корінний маршрут, при переході на нього користувача буде перенаправлений на маршрут `'/sorting'`;
- `'/sorting'` – маршрут розділу сортування, обробляється компонентом `MainTable`;
- `'/map'` - маршрут мапи, обробляється компонентом `MapContainer`;
- `'/search'` – маршрут розділу пошуку, обробляється компонентом `SearchTable`;
- `'/sorting/result'` – маршрут сторінки відсортованих відходів, обробляється компонентом `SortedList`;
- `'/sorting/:garbageType'` – маршрут сторінки вибраного типу вторсировини `garbageType`, обробляється компонентом `GarbageDetailsTable`;
- якщо вхідний запит не задовольняє жодного з вище наведених маршрутів, то він буде оброблений компонентою `NotFound`.

```

<Switch>
  <Route exact path="/"><Redirect to="/sorting" /></Route>
  <Route exact path="/sorting" component={MainTable} />
  <Route path="/map" component={MapContainer} />
  <Route path="/search" component={SearchTable} />
  <Route exact path="/sorting/result" component={SortedList} />
  <Route path="/sorting/:garbageType" component={GarbageDetailsTable} />
  <Route component={NotFound} />
</Switch>

```

Рисунок 3.6 – Роутинг проєкту

Повний алгоритм роутингу наведений у додатку 4.

Використання роутингу забезпечує можливість передавання посилання на конкретну сторінку, а не просто на вебресурс. Адже відображення тих чи інших компонентів залежить від URL. URL може бути змінена як деяким компонентом, так і користувачем у браузері. Так для переходу на іншу сторінку компонент змінює URL, потім роутер порівнює маршрут із заданими і передає керування на відповідний компонент (рис. 3.7). Як результат користувач спостерігає зміну сторінки.

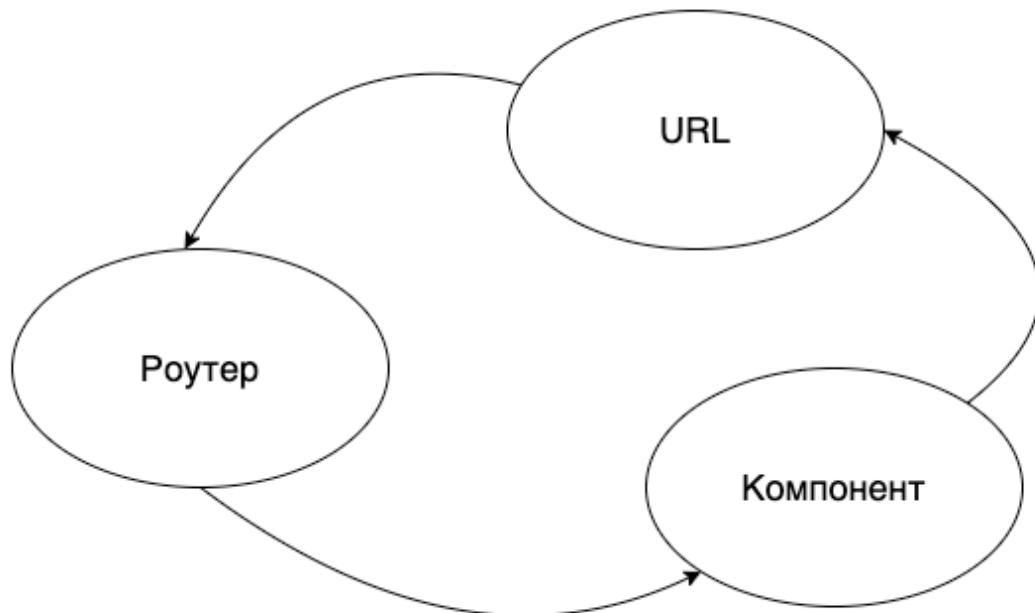


Рисунок 3.7 – Процес переходу між сторінками

3.2.3 Опис розділу сортування

Початкова сторінка даного розділу знаходиться за маршрутом ‘/sorting’ і відображає такі типи вторинної сировини (рис. 3.8):

- одноразовий посуд;

Змін.	Арк.	№ докум.	Підпис	Дата

- пластик;
- папір і картон;
- скло;
- метал;
- tetra Pak;
- батарейки;
- інше.

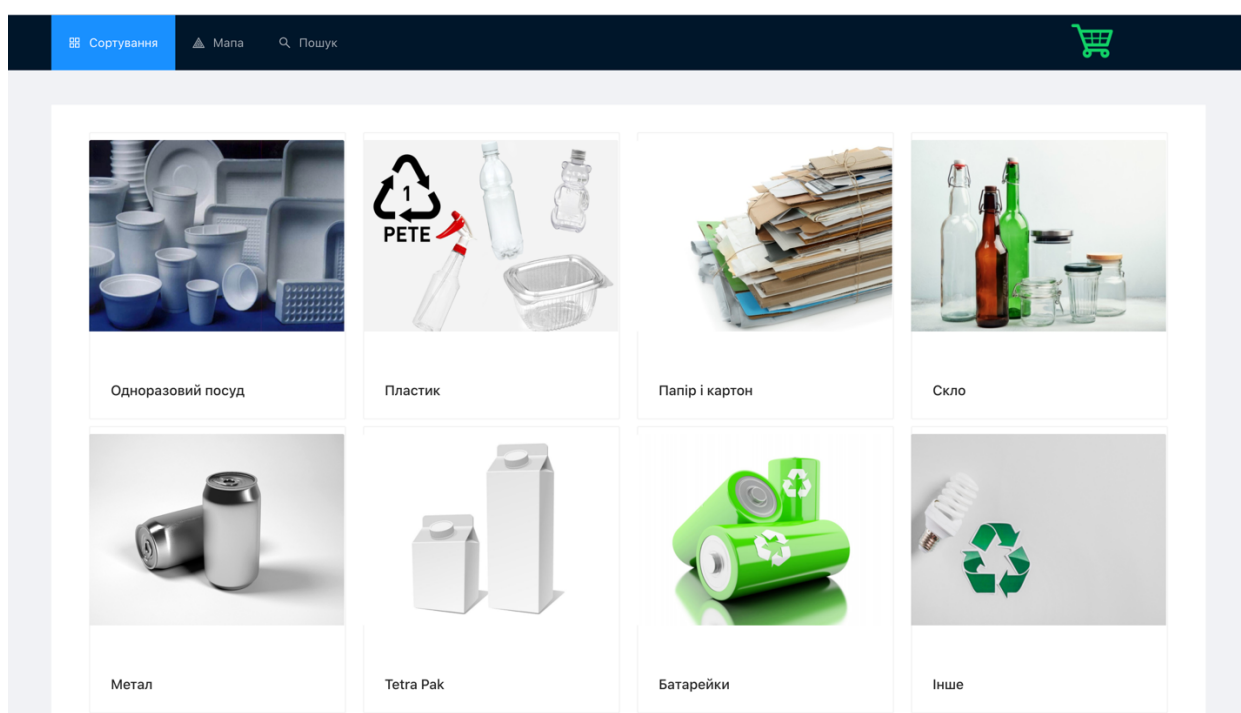


Рисунок 3.8 – Початкова сторінка розділу сортування

Вибравши один із типів вторинної сировини користувач перенаправляється на маршрут '/sorting/:garbageType', де garbageType – тип вибраної сировини. Так, вибравши пластик, користувач перенаправляється на маршрут '/sorting/plastic' та спостерігає сторінку із списками підтипів та безпосередньо самої вторинної сировини вибраного

Змін.	Арк.	№ докум.	Підпис	Дата

типу. Прототип сторінки, що відображає детальну інформацію про обраний тип відходу, наслідується від визначеного прототипу сторінки та має вигляд зображений на рис. 3.9. Сторінка, що безпосередньо відображає інформацію про пластик має вигляд зображений на рис. 3.10.

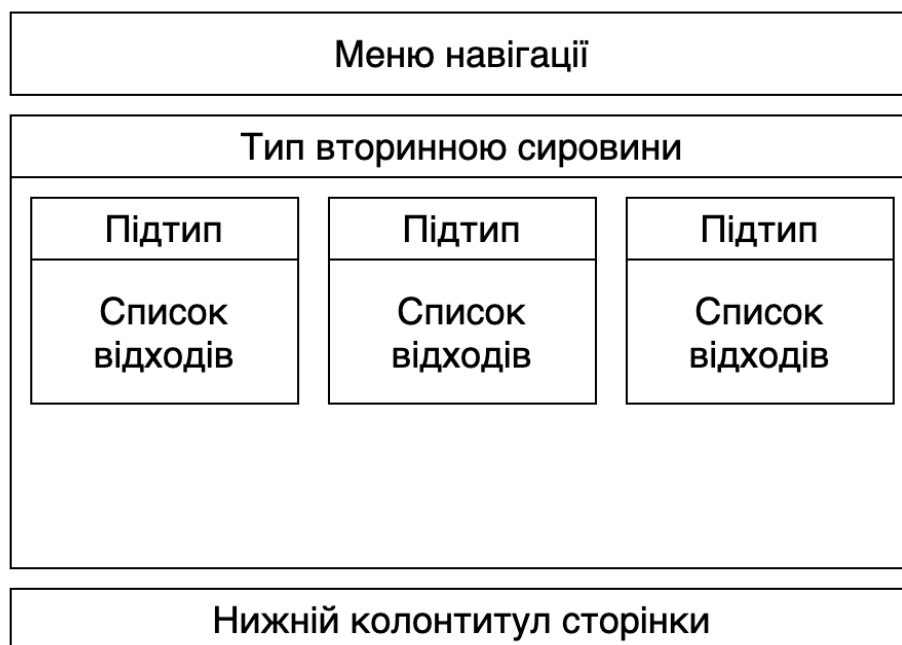


Рисунок 3.9 – Прототип сторінки, що відображає детальну інформацію про обраний тип сировини

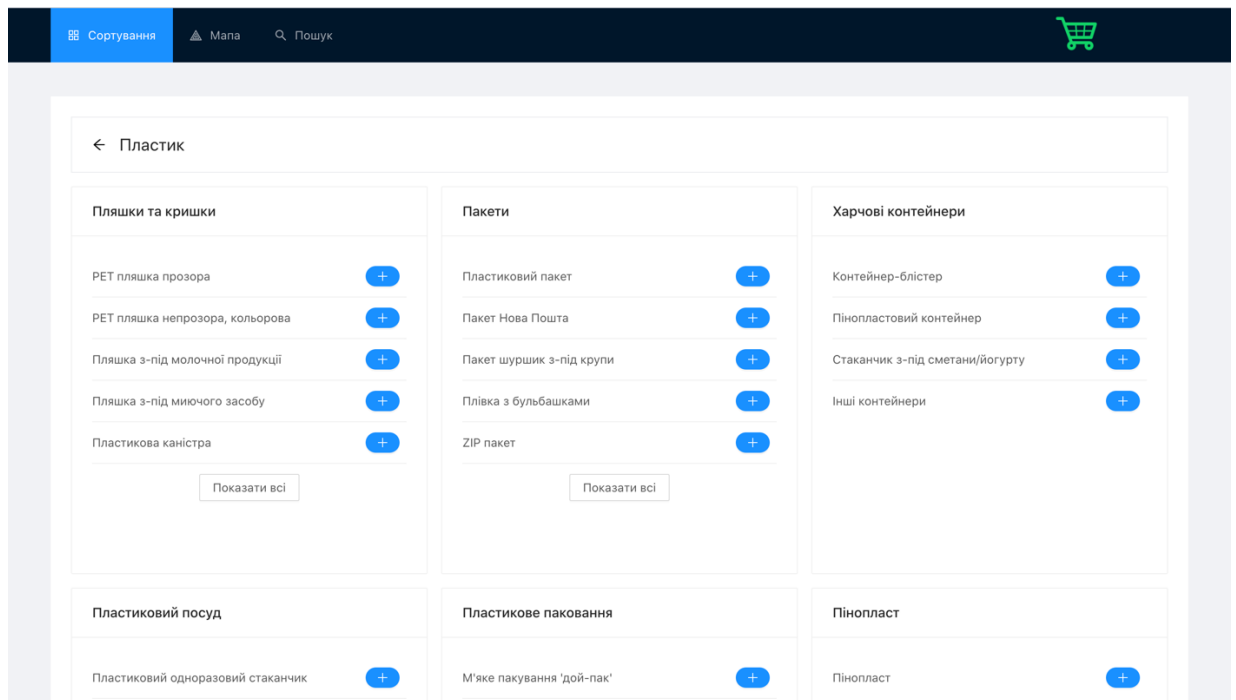


Рисунок 3.10 – Сторінка, що відображає детальну інформацію про пластик

Сторінки, що відображають детальну інформацію про вибраний тип сировини мають цілу низку UX особливостей спрямованих на спрощення та покращення взаємодії користувача із ресурсом. Серед них:

- відображення лише п’яти сутностей вторинної сировини для не нагромадження сторінки, адже користувача може цікавити зовсім інший підтип;
- можливість показати/сховати всі сутності деякого підтипу за допомогою кнопок ‘Показати всі’ / ‘Сховати всі’ (рис. 3.11);
- демонстрування користувачеві, що на сутність можна натиснути для перегляду подальшої інформації, шляхом підсвічення при наведенні та зміни курсору на pointer (рис. 3.12);
- відображення підказки про можливість додавання сутності у кошик при наведенні на відповідну кнопку (рис. 3.13);

- відображення відповідного елемента при додаванні у кошик, який дає зрозуміти, що сутність була успішно додана (рис. 3.14);
- відображення мітки біля кошику в навігаційному меню, яка показує кількість доданих сутностей (рис. 3.15);
- відображення інформації про сортування та можливість додавання/видалення вибраної сировини, при натисканні на неї (рис. 3.16).

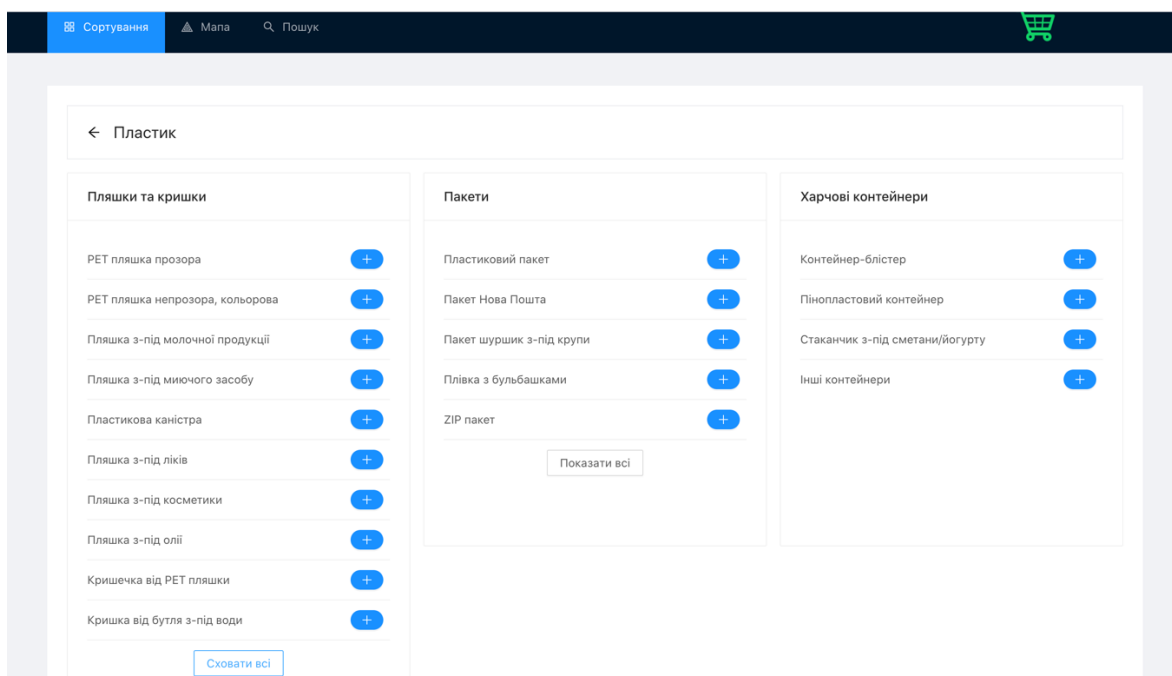


Рисунок 3.11 – Відображення всіх сутностей вибраного підтипу

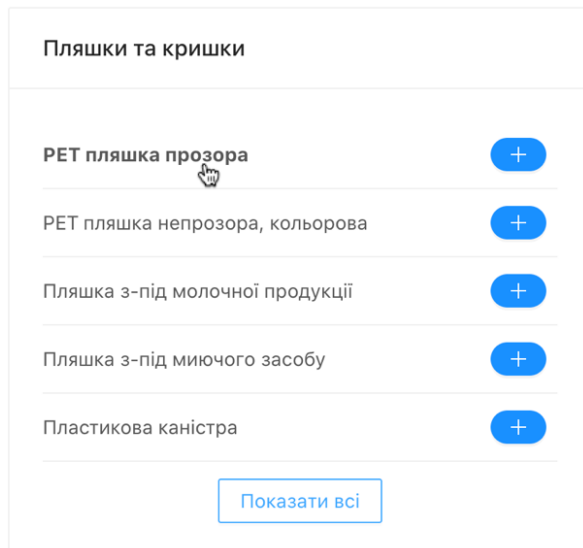


Рисунок 3.12 – Підсвічення при наведенні на сутність

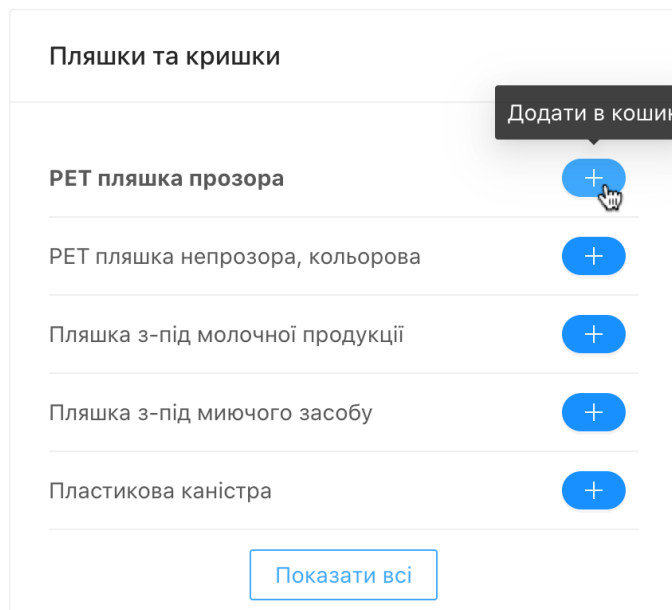


Рисунок 3.13 – Відображення підказки про можливість додавання у кошик

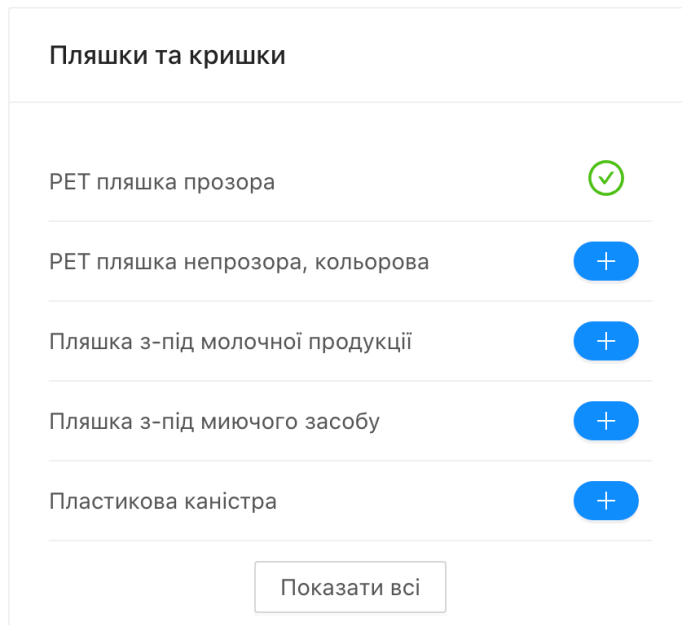


Рисунок 3.14 – Відображення елемента при додаванні сутності до кошику

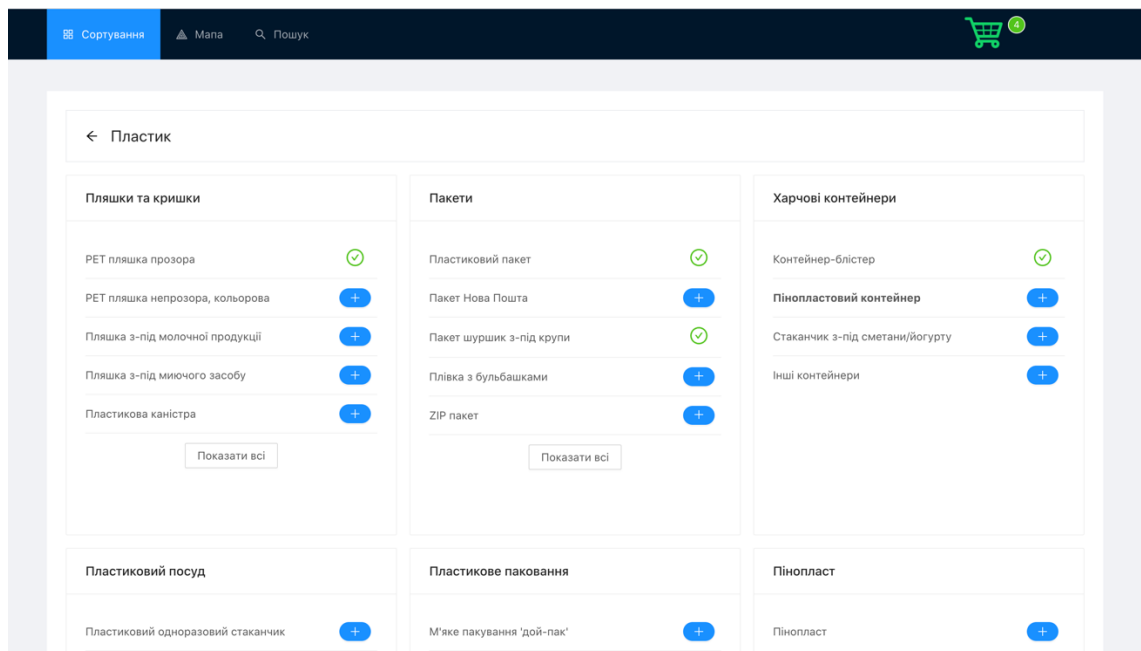


Рисунок 3.15 – Відображення кількості доданих сутностей біля кошику

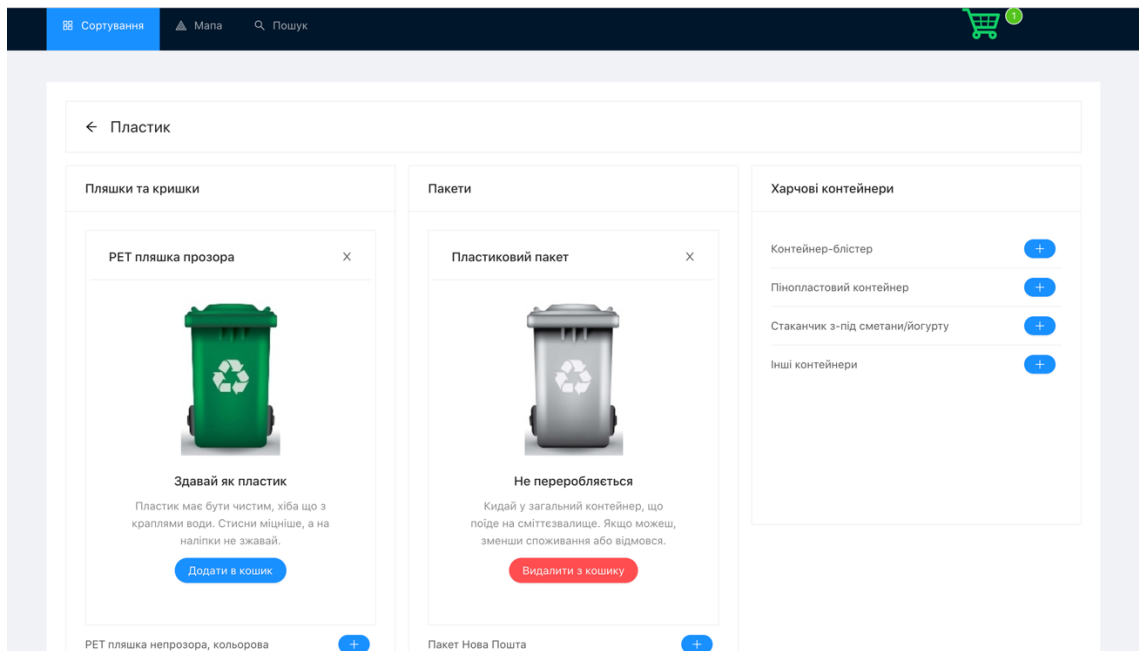


Рисунок 3.16 – Відображення кількості доданих сутностей біля кошику

За допомогою реалізованого функціоналу користувач має змогу як знайти інформацію про вторинну сировину яка його цікавить так і додати в кошик всі необхідні сутності для подальшого їх сортування. При натисканні на зображення кошику в навігаційному меню користувач має змогу відслідкувати всі додані сутності (рис. 3.17).

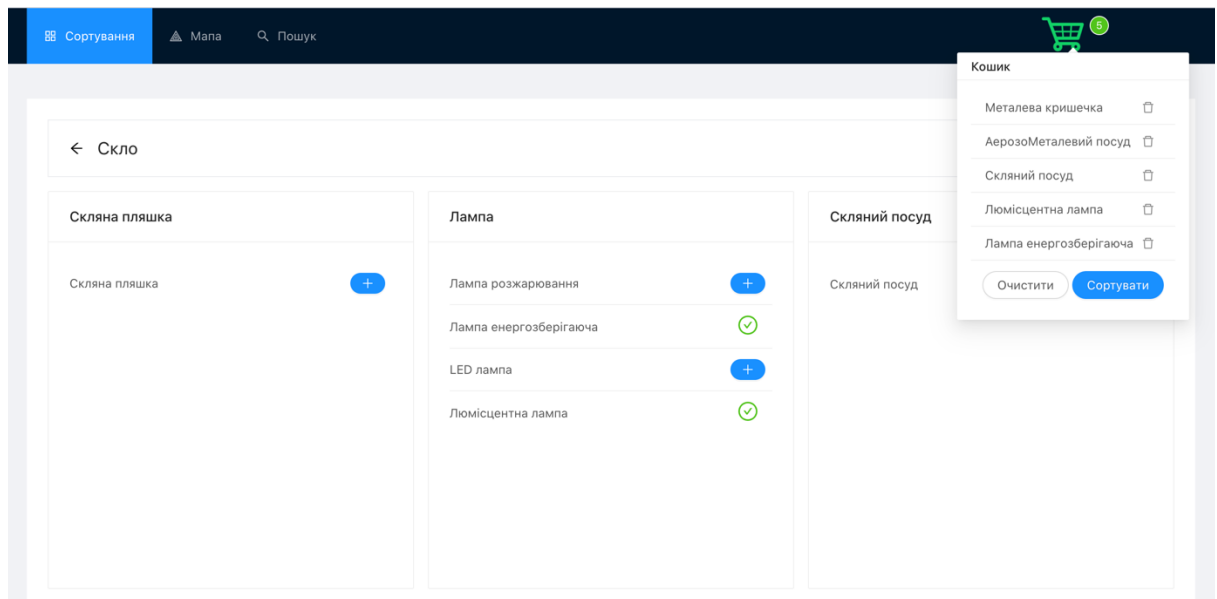
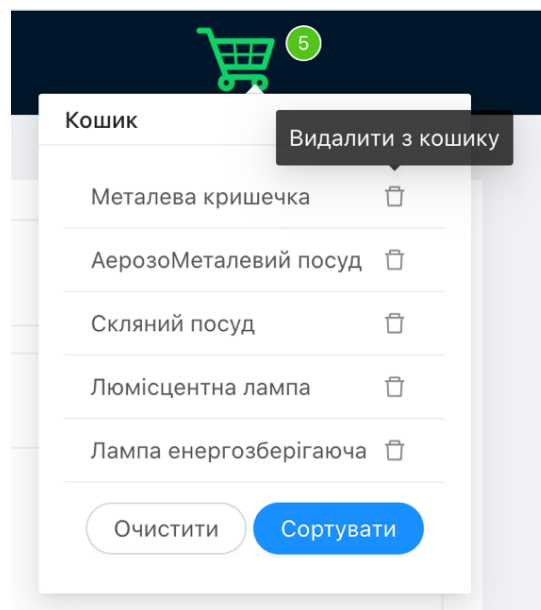


Рисунок 3.17 – Кошик з вибраними сутностями

У кошику користувач може:

- видалити одну із сутностей (рис. 3.18);
- очистити весь кошик;
- відсортувати вибрану сировину.



Змін.	Арк.	№ докум.	Підпис	Дата

Рис. 3.18 – Видалення сутності з кошику

Натиснувши кнопку 'Сортувати' користувач буде перенаправлений на маршрут '/sorting/result'. На відповідній сторінці зображенні погруповані сутності з описом і деякими рекомендаціями по сортуванню. Для кожної відсортованої групи, що підлягає здачі, наявна кнопка 'Показати місця прийому на мапі' (рис. 3.19). Натиснувши на неї, користувач перенаправиться на маршрут 'map?searchType=:type', де :type – тип вибраної сировини. Як результат користувач матиме змогу спостерігати місця прийому вибраної сировини на мапі. Алгоритм сортування вибраної сировини зображений у додатку 2.

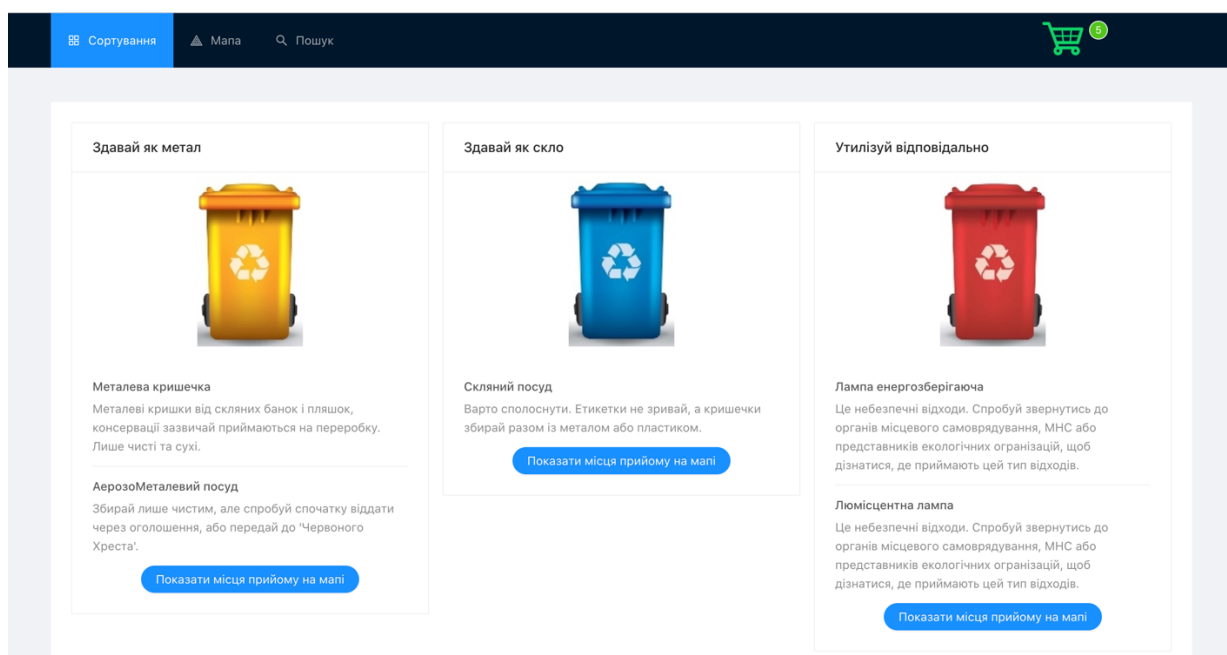


Рисунок 3.19 – Сторінка з відсортованою сировиною

Змін.	Арк.	№ докум.	Підпис	Дата

3.2.4 Опис розділу мапи

У даному розділі користувач має змогу знайти місце прийому вторинної сировини на мапі. Розділ складається з однієї сторінки. Прототип даної сторінки наслідує визначений прототип сторінки та має вигляд зображений на рис. 3.20. Сама сторінка зображена на рис. 3.21.

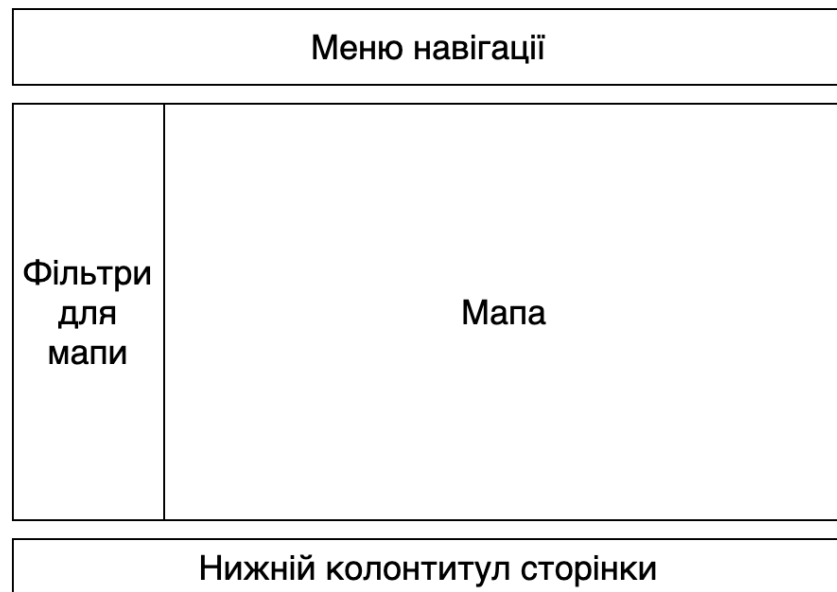


Рисунок 3.20 – Прототип сторінки з мапою

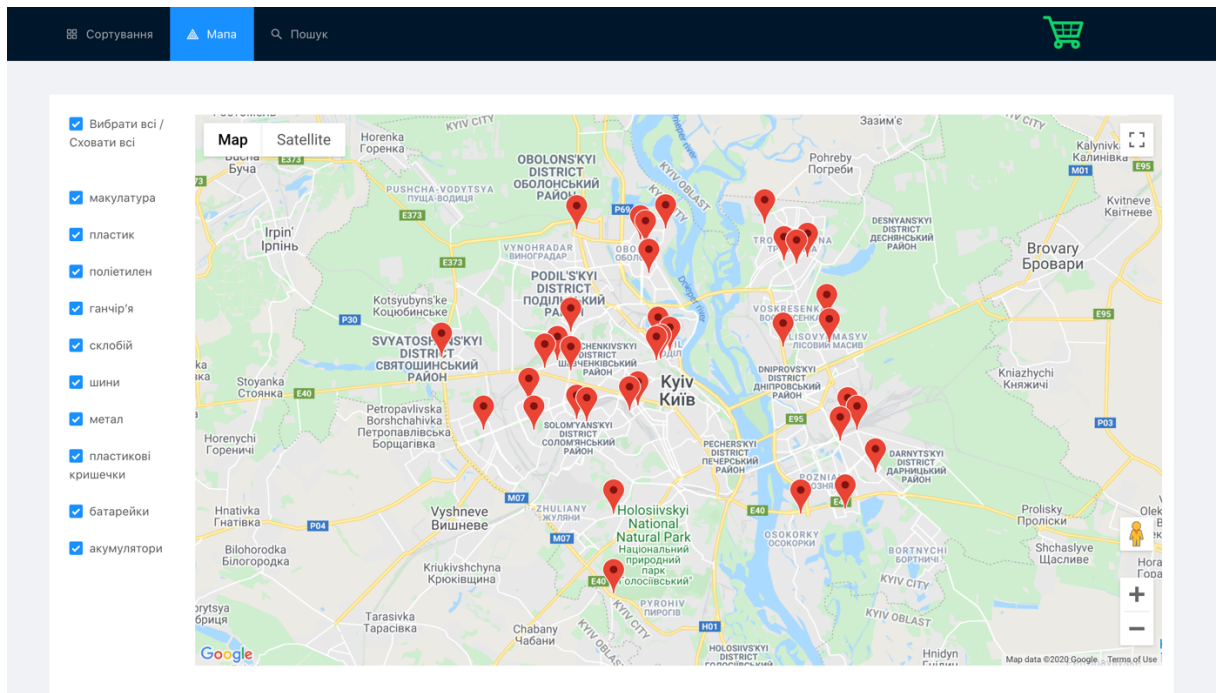


Рисунок 3.21 – Сторінка з мапою

Для зручності пошуку було реалізовано фільтрацію місць прийому за типом сировини, що приймається. Так, забезпечена можливість фільтрації за наступними типами:

- макулатура;
- пластик;
- поліетилен;
- ганчір'я;
- склобій;
- шини;
- метал;
- пластикові кришечки;
- батарейки;
- акумулятори.

Змін.	Арк.	№ докум.	Підпис	Дата

Для зручності користування до фільтрів доданий фільтр ‘Вибрати всі / Сховати всі’, який надає можливість обрати або сховати всі типи вторинної сировини. За замовчуванням усі фільтри активні, тобто мапа відображає всі точки. Алгоритм фільтрації точок зображений у додатку 3.

Після проходження розділу сортування, натиснувши кнопку ‘Показати місця прийому на мапі’ користувач перенаправляється на мапу, яка в свою чергу відображатиме лише ті місця, де приймають вибрану сировину (рис. 3.22).

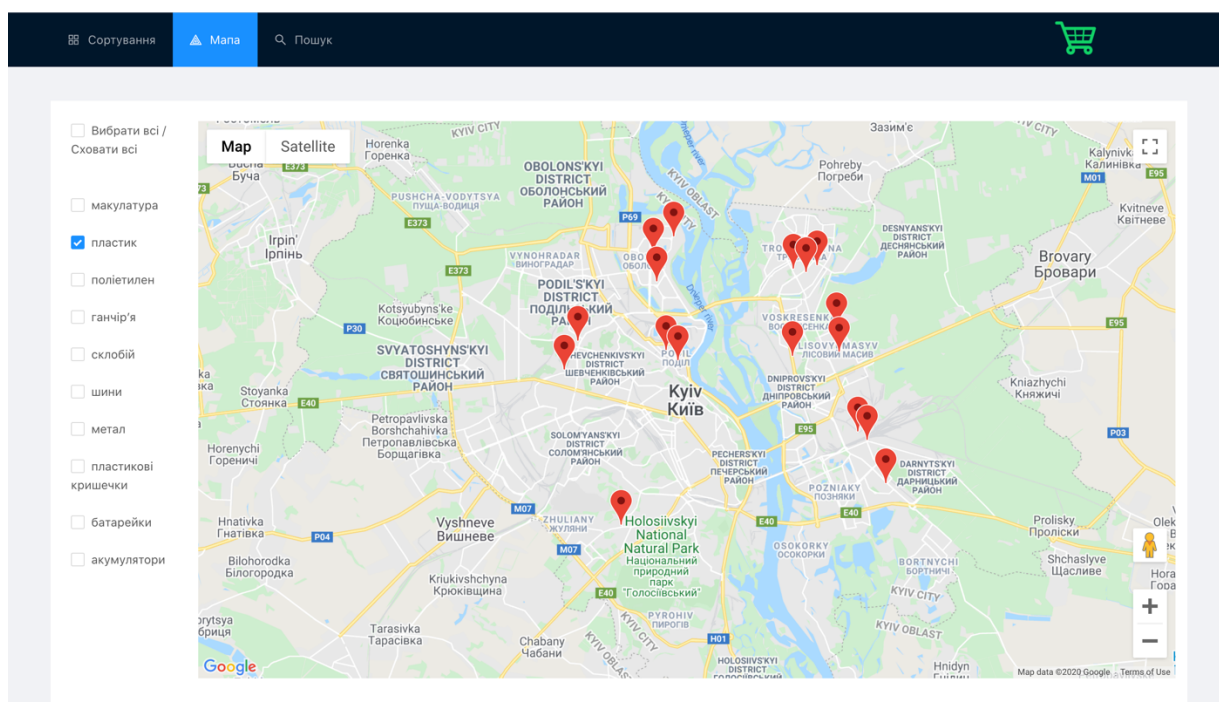


Рисунок 3.22 – Відображення місць прийому пластику

При натисканні на маркер на мапі з’являється вікно з детальною інформацією про місце прийому (рис. 3.23).

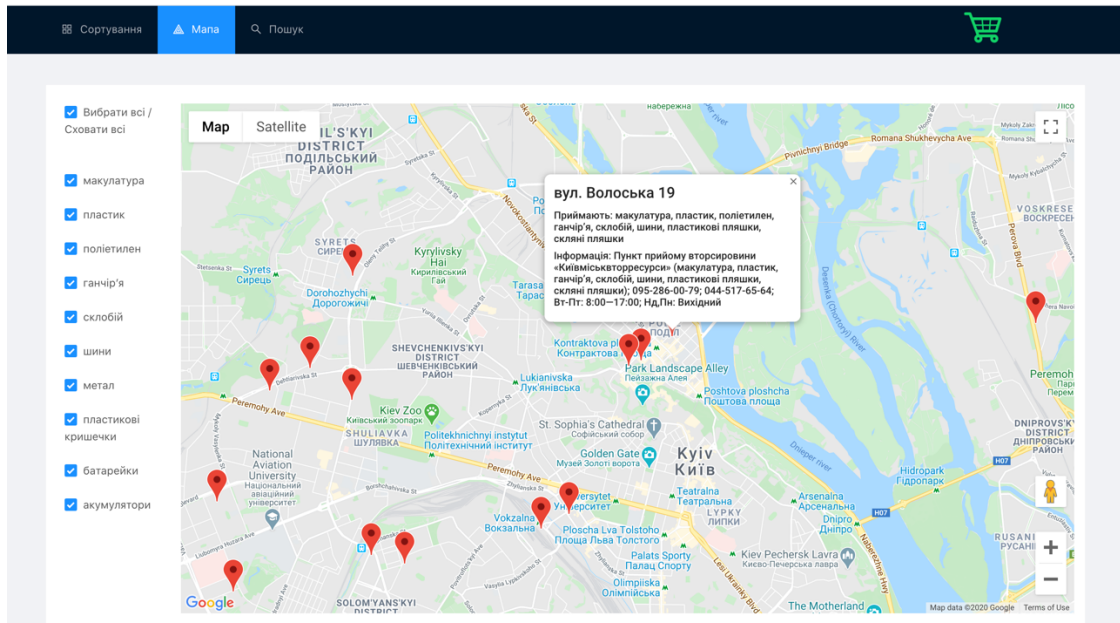


Рисунок 3.23 – Детальна інформація про місце прийому

Вікно з детальною інформацією містить такі дані (рис. 3.24):

- адреса;
- тип сировини, що приймають;
- інформація про місце (контакти, робочі дні, дані організації).

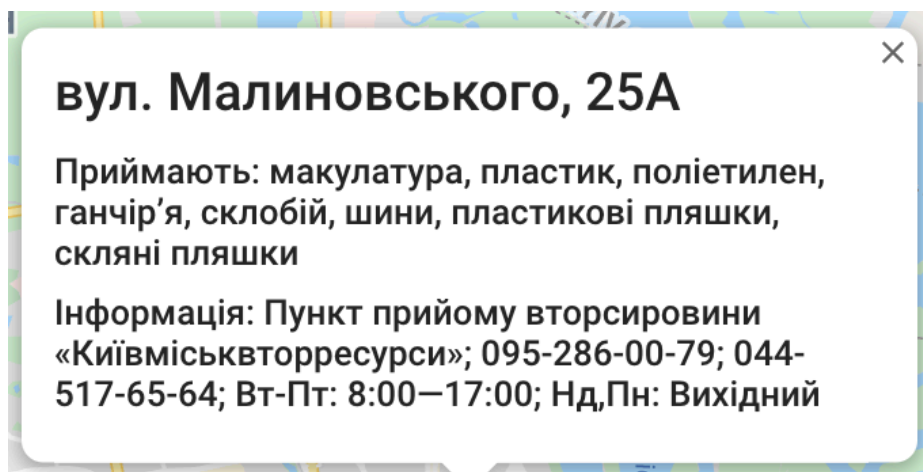


Рисунок 3.24 – Дані, що містяться в інформації про локацію

3.2.5 Опис розділу пошуку

Розділ пошуку являє собою сторінку із таблицею в якій наведені усі сутності вторинної сировини, наявні в базі даних. Дана таблиця містить такі колонки (рис. 3.25):

- назва відходу;
- метод здачі;
- опис сировини;
- кнопка для додавання сировини в кошик.

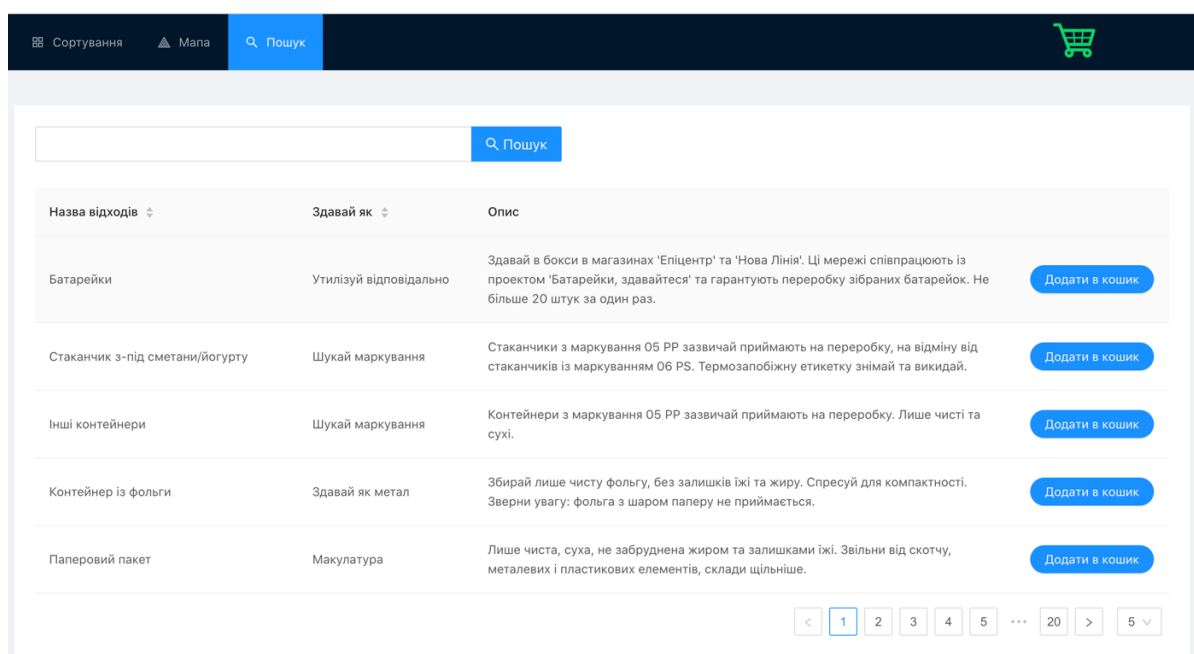


Рисунок 3.25 – Сторінка пошуку

Для зручності пошуку необхідної сировини реалізовано сортування в таблиці сутностей за назвою або методом здачі (рис. 3.26). Також доступний пошук, за допомогою якого легко можна знайти необхідну сировину (рис. 3.27).

Назва відходів	Здавай як	Опис	
Автомобільні оливи та рідини	Віддай на СТО	Відпрацьовані автомобільні оливи можуть стати у нагоді на найближчій СТО.	Додати в кошик
Аерозольний балон	Збирай окремо	Збирай лише чистим, залишивши, хіба що, етикетки. Не намагайся розбирати.	Додати в кошик
Аерозолі/Металевий посуд	Здавай як метал	Збирай лише чистим, але спробуй спочатку віддати через оголошення, або передай до 'Червоного Хреста'.	Додати в кошик
Алюмінієва бляшанка	Здавай як метал	Алюмінієві бляшанки зазвичай приймають на переробку. Промий водою, спробуй стиснути. Зверни увагу: деякі пивні бляшанки зроблені з жерсті. Перевір за допомогою магніту: жерсть магнітиться, алюміній - ні.	Додати в кошик
Банка з-під фарби	Здавай як метал	Металеві банки з-під фарб зазвичай приймають на переробку. Банка має бути пустою, накритою кришкою. Етикетку можеш не знімати.	Додати в кошик

Рисунок 3.26 – Таблиця відсортована за назвою сировини

Назва відходів	Здавай як	Опис	
Картонна коробка	Макулатура	Лише чиста, суха, не забруднена жиром та залишками їжі. Звільни від скотчу, металевих і пластикових елементів, склади щільніше.	Додати в кошик
Коробка з-під піци	Макулатура	Лише чиста, суха, не забруднена жиром та залишками їжі. Звільни від скотчу, металевих і пластикових елементів, склади щільніше.	Додати в кошик

Рисунок 3.27 – Пошук в таблиці

3.3. Розробка серверної частини проєкту

Як згадувалось раніше у даному вебресурсі реалізована дворівнева 'клієнт-серверна' архітектура. Серверна частина складається з одного серверу, який обробляє всі запити. Клієнтом є браузер. Клієнт і сервер

взаємодіють між собою за допомогою протоколу HTTP. Клієнт відправляє HTTP повідомлення, в якому вказано яку інформацію та в якому вигляді він хоче отримати від серверу. Сервер у свою чергу, отримавши таке повідомлення, відправляє клієнту відповідь з необхідною інформацією. В основі реалізованої взаємодії клієнта і сервера лежить принцип, що будь-яку взаємодію починає клієнт. Сервер лише відповідає клієнту, повідомляє, чи може він виконати запитовані дії або надати необхідну інформацію.

3.3.1 Створення серверу

Сервер створений за допомогою Nodejs та бібліотеки Express.js. У корінній папці додатку створений файл server.js, в якому ініціалізується сервер:

```
app.listen('8000', error => {  
  if (error) {  
    console.error(error);  
  } else {  
    console.info('server is running on port 8000')  
  };  
});
```

3.3.2 Створення API

Для доступу до бази даних створено API. API додатку винесено в окремий файл api.js і являє собою перерахунок всіх можливих запитів, що може отримувати сервер від клієнта. Перераховані такі запити:

- `app.get('garbageTypes')` – для отримання всіх типів вторинної сировини;
- `app.get('garbages')` – для отримання всіх сутностей вторинної сировини;
- `app.get('garbages/:type')` – для отримання сутностей вторинної сировини запитованого типу;
- `app.get('garbages/:name')` – для отримання сутності вторинної сировини за її назвою;
- `app.get('garbages/:sortAs')` – для отримання сутності вторинної сировини за методом її сортування;
- `app.get('locations')` – для отримання всіх сутностей локацій;
- `app.get('locations/:id')` – для отримання сутності локації за її ідентифікатором.

При зверненні до API виконуються такі кроки:

- підключення до бази даних;
- здійснення необхідних маніпуляцій;
- відправка даних у відповідь на запит.

3.3.3 Використання бази даних

У базі даних ми всього зберігаємо три колекції:

- `garbageTypes` – колекція, яка містить типи вторинної сировини, відображаються на початковій сторінці розділу сортування;
- `garbages` – колекція, яка містить всі сутності вторинної сировини;
- `locations` – колекція, яка містить всі місця прийому вторинної сировини.

Приклад сутності `garbageTypes`:

```
{  
  "name": "plastic",  
  "title": "Пластик",  
  "image": "source/plastic.jpg"  
}
```

Приклад сутності garbages:

```
{  
  "name": "Скляна пляшка",  
  "sortAs": "Здавай як скло",  
  "description": "Варто сполоснути. Етикетки не зривай, а кришечки  
збирай разом із металом або пластиком.",  
  "path": "Скло/Скляна пляшка"  
}
```

Приклад сутності locations:

```
{  
  "lat": 50.4317838,  
  "lng": 30.651661,  
  "address": "вул. Бориспільська, 1",  
  "sortAs": ["макулатура", "пластик", "поліетилен", "ганчір'я",  
"склобій", "шини", "пластикові пляшки", "скляні пляшки"],  
}
```

"description": "Пункт прийому вторсировини
«Київміськвотресурси»; 095-286-00-79; 044-517-65-64; Вт-Пт: 8:00—
17:00; Нд,Пн: Вихідний"

}

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		51

ВИСНОВКИ

Метою даного дипломного проекту було розроблення веборієнтованого сервісу для сортування та пошуку місць прийому вторинної сировини. Розроблена система надає користувачам можливість легко і зручно знаходити інформацію про сортування та місця прийому різних відходів.

У першій частині дипломного проекту для обґрунтування вибору теми та показу необхідності даного сервісу були розглянуті такі питання:

- чинне законодавство України;
- поточний процес сортування сміття;
- пошук інформації щодо сортування та місць прийому вторинної сировини.

Після аналізу була виявлена фактична відсутність конкурентів для розробленого сервісу.

У другому розділі було детально розглянуто питання вибору технологій для реалізації проекту. Вибір технологій був здійснений так, щоб вони забезпечували:

- підвищення швидкодії;
- спрощення реалізації та подальшої підтримки проекту;
- достатній рівень надійності;
- простота використання.

У третій частині було розглянуто реалізацію проекту. Було детально описано як загальну структуру додатку, так і окремі розділи та функції, які вони виконують. Значна увага приділяється опису розділу сортування, так як спрощення процесу сортування є головною метою даного проекту.

Як результат, всі поставлені завдання було виконано та розроблено зручний та ефективний вебдодаток, який реалізовує такі функції:

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		52

- пошук інформації про сортування вторинної сировини;
- сортування всіх вибраних відходів;
- відображення на мапі місць прийому;
- фільтрація місць прийому за необхідним типом сировини.

У майбутньому розроблений сервіс може бути покращеним. Зокрема можливо додати функції авторизації на сайті із подальшою оцінкою та коментуванням місць прийому. А також можливість для користувачів пропонувати нові місця прийому.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		53

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Закон України про відходи. – Електрон. дані (1 файл) – Режим доступу: <https://zakon.rada.gov.ua/laws/show/187/98-%D0%B2%D1%80> (дата звернення 15.03.2020) – Назва з екрану.
2. Ecomapa.gov - Електрон. дані (1 файл) – Режим доступу: <https://ecomapa.gov.ua/?layer=punkti> (дата звернення 05.04.2020) - Назва з екрану.
3. Epochtimes.com.ua - Електрон. дані (1 файл) – Режим доступу: <https://www.epochtimes.com.ua/maps/recycle/kyiv.php> (дата звернення 05.04.2020) - Назва з екрану.
4. Google.com - Електрон. дані (1 файл) – Режим доступу: <https://www.google.com/maps> (дата звернення 05.04.2020) - Назва з екрану.
5. 2gis.ua – Електрон. Дані (1 файл) – Режим доступу: <https://2gis.ua/uk/> (дата звернення 05.04.2020) – Назва з екрану.
6. sites.google.com/site – Електрон. Дані (1 файл) – Режим доступу: <https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti> (дата звернення 12.04.2020) – Назва з екрану.
7. reactjs.org/docs/ – Електрон. Дані (1 файл) – Режим доступу: <https://reactjs.org/docs/> (дата звернення 12.04.2020) – Назва з екрану.
8. <https://nodejs.org/en/docs> – Електрон. Дані (1 файл) – Режим доступу: <https://nodejs.org/en/docs/> (дата звернення 13.04.2020) – Назва з екрану.
9. <https://docs.mongodb.com/cloud> – Електрон. Дані (1 файл) – Режим доступу: <https://docs.mongodb.com/cloud/> (дата звернення 14.04.2020) – Назва з екрану.

10. Компоненты сетевого приложения. Клиент-серверное взаимодействие и роли серверов. – Электрон. дані (1 файл). – Режим доступа: <http://www.4stud.info/networking/lecture5.html> (дата звернення 07.05.2020). – Назва з екрану.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		55