

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Інтернет-платформа «Саллетрон»»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ.мз-91с Губанов Віталій Васильович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«__» грудня 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Ващенко С.М.

Голова комісії

(підпис)

Шифрін Д.М.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«___» _____ 2020 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентові

Губанов Віталій Васильович

(прізвище, ім'я, по батькові)

1 Тема проекту Інтернет-платформа «Саллетрон»

затверджена наказом по університету від «16» листопада 2020 р. №1773-III

2 Термін здачі студентом закінченого проекту « 07 » грудня 2020 р.

3 Вхідні дані до проекту технічне завдання на розробку Інтернет-платформи

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі та вибір інструментів для реалізації Інтернет-платформи, реалізація Інтернет-платформи «Саллетрон»

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність проблеми, аналіз аналогів, мета дипломної роботи, задачі проекту, реалізація Інтернет-платформи «Саллетрон»

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |

Дата видачі завдання _____.

Керівник _____
(підпис)Завдання прийняв до виконання _____
(підпис)**КАЛЕНДАРНИЙ ПЛАН**

| № п/п | Назва етапів випускної проекту | Термін виконання етапів проекту | Примітка |
|-------|--|---------------------------------|----------|
| 1 | Ознайомлення з предметною областю | 01.09.20 – 03.09.20 | |
| 2 | Обґрунтування потреби в платформі | 04.09.20 – 09.09.20 | |
| 3 | Аналіз існуючих Інтернет-платформ | 10.09.20 – 16.09.20 | |
| 4 | Визначення вимог до розробленої платформи | 17.09.20 – 23.09.20 | |
| 5 | Визначення інструментарію для реалізації проекту | 24.09.20 – 29.09.20 | |
| 6 | Планування WBS | 30.09.20 – 30.09.20 | |
| 7 | Планування OBS | 01.10.20 – 01.10.20 | |
| 8 | Складання календарного плану розробки Інтернет-платформи | 02.10.20 – 02.10.20 | |
| 9 | Визначення ризиків | 05.10.20 – 06.10.20 | |
| 10 | Розробка прототипу Інтернет-платформи | 07.10.20 – 12.10.20 | |
| 11 | Розробка каркасу десктопної версії Інтернет-платформи | 13.10.20 – 21.10.20 | |
| 12 | Розробка необхідних функціональних модулів | 22.10.20 – 04.11.20 | |
| 14 | Тестування Інтернет-платформи | 11.11.20 – 11.11.20 | |
| 15 | Оформлення супровідної документації | 10.09.20 – 13.11.20 | |
| 16 | Архівація проекту та введення в експлуатацію | 16.11.20 – 16.11.20 | |

Магістрант _____

Губанов В.В.

Керівник роботи _____

к.т.н., доц. Ващенко С.М.

РЕФЕРАТ

Тема дипломної роботи «Інтернет-платформа «Саллетрон»».

Дипломна робота складається зі вступу, чотирьох розділів, висновку, списку літератури та додатків.

Пояснювальна записка містить 91 с., 54 рис., 4 табл., 2 додатків, 45 джерел.

Перший розділ присвячений детальному дослідженню питання необхідності Інтернет-платформи та аналізу існуючих аналогів.

У другому розділі виконується конкретизація мети проекту, формування переліку задач для реалізації проекту, вибір засобів реалізації та планування робіт.

Третій розділ присвячений моделюванню процесу проектування Інтернет-платформи з використанням діаграм нотації IDEF0 та Use Case.

Четвертий розділ присвячений детальному процесу реалізації проекту: макетування платформи та підбір/редагування графічних матеріалів, розробка каркасу ресурсу, розробка функціональних модулів платформи, тестування розробленого проекту.

Результатом дипломної роботи є Інтернет-платформа «Саллетрон», що підвищить показник Інтернет-купівлі/продажу на території України.

Ключові слова: ІНТЕРНЕТ-ПЛАТФОРМА, JAVASCRIPT, МОДУЛЬ, ТЕСТУВАННЯ, МАКЕТУВАННЯ.

ЗМІСТ

| | |
|---|----|
| Вступ..... | 6 |
| 1 Аналіз предметної області | 7 |
| 1.1 Актуальність проблеми..... | 7 |
| 1.2 Аналіз існуючих аналогів | 9 |
| 2 Постановка задачі та методи дослідження..... | 16 |
| 2.1 Мета та задачі дослідження..... | 16 |
| 2.2 Вибір інструментів для реалізації Інтернет-платформи..... | 18 |
| 3 Проектування Інтернет-платформи «Саллетрон» | 22 |
| 3.1 Діаграми нотації IDEF0..... | 22 |
| 3.2 Діаграма Use Case..... | 32 |
| 4 Реалізація Інтернет-платформи «Саллетрон» | 34 |
| 4.1 Створення макету платформи | 34 |
| 4.2 Реалізація платформи | 45 |
| Висновки | 58 |
| Список використаних джерел | 60 |
| Додаток А..... | 64 |
| Додаток Б..... | 75 |

ВСТУП

Актуальність: В сучасному світі найбільший показник продажі-купівлі припадає на онлайн-ресурси. Актуальність розробки інтернет платформи на даний час зростає у геометричній прогресії, проте більшість існуючих аналогів мають недоліки у використанні як з боку продавця, так і з боку покупця. Тому було прийнято рішення про розробку юзабільної та ефективної у використанні Інтернет платформи без наявності виявлених недоліків у існуючих аналогів.

Тема: Інтернет-платформа «Саллетрон».

Мета: Розробити Інтернет-платформу «Саллетрон» для реалізації процесу продажу-купівлі товарів побутового вжитку на території України.

Об'єкт дослідження: Дослідження Інтернет-платформи «Саллетрон» для продажу речей повсякденного використання.

Предмет дослідження: Процес розробки Інтернет-платформи «Саллетрон».

Практична цінність: Створення універсальної Інтернет-платформи «Саллетрон» для підвищення показника ефективності процесу купівлі-продажу на території України.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

У світі потужного розвитку інформаційних технологій користувач має можливість виконувати будь-які операції з використанням Інтернету та електронного пристрою. На сьогоднішній день існують онлайн-платформи будь-якої сфери для покращення показника кількості користувачів та для збільшення показника прибутковості підприємства. Наприклад, будь-який банк України має власний мобільний додаток, за допомогою якого користувач може виконувати будь-які фінансові операції без наочного контакту з оператором банку. Також більшість ресторанів та кафе мають власні сайти, де користувач може ознайомитись з інтер'єром, меню та за можливістю виконати доставку страв. Також на даний час зростає популярність використання Інтернет-платформ для реалізації процесу купівлі-продажу[1,2].

Інтернет-платформа – це посередник між користувачем-продавцем та користувачем-покупцем [3,4,5]. Користувач даного виду ресурсу може бути як фізичне обличчя, так і юридичне. У зв'язку з цим процес будь-якого товару та послуги на просторах Інтернету став набагато легшим та швидшим [6].

Основною ціллю будь-якої Інтернет-платформи є збільшення кількості користувачів та збільшення кількості угод з мінімальними вхідними ресурсами. Найголовніша ціль даного ресурсу – це швидка «конвертація» покупця в продавця. При наявності юзабельного інтерфейсу користувача показник кількості зацікавленої аудиторії стрімко зростає [7,8,9].

Використання Інтернет-платформи з боку продавців має свої переваги:

– розміщення об'яв товару або послуги в більшості випадків не вимагає витрачання коштів;

- можливість укладення угоди про купівлю-продаж в швидкий строк;
- продавець має доступ до великої кількості зацікавлених покупців;
- більшість платформ має достатній функціонал для проведення аналітики ефективності продаж, за рахунок чого продавець може провести аналіз популярних товарів або послуг для збільшення показника продажу [10,11].

Проте існують недоліки у використанні Інтернет-платформ з боку продавців:

- наявність комісії за товар;
- наявність сплати за «перехід» на об'яву продавця;
- наявність щомісячної або щорічної підписки;
- наявність плати за рекламу товару або послуги [3].

З точки зору користувача-покупця використання Інтернет-площадки має свої переваги також:

- швидкий пошук бажаного товару або послуг з використанням фільтрації;
- можливість детального ознайомлення з товаром за рахунок використання доданої галереї;
- перегляд товарів зі знижками, акціями;
- розрахунок точної вартості товару або послуги з урахуванням доставки;
- можливість підтримки контакту з продавцем за рахунок використання вбудованого чату;
- можливість перегляду статусу обраного товару [12,13].

З урахуванням вище розглянутих переваг використання Інтернет-платформ з боку продавців та покупців, актуальність розробки даного виду ресурсу зростає на даний час. Тому було прийнято рішення про розробку універсальної та юзабельної Інтернет-платформи «Саллетрон» для підвищення показника продажів товарів повсякденного вжитку на території України.

Даний проект буде нести попит у сфері купівлі-продажу з використанням Інтернет ресурсів.

1.2 Аналіз існуючих аналогів

Визначивши актуальність розробки Інтернет-платформ для реалізації процесу купівлі-продажу, необхідно було провести аналіз існуючих аналогів.

На сьогоднішній день найбільш популярними Інтернет-ресурсами даної сфери є OLX.ua, Prom.ua та Klumba.com. Для проведення експертного аналізу були обрані саме популярні Інтернет-платформи на території України, оскільки метою магістерської роботи є розробка Інтернет-платформи для підвищення показника ефективності та легкості процесу купівлі-продажу на даній території.

Першим розглянутим аналогом був OLX.ua. Даний ресурс виступає у ролі платформи оголошень будь-яких товарів та послуг. На даний час на ресурсі OLX.ua існує велика кількість товарів, які можна класифікувати на такі групи: «Одяг», «Нерухомість», «Транспорт», «Робота», «Тварини», «Хоббі» та інші [14,15].

Дана Інтернет-платформа надає можливість будь-яким авторизованим користувачам розмістити об'яви товару або послуг. На офіційній сторінці OLX.ua зазначено, що новий користувач буде мати прибутковість у розмірі 15 угод за хвилину та те, що нова об'ява буде успішно закрита через 14 днів [16]. Також розробники платформи анонсують той факт, що ніяких додаткових коштів на початку старту бізнесу користувачу не потрібно. Проте провівши аналіз відгуків користувачів платформи були отримані такі результати:

- можливість розміщення обмеженої кількості об'яв з профілю користувача, максимальна кількість 12.
- необхідність вкладання коштів у рекламу, оскільки нові об'яви без підтримки платформи не відображаються при пошуку для інших користувачів.

– необхідність використання якісних фото, які відповідають корпоративній етиці платформи.

До того ж, з боку звичайних користувачів на даний час платформа OLX.ua втрачає популярність у зв'язку з редизайном власної оболонки. На думку більшості користувач, минула версія платформи була більш зручною та зрозумілою для недосвідченого користувача, ніж сучасна. На рис. 1.1 представлена стара версія ресурсу, на рис.1.2 – сучасна.



Рисунок 1.1 – Стара версія інтерфейсу «OLX.ua»

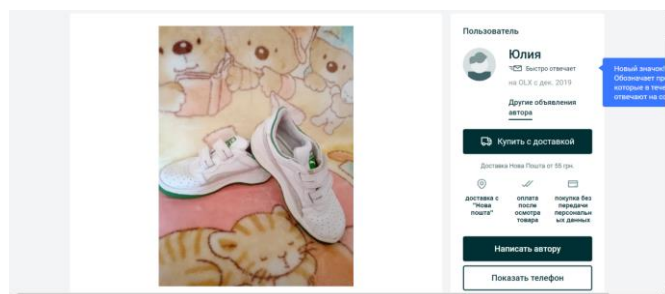


Рисунок 1.2 – Нова версія інтерфейсу «OLX.ua»

Крім того, що розробниками платформи були змінені корпоративні кольори, які були популярними та знайомими для більшості користувачів України, був змінений сам механізм пошуку та перегляду об'яв. За даними більшості

опитувальників користувачі скаржаться на відсутність зручної системи фільтрації. Оскільки пошук потрібного товару на певний час став довшим та складнішим. Також більшість користувачів скаржаться на відсутність зручної функції «скролінгу» галереї товару. В старій версії платформи розробниками була передбачена карусель з використанням інтерактивних кнопок або жестами, сучасна версія передбачає використання перегляд галереї лише з використанням кнопок у вигляді занадто тонких стрілок. Для звичайного користувача, який прагне комфортної роботи з ресурсу, даний варіант механізму не є прийнятним. На рис. 1.3 представлена панель перегляду фото-галереї товару.

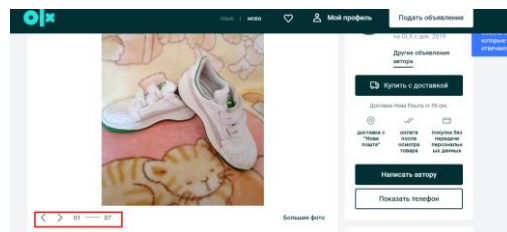


Рисунок 1.3 – Механізм перегляду фото-галереї

Наступний аналог – Prom.ua. Prom.ua – друга за популярністю Інтернет-платформа на території України. За статистикою на 2019 р. платформа Prom.ua входить до ТОП-10 найбільш відвідуваних сайтів України [17]. На рис. 1.4 представлена на стартова сторінка ресурсу.

Проаналізувавши дизайн платформи з точки зору принципів UI/UX, можна з впевненістю затвердити, що всі вони порушені. Більшість користувачів звикли, що головні групи товарів розміщені в центральній частині контенту, бажано вгорі. Проте розробниками Prom.ua були вирішено про розміщення системи фільтрації в боковій панелі (рис. 1.5).

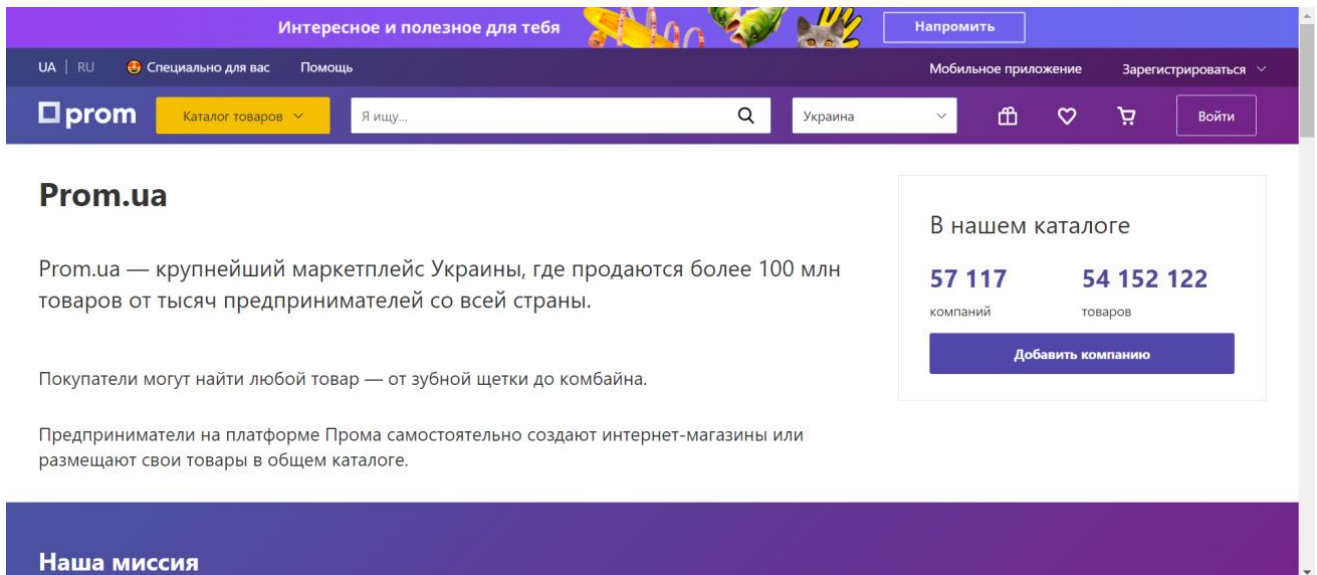


Рисунок 1.4 – Стартова сторінка Prom.ua

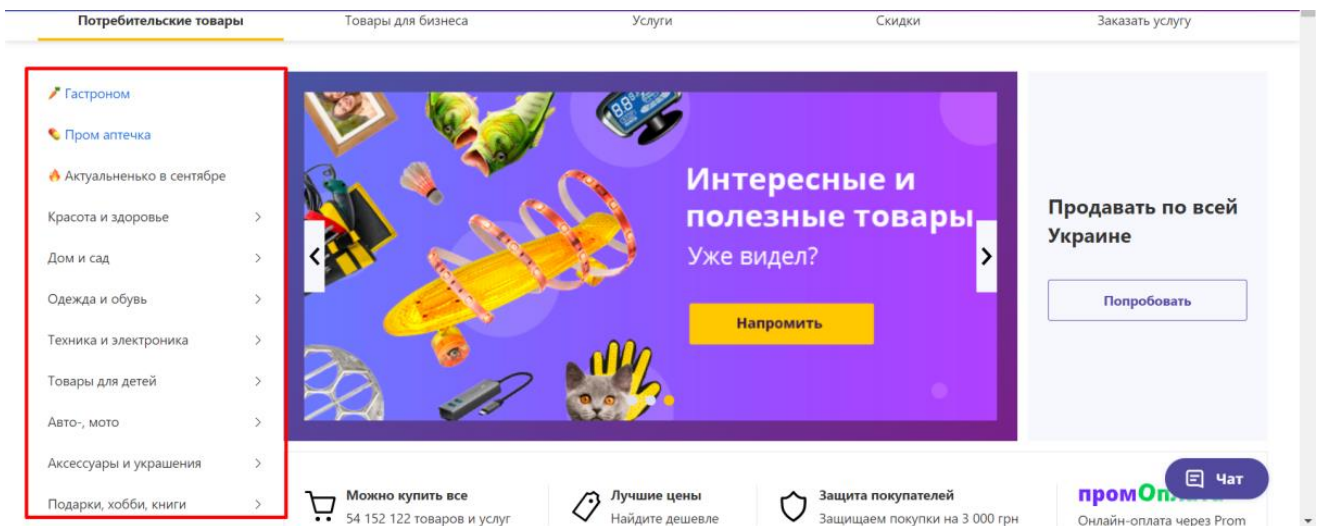


Рисунок 1.5 – Система фільтрації

До того, ж для більшості користувачів не є прийнятним той факт, що один і той самий продукт може знаходитися в різних секціях. Дана система групування товарів збільшує час пошуку з боку користувача та зменшує показник ефективності роботи з ресурсом (рис. 1.6).

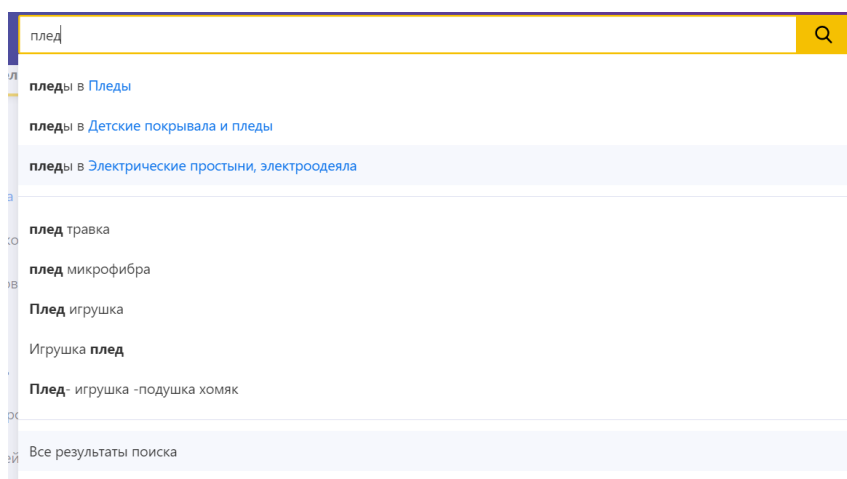


Рисунок 1.6 – Система групування товарів

До того, ж провівши аналіз відгуків користувачів, були отримані такі результати:

- некоректна робота системи фільтрації при виборі «приватне обличчя» та «оптові закупівлі»;
- велика кількість випадків витрати зворотного зв'язку з боку продавця;
- відсутність функції «повернення» при отриманні пошкодженого товару;
- завеликі цітники на рекламу приватних об'яв;
- жорсткі корпоративні правила для розміщення об'яв товару [19].

Останній аналог – Klumba.com. Даний ресурс розроблений з урахуванням більшості принципів юзабільності. Панель фільтрації знаходиться в центральній частині контенту, що звичайно пришвидшує пошук необхідного товару серед користувачів (рис.1.7). До того ж групування товарів відбувається у більш зручній формі та логічній для користувачі. Оскільки даний ресурс позиціонує себе як платформа продажу одягу для всієї родини, то групування відбувається лише за трьома групами «Жінки», «Чоловіки» та «Діти» (рис. 1.8) [20].

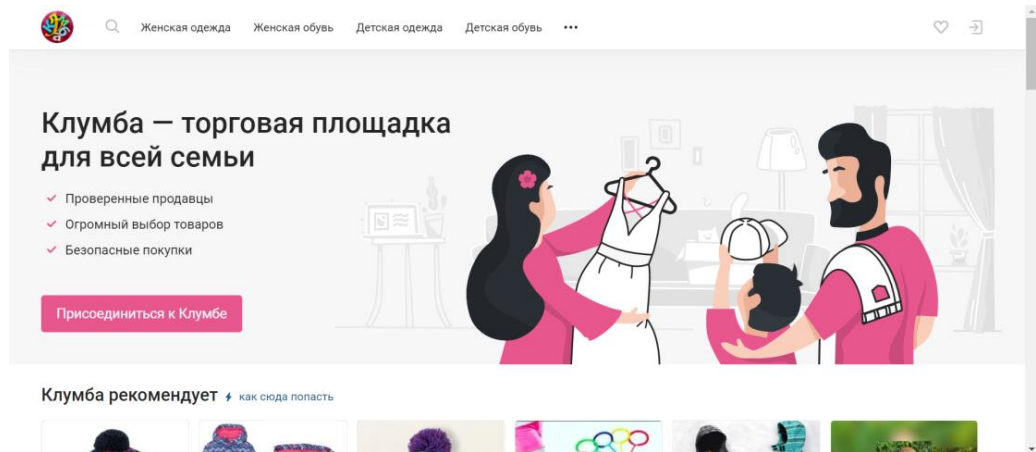


Рисунок 1.7 – Стартовое вікно ресурсу

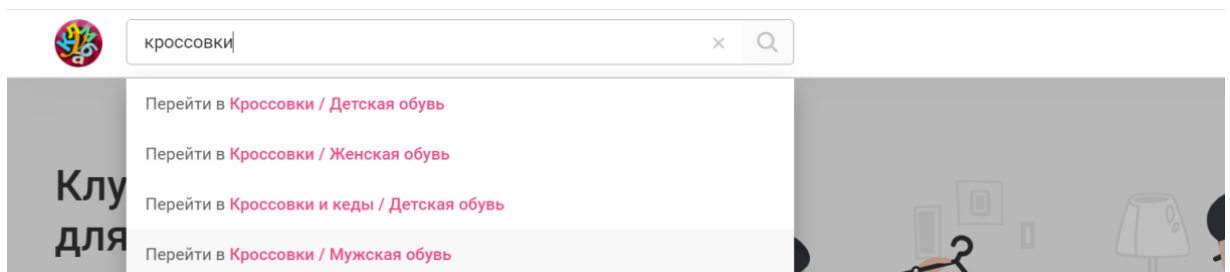


Рисунок 1.8 – Система групування товарів

Провівши аналіз відгуків користувачів даного ресурсу, були отримані результати:

- великий асортимент одягу за дуже малими цінами;
- відсутність системи повернення за необхідністю;
- середній показник завершення угод з випадком не прибуття товару або не виплати коштів.

Після проведення аналізу існуючих аналогів, було проведено експертне порівняння можливостей та недоліків використання розглянутих ресурсів та розробленого проекту «Саллетрон». У табл. 1.1 представлені результати експертного аналізу.

Таблиця 1.1 – Порівняльна характеристика аналогів

| Назва критерію | Назва ресурсу | | | |
|--|---------------|---------|------------|-----------|
| | OLX.ua | Prom.ua | Klumba.com | Саллетрон |
| Зручна навігація по сайту | - | - | + | + |
| Можливість розміщення товару з мінімальним вкладом | - | - | + | + |
| Відсутність реклами | - | - | - | + |
| Адаптивність сайту під мобільну версію | + | + | - | + |

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Метою магістерської роботи є розробка Інтернет-платформи «Саллетрон» для реалізації процесу продажу-купівлі товарів побутового вжитку на території України. Даний проект буде нести соціально-економічну вигоду для тих користувачів, які прагнуть без мінімальних вкладів на старті розпочати власний бізнес та для тих користувачів, які прагнуть швидко та зручно віднайти потрібний товар.

Також при збільшенні аудиторії та запитів на функціонал, даний проект може бути розширеним для користувачів ЄС.

Розроблений проект «Саллетрон» повинен підтримувати такі функції:

- наявність форми логізації/авторизації користувача;
- можливість швидкого пошуку товару;
- наявність зручної системи фільтрації;
- наявність форми для додавання власної об'яви.

Функціональні вимоги Інтернет-платформи:

- функція мультимовності;
- наявність інтерактивності перегляду галереї товару.

Нефункціональні вимоги:

- дизайн повинен відповідати сучасним тенденціям;
- зручна системи навігації;
- наявність адаптивної мобільної версії.

Інтернет-платформа «Саллетрон» повинна бути розроблена у форматі лендінгу з урахуванням останніх тенденцій кольорових гам. Мобільна версія

повинна бути розроблена з урахуванням всіх принципів UI/UX дизайну мобільних додатків та мати також легку систему навігації.

Для досягнення мети науково-дослідницької роботи необхідно виконати перелік задач:

- провести аналіз проблеми реалізації Інтернет-платформи;
- провести експертний аналіз існуючих Інтернет-платформ;
- розробити макет платформи та виконати реалізацію каркасу Інтернет-платформи;
- виконати розробку необхідних функціональних модулів платформи з урахуванням адаптації у мобільну версію сайту;
- виконати тестування платформи на всіх видах електронних пристроїв.

Розробка прототипу Інтернет-платформи дозволить узгодити разом з замовником дизайн та принцип роботи платформи. Інтернет-платформа «Саллетрон» повинна бути розроблена у форматі лендінгу з урахуванням останніх тенденцій кольорових гам. Мобільна версія повинна бути розроблена з урахуванням всіх принципів UI/UX дизайну мобільних додатків та мати також легку систему навігації.

Після того, як прототип буде затверджений, розробник проекту може виконати реалізацію каркасу десктопної версії платформи з урахуванням затверджених макетів. Далі необхідно виконати реалізацію функціональних модулів платформи:

- логінізація/авторизація користувачів;
- відображення категорій товару та їх перегляд;
- панель додавання товару.

Далі необхідно виконати адаптацію десктопної версії в мобільну. Для цього необхідно додатковий модуль, який виконає розрахунок кількості компонентів, їх розмірів в залежності від розміру екрану пристрою.

Останнім етапом розробки є тестування платформи на різних видах електронних пристроях. Даний етап дозволить перевірити коректність роботи всіх компонентів додатку в залежності від версії. Для коректного та своєчасного процесу реалізації проекту було виконано етап планування робіт, що представлений в Додатку А.

2.2 Вибір інструментів для реалізації Інтернет-платформи

Наступним кроком після визначення мети та задач проекту був вибір засобів реалізації Інтернет-платформи. Вибір засобів інструментів виконувався з урахуванням основних задач проекту. Першим етапом реалізації Інтернет-платформи «Саллетрон» є реалізація макету платформи. Процес макетування полягає у визначенні розташування основних блоків веб-ресурсу, наповнення їх графічними матеріалами та приведення всього контенту до єдиної кольорової гами. Тому для даного етапу було обрано програмний засіб Adobe Photoshop.

Adobe Photoshop – багатофункціональний графічний редактор від компанії Adobe. Перш за все Adobe Photoshop був реалізований для редагування графічних зображень [21].

Проте з більш великою частотою оновлення версій програмного забезпечення функціонал Adobe Photoshop також оновлювався та доповнювався. На даний час за допомогою Adobe Photoshop користувач має можливість:

- виконувати роботу з ескізами та їх доповнювати;
- виконувати післяобробку 2d та 3d моделей;
- маніпулювати формою моделей;
- створювати макети сайтів з авторськими елементами [22].

Функціонал Adobe Photoshop версії 2020 року має зовсім новий функціонал, за рахунок якого користувач з легкістю може виконати розробку макету та навіть його анімувати у форматі gif [23].

До того ж урахувуючи те, що продукти Adobe створюють своєрідну екосистему, існує можливість постійної інтеграції результуючих даних з одного програмного засобу в інший. Таким чином, за бажанням, користувач може виконати прототипування макету з використанням Adobe Xd. Проте лише за умови, що макети будуть реалізовані засобами програмного забезпечення Adobe Photoshop.

Наступною задачею проекту є розробка каркасу Інтернет-платформи. В даному випадку, оперуючи базою знань розробника, було прийнято рішення про застосування мови html. HTML – мова розмітки веб-сторінки, складовою якої є теги. Кожний тег відповідає за той чи інший компонент веб-сторінки [24].

Перевагами використання мови програмування HTML є:

- простота у використанні;
- легка мова для вивчення;
- безкоштовний доступ до всього функціоналу мови;
- підтримується майже в кожному браузері;
- мова програмування широко застосовується в браузерах;
- можливість використання сторонніх шаблонів [25].

Проте дана мова програмування має свої недоліки:

- користувач може створювати лише статичні та звичайні веб-сторінки;
- необхідно написати багато коду для створення простої веб-сторінки;
- HTML не має функцій безпеки веб-сторінки [26].

Як вже було описано раніше, HTML застосовується лише створення статичних веб-сторінок, тому для реалізації функціоналу та додавання динаміки до веб-сторінки було обрано мову програмування JavaScript.

Обрана мова програмування має свої переваги в експлуатації:

- Швидкість (Клієнтський JavaScript дуже швидкий, оскільки його можна негайно запустити в клієнтському браузері. Якщо не потрібні зовнішні ресурси, JavaScript не перешкоджає мережевим викликам до серверного сервера);
- простота (JavaScript порівняно простий у вивченні та реалізації);
- популярність (JavaScript широко використовується в Інтернеті);
- Сумісність (JavaScript чудово поєднується з іншими мовами і може бути використаний у величезній кількості програм);
- навантаження сервера (Будучи на стороні клієнта, зменшується попит на сервері веб-сайту);
- JavaScript дає можливість створювати розширені інтерфейси [27].

Проте існують недоліки у використанні JavaScript:

- захист на стороні клієнта (Оскільки код JavaScript виконується на стороні клієнта, помилки та недогляди іноді можуть бути використані для зловмисних цілей. Через це деякі люди вирішують повністю вимкнути JavaScript);
- підтримка браузера (Хоча сценарії на стороні сервера завжди дають однакові результати, різні браузери іноді інтерпретують код JavaScript по-різному. У наш час відмінності мінімальні, і користувачу не слід турбуватися про це, поки йде тестування сценарію у всіх основних браузерах) [28].

Також для того, щоб виконувати швидку та ефективну збірку функціональних модулів було прийнято рішення про використання допоміжного та стороннього модуля webpack. Перевагами використання модуля є :

- застосування окремих стилів для обраного модуля;
- формування під кожну сторінку власний стиль в окремому файлі;
- швидкість відображення сайту та його рендеру [29].

Для того, щоб Інтернет-платформа мала власне сховище даних, було обрано ресурс Firebase. На даний час Firebase є досить популярним засобом використання в галузі веб-розробки, оскільки поєднує в собі функціонал для серверу, для бази

даних, для хостингу. За рахунок наданого API розробник може миттєво виконати синхронізацію даних клієнтів з сервером [30].

3 ПРОЕКТУВАННЯ ІНТЕРНЕТ-ПЛАТФОРМИ «САЛЛЕТРОН»

3.1 Діаграми нотації IDEF0

Після того, як була виконана низка задач:

- детальний аналіз необхідності в реалізації Інтернет-платформи для покращення процесу продажу-купівлі товарів побутового вжитку на території України;
- експертний аналіз існуючих платформ для продажу товарів та послуг на території України;
- сформована детальна мета дипломного проекту з зазначенням результуючим ефектом застосування продукту;
- сформований перелік необхідних задач для реалізації проекту;
- проведено аналіз та вибір засобів реалізації;
- виконано планування робіт проекту,

розробник проекту перейшов до наступного етапу робіт – проектування Інтернет-платформи.

Перший етап проектування проекту – створення деталізованих діаграм нотації IDEF0.

IDEF0 – графічна нотація функціонального моделювання для відображення бізнес-процесів. Діаграми нотації IDEF0 являють собою набір блоків, кожних з яких має вхідні та вихідні дані. Кожний з блоків діаграми розбивається на підзадачі, кожні в яких приводять до результату – батьківського блоку [31].

Діаграми нотації IDEF0 набули широкого застосування, по – перше, завдяки простоті графічного представлення. Головними складовими моделі є примітивні фігури: прямокутник та стрілка. Прямокутниками відображаються основні процеси

розробки або експлуатації системи, за рахунок стрілок відображається зв'язок системи з зовнішніми факторами та зв'язок процесів між собою [32].

Перевагами використання діаграм нотації IDEF0 є:

- детальний опис бізнес-процесів;
- детальна декомпозиція процесів;
- простота відображення інформації;
- можливість корегування кількості процесів.

Проте застосування діаграм нотації IDEF0 при проектуванні проекту має недоліки, а саме:

- надлишок стрілок при відображенні всіх зв'язків;
- велика кількість блоків при декомпозиції [33];

Першим структурним елементом діаграми нотації IDEF0 – контекстна діаграма А-0. При розробці контекстної діаграми розробник проекту має виявити головну мету (задачу), що реалізує бізнес-процес. За рахунок контекстної діаграми виконується відображення загальної задачі бізнес-процесу: кількість процесів, фактори-впливу, необхідний інструментарій, вхідні дані. Результат проекту [34].

На основі опису основних компонентів діаграми А-0 були виявлені дані для контекстної діаграми проекту «Розробка Інтернет-платформи «Саллетрон»»:

- Вхідні дані: технічне завдання.
- Управління: загально прийняті принципи макетування, принципи модульного програмування.
- Механізми: Adobe Photoshop, Firebase, SublimeText, JavaScript, webpack.
- Вихідні дані: Інтернет-платформа «Саллетрон».

На рис.3.1 представлена контекстна діаграма проекту «Розробка Інтернет-платформи «Саллетрон»».

Після того, як була сформована діаграма А-0, була виконана її декомпозиція на основні процеси проекту:

- створення макету платформи;
- реалізація платформи.

Для кожного етапу були зазначені дані, які були взяті з попередньо створеної діаграми А-0.

Дані для процесу «Створення макету платформи»:

- Вхідні дані: технічне завдання.
- Управління: загально прийняті принципи макетування.
- Механізми: Adobe Photoshop.
- Вихідні дані: макет платформи.

Дані для процесу «Реалізація платформи»:

- Вхідні дані: технічне завдання, макет платформи.
- Управління: принципи модульного програмування.
- Механізми: Firebase, SublimeText, JavaScript, webpack.
- Вихідні дані: Інтернет-платформа «Саллетрон».

На основі зазначених даних для базових процесів проекту була створена діаграма другого рівня, що представлена на рис. 3.2.

Далі необхідно виконати декомпозицію кожного процесу на основі підзадачі таким чином, щоб результат попередньої задачі був вхідними даними для наступної задачі.

Процес «Створення макету платформи» було розбито на підзадачі:

- підготовка растрових та векторних зображень;
- макетування платформи.

Для підпроцесу «Підготовка растрових та векторних зображень» були обрані дані:

- Вхідні дані: технічне завдання.
- Управління: загально прийняті принципи макетування.
- Механізми: Adobe Photoshop.
- Вихідні дані: графічний матеріал.

Для підпроцесу «Макетування платформи» були обрані дані:

- Вхідні дані: технічне завдання, графічний матеріал.
- Управління: загально прийняті принципи макетування.
- Механізми: Adobe Photoshop.
- Вихідні дані: макет платформи.

На основі детальної декомпозиції процесу «Створення макету платформи» була створена діаграма, що представлена на рис. 3.3.

Процес «Реалізація платформи» було розбито на підзадачі:

- структуризація проекту;
- реалізація модулів;
- відладка модулів;
- тестування платформи.

Для підпроцесу «Структуризація проекту» були обрані дані:

- Вхідні дані: технічне завдання, макет платформи.
- Управління: принципи модульного програмування.
- Механізми: Firebase, SublimeText, JavaScript, webpack.
- Вихідні дані: структура проекту.

Для підпроцесу «Реалізація модулів» були обрані дані:

- Вхідні дані: технічне завдання, макет платформи, структура проекту.
- Управління: принципи модульного програмування.
- Механізми: Firebase, SublimeText, JavaScript, webpack.
- Вихідні дані: функціональні модулі.

Для підпроцесу «Відладка модулів» були обрані дані:

- Вхідні дані: функціональні модулі.
- Управління: принципи модульного програмування.
- Механізми: Firebase, SublimeText, JavaScript, webpack.
- Вихідні дані: цілісний проект.

Для підпроцесу «Тестування платформи» були обрані дані:

- Вхідні дані: цілісний проект.
- Вихідні дані: цілісний проект.

На основі зазначених даних декомпозицій процесу «Реалізація платформи» була створена діаграма, що представлена на рис. 3.4.

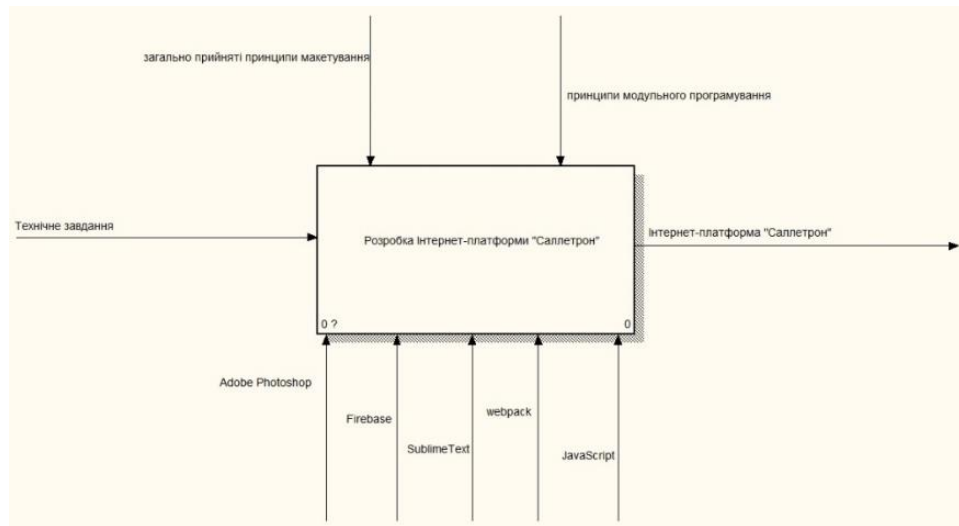


Рисунок 3.1 – Діаграма А-0

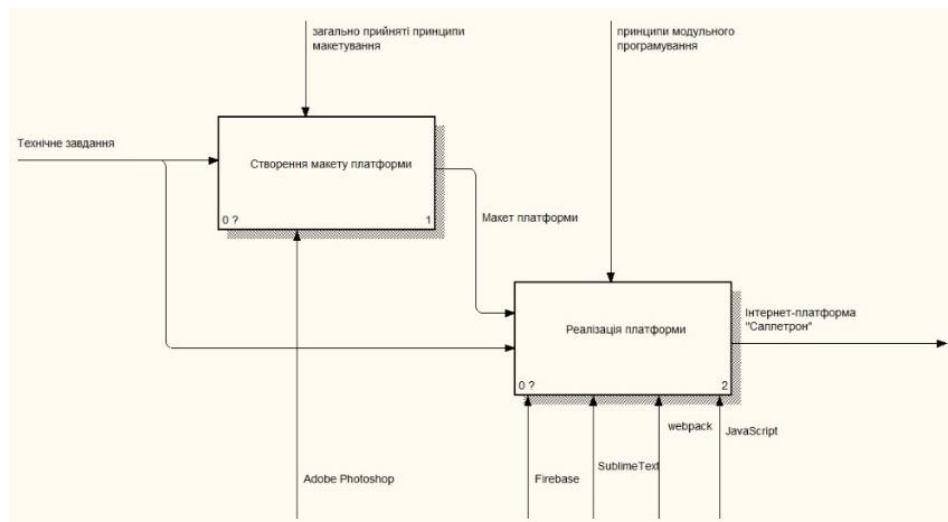


Рисунок 3.2 – Діаграма другого рівня

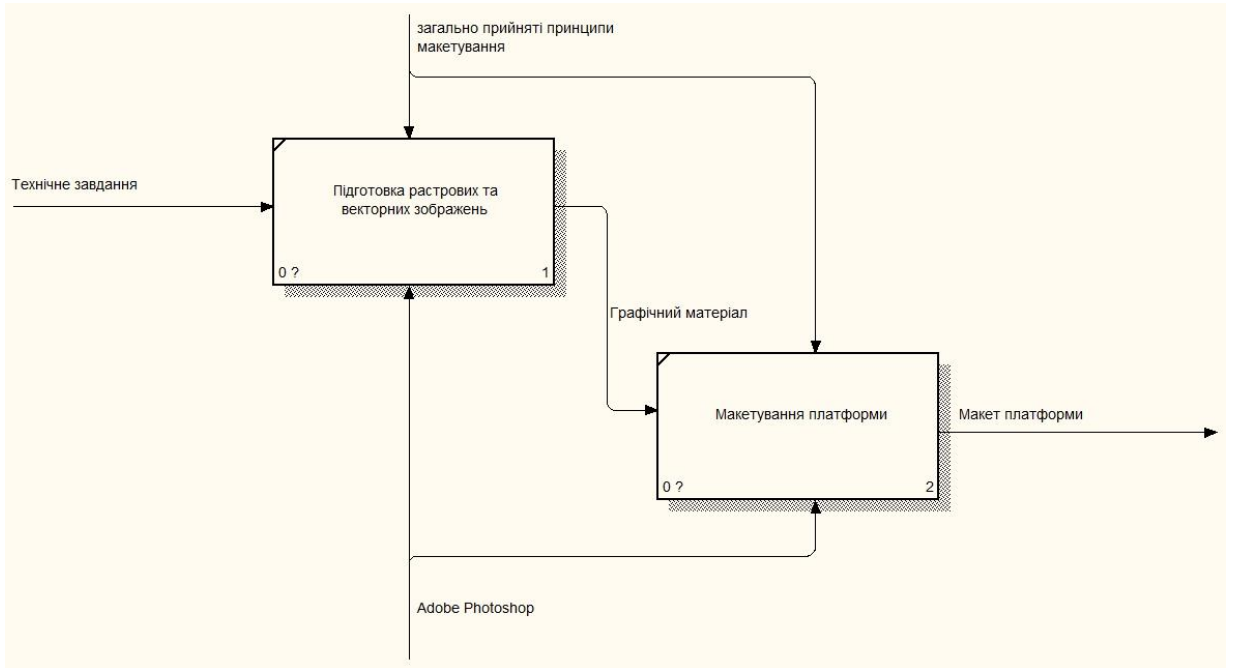


Рисунок 3.3 – Діаграма декомпозиція процесу «Створення макету платформи»

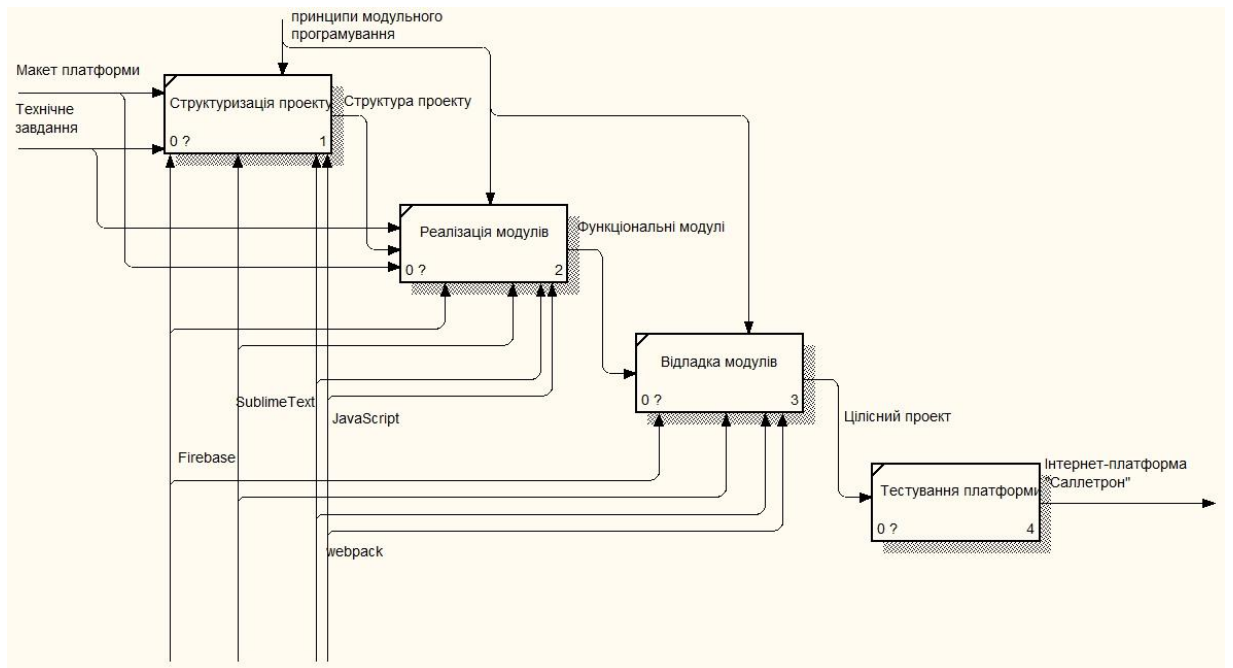


Рисунок 3.4 – Діаграма декомпозиція процесу «Реалізація платформи»

Далі на основі розроблених діаграм нотації IDEF0 процесу розробки було виконано проектування діаграм процесу використання платформи з боку користувача, що прагне додати новий товар для продажу. Для контекстної діаграми даного процесу були визначені дані, що наведені нижче:

- Вхідні дані: персональні дані користувача, опис товару, графічні зображення товару.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.
- Вихідні дані: оновлений перелік товарів для продажу, інформація про поточні товари.

На рис. 3.5 представлена контекстна діаграма процесу «Використання Інтернет-платформи «Саллетрон»».

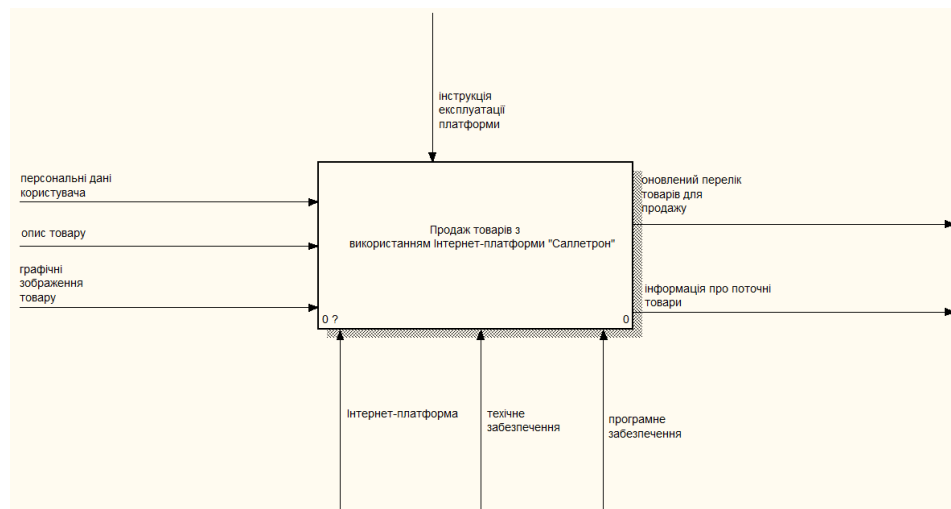


Рисунок 3.5 – Контекстна діаграма процесу «Використання Інтернет-платформи «Саллетрон»»

Далі була проведена декомпозиція контекстної діаграми на три основних підпроцесу:

- реєстрація/авторизація користувача;

- додавання інформації про товар;
- перегляд товарів на платформі.

Для підпроцесу «Реєстрація/авторизація користувача» були визначені наступні дані:

- Вхідні дані: персональні дані користувача.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.
- Вихідні дані: авторизований користувач.

Для підпроцесу «Додавання інформації про товар» були визначені наступні дані:

- Вхідні дані: авторизований користувач, опис товару, штрафні зображення товару.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.

- Вихідні дані: необхідність перегляду товарів платформи, оновлений перелік товарів для продажу.

Для підпроцесу «Перегляд товарів на платформі» були визначені наступні дані:

- Вхідні дані: необхідність перегляду товарів платформи.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.

- Вихідні дані: інформація про поточні товари.

На рис. 3.6 представлена діаграма декомпозиції першого рівня.

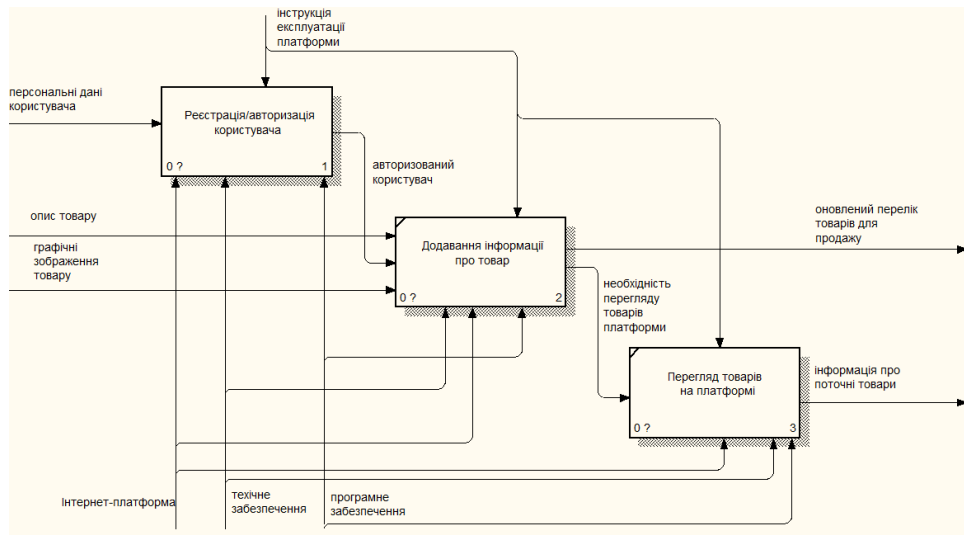


Рисунок 3.6 – Діаграма декомпозиції першого рівня

Далі була проведена декомпозиція підпроцесу «Реєстрація/авторизація користувача» на чотири взаємозалежних процесу:

- введення логіну та паролю;
- пошук співпадінь в БД;
- реєстрація нового користувача;
- внесення інформації до БД.

Для процесу «Введення логіну та паролю» були визначені наступні дані:

- Вхідні дані: персональні дані користувача.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.

- Вихідні дані: запит до БД.

Для процесу «Пошук співпадінь в БД» були визначені наступні дані:

- Вхідні дані: запит до БД.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.

- Вихідні дані: відсутність співпадінь, авторизований користувач.

Для процесу «Реєстрація нового користувача» були визначені наступні дані:

- Вхідні дані: відсутність співпадінь, персональні дані користувача.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.

- Вихідні дані: дані нового користувача.

Для процесу «Внесення інформації до БД» були визначені наступні дані:

- Вхідні дані: дані нового користувача.
- Управління: інструкція експлуатації платформи.
- Механізми: Інтернет-платформа, технічне забезпечення, програмне забезпечення.

- Вихідні дані: авторизований користувач.

На рис. 3.7 представлена діаграма декомпозиції підпроцесу «Реєстрація/авторизація користувача».

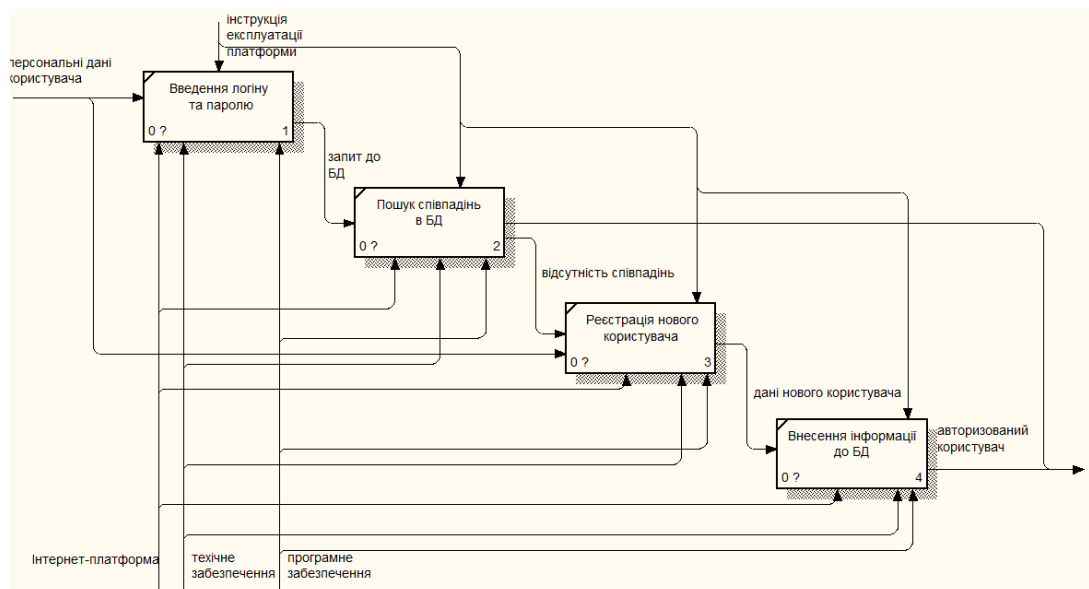


Рисунок 3.7 – Декомпозиція підпроцесу «Реєстрація/авторизація користувача»

3.2 Діаграма Use Case

Наступним етапом проектування бізнес-процесу «Інтернет-платформа «Саллетрон»» була розробка діаграми Use Case.

Діаграма Use Case або діаграма варіантів використання – діаграма, що відображає функціональні можливості система в ході її експлуатації. Під час розробки даної діаграми зазначаються всі можливі актори, які можуть взаємодіяти з системою. До того ж акторами можуть бути як фізичні особи, так і механізми [35].

Використання діаграми варіантів використання при процесі проектування та реалізації проекту має свої переваги:

- визначені актори системи;
- зручний варіант спілкування між розробником та замовником;
- початковий шаблон для проведення тестування [36].

Для проекту «Інтернет-платформа «Саллетрон»» були визначені варіанти використання системи з боку користувача:

- виконати логінізацію;
- виконати авторизацію;
- переглянути товари;
- виконати фільтрацію товарів;
- виконати пошук товару;
- додати товар до кошику;
- поділитися оголошенням на товар.

До того, ж при розробці були визначені залежності варіантів використання між собою. Наприклад, лише при виконанні авторизації або логінізації на платформу користувач має можливість виконувати додавання товару до кошику або ділитися оголошенням у соц.мережі. Також лише при авторизації або

логізації до системи користувач має доступ до додавання оголошення на власний товар.

На рис. 3.8 представлена діаграма варіантів використання проекту «Інтернет-платформа «Саллетрон»».

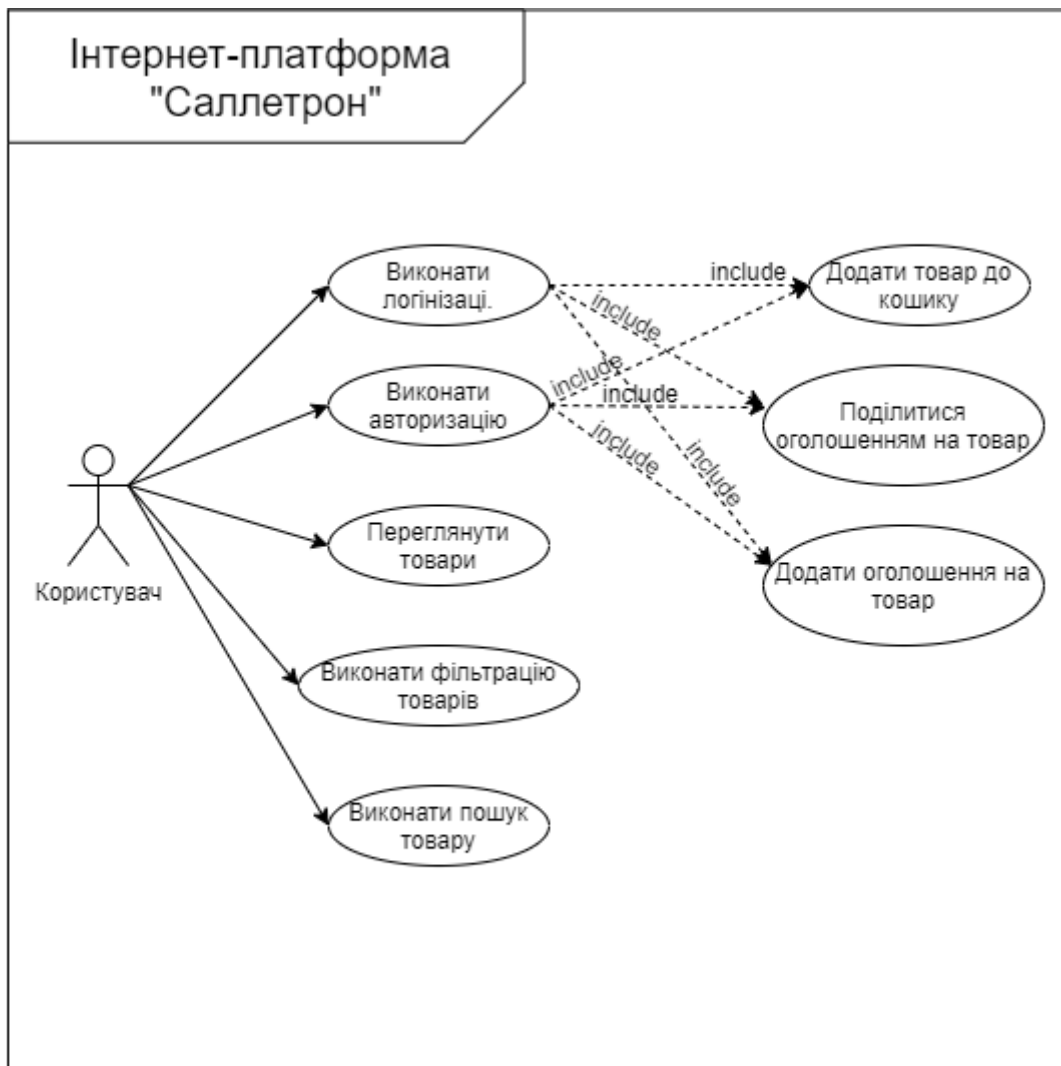


Рисунок 3.8 – Діаграма Use Case

4 РЕАЛІЗАЦІЯ ІНТЕРНЕТ-ПЛАТФОРМИ «САЛЛЕТРОН»

4.1 Створення макету платформи

Провівши детальний процес проектування Інтернет-платформи «Саллетрон», був сформований детальний план процесу реалізації проекту.

Першим етапом реалізації Інтернет-платформи «Саллетрон» є підготовка растрових та векторних зображень. На даному етапі робіт необхідно було виконати розробку логотипу Інтернет-платформи, виконати пост обробку растрових зображень для демонстрації товарів платформи, виконати пошук векторних зображень для елементів навігації платформи.

Для реалізації даного процесу був використаний функціонал програмного забезпечення Adobe Photoshop. На рис. 4.1 представлений процес розробки логотипу платформи.

Далі була проведена пост обробка зображень для тестових варіантів представлення товарів. В даному випадку було прийнято рішення про використання товарів різноманітних груп побутового вжитку: книги, машини, побутова техніка.

Оскільки Інтернет-платформа «Саллетрон» буде перш за все розроблена для покращення продажу та купівлі товарів повсякденного вжитку, але як можливість подальшого розвитку експлуатації платформи, існує варіант розширення функціоналу та політики продажів на ресурсі. На рис. 4.2 представлений процес обробки графічного зображення, що буде використано для тестового представлення товару.

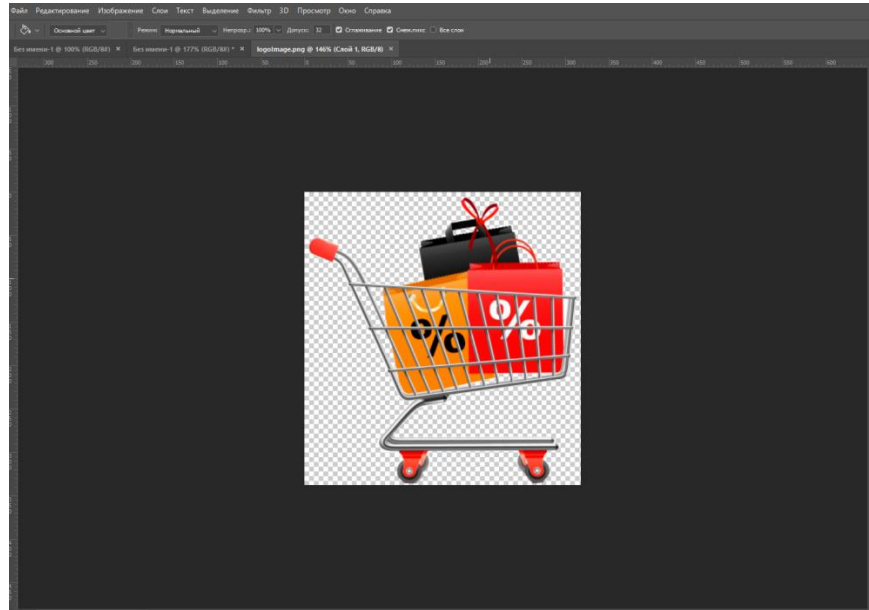


Рисунок 4.1 – Розробка логотипу платформи



Рисунок 4.2 – Пост обробка графічного зображення

Далі було проведено пошук векторних зображень для навігаційних елементів платформи, які будуть уніфікованими з існуючими розробками, а саме:

- виконати пошук іконки «кошик»;
- виконати пошук іконки «оновлення»;
- виконати пошук іконки «пошук»;
- виконати пошук іконки «обране»;
- виконати пошук іконки «поширити».

Оскільки знайдені матеріали не потребували пост обробки засобами програмного забезпечення Adobe Illustrator, було прийнято рішення про використання лише функціоналу програми Adobe Photoshop. Також у зв'язку з тим, що Adobe Photoshop не підтримує функціонал для роботи з векторними зображення у форматі SVG, було прийнято рішення про пошук веб-ресурсу, де користувач може обрати бажану іконку та завантажити її у форматі png.

На даний час найбільш популярними веб-ресурсами для пошуку іконок є The Noun Project та Flaticon. Кожний із цих ресурсів є англomовними, тому інколи виникають складності у пошуку бажаної іконки через наявність мовного бар'єру. До того ж, для завантаження бажаного зображення користувачу необхідно пройти реєстрацію на цих ресурсах. До того ж, мінусом Flaticon є те, що 50% зображень є платними при відсутності преміум-аккаунту у користувача. У той час, коли The Noun Project на 100% містить лише безкоштовні зображення. На рис. 4.3 представлений пошук на ресурсі Flaticon за пошуком «гео» для демонстрації співвідношення безкоштовного та платного контенту.

Проте The Noun Project має також суттєвий недолік, а саме те, що всі зображення представлені у чорно білій версії іконок. Натомість ресурс Flaticon надає можливість користувачу обрати іконку у контурному представленні або у двох-трьох варіантах забарвлення. На рис. 4.4 представлений пошук іконки «гео» на ресурсі The Noun Project. На рис. 4.5 представлений процес пошуку іконки «гео» для демонстрації різнобарв'я зображень. На рис. 4.6 представлений процес вибору

різного формату представлення зображення на ресурсі Flaticon.

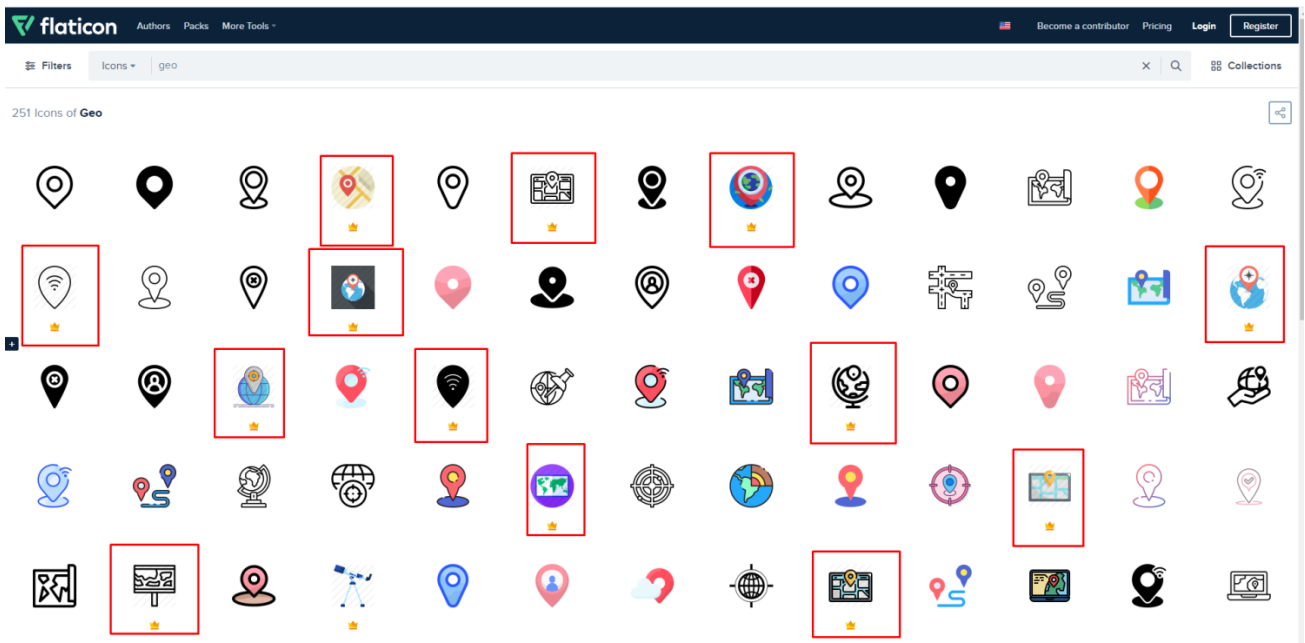


Рисунок 4.3 – Пошук на ресурсі Flaticon за пошуком «geo» для демонстрації співвідношення безкоштовного та платного контенту

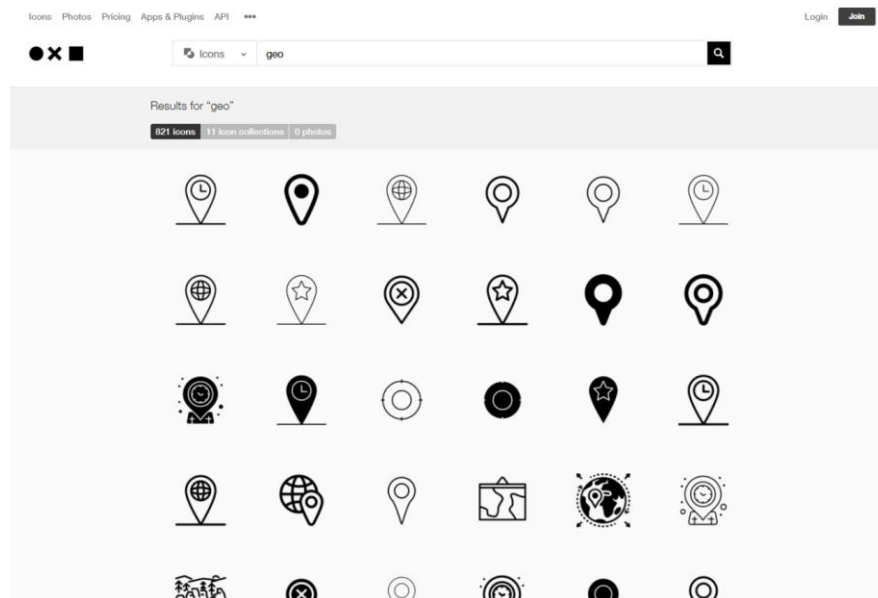


Рисунок 4.4 – Пошук іконки «geo» на ресурсі The Noun Project

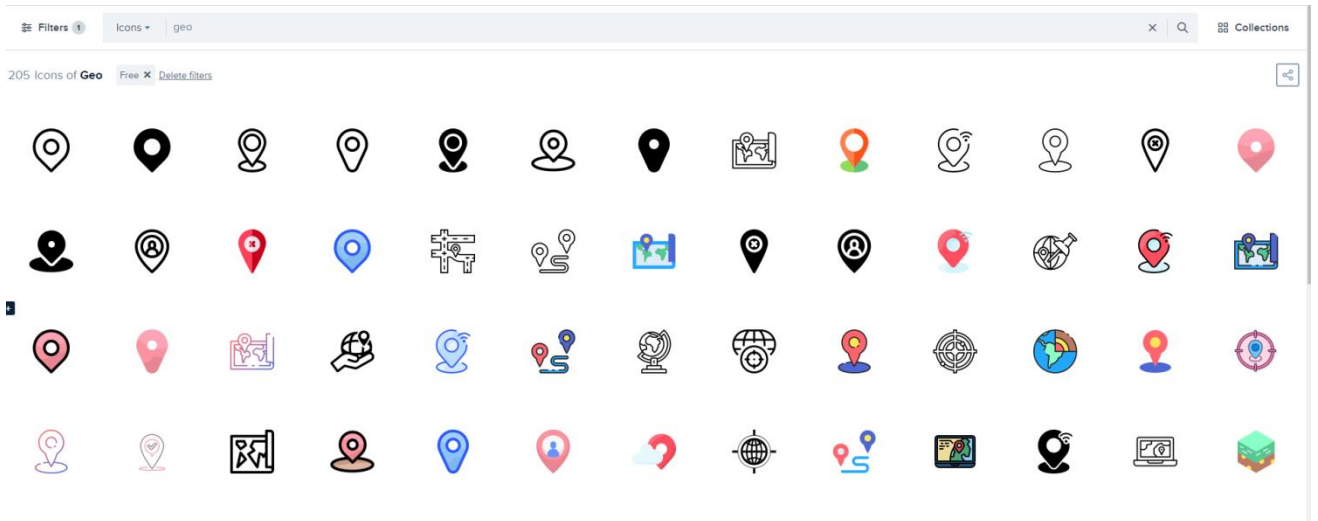


Рисунок 4.5 – Процес пошуку іконки «geo» на ресурсі Flaticon для демонстрації різнобарв'я зображень

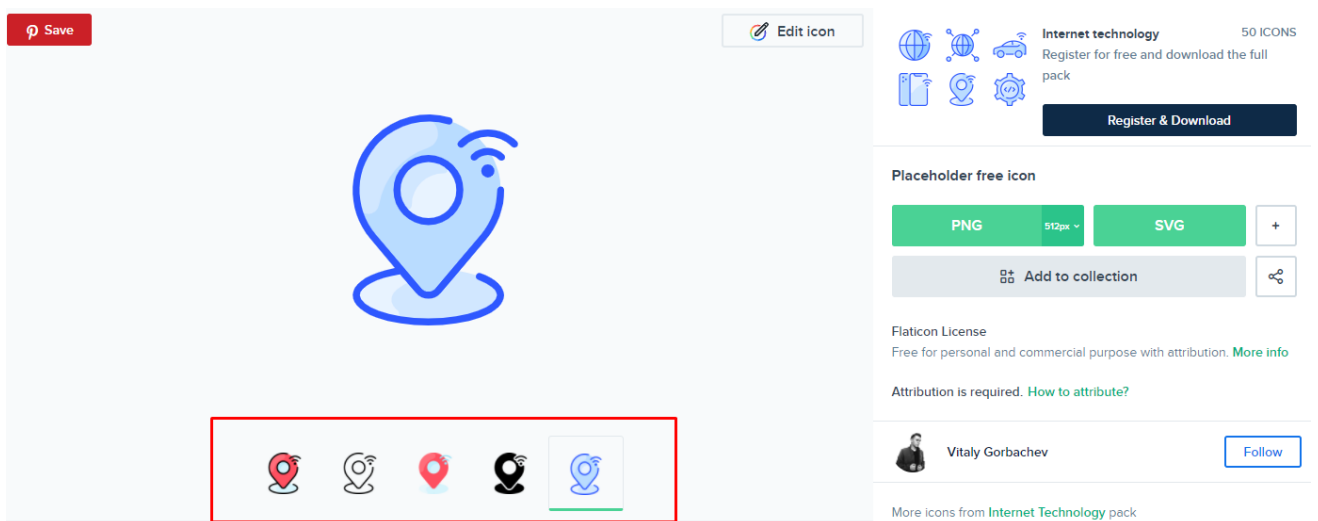


Рисунок 4.6 – Процес вибору різного формату представлення зображення на ресурсі Flaticon

Провівши аналіз обох ресурсів, було отримано результати, що описані нижче:

- The Noun Project містить лише безкоштовні зображення;
- Flaticon містить як безкоштовний, так і платний контент;
- The Noun Project пропонує для користувача лише один варіант

представлення, чорно білий;

- Flaticon пропонує користувачу як чорно білі версії зображень, так і різнокольорові;
- Flaticon надає можливість користувачу обрати бажане представлення обраного зображення;
- The Noun Project та Flaticon однаково надають можливість завантаження зображення у форматі SVG та PNG.

На основі отриманих зображень було прийнято рішення про вибір ресурсу Flaticon для пошуку та завантаження необхідних іконок для платформи. Також було прийнято рішення про застосування функціоналу програмного забезпечення Adobe Photoshop для зміни кольору іконки, у разі не виявлення іконки бажаного забарвлення на ресурсі. На рис. 4.7 представлений процес зміни забарвлення іконки, завантаженої з ресурсу Flaticon.

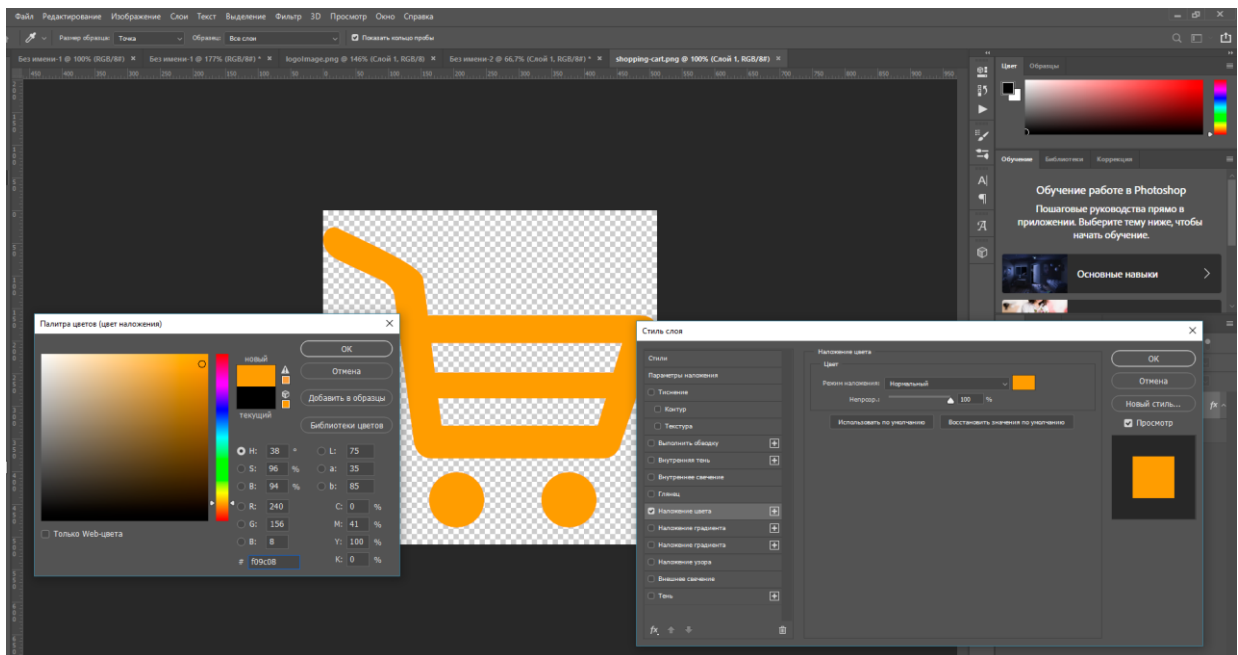


Рисунок 4.7 – Процес зміни забарвлення іконки,
завантаженої з ресурсу Flaticon

Виконавши розробку логотипу, провівши пост обробку фотографій, виконавши пошук зручного ресурсу для пошуку необхідних іконок та провівши редагування іконок під стиль розробленого логотипу, розробник проекту перейшов до наступного етапу робіт – макетування платформи.

На даному етапі необхідно було визначити структуру платформи, визначити основні та акцентуючі кольори, визначити основний шрифт платформи та розробити макет платформи з урахуванням всіх попередніх результатів підготовчого процесу.

Перш за все було прийнято рішення про розробку Інтернет-платформи у форматі landing page для уникнення навантаження на сервер та для комфортної експлуатації ресурсу користувача з будь-якого виду пристрою.

Далі був проведений аналітичний пошук та прийнято рішення про кількість та розташування основних блоків платформи. На рис. 4.8 представлена розроблена схема структурних блоків Інтернет-платформи «Саллетрон».

На основі розробленої структури платформи розробником проекту було проведено чорновий варіант макетування у вигляді представлення основних блоків та їх розміщення. На рис. 4.9 представлений чорновий варіант макетування.

Далі опираючись на попередні проміжні результати, необхідно було обрати основні кольори платформи та одночасно обрати акцентуючі кольори, які б гармонічно поєднувались та мінімізували появи когнітивного дисонансу у користувача під час процесу використання платформи.

Основним кольором платформи було обрано відтінок за кодуванням #1e2939, що представлений на рис. 4.10. У якості акцентуючого кольору було обрано відтінок #ff6600, що представлений на рис. 4.11.

Після того, як було обрано основну колористику платформи, необхідно було прийняти рішення стосовно вибору основного шрифту платформи. В даному випадку аналіз шрифтів базувався на основному критерію коректності відображення шрифту на всіх можливих сучасних браузерях. В результаті аналізу було обрано шрифт Montserrat з типами відображення, що наведені нижче:

- Bold;
- Regular;
- SemiBold.

Після проектування розміщення основних блоків ресурсу, визначення колористики платформи та шрифту, розробник проекту перейшов до макетування з додаванням контенту та всіх допоміжних графічних елементів. На рис. 4.12 представлений результат макетування.

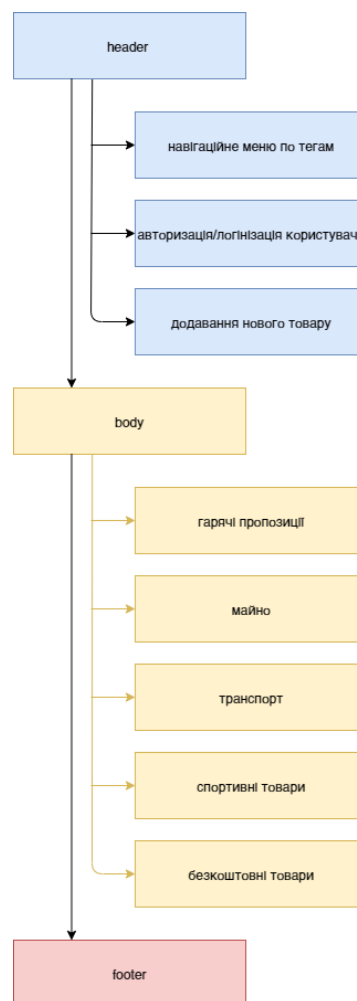


Рисунок 4.8 – Розроблена схема структурних блоків Інтернет-платформи «Саллетрон»

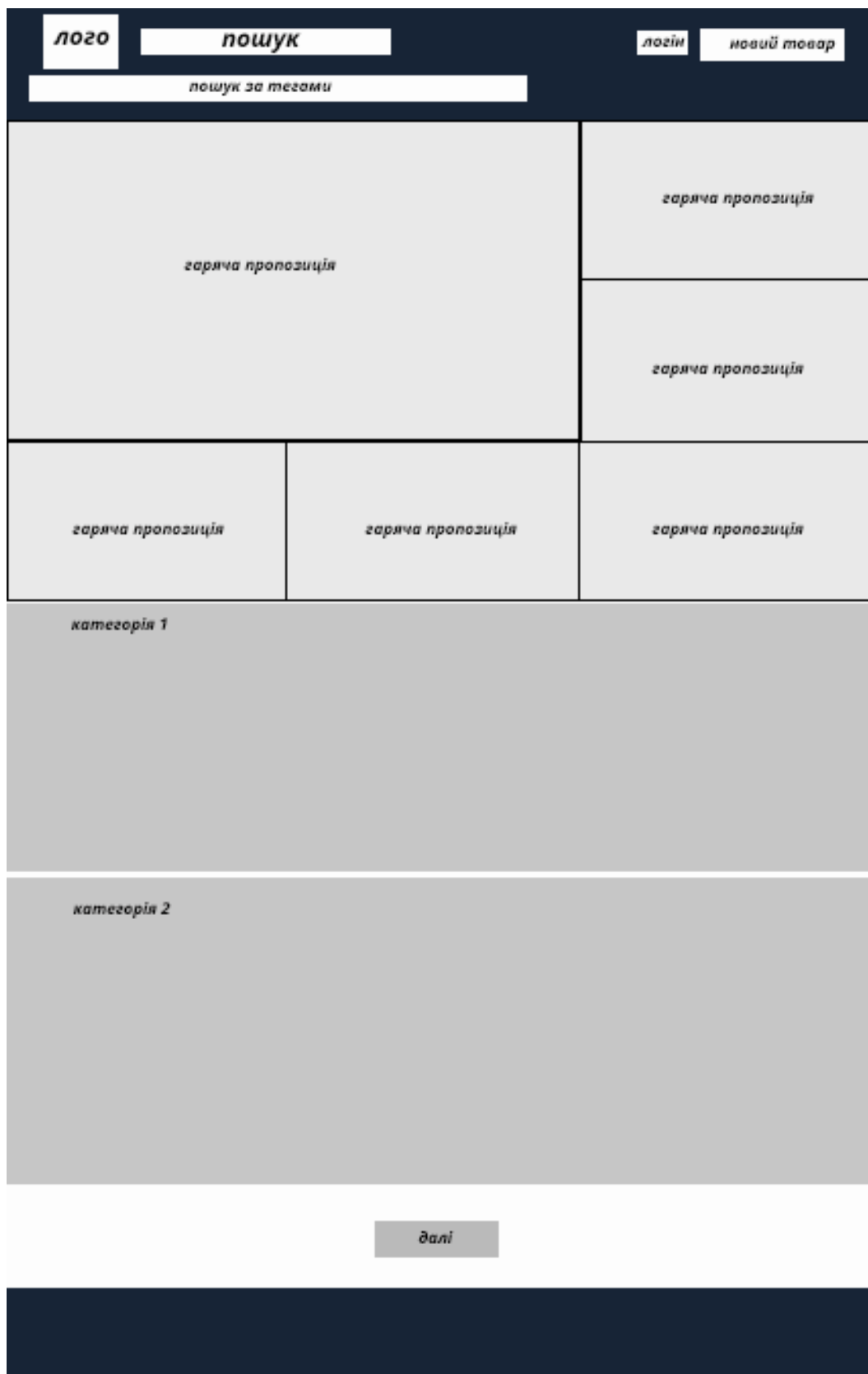


Рисунок 4.9 – Чорновий варіант макетування платформи

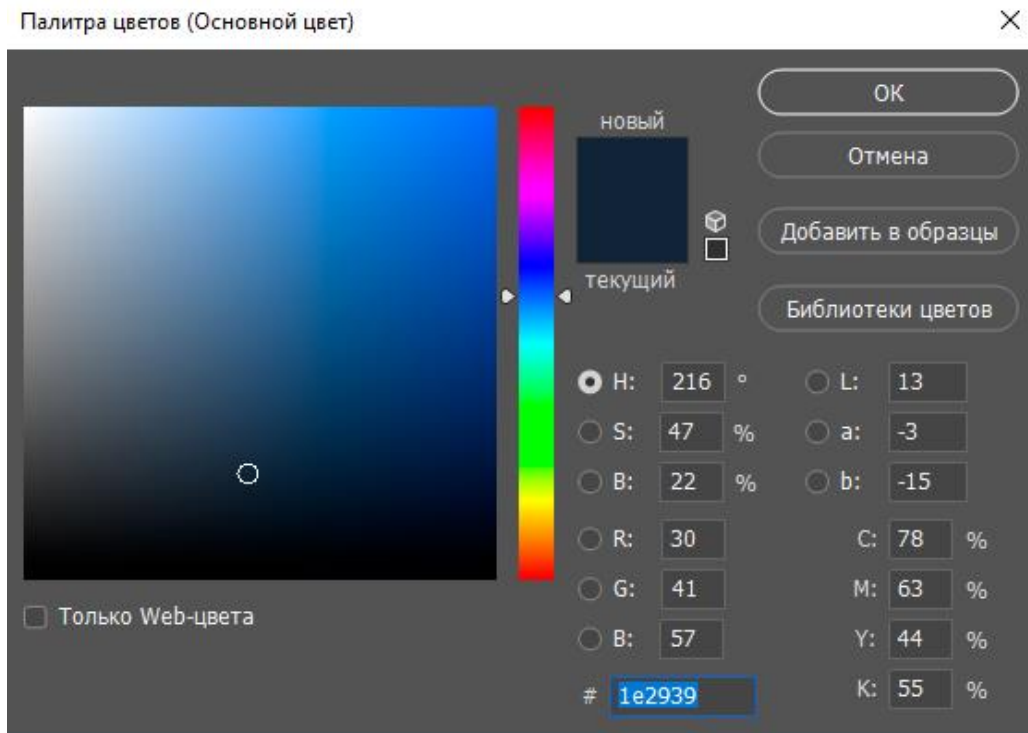


Рисунок 4.10 – Основной колір платформи

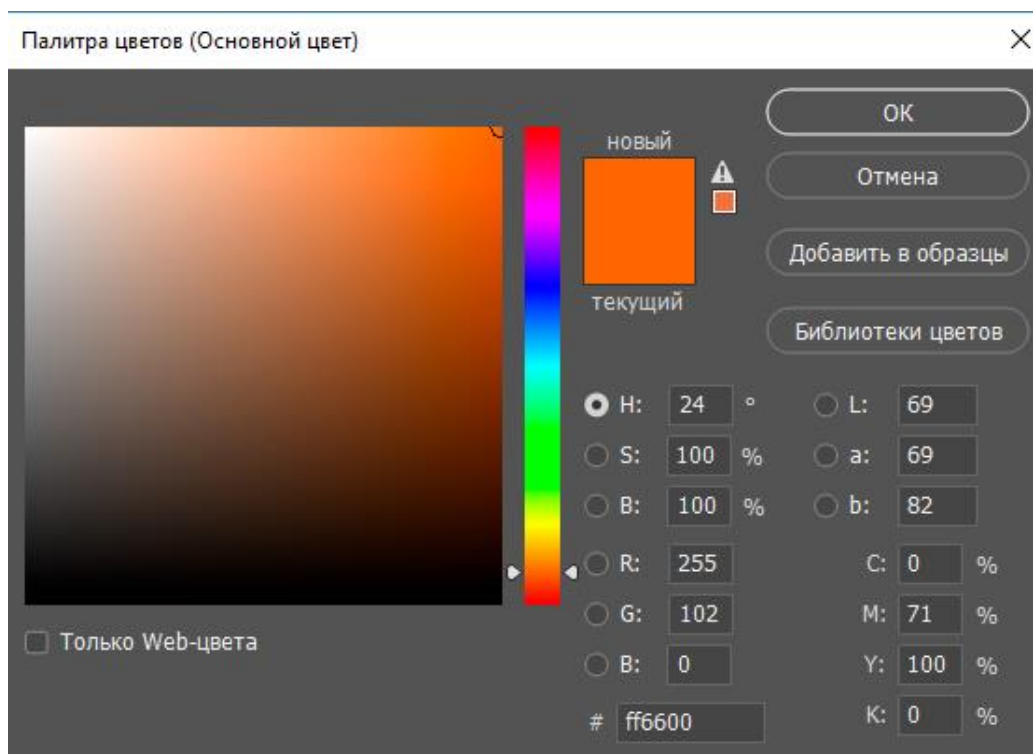


Рисунок 4.11 – Акцентующий колір платформи

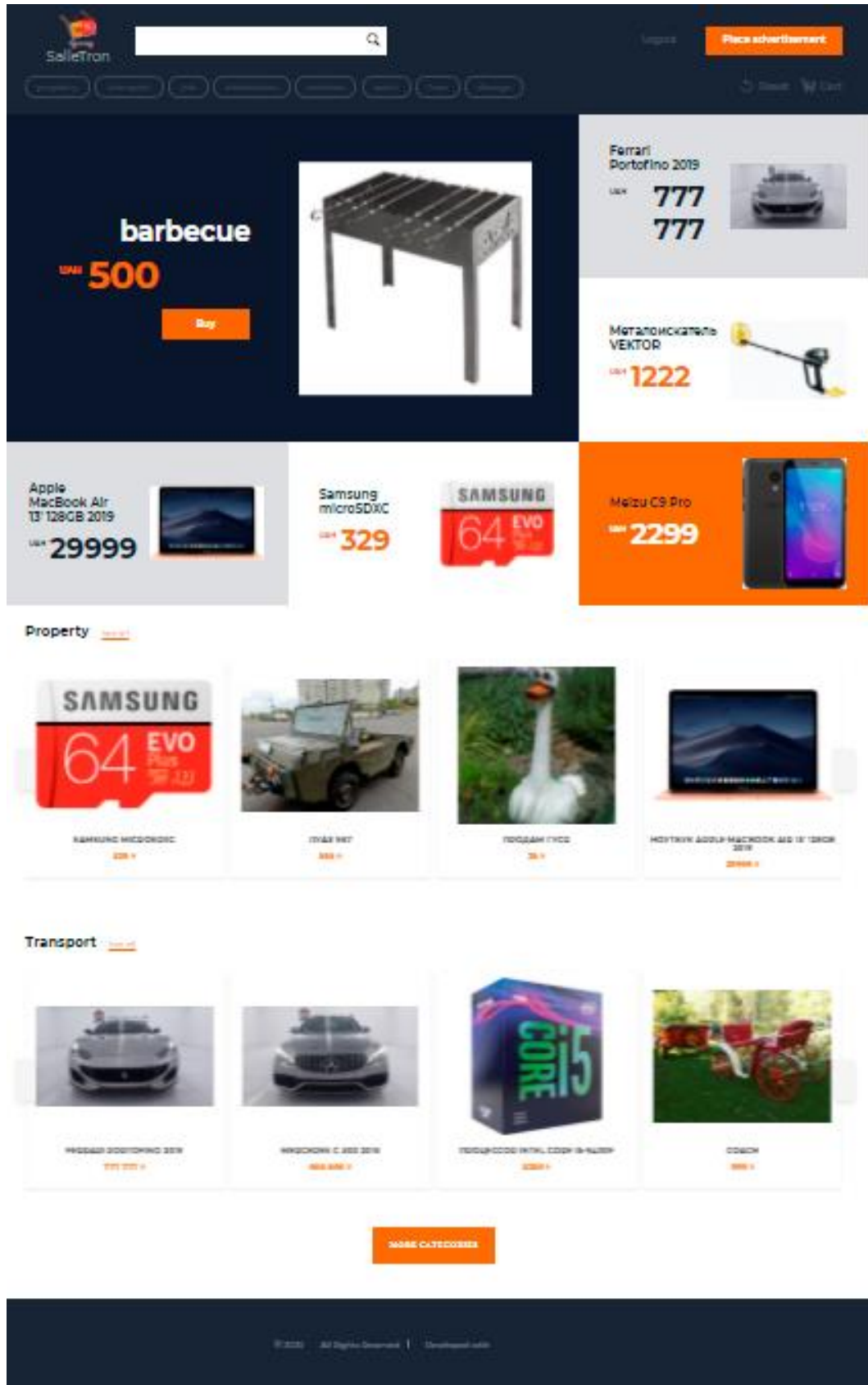


Рисунок 4.12 – Макет платформы

4.2 Реалізація платформи

Згідно з матеріалами Розділу 3, рис. 3.4, встановлено, що платформа буде розроблена в 4 головних крока.

Підготовчим кроком до безпосередньо реалізації платформи є структуризація проекту.

На цьому етапі були сформовані та реалізовані наступні підзадачі:

- створення ієрархічного дерева, з основним функціями;
- підключення веб пака;
- підключення до сховища даних;
- розробка каркасу платформи;
- створення файлу `index.js` який містить всі необхідні імпортовані бібліотеки та майбутнього розроблені компоненти для коректного та повноцінного функціонування.

На рисунку 4.13 наведено структура проекту, яка відображає ієрархічне дерево проекту.

На етапі підключення `webpack` (рис.4.14) була виконана інтеграція функціоналу `webpack` до існуючого проекту «Саллетрон».

Далі необхідно було розробити файл «`config.js`» для встановлення взаємодії між файлами проекту та файлами `webpack` (рис.4.15).

Наступними кроками було створення файл `api.js` (рис.4.16), який виконує функцію підключення до сховища даних `Firebase` (рис.4.17)

Файл `api.js` виконує ще функцію серверу, отримання даних та надання необхідної інформації та контенту в залежності від дій користувача з платформою.

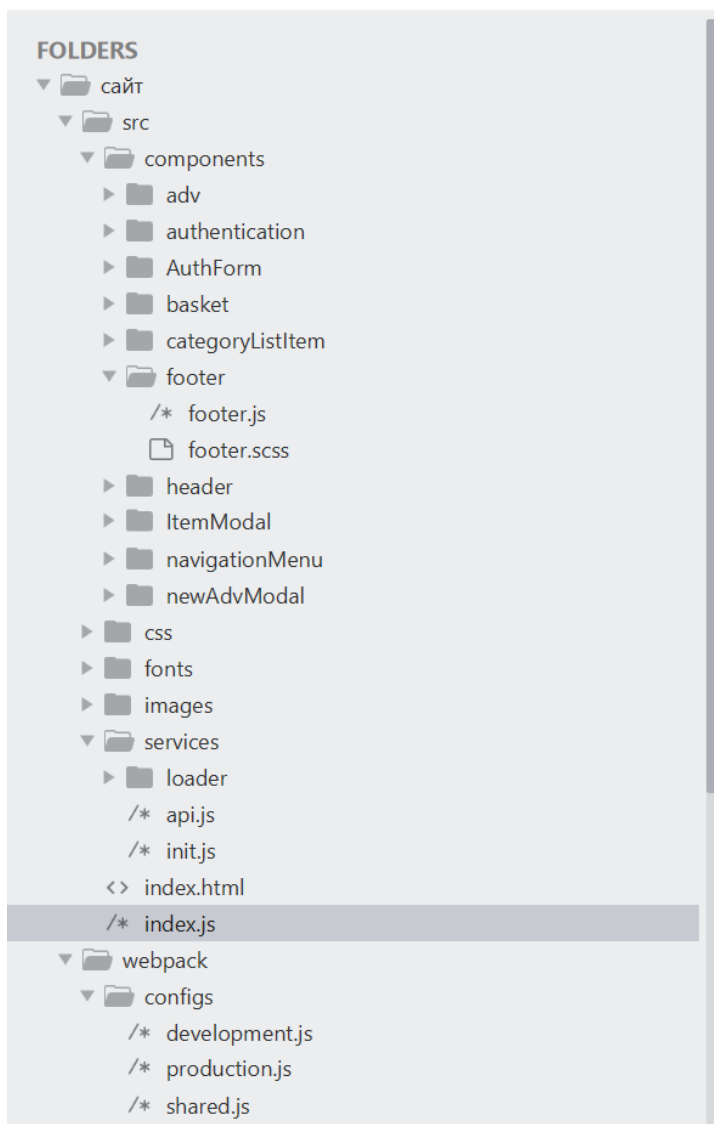


Рисунок 4.13 – Структура проекту

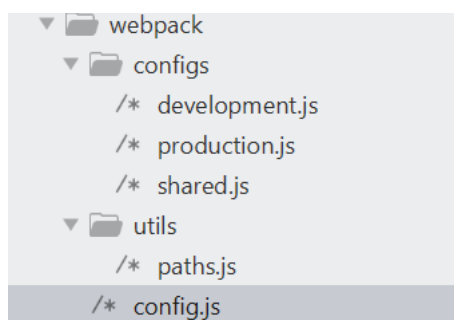


Рисунок 4.14 – Підключення webpack

```

config.js
const webpackMerge = require('webpack-merge');
const loadSharedConfig = require('./configs/shared');

const loadModeConfig = env => require(`./configs/${env.mode}`)(env);

module.exports = env =>
  webpackMerge(loadSharedConfig(env), loadModeConfig(env));

```

Рисунок 4.15 – Головний код запуску webpack

```

api.js
import axios from 'axios';

axios.defaults.baseURL = 'https://saletrontest.firebaseio.com';

const apiKey = 'AIzaSyDGixM-3-XeusqOkskulTpiQQP2BLo-jPs';

const convertData = data => {
  const [values] = Object.values(data.data);
  return values;
};

const transformedData = categories => {
  const arr = [];
  const values = Object.values(categories);
  for (let value of values) {
    const keys = Object.keys(value);
    for (let key of keys) {
      arr.push({
        id: key,
        ...value[key],
      });
    }
  }
  return arr;
};

```

Рисунок 4.16 – Створення файл арі

```
axios.defaults.baseURL = 'https://saletrontest.firebaseio.com';
```

Рисунок 4.17 – Підключення firebase

На рис.4.18 надано каркас інтернет платформи, який був реалізований мовою програмування HTML. При розробці каркасу було визначено розташування основних функціональних блоків, кожний з яких має унікальний ідентифікатор. За рахунок наявності ідентифікаторів розробником проекту буде реалізоване динамічне підключення функціональних модулів платформ.

Останнім етапом структуризації проекту була реалізація файлу «index.js» (рис.4.19).

```
index.html x
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <title>SalleTRON</title>
</head>

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <link rel="shortcut icon" type="image/png" href="./images/logoImage.png">
  <title>SalleTRON</title>
</head>

<body>
  <!-- HEADER SECTION -->
  <header class="header">
    <nav class="header__nav container">
      <!-- burger -->
      <div class="header__burger">
        <button class="hamburger hamburger--emphatic id="burgerButton" type="button">
          <span class="hamburger-box">
            <span class="hamburger-inner"></span>
          </span>
        </button>
      </div>
      <div class="header__burgerMenu js-tabletBurgerMenu ">
      </div>
      <!-- Logo -->
      <a href="#" class="header__logo">
        
        <p class="header__logo-text">SalleTron</p>
      </a>
      <!-- search -->
      <input class="header__search" type="input" name="headerInput">
      <!-- Login -->
      <form class="header__form">
        <!-- <button class="header__form-register" type="button" name="register">Registration</button> -->
        <button class="header__form-login" type="button" name="login">Login</button>
        <button class="header__form-logout" type="button" name="login">Logout</button>
      </form>
    </nav>
  </header>

```

Рисунок 4.18 – Каркас платформи


```
index.js x
import './services/loader/loader.scss';
import './css/styles.scss';
import './services/api';
import './components/newAdvModal/newAdvModal';
import './components/header/header';
import './components/navigationMenu/navigationMenu';
import './components/authentication/authServices';
import './components/AuthForm/authOpenModal';
import './components/basket/basket';
import './components/adv/js/adv';
import './components/categoryListItem/renderCategoryList';
import './components/categoryListItem/categoryListItem';
import './components/ItemModal/ItemModal';
import './components/footer/footer';
import './services/init';
```

Рисунок 4.19 – index.js

Проект «Інтернет-платформа «Саллетрон»» реалізовано за принципами модульного програмування. Функціональна частина проекту розроблена у вигляді набору незалежних модулів.

Розглянемо реалізацію модуля «AuthForm» (логізація/авторизація).

Процес розробки модуля складається з чотирьох складових:

- формування каркасу форми авторизації (рис.4.20);
- розробка логіки роботи процесу логізації/авторизації користувача (рис.4.21);
- розробка каркасу модульного вікна (рис.4.22);
- відображення модульного вікна під форму (рис.4.23).

Усі інші модулі та їх програмний код надано у Додатку Б.

```

authForm.js x
const getAuthForm = () => {
  return `
    <form class="js-form" name="authForm">
      <input
        name="email"
        type="text"
        class="emailInput input"
        placeholder="E-mail"
        required
      />
      <p class="errorEmail error"></p>
      <input
        name="password"
        type="password"
        class="passwordInput input"
        placeholder="Password"
        required
      />
      <p class="errorPassword error"></p>
      <div class="buttons">
        <button class="login btn">Log in</button>
        <button class="registration btn">New account</button>
      </div>
    </form>
  `;
};

export default getAuthForm;

```

Рисунок 4.20 – Каркас форми авторизації

```

authFormLogic.js x
const authFormLogic = () => {
  const checkForm = document.querySelector('.js-form');
  checkForm.addEventListener('submit', checkValue);
  function checkValue(e) {
    e.preventDefault();
    const errorEmail = document.querySelector('.errorEmail');
    const errorPassword = document.querySelector('.errorPassword');

    errorEmail.textContent = '';
    errorPassword.textContent = '';

    const valueEmail = e.currentTarget.elements.email.value;
    if (!(valueEmail.includes('@') && valueEmail.includes('.'))) {
      errorEmail.textContent = 'Incorrect e-mail!';
      return;
    }

    if (valueEmail.length < 5) {
      errorEmail.textContent = 'Incorrect e-mail!';
      return;
    }

    const valuePassword = e.currentTarget.elements.password.value;

    if (valuePassword.length < 6) {
      errorPassword.textContent = 'Password must be more than 5 symbols!';
    }
    e.currentTarget.reset();

    if (
      valueEmail.includes('@') &&
      valueEmail.includes('.') &&
      valueEmail.length >= 5 &&
      valuePassword.length >= 6
    ) {
      errorEmail.textContent = '';
      errorPassword.textContent = '';
    }
  }
};

export default authFormLogic;

```

Рисунок 4.21 – Логіка роботи форми

```

forOpenModal.js x
import './stylesAuth.scss';
import getAuthForm from './authForm';

const openModal = () => {
  return `
<div class="modalWindow">
  <button class="closeModalWindow">
    <svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
    viewBox="0 0 492 492" style="enable-background:new 0 0 492 492;" xml:space="preserve">
      <g>
      <g>
      <path d="M300.188,246L484.14,62.04c5.06-5.06,7.852-11.82,7.86-19.024c0-7.208-2.792-13.972-7.86-19.028L468.02,7.872
      c-5.068-5.076-11.824-7.856-19.036-7.856c-7.2,0-13.956,2.78-19.024,7.856L246.008,191.82L62.048,7.872
      c-5.06-5.076-11.82-7.856-19.028-7.856c-7.2,0-13.96,2.78-19.02,7.856L7.872,23.988c-10.496,10.496-10.496,27.568,0,38.052
      L191.828,246L7.872,429.952c-5.064,5.072-7.852,11.828-7.852,19.032c0,7.204,2.788,13.96,7.852,19.028L116.124,16.116
      c5.06,5.072,11.824,7.856,19.02,7.856c7.208,0,13.968-2.784,19.028-7.856L183.96-183.952L183.952,183.952
      c5.068,5.072,11.824,7.856,19.024,7.856h0.008c7.204,0,13.96-2.784,19.028-7.856L116.12-16.116
      c5.06-5.064,7.852-11.824,7.852-19.028c0-7.204-2.792-13.96-7.852-19.028L300.188,246z"/>
      </g>
    </svg>
  </button>
  <p class="textGoogle text">
    For authorization you can use the Google or Facebook Account:
  </p>
  <div class="buttonsAuth">
    <button class="googleBtn authBtn" id="google">
      <svg class="googleSvg" enable-background="new 0 0 512 512" height="20" viewBox="0 0 512 512"
      width="20" xmlns="http://www.w3.org/2000/svg"><g><path d="m120 256c0-25.367 6.989-49.13 19.131-69.477v-86.308h-86.308c-34.255
      <span class="spanAuth">Google</span>
    </button>
    <button class="facebookBtn authBtn" id="facebook">
      <svg class="facebookSvg" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 48 48" width="25px" height="25px"><linearGradient id=
      <span class="spanAuth">Facebook</span>
    </button>
  </div>
  <p class="textAuth text">Or enter with e-mail and password:</p>
  <div>
    <input type="text" value="" />
    <input type="password" value="" />
    <input type="button" value="Login" />
  </div>
  </div>`;
};

export default openModal;

```

Рисунок 4.22 – Каркас модульного вікна

```

authOpenModal.js x
import './stylesAuth.scss';
import openModal from './forOpenModal';
import authFormLogic from './authFormLogic';
import '../../../../node_modules/basiclightbox/dist/basiclightbox.min.css';
import * as basiclightbox from 'basiclightbox';
import { addButtonListeners, closeWindowOnLogin } from '../authentication/refs';

function showModal() {
  const instance = basiclightbox.create(
    `<div class="modalAuthForm">${openModal()}</div>`,
  );

  instance.show();
  authFormLogic();
  addButtonListeners();
  closeWindowOnLogin(instance)
  document
    .querySelector('.closeModalWindow')
    .addEventListener('click', instance.close);
}

const loginModal = document.querySelector('.header__form-login');
loginModal.addEventListener('click', showModal);
// console.log(registrationModal);

export default showModal;

```


Рисунок 4.23 – Відображення модульного вікна під форму

Далі була проведена відладка модулів. Цей крок розглядаємо тільки як процес пошуку і виправлення помилок (без тестування) у проекті.


За допомогою функціоналу webpack зроблено злиття функціональних модулів в єдиний проект, та виконано процес відлагодження помилок у разі виявлення конфліктів між змінними та назв функціональних методів.


Процес тестування проекту «Інтернет-платформа «Саллетрон»» виконувався з урахуванням варіантів ресурсу, що були описані в Розділі 3.

Виконано перевірку форми процесу логізації/авторизації користувача. Це надасть впевненість в тому, що сховище даних коректно працює на запити користувача. На рис. рисунок 4.24 представлено процес авторизації нового користувача. Проміжний крок – валідація даних користувача (рис. 4.25). На рис. 4.26 підтвердження коректного попереднього процесу.



For authorization you can use the Google or Facebook Account:

 Google

 Facebook

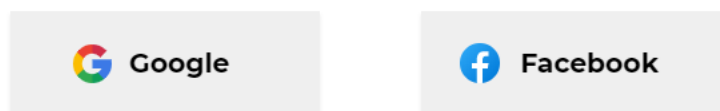
Or enter with e-mail and password:

LOG IN

NEW ACCOUNT

Рисунок 4.24 – Процес авторизації нового користувача

For authorization you can use the Google or Facebook Account:



Or enter with e-mail and password:

Password must be more than 5 symbols!

Рисунок 4.25 – Валідація даних користувача

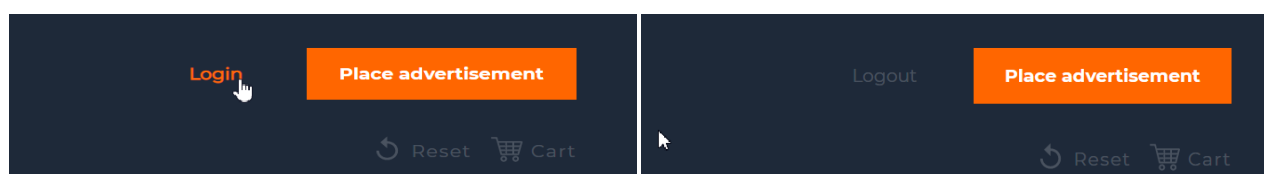
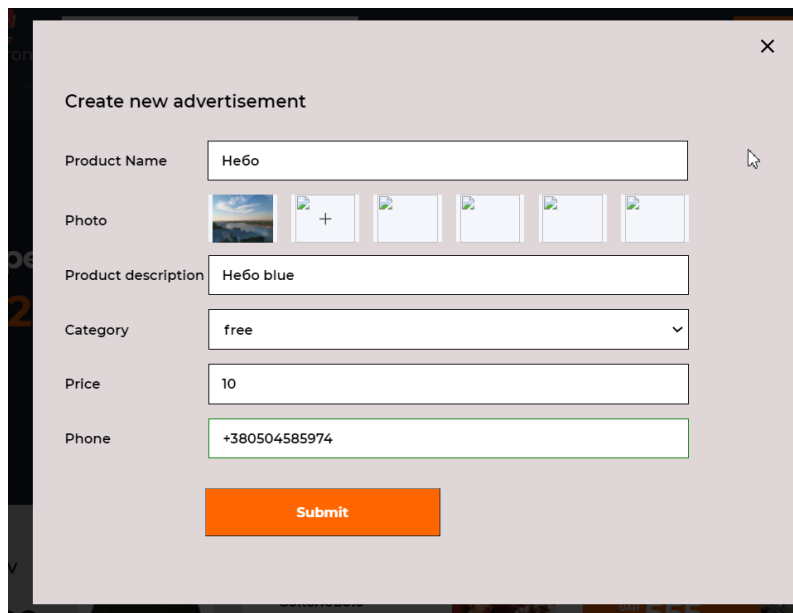


Рисунок 4.26 – Підтвердження реєстрації

Однім з варіантів використання є можливість додавання товару зареєстрованим користувачам (рис.4.27 – 4.28).

Також була проведена перевірка адаптивного відображення інтернет-платформи на мобільному пристрою . На рис. 4.29 представлено стартову сторінку Інтернет-платформи на мобільному пристрою. З урахуванням адаптації каракасу,

розробником проекту було реалізоване бокове меню, що виконує навігаційний інструмент для користувача та містить форму авторизації/логінізації (рис. 4.30). На рис. 4.31 продемонстрована адаптивна карточка товару.



The screenshot shows a form titled "Create new advertisement" with the following fields:

- Product Name: HeBo
- Photo: A gallery with one image selected and four empty slots.
- Product description: HeBo blue
- Category: free (dropdown menu)
- Price: 10
- Phone: +380504585974

A red "Submit" button is located at the bottom of the form.

Рисунок 4.27 – Додавання нового товару

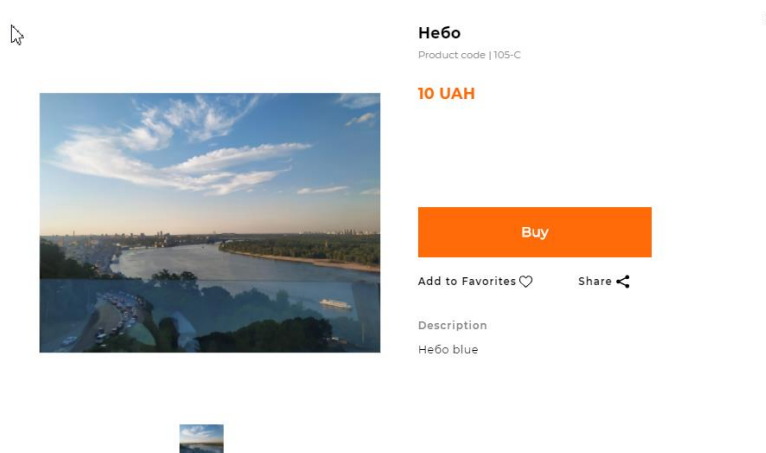


Рисунок 4.28 – Перевірка наявності нового товару

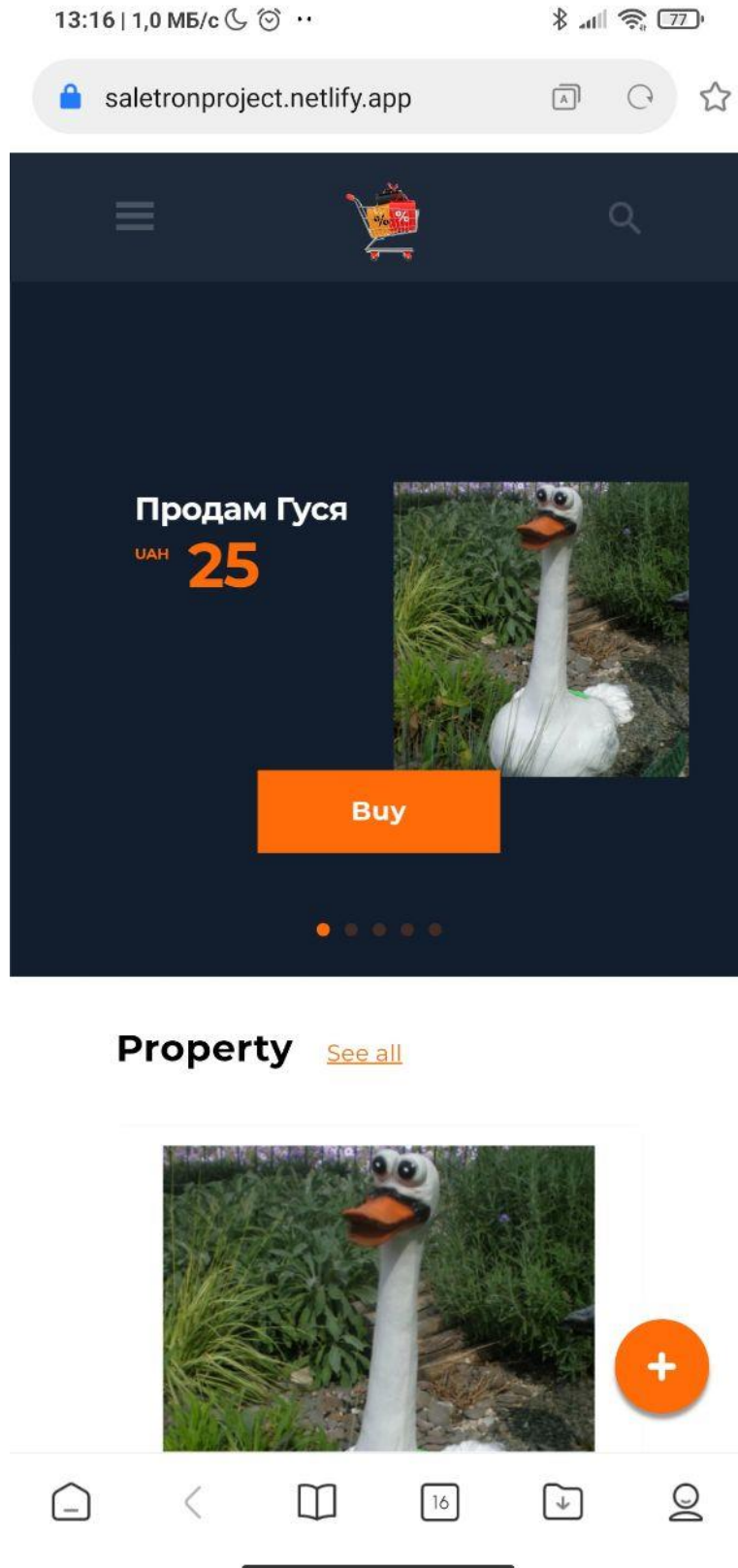


Рисунок 4.29 – Стартова сторінка Інтернет-платформи на мобільному пристрою

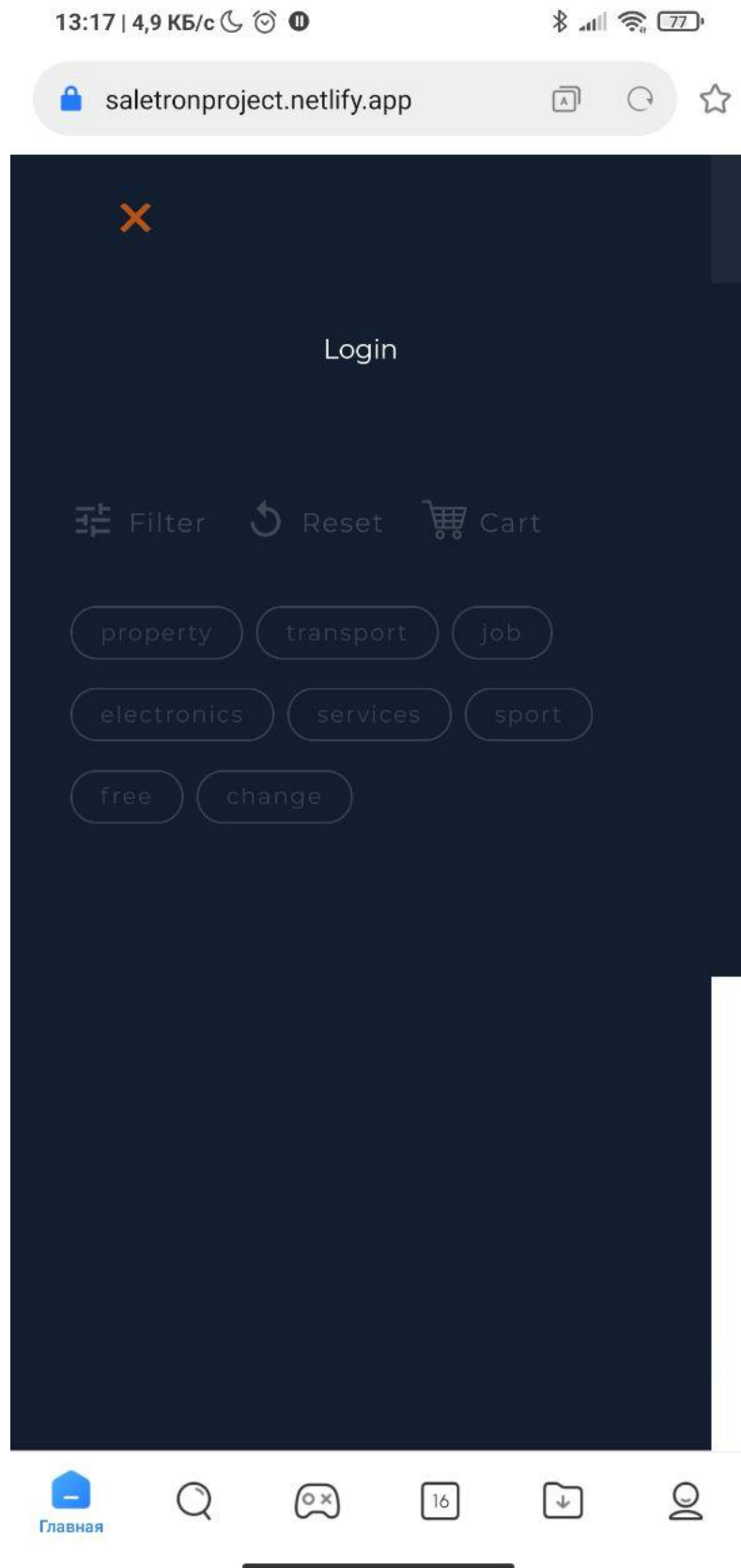


Рисунок 4.30 – Боковое меню у вигляді з мобільного пристрою

13:17 | 0,4 КБ/с

Bluetooth, Cellular, Wi-Fi, 77%

25 UAH



Buy

Add to Favorites

Share

Description

Качество гарантируем!

Главная



Рисунок 4.31 – Адаптивна карточка товару

ВИСНОВКИ

В ході виконання дипломного проекту була розроблена Інтернет-платформа «Саллетрон» для реалізації процесу продажу-купівлі товарів побутового вжитку на території України.

Розроблена Інтернет-платформа «Саллетрон» відповідає всім поставленим вимогам:

- наявність форми логізації/авторизації користувача;
- можливість швидкого пошуку товару;
- наявність зручної системи фільтрації;
- наявність форми для додавання власної об'яви.

Даний проект буде нести соціально-економічну для тих користувачів, які прагнуть без мінімальних вкладів на страті розпочати власний бізнес та для тих користувачів, які прагнуть швидко та зручно віднайти потрібний товар.

В ході виконання дипломного проекту був проведений детальний аналіз питання необхідності розробки зручної Інтернет-платформи для користувачів території України. Також на даному етапі проекту був проведений аналіз існуючих та популярних Інтернет-платформ України. Даний аналіз показав, що більшість розглянутих платформ потребують у модернізації з точки зору дизайну та з точки зору функціональних можливостей з боку користувача.

Розробником проекту був виконаний етап визначення мети та задач проекту. На даному етапі для конкретизації мети та задач, необхідно було провести планування робіт. За рахунок розроблених діаграм структури та організації проекту, діаграми Ганта та PERT-мережі, розробник проекту отримав загальний об'єм проекту у часі та кількості задач з урахуванням додаткового часу на кожну підзадачу.

В ході виконання дипломного проекту були застосовані інструменти: Adobe Photoshop для розробки макету платформи, HTML для формування каркасу ресурсу, JavaScript для розробки функціональної частини платформи, Firebase для наявності сховища даних.

Після реалізації проекту було проведено детальне тестування платформи з урахуванням всіх функціональних можливостей сайту та з урахуванням наявності мобільної версії. Під час проведення тестування не було виявлено недоліків в функціональних блоках платформи. Тестування мобільної версії показало, що розробником проекту була коректно організована адаптація макету.

В результаті виконання всіх етапів дипломного проекту була розроблена Інтернет-платформа «Саллетрон», що підвищить показник Інтернет-купівлі/продажу на території України.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Application of Information Technology in Our Daily Life [Електронний ресурс] – режим доступу: <https://cutt.ly/WgR3Ajl>
2. Technology in Our Life Today and How It Has Changed [Електронний ресурс] – режим доступу: <https://cutt.ly/xgR3Jko>
3. 10 лучших торговых площадок в интернете [Електронний ресурс] – режим доступу: <https://cutt.ly/Qf3wqrD>
4. What is an “online platform”? [Електронний ресурс] – режим доступу: <https://cutt.ly/EgR3ZL4>
5. So What Is an Internet Platform? [Електронний ресурс] – режим доступу: <https://cutt.ly/WgR3BkR>
6. Торговые площадки Украины [Електронний ресурс] – режим доступу: <https://cutt.ly/uf3wePw>
7. Юзабилити интернет-магазина: актуальность и влияние на продажи — Lemarbet [Електронний ресурс] – режим доступу: <https://cutt.ly/Tf3wu2l>
8. Benefits of online platforms [Електронний ресурс] – режим доступу: <https://cutt.ly/QgR31gO>
9. Making sense of Internet of Things platforms [Електронний ресурс] – режим доступу: <https://cutt.ly/mgR320b>
10. Why You Should Sell Online: 7 Benefits Of Ecommerce Websites [Електронний ресурс] – режим доступу: <https://cutt.ly/wgR383W>
11. The Top Advantages of eCommerce: Why You Should Sell Online [Електронний ресурс] – режим доступу: <https://cutt.ly/NgR35AF>
12. 7 лучших площадок для продаж - Блог MSROP [Електронний ресурс] – режим доступу: <https://cutt.ly/hf3wo1e>

13. Top Ukraine Shopping Websites [Электронный ресурс] – режим доступа: <https://cutt.ly/KgR8ef5>
14. OLX — Википедия [Электронный ресурс] – режим доступа: <https://cutt.ly/of3wsuP>
15. Сервис объявлений OLX: сайт объявлений в Украине [Электронный ресурс] – режим доступа: <https://cutt.ly/VgR8yTT>
16. Бизнес на OLX: создайте ваш интернет-магазин на OLX [Электронный ресурс] – режим доступа: <https://cutt.ly/tf3wfdi>
17. Prom.ua — Вікіпедія [Электронный ресурс] – режим доступа: <https://cutt.ly/of3wg6s>
18. Prom.ua [Электронный ресурс] – режим доступа: <https://cutt.ly/HgR8pc3>
19. Prom.ua плюсы и минусы - недостатков больше чем достоинств - Первый независимый сайт отзывов Украины [Электронный ресурс] – режим доступа: <https://cutt.ly/jf3wke0>
20. Клумба - товары для всей семьи [Электронный ресурс] – режим доступа: <https://cutt.ly/ugR8dIF>
21. Графический редактор Photoshop - лучшая программа для редактирования изображений [Электронный ресурс] – режим доступа: <https://cutt.ly/HgPPziN>
22. Возможности Adobe Photoshop: обзор версий и достоинств программы [Электронный ресурс] – режим доступа: <https://cutt.ly/YgPPxno>
23. Как сделать макет или дизайн сайта в Photoshop [Электронный ресурс] – режим доступа: <https://cutt.ly/agPPcCD>
24. What is HTML [Электронный ресурс] – режим доступа: <https://cutt.ly/JgPPvXy>
25. What Is HTML? Advantage and Disadvantage of HTML [Электронный ресурс] – режим доступа: <https://cutt.ly/JgPPbXt>
26. What are the pros and cons of HTML? [Электронный ресурс] – режим доступа: <https://cutt.ly/AgAf2px>

27. Advantages and Disadvantages of JavaScript [Электронный ресурс] – режим доступа: <https://cutt.ly/PgPPnC2>
28. The Advantages and Disadvantages of JavaScript [Электронный ресурс] – режим доступа: <https://cutt.ly/DgPPIO2>
29. Webpack: When To Use And Why [Электронный ресурс] – режим доступа: <https://cutt.ly/wgPPTxe>
30. Что такое Firebase ? [Электронный ресурс] – режим доступа: <https://cutt.ly/wgPPTxe>
31. Знакомство с нотацией IDEF0 и пример использования [Электронный ресурс] – режим доступа: <https://cutt.ly/UgAkxRi>
32. Методология IDEF0 [Электронный ресурс] – режим доступа: <https://cutt.ly/ogAkbQv>
33. Преимущества и недостатки использования ideo для описания бизнес-процессов [Электронный ресурс] – режим доступа: <https://cutt.ly/OgAkm35>
34. Нотация IDEF0, или Матрёшка для бизнес-аналитика [Электронный ресурс] – режим доступа: <https://cutt.ly/TgAkevK>
35. Диаграмма вариантов использования (use case diagram) [Электронный ресурс] – режим доступа: <https://cutt.ly/JgAkUuq>
36. Построение диаграммы вариантов использования [Электронный ресурс] – режим доступа: <https://cutt.ly/0gAkODD>
37. Цели SMART: Расшифровка + 10 примеров (на все случаи) [Электронный ресурс] – режим доступа: <https://cutt.ly/UgeWaYr>
38. SMART(СМАРТ) - технология постановки целей с примерами [Электронный ресурс] – режим доступа: <https://cutt.ly/PgeWddG>
39. Преимущества Иерархической Структуры Работ (WBS) для менеджеров ИТ проектов [Электронный ресурс] – режим доступа: <https://cutt.ly/9geWf07>

40. Разработка иерархической структуры работ (WBS) проекта, разработка сетевой диаграммы (PND) [Электронный ресурс] – режим доступа: <https://cutt.ly/kgeWhjl>

41. Структура разбивки работ [Электронный ресурс] – режим доступа: <https://cutt.ly/egeWkY7>

42. Иерархические структуры декомпозиции [Электронный ресурс] – режим доступа: <https://cutt.ly/NgeWxxE>

43. Как диаграммы Ганта упрощают работу с проектами [Электронный ресурс] – режим доступа: <https://cutt.ly/3geWvhc>

44. Метод PERT в сетевом планировании: суть, формула, примеры [Электронный ресурс] – режим доступа: <https://cutt.ly/AgeWm27>

45. Управление рисками [Электронный ресурс] – режим доступа: <https://cutt.ly/VgeWW4f>

Додаток А

Планування робіт

Ідентифікація мети ІТ-проекту. Результатом дипломного проекту є розроблена Інтернет-платформа «Саллетрон». Для того, щоб провести детальний аналіз мети проекту, була застосована методологія SMART. SMART – це метод аналізу мети проекту, включаючи в себе: конкретизація, вимірюваність, досяжність, важливість та визначення термінів виконання робіт [37].

Методологія SMART має як плюси, так і мінуси у використанні. До позитивних аспектів використанні методології відносяться:

- багатofункціональність;
- поглибленість у проект;
- зрозумілість проекту.

До негативних аспектів використання методології відноситься неможливість спрогнозувати всі можливі зовнішні фактори впливу на розвиток проекту [38].

У табл. А.1 відображений результат аналізу мети проекту з використанням методології SMART.

Таблиця А.1 – Аналіз мети проекту за методологією SMART

| | |
|-------------|--|
| Specific | Розробка Інтернет-платформи «Саллетрон» |
| Measurable | Збільшення показника якості використання Інтернет-платформ з боку продавців та збільшення потенційної аудиторії. |
| Agreed Upon | Реалізація Інтернет-платформи виконується за допомогою методів мови програмування JavaScript |

Продовження таблиці А.1.

| | |
|--------------|---|
| Realistic | Розробник проекту має достатню кваліфікацію для реалізації платформи та її подальшої підтримки. |
| Time-related | Платформа має бути реалізована згідно з календарним планом. |

Планування змісту структури робіт ІТ-проекту. Наступним етапом планування проекту був опис фаз розробки. Для цього процесу було використано WBS-діаграму. Ієрархічна структура процесів проекту WBS або структура декомпозиції робіт проекту – схема, що відображає задачі проекту та відносини між собою [39].

За допомогою WBS-діаграма менеджери проекти виконують структурування та визначення вмісту проекту. Крупні задачі проекту розбиваються на декілька дрібних підзадач, де додається більш детальний опис процесу [40].

Переваги використання WBS-діаграми:

- повна організація проекту;
- чіткий перелік етапів проекту;
- ілюстрація загального обсягу проекту [41].

На рис.А.1 представлена WBS-діаграма проекту «Інтернет-платформа «Саллетрон»».

Побудова календарного графіку виконання ІТ-проекту. Наступний етап – планування структури організації. Для цього було сформовано OBS-діаграму на основі раніше розробленої WBS-діаграми. OBS-діаграма дозволяє на основі декомпозиції проектів, розробленої в WBS-діаграмі, встановити відносини між учасниками проекту. Для кожної підзадачі зазначаються члени проекту, що будуть відповідати за даний етап роботи [42]. За допомогою даної діаграми відбувається контроль робіт між учасниками проекту і за необхідністю виконується

реорганізація структури робіт. На рис. А.2 представлена розроблена OBS-діаграма проекту.

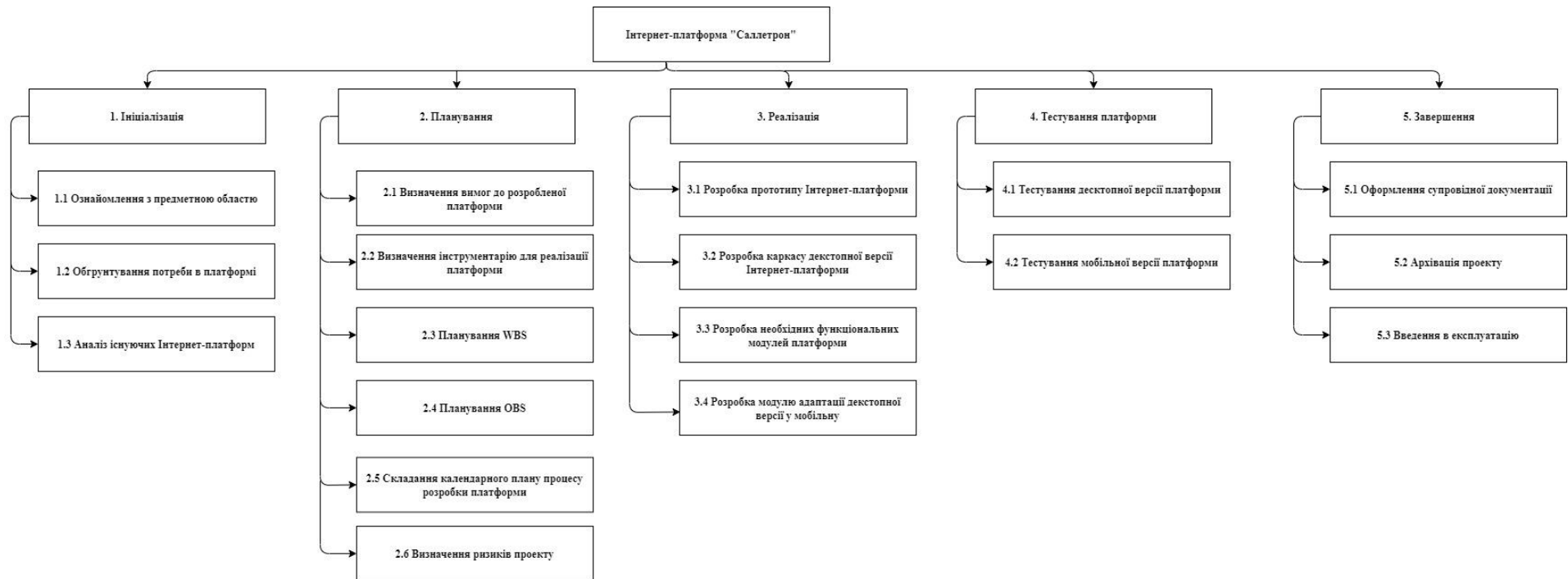


Рисунок А.1 – WBS-діаграма проекту «Інтернет-платформа «Саллетрон»»

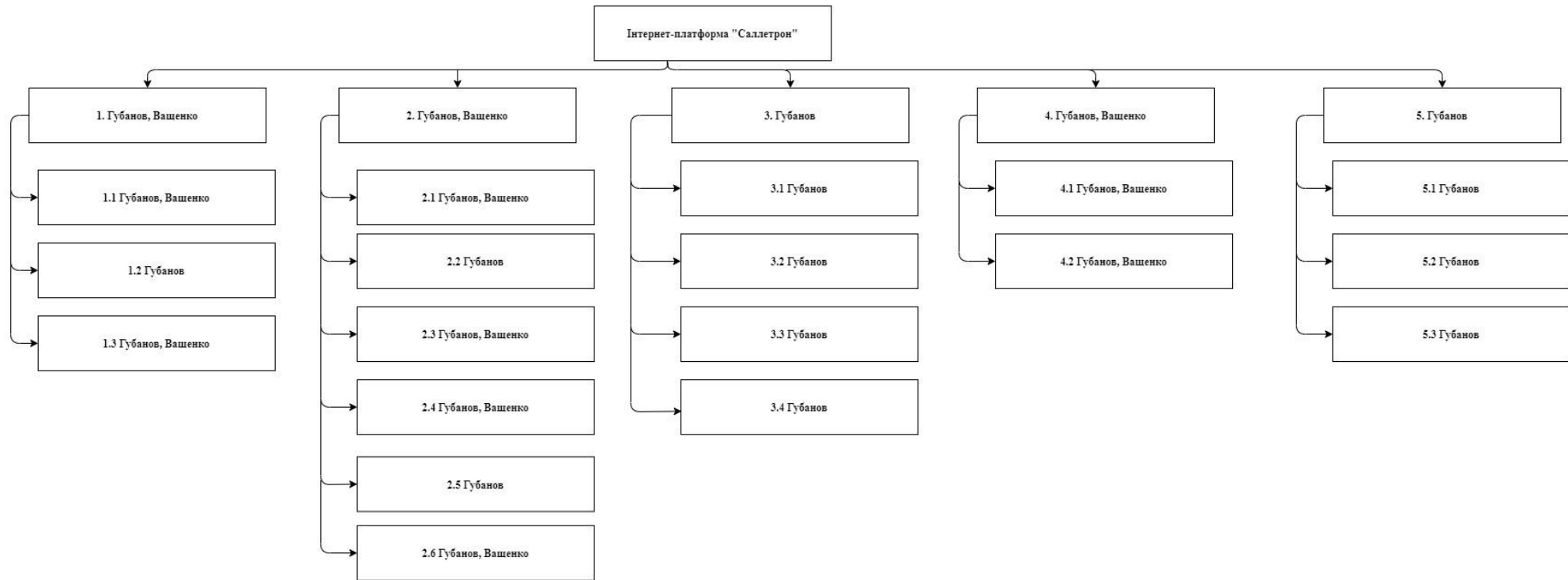


Рисунок А.2 – OBS-діаграма проекту «Интернет-платформа «Салнетрон»»

Планування ризиків ІТ-проекту. Наступним етапом планування проекту було складання календарного плану та визначення критичного шляху. Для формування календарного плану був використаний програмний додаток GanttProject, де було створено діаграму Ганта. Діаграма Ганта – це візуальне представлення задач, що були запланові під час організації проекту. Діаграма Ганта один із видів горизонтальних графіків, який відображає задачу, що виконується, та тривалість задачі у днях [43].

За допомогою діаграми Ганта можна відслідкувати:

- які задачі виконані та які плануються;
- дата початку та кінця кожної задачі;
- загальний час виконання кожної задачі;
- загальний термін виконання проекту.

На основі розробленої діаграми Ганта виконується побудова PERT-мережі, за допомогою якої розраховується критичний шлях виконання проекту. Модель представляє собою мережевий графік, до складу якого входять задачі проекту та зв'язки між ними [44].

На рис. А.3 представлений календарний план робіт, на рис. А.4 відображена розроблена діаграма Ганта проекту на основі календарного плану. На рис.А.5 – А.7 відображена PERT-мережа.

| | | |
|--|----------|----------|
| • Ознайомлення з предметною областю | 01.09.20 | 03.09.20 |
| • Обґрунтування потреби в платформі | 04.09.20 | 09.09.20 |
| • Аналіз існуючих Інтернет-платформ | 10.09.20 | 16.09.20 |
| • Визначення вимог до розробленої платфо... | 17.09.20 | 23.09.20 |
| • Визначення інструментарію для реалізації... | 24.09.20 | 29.09.20 |
| • Планування WBS | 30.09.20 | 30.09.20 |
| • Планування OBS | 01.10.20 | 01.10.20 |
| • Складання календарного плану розробки ... | 02.10.20 | 02.10.20 |
| • Визначення ризиків проекту | 05.10.20 | 06.10.20 |
| • Розробка прототипу Інтрнет-платформи | 07.10.20 | 12.10.20 |
| • Розробка каркасу десктопної версії Інтерн... | 13.10.20 | 21.10.20 |
| • Розробка необхідних функціональних мо... | 22.10.20 | 04.11.20 |
| • Розробка модулю адаптації десктопної ве... | 05.11.20 | 10.11.20 |
| • Тестування десктопної версії Інтернет-пла... | 11.11.20 | 11.11.20 |
| • Тестування мобільної версії Інтернет-плат... | 12.11.20 | 12.11.20 |
| • Оформлення супровідної документації | 10.09.20 | 13.11.20 |
| • Архівація проекту | 16.11.20 | 16.11.20 |
| • Введення в експлуатацію | 17.11.20 | 17.11.20 |

Рисунок А.3 – Календарний план проекту «Інтернет-платформа «Саллетрон»»

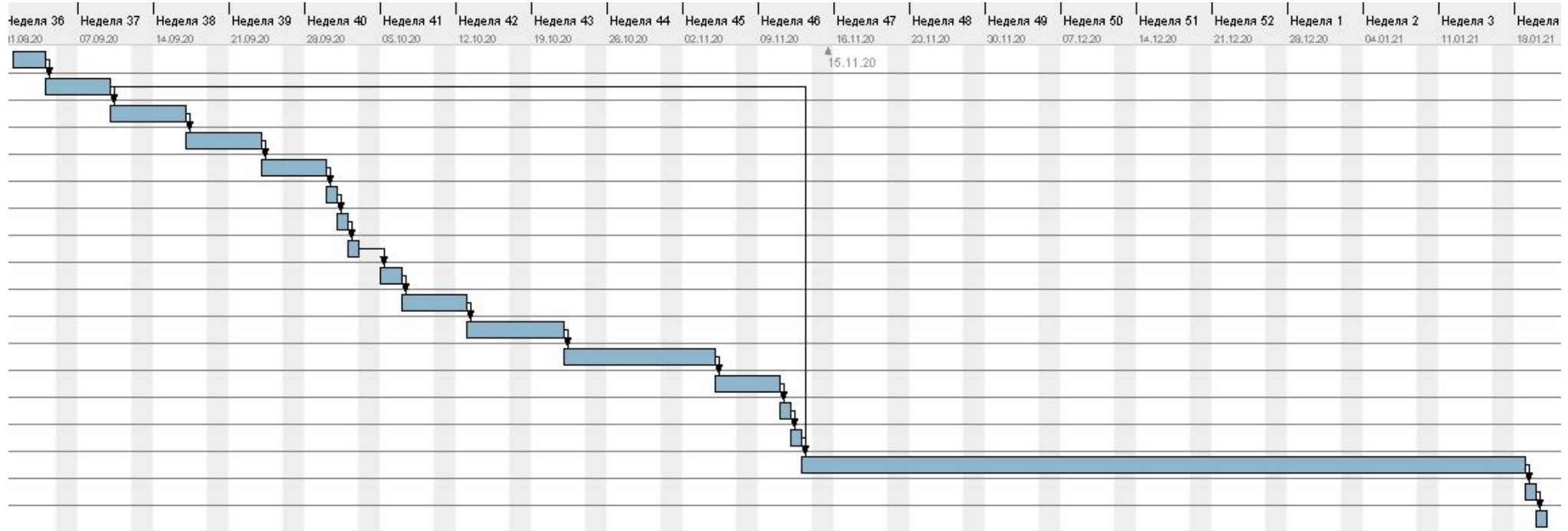


Рисунок А.4 – Диаграмма Ганта проекту «Интернет-платформа «Саллетрон»»



Рисунок А.5 – PERT-мережа проекту

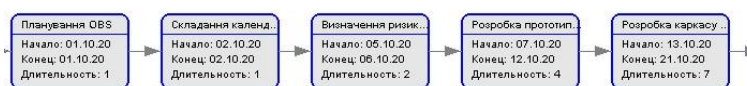


Рисунок А.6 – Продовження PERT-мережі



Рисунок А.7 – Продовження PERT-мережі

Останній етап планування проекту – визначення ризиків та розроблення сценарію дій їх подолання та мінімізації. Рекомендується виконати управління ризиками в 4 етапи:

- ідентифікація ризиків;
- аналіз силу впливу ризиків на проект;
- планування найбільш вагомих ризиків;
- відслідковування та контроль ризиків [45].

Процес управління ризиками починається на початку проекту та триває впродовж всього циклу робіт. За допомогою коректного та своєчасного сформованого плану реагування на ризики існує ймовірність мінімізації ресурсів людських, часу та програмних. У табл. А.2 представлений сформований перелік

можливих ризиків проекту «Інтернет-платформа «Саллетрон»» з зазначення ймовірності настання та сили впливу.

Таблиця А.2 – Ідентифікація ризиків та їх вплив на проект

| Ризики | Ймовірність настання | Вплив ризику |
|--|----------------------|--------------|
| Технічні ризики | Середній | Високий |
| Ризики оцінки термінів | Низький | Середній |
| Ризики неприйняття Інтернет-платформи замовником | Низький | Низький |
| Технологічні ризики | Середній | Високий |
| Виникнення помилок при тестування проекту | Низький | Середній |

На основі сформованого переліку можливих ризиків проекту був розроблений сценарій реагування команди проекту на ризики для мінімізації втрат ресурсів. У табл. А.3 представлений план реагування на ризики проекту «Інтернет-платформа «Саллетрон»».

Таблиця А.3 — План реагування на ризики

| Ризики | Ймовірність настання | Вплив на проект | Рейтинговий бал | Стратегія реагування | План дій |
|------------------------|----------------------|-----------------|-----------------|----------------------|---|
| Технічні ризики | Середній | Високий | 4 | Попередження | Попереднє тестування обладнання; |
| Ризики оцінки термінів | Низький | Середній | 4 | Пом'якшення | Перебудова плану виконання робіт з можливістю додавання людських ресурсів |

Продовження таблиці А.3

| Ризики | Ймовірність настання | Вплив на проект | Рейтинговий бал | Стратегія реагування | План дій |
|--|----------------------|-----------------|-----------------|----------------------|--|
| Ризики неприйняття Інтернет-платформи замовником | Низький | Низький | 1 | Прийняття | Проведення попереднього детального аналізу предметної області |
| Технологічні ризики | Середній | Високий | 7 | Пом'якшення | Виділення додаткового часу на пошук нових методів або нової технології для реалізації проекту. |
| Виникнення помилок при тестуванні проекту | Низький | Середній | 5 | Попередження | Контроль проміжних результатів під час реалізації проекту |

Додаток Б

Програмний код функціональних модулів

authServices.js

```
import './styles.scss';
import axios from 'axios';
import { addToLocalStorage, logErrors, hideMenue, logOut } from './services';
import { stateOfAuth } from './refs';
import apiServices from '../../services/api';

logOut();
hideMenue();

export const firebaseConfig = {
  apiKey: 'AIzaSyDGixM-3-XeusqOkskuLtpiQQP2BLo-jPs',
  authDomain: 'saletrontest.firebaseio.com',
  databaseURL: 'https://saletrontest.firebaseio.com',
  projectId: 'saletrontest',
  storageBucket: 'saletrontest.appspot.com',
  messagingSenderId: '733293305057',
  appId: '1:733293305057:web:295b4d8a33b980caa26ba0',
};

const user = {
  email: '',
  password: '',
};

export const userLoginHandler = e => (user[e.target.name] = e.target.value);

const resetUserData = e => {
  user.email = '';
  user.password = '';
};

const fetch = async action => {
  const errorPassword = document.querySelector('.errorPassword');
  const errorEmail = document.querySelector('.errorEmail');
  try {
    const result = await axios.post(
```

```

`https://identitytoolkit.googleapis.com/v1/accounts:${action}?key=${firebaseConfig.ap
iKey}` ,
  { ...user, returnSecureToken: true },
);

if (result.status === 200) {
  console.log(result);
  stateOfAuth.instance.close();
  stateOfAuth.isUserAuthenticated = true;
  addToLocalStorage(token, userID);
}

const token = result.data.idToken;
const userID = result.data.localId;

hideMenu();
} catch (error) {
  logErrors(error);
  switch (error.response.data.error.message) {
    case 'INVALID_PASSWORD':
      errorPassword.textContent = 'Wrong password';
      break;
    case 'EMAIL_NOT_FOUND':
      errorEmail.textContent = 'User not found';
      break;
    case 'EMAIL_EXISTS':
      errorEmail.textContent = 'User already exist';
      break;
    case 'TOO_MANY_ATTEMPTS_TRY_LATER':
      errorEmail.textContent =
        'Too many unsuccessful login attempts. Please try again later.';
      break;

    default:
      break;
  }
}
};

const submitData = async () => {
  const result = await apiServices.signUpUser(user);

```

```

    if (result.statusCheck.status === 200) {
      console.log(result);
      stateOfAuth.instance.close();
      stateOfAuth.isUserAuthenticated = true;
    }
    resetUserData();
    hideMenu();
  };
const checkData = async () => {
  const result = await apiServices.signInUser(user);

  if (result.statusCheck.status === 200) {
    console.log(result);
    stateOfAuth.instance.close();
    stateOfAuth.isUserAuthenticated = true;
  }
  resetUserData();
  hideMenu();
};

export function submitForm(e) {
  e.preventDefault();
  switch (e.submitter.classList[0]) {
    case 'login':
      checkData();
      break;
    case 'registration':
      submitData();

      break;
  }
}

```

basket.js

```

import axios from 'axios';
import basketMarkup from './basket.hbs';
import openItemModal from '../ItemModal/ItemModal';
import './basket.scss';

const refs = {

```

```

basketModal: document.querySelector('.basketModal'),
basket: document.querySelector('.basket'),
input: document.querySelector('input[name="itemById"]'),
button: document.querySelector('.buyButton'),
show: document.querySelector('.show'),
total: document.querySelector('.basket__total'),
clear: document.querySelector('.basket__clear'),
cart: document.querySelector('.buttonCart'),
};

async function getProduct(e) {
  // const response = await axios.get(
  //   `https://salletron.firebaseio.com/products/change/-M8zgqzxq0ZHiAo3xxzY.json`,
  // );

  if (e.target.className === 'modal__buyItem') {
    const parentRef = e.target.parentNode.parentNode;
    const itemImage = parentRef.children[0].children[0].children[0].currentSrc;
    const itemName = parentRef.children[1].children[1].textContent;
    const itemPrice = parentRef.children[1].children[3].textContent;
    let check = 0;

    refs.basket.children.forEach(element => {
      if (
        element.nodeName === 'DIV' &&
        element.children[1].textContent === itemName
      ) {
        element.children[2].children[1].textContent =
          Number(element.children[2].children[1].textContent) + 1;

        refs.total.textContent =
          Number(refs.total.textContent) + parseInt(itemPrice);
        check = 1;
        return;
      }
    });

    if (check === 1) return;

    const data = {
      itemName,

```

```

    itemImage,
    itemPrice,
  };

  refs.basket.insertAdjacentHTML('afterbegin', basketMarkup(data));

  refs.removeProduct = document.querySelector('.basket__itemDelete');
  refs.removeProduct.addEventListener('click', removeProduct);

  refs.productQuantity = document.querySelector('.item__quantity');
  refs.productIncrement = document.querySelector('.increment');
  refs.productDecrement = document.querySelector('.decrement');

  refs.productIncrement.addEventListener('click', changeQuantity);
  refs.productDecrement.addEventListener('click', changeQuantity);

  refs.total.textContent =
    Number(refs.total.textContent) + parseInt(data.itemPrice);
}
}

function removeProduct(e) {
  e.target.parentNode.remove();

  const productQuantity =
    e.target.parentNode.children[2].children[1].textContent;
  const productPrice = parseInt(e.target.parentNode.children[3].textContent);
  refs.total.textContent =
    Number(refs.total.textContent) - productPrice * Number(productQuantity);

  if (refs.basket.children.length === 2) closeBasket();
}

function changeQuantity(e) {
  const productQuantity = e.target.parentNode.children[1];
  const productPrice = parseInt(
    e.target.parentNode.parentNode.children[3].textContent,
  );

  if (e.target.className === 'increment') {
    productQuantity.textContent = Number(productQuantity.textContent) + 1;

```

```

    refs.total.textContent = Number(refs.total.textContent) + productPrice;
    return;
} else if (
    e.target.className === 'decrement' &&
    Number(productQuantity.textContent) > 1
) {
    productQuantity.textContent = Number(productQuantity.textContent) - 1;
    refs.total.textContent = Number(refs.total.textContent) - productPrice;
}
}

```

```

function clearBasket() {
    while (refs.basket.firstChild.nodeName === 'DIV') {
        refs.basket.firstChild.remove();
    }
    refs.total.textContent = 0;
    closeBasket();
}

```

```

function openBasket() {
    if (refs.basket.children.length === 2) {
        alert('basket is empty');
        return;
    }
    refs.basket.parentElement.classList.add('basketModal-show');
    refs.basketModal.addEventListener('click', closeBasket);
    document.addEventListener('keydown', closeBasket);
}

```

```

function closeBasket(e) {
    if (
        refs.basket.children.length === 2 ||
        e.target === e.currentTarget ||
        e.code === 'Escape'
    ) {
        refs.basket.parentElement.classList.remove('basketModal-show');
        refs.basketModal.removeEventListener('click', closeBasket);
        document.removeEventListener('keydown', closeBasket);
    }
}

```



```
document.addEventListener('click', getProduct);
refs.cart.addEventListener('click', openBasket);

refs.clear.addEventListener('click', clearBasket);
```

categoryListItem.js

```
import '@glidejs/glide/dist/css/glide.core.min.css';
import '@glidejs/glide/dist/css/glide.theme.css';
import Glide from '@glidejs/glide';
import apiServices from '../services/api';
import markup from './categoryListItemMarkup';
import openItemModal from '../ItemModal/ItemModal';
import './categoryListItem.scss';

const categoryList = document.querySelector('.categoryList');
const spinner = document.querySelector('.spinner');

categoryList.addEventListener('click', e => {
  if (e.target.nodeName === 'IMG' || e.target.nodeName === 'P') {
    openItemModal(e.target.dataset.id);
  }
});

export function getCategoryListItem(category) {
  spinner.classList.add('spinner__show');

  const categoryItems = apiServices.getProductsByCategory(category);

  const products = {
    visible: 12,
  };

  const itemMarkup = categoryItems.reduce((acc, item) => {
    acc += markup.getItemMarkup(item);
    return acc;
  }, '');

  categoryList.insertAdjacentHTML(
    'beforeend',
    markup.mainMarkup(category, itemMarkup, categoryItems),
  );
}
```

```

const categoryContainer = document.querySelector(
  `[data-replace=${category}]`,
);
categoryContainer.innerHTML = markup.withSlider(
  category,
  itemMarkup,
  categoryItems,
);
getSlider();

categoryContainer.addEventListener('click', seeAllProducts);

function seeAllProducts(e) {
  if (e.target.dataset.btnseeall === `${category}`) {
    const categoryCont = document.querySelector(
      `[data-replace=${e.target.dataset.btnseeall}]`,
    );
    categoryCont.innerHTML = markup.withoutSlider(e.target.dataset.btnseeall);

    const categoryContent = document.querySelector(
      `[data-content=${e.target.dataset.btnseeall}]`,
    );
    categoryContent.innerHTML = innerMarkup(
      categoryItems,
      0,
      products.visible,
    );

    const loadMoreBtn = document.querySelector(
      `[data-loadmore=${e.target.dataset.btnseeall}]`,
    );
    loadMoreBtn.addEventListener('click', loadMoreProducts);
    endOfCategoryHandler(categoryItems, loadMoreBtn, products);

    function loadMoreProducts() {
      loadMoreBtn.classList.add('button--loading');

      const addedProducts = innerMarkup(
        categoryItems,
        products.visible,
        products.visible + 12,

```

```

    );
    products.visible += 12;

    const categoryContent = document.querySelector(
      `[data-content=${e.target.dataset.btnseeall}]`,
    );
    categoryContent.insertAdjacentHTML('beforeend', addedProducts);
    loadMoreBtn.classList.remove('button--loading');
    endOfCategoryHandler(categoryItems, loadMoreBtn, products);
  }
}

if (e.target.dataset.btnseeless === `${category}`) {
  const categoryCont = document.querySelector(
    `[data-replace=${e.target.dataset.btnseeless}]`,
  );
  categoryCont.innerHTML = markup.withSlider(
    e.target.dataset.btnseeless,
    itemMarkup,
    categoryItems,
  );
  getSlider();
}
}
}

function getSlider() {
  const sliders = document.querySelectorAll('.glide');

  for (let i = 0; i < sliders.length; i += 1) {
    const glide = new Glide(sliders[i], {
      type: 'carousel',
      perView: 4,
      dots: '#dots',
      autoplay: 6000,
      breakpoints: {
        1279: {
          gap: 26,
          perView: 2,
        },
        767: {

```

```

        gap: 30,
        perView: 1,
    },
},
});
glide.mount();
}
}

```

```

function innerMarkup(categoryItems, start, end) {
    return `
    ${categoryItems.slice(start, end).reduce((acc, item) => {
        return (acc += markup.getItemMarkup(item));
    }, '')}`;
}

```

```

function endOfCategoryHandler(categoryItems, loadMoreBtn, products) {
    if (products.visible >= categoryItems.length) {
        loadMoreBtn.disabled = true;
        loadMoreBtn.classList.replace('button', 'button--inactive');
        loadMoreBtn.querySelector('span').textContent = 'No more here';
    }
}

```

categoryListItemMarkup.js

```

const markup = {
    getItemMarkup(product) {
        return `
            <li class="category__list-item glide__slide">
                <div class='category__list-item-img-wrapper'>
                    <img class = "category__list-item-img"
                    src = "${product.images[0]}"
                    alt = "item" data-id=${product.id}>
                </div>
                <div class="category__list-item-info">
                    <p class = "category__list-item-name"
                    data-id=${product.id}>
                        ${product.name}
                    </p>
                    <p class = "category__list-item-price" data-id=${product.id}>

```

```

                ${product.price}
                <span                                class="category__list-item-price-
suffix">&#8372;</span>
            </p>
        </div>
    </li>
`;
},

mainMarkup(category) {
    return `
    <div class = "category container">
        <div class='replacableContainer' data-replace=${category}></div>
    </div>
    `;
},

withoutSlider(category) {
    return `
        <h3 class='category__name'>${capitalize(`${category}`)}</h3>
        <button            class="category__btn            category__btn-see-less"            data-
btnseeless=${category}>See less</button>
        <div class='category__content' data-content=${category}></div>
        <button class='button loadMoreBtn' data-loadmore=${category} id='call-to-
action'>
            <span class='loadMoreBtnText'>Click for magic</span>
        </button>
    `;
},

withSlider(category, itemMarkup, categoryItems) {
    return `
        <h3 class='category__name'>${capitalize(`${category}`)}</h3>
        <button            class="category__btn            category__btn-see-all"            data-
btnseeall=${category}>See all</button>
        <div class="glide" data-glide=${category}>
            <div data-glide-el="track" class="glide__track">
                <div class="category__list glide__slides" data-category=${category}>
                    ${itemMarkup}
                </div>
            </div>
        </div>
    `;
},

```

```

<div class="glide__arrows" data-glide-el="controls">
  <button button class = "glide__arrow glide__arrow--left prev" data-
glide-dir="<">
    <svg          class="svg-arrow"          version="1.1"          id="Capa_1"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px"
y="0px" viewBox="0 0 477.175 477.175" style="enable-background: new 0 0 477.175
477.175;" xml:space="preserve">
      <path d="M145.188,238.575l215.5-215.5c5.3-5.3,5.3-13.8,0-19.1s-
13.8-5.3-19.1,0l-225.1,225.1c-5.3,5.3-5.3,13.8,0,19.1l225.1,225
c2.6,2.6,6.1,4,9.5,4s6.9-1.3,9.5-4c5.3-5.3,5.3-13.8,0-19.1L145.188,238.575z" />
    </svg>
  </button>
  <button button class = "glide__arrow glide__arrow--right next" data-
glide-dir=">">
    <svg          class="svg-arrow"          version="1.1"          id="Capa_1"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px"
y="0px" viewBox="0 0 477.175 477.175" style="enable-background: new 0 0 477.175
477.175;" xml:space="preserve">
      <path          d="M360.731,229.075l-225.1-225.1c-5.3-5.3-13.8-5.3-
19.1,0s-5.3,13.8,0,19.1l215.5,215.5l-215.5,215.5
c-5.3,5.3-
5.3,13.8,0,19.1c2.6,2.6,6.1,4,9.5,4c3.4,0,6.9-1.3,9.5-4l225.1-
225.1C365.931,242.875,365.931,234.275,360.731,229.075z" />
    </svg>
  </button>
</div>
<div class="glide__bullets" data-glide-el="controls[nav]">
  ${categoryItems
.map(
  (item, idx) =>
    `<button class="glide__bullet" data-glide-dir="=${idx}"></button>`,
  )
.join('')}
</div>
</div>

`;
},
};

function capitalize(str) {
  return str.charAt(0).toUpperCase() + str.slice(1);
}

```

```

}
export default markup;

```

renderCategoryList.js

```

import ApiService from '../services/api';
import { getCategoryListItem } from './categoryListItem';

export const categoriesCount = { count: 0 };

const renderBtn = document.querySelector('.renderBtn');
const spinner = document.querySelector('.spinner');

export function paginationCategori(num) {
  spinner.classList.add('spinner__show');
  ApiService.getCategoriesList().then(data => {
    for (let i = categoriesCount.count; i < num + categoriesCount.count; i++) {
      if (data[i]) {
        getCategoryListItem(data[i]);
      }
      renderBtn.classList.remove('button--loading');
      checkForEndOfData(data, num, categoriesCount.count);
    }

    categoriesCount.count += num;
    spinner.classList.remove('spinner__show');
  });
}

renderBtn.addEventListener('click', paginationCategori);

function paginationCategori1() {
  renderBtn.classList.add('button--loading');
  paginationCategori(2);
}

function checkForEndOfData(data, num, count) {
  if (data.length <= num + count) {
    renderBtn.disabled = true;
    renderBtn.classList.replace('button', 'button--inactive');
    renderBtn.querySelector('span').textContent = 'No more categories';
  }
}

```

```

    }
  }

export function checkForStartOfData() {
  renderBtn.disabled = false;
  renderBtn.classList.replace('button--inactive', 'button');
  renderBtn.querySelector('span').textContent = 'More categories';
}

```

footer.js

```

import './footer.scss';

const footerMarkup = `
<div class="footer__info container">
<p class="footer__info-r1">
<span> &#169; 2020</span>
<span>
  <svg
    width="1px" height="12px">
    <path fill-rule="evenodd" opacity="0.502" fill="rgb(206, 210, 209)"
      d="M1.000,20.000 L-0.000,20.000 L-0.000,-0.000 L1.000,-0.000 L1.000,20.000 Z"/>
  </svg>
</span>
  <span class="br">All Rights Reserved</span>
  <span>
    <svg
      width="1px" height="12px">
      <path fill-rule="evenodd" opacity="0.502" fill="rgb(206, 210, 209)"
        d="M1.000,20.000 L-0.000,20.000 L-0.000,-0.000 L1.000,-0.000 L1.000,20.000 Z"/>
    </svg>
  </span>
</p>
<p class="footer__info-r1">
  <span> Developed with</span>
  <span>
    <svg
      width="10px" height="10px">
      <image
        x="0px" y="0px" width="10px" height="10px"
        xlink:href="data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAABAAAAAQCAMAAAAoLQ9TAAAABGdBT
        UEAAALGPC/xhBQAAACBjSFJNAAB6JgAAgIQAAAPoAAACA6AAAAdTAAAOpgAAA6mAAAF3CculE8AAAAjVBMVEX/aw

```



```

function toggleBurger(e) {
  e.preventDefault();
  e.stopPropagation();
  if (!e.currentTarget) return;
  if (e.currentTarget) {
    hamburger.classList.toggle('is-active');
    burgerMenu.classList.toggle('expanded');
  }
}
// -----
function toggleTabletFilter(e) {
  e.preventDefault();
  e.stopPropagation();
  if (!e.currentTarget) return;
  if (e.currentTarget) {
    burgerMenu.classList.toggle('filterIsActiveForTablet');
  }
}
// -----
async function searchCategory(e) {
  spinner.classList.add('spinner__show');

  const inputValue = e.target.value;
  let isFound = false;
  categoryList.innerHTML = '';
  if (inputValue === '') {
    paginationCategore(2);
  }

  const getCategoryData = await apiServices.getCategoriesList();
  await getCategoryData.forEach(category => {
    if (
      category.toLowerCase().includes(inputValue.toLowerCase()) &&
      inputValue.length >= 3
    ) {
      categoryList.insertAdjacentHTML(
        'beforeend',
        searchBarHbs(apiServices.productsArr),
      );

      isFound = true;
    }
  });
}

```

```

        spinner.classList.remove('spinner__show');

        return;
    }
});

if (!isFound) {
    apiServices.getProducts().then(data => {
        if (inputValue.length >= 3) {
            const filteredProducts = data.filter(product =>
                product.name.toLowerCase().includes(inputValue.toLowerCase()),
            );
            categoryList.insertAdjacentHTML(
                'beforeend',
                searchBarHbs(filteredProducts),
            );
        }
        spinner.classList.remove('spinner__show');

        return;
    });
}
}

```

ItemModal.js

```

import './ItemModal.scss';

import '@glidejs/glide/dist/css/glide.core.min.css';
import '@glidejs/glide/dist/css/glide.theme.css';
import Glide from '@glidejs/glide';
import ItemModalHbs from './ItemModal.hbs';
import tabletItemPictureMarkup from './tabletItemPictureMarkup.hbs';
import apiService from '../services/api';
import * as basicLightbox from 'basiclightbox';
const spinner = document.querySelector('.spinner');

// apiService.signUpUser({
//   email:'master-321111@i.ua',
//   password:'111111'
// })

```

```

// apiService.signInUser({
//   email: 'master-321@i.ua',
//   password: '111111'
// })
const openItemModal = id => {
  spinner.classList.add('spinner__show');

  apiService.getProductById(id).then(data => {
    const glide = new Glide('.glideItemModal', {
      type: 'carousel',
      perView: 1,
      dots: '#dots',
    });

    const instance = basicLightbox.create(
      `
      <div>${ItemModalHbs(data)}</div>
    `,
    {
      closable: false,
    },
  );

  instance.show();
  spinner.classList.remove('spinner__show');

  const favoriteBtn = document.querySelector('.modal__share-item-heart-cvg');
  if (
    JSON.parse(localStorage.getItem('user')) &&
    JSON.parse(localStorage.getItem('user')).favorites.includes(id)
  ) {
    favoriteBtn.innerHTML = `
    <path
                                d="M255,489.61-35.7-
35.7C86.7,336.6,0,257.55,0,160.65C0,81.6,61.2,20.4,140.25,20.4c43.35,0,86.7,20.4,114.
75,53.55
    C283.05,40.8,326.4,20.4,369.75,20.4C448.8,20.4,510,81.6,510,160.65c0,96.9-
86.7,175.95-219.3,293.25L255,489.6z"/>
    `;
  }
}

```

```

const closeItemModalBtn = document.querySelector('.btnClose');
closeItemModalBtn.addEventListener('click', () => instance.close());

const backItemModalBtn = document.querySelector('.btnBack');
backItemModalBtn.addEventListener('click', () => instance.close());

favoriteBtn.addEventListener('click', e => {
  if (JSON.parse(localStorage.getItem('user'))) {
    if (!JSON.parse(localStorage.getItem('user')).favorites.includes(id)) {
      console.log('add');
      e.currentTarget.innerHTML = `
<path
                                                                    d="M255,489.61-35.7-
35.7C86.7,336.6,0,257.55,0,160.65C0,81.6,61.2,20.4,140.25,20.4c43.35,0,86.7,20.4,114.
75,53.55

C283.05,40.8,326.4,20.4,369.75,20.4C448.8,20.4,510,81.6,510,160.65c0,96.9-
86.7,175.95-219.3,293.25L255,489.6z"/>
`;
      apiService.addUserFavorite(id);
    } else {
      console.log('remove');
      e.currentTarget.innerHTML = `
<path
d="m256 455.515625c-7.289062 0-14.316406-2.640625-19.792969-7.4375-20.683593-
18.085937-40.625-35.082031-58.21875-50.0742191-.089843-.078125c-51.582032-43.957031-
96.125-81.917969-127.117188-119.3125-34.644531-41.804687-50.78125-81.441406-50.78125-
124.742187 0-42.070313 14.425781-80.882813 40.617188-109.292969 26.503906-28.746094
62.871093-44.578125 102.414062-44.578125 29.554688 0 56.621094 9.34375 80.445312
27.769531 12.023438 9.300781 22.921876 20.683594 32.523438 33.960938 9.605469-
13.277344 20.5-24.660157 32.527344-33.960938 23.824218-18.425781 50.890625-27.769531
80.445312-27.769531 39.539063 0 75.910156 15.832031 102.414063 44.578125 26.191406
28.410156 40.613281 67.222656 40.613281 109.292969 0 43.300781-16.132812 82.9375-
50.777344 124.738281-30.992187 37.398437-75.53125 75.355469-127.105468 119.308594-
17.625 15.015625-37.597657 32.039062-58.328126 50.167969-5.472656 4.789062-12.503906
7.429687-19.789062 7.429687zm-112.96875-425.523437c-31.066406 0-59.605469 12.398437-
80.367188 34.914062-21.070312 22.855469-32.675781 54.449219-32.675781 88.964844 0
36.417968 13.535157 68.988281 43.882813 105.605468 29.332031 35.394532 72.960937
72.574219 123.476562 115.6251.09375.078126c17.660156 15.050781 37.679688 32.113281
58.515625 50.332031 20.960938-18.253907 41.011719-35.34375 58.707031-50.417969
50.511719-43.050781 94.136719-80.222656 123.46875-115.617188 30.34375-36.617187
43.878907-69.1875 43.878907-105.605468 0-34.515625-11.605469-66.109375-32.675781-

```

```

88.964844-20.757813-22.515625-49.300782-34.914062-80.363282-34.914062-22.757812 0-
43.652344 7.234374-62.101562 21.5-16.441406 12.71875-27.894532 28.796874-34.609375
40.046874-3.453125 5.785157-9.53125 9.238282-16.261719 9.238282s-12.808594-3.453125-
16.261719-9.238282c-6.710937-11.25-18.164062-27.328124-34.609375-40.046874-18.449218-
14.265626-39.34375-21.5-62.097656-21.5zm0 0" />`;
    apiService.deleteUserFavorite(id);
  }
} else {
  basicLightbox
    .create(`<div class="modalSignInAlert">please sign in</div>`)
    .show();
}
});

glide.mount();

const tabletItemImage = document.querySelector('.tablet_picture');
const sMarkup = `
<div class="itemModal__MainImageWrapper">
  
</div>
`;
tabletItemImage.insertAdjacentHTML('beforeend', sMarkup);
tabletItemImage.insertAdjacentHTML(
  'beforeend',
  tabletItemImageMarkup(data),
);
const clickedPicture = document.querySelectorAll('.itemSliderPicture');
const itemModal__MainImage = document.querySelector(
  '.itemModal__MainImage',
);
clickedPicture.forEach(picture => {
  picture.addEventListener('click', e => {
    itemModal__MainImage.setAttribute('src', e.target.getAttribute('src'));
  });
});
});
};
export default openItemModal;

```

navigationMenu.js

```

import './ItemModal.scss';

import '@glidejs/glide/dist/css/glide.core.min.css';
import '@glidejs/glide/dist/css/glide.theme.css';
import Glide from '@glidejs/glide';
import ItemModalHbs from './ItemModal.hbs';
import tabletItemPictureMarkup from './tabletItemPictureMarkup.hbs';
import apiService from '../../services/api';
import * as basicLightbox from 'basiclightbox';
const spinner = document.querySelector('.spinner');

// apiService.signUpUser({
//   email:'master-321111@i.ua',
//   password:'111111'
// })

// apiService.signInUser({
//   email: 'master-321@i.ua',
//   password: '111111'
// })
const openItemModal = id => {
  spinner.classList.add('spinner__show');

  apiService.getProductById(id).then(data => {
    const glide = new Glide('.glideItemModal', {
      type: 'carousel',
      perView: 1,
      dots: '#dots',
    });

    const instance = basicLightbox.create(
      `
      <div>${ItemModalHbs(data)}</div>
      `
      ,
      {
        closable: false,
      }
    );
  });
};

```

```

instance.show();
spinner.classList.remove('spinner__show');

const favoriteBtn = document.querySelector('.modal__share-item-heart-cvg');
if (
  JSON.parse(localStorage.getItem('user')) &&
  JSON.parse(localStorage.getItem('user')).favorites.includes(id)
) {
  favoriteBtn.innerHTML = `
<path                                     d="M255,489.61-35.7-
35.7C86.7,336.6,0,257.55,0,160.65C0,81.6,61.2,20.4,140.25,20.4c43.35,0,86.7,20.4,114.
75,53.55
C283.05,40.8,326.4,20.4,369.75,20.4C448.8,20.4,510,81.6,510,160.65c0,96.9-
86.7,175.95-219.3,293.25L255,489.6z"/>
`;
}

const closeItemModalBtn = document.querySelector('.btnClose');
closeItemModalBtn.addEventListener('click', () => instance.close());

const backItemModalBtn = document.querySelector('.btnBack');
backItemModalBtn.addEventListener('click', () => instance.close());

favoriteBtn.addEventListener('click', e => {
  if (JSON.parse(localStorage.getItem('user'))) {
    if (!JSON.parse(localStorage.getItem('user')).favorites.includes(id)) {
      console.log('add');
      e.currentTarget.innerHTML = `
<path                                     d="M255,489.61-35.7-
35.7C86.7,336.6,0,257.55,0,160.65C0,81.6,61.2,20.4,140.25,20.4c43.35,0,86.7,20.4,114.
75,53.55
C283.05,40.8,326.4,20.4,369.75,20.4C448.8,20.4,510,81.6,510,160.65c0,96.9-
86.7,175.95-219.3,293.25L255,489.6z"/>
`;
      apiService.addUserFavorite(id);
    } else {
      console.log('remove');
      e.currentTarget.innerHTML = `
<path

```



```

d="m256 455.515625c-7.289062 0-14.316406-2.640625-19.792969-7.4375-20.683593-
18.085937-40.625-35.082031-58.21875-50.0742191-.089843-.078125c-51.582032-43.957031-
96.125-81.917969-127.117188-119.3125-34.644531-41.804687-50.78125-81.441406-50.78125-
124.742187 0-42.070313 14.425781-80.882813 40.617188-109.292969 26.503906-28.746094
62.871093-44.578125 102.414062-44.578125 29.554688 0 56.621094 9.34375 80.445312
27.769531 12.023438 9.300781 22.921876 20.683594 32.523438 33.960938 9.605469-
13.277344 20.5-24.660157 32.527344-33.960938 23.824218-18.425781 50.890625-27.769531
80.445312-27.769531 39.539063 0 75.910156 15.832031 102.414063 44.578125 26.191406
28.410156 40.613281 67.222656 40.613281 109.292969 0 43.300781-16.132812 82.9375-
50.777344 124.738281-30.992187 37.398437-75.53125 75.355469-127.105468 119.308594-
17.625 15.015625-37.597657 32.039062-58.328126 50.167969-5.472656 4.789062-12.503906
7.429687-19.789062 7.429687zm-112.96875-425.523437c-31.066406 0-59.605469 12.398437-
80.367188 34.914062-21.070312 22.855469-32.675781 54.449219-32.675781 88.964844 0
36.417968 13.535157 68.988281 43.882813 105.605468 29.332031 35.394532 72.960937
72.574219 123.476562 115.6251.09375.078126c17.660156 15.050781 37.679688 32.113281
58.515625 50.332031 20.960938-18.253907 41.011719-35.34375 58.707031-50.417969
50.511719-43.050781 94.136719-80.222656 123.46875-115.617188 30.34375-36.617187
43.878907-69.1875 43.878907-105.605468 0-34.515625-11.605469-66.109375-32.675781-
88.964844-20.757813-22.515625-49.300782-34.914062-80.363282-34.914062-22.757812 0-
43.652344 7.234374-62.101562 21.5-16.441406 12.71875-27.894532 28.796874-34.609375
40.046874-3.453125 5.785157-9.53125 9.238282-16.261719 9.238282s-12.808594-3.453125-
16.261719-9.238282c-6.710937-11.25-18.164062-27.328124-34.609375-40.046874-18.449218-
14.265626-39.34375-21.5-62.097656-21.5zm0 0" />`;
    apiService.deleteUserFavorite(id);
  }
} else {
  basicLightbox
    .create(`<div class="modalSignInAlert">please sign in</div>`)
    .show();
}
});

glide.mount();

const tabletItemImage = document.querySelector('.tablet_picture');
const sMarkup = `
<div class="itemModal__MainImageWrapper">
  
</div>
`;
tabletItemImage.insertAdjacentHTML('beforeend', sMarkup);

```

```
tabletItemPicture.insertAdjacentHTML(  
  'beforeend',  
  tabletItemPictureMarkup(data),  
);  
const clickedPicture = document.querySelectorAll('.itemSliderPicture');  
const itemModal__MainImage = document.querySelector(  
  '.itemModal__MainImage',  
);  
clickedPicture.forEach(picture => {  
  picture.addEventListener('click', e => {  
    itemModal__MainImage.setAttribute('src', e.target.getAttribute('src'));  
  });  
});  
});  
export default openItemModal;
```