

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРНІГІВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ
УНІВЕРСИТЕТ

Навчально-науковий інститут електронних та інформаційних технологій
Кафедра інформаційних та комп'ютерних систем

Допущено до захисту

Завідувач кафедри
к.е.н., доцент Базилевич В.М.

« _____ » _____ 2020 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за освітньо-професійною програмою бакалавра

Інформаційно-комп'ютерна система обробки замовлень та аналізу
попиту споживачів

Спеціальність 123 – Комп'ютерна інженерія
Галузь знань 12 – Інформаційні технології

Виконавець:
студент гр. КІ–161

Довгопол Тимофій Вікторович

(підпис)

Керівник:

Бичко Володимир Анатолійович

(підпис)

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання кваліфікаційної роботи
Довгопол Т. В., гр. КІ - 161

**Тема кваліфікаційної роботи: ІНФОРМАЦІЙНО-КОМП'ЮТЕРНА
СИСТЕМА ОБРОБКИ ЗАМОВЛЕНЬ ТА АНАЛІЗУ ПОПИТУ
СПОЖИВАЧІВ**

Передбачувані технічні та експлуатаційні результати роботи

Інформаційно-комп'ютерна система обробки замовлень та аналізу попиту споживачів представлена у вигляді веб сайту (інтернет-магазину) як SPA додатку. Сайт планується бути кросбраузерним та адаптованим під мобільні пристрої. Буде використано такий стек технологій: HTML5, CSS3, SCSS, Bootstrap4, JavaScript, jQuery, PHP, RedBeanPHP, JSON, MySQL, Webpack та сторонні плагіни. Під час розробки, збереження версій проекту буде надано SCV GIT та GitHub.

Метою даної роботи є:

- Закріплення знань за допомогою практики у використанні актуальних інструментів для розробки сайтів.
- Розробка інтернет-магазину та збір даних користувачів для статистики.
- Поєднати знання у сфері дизайну, фронтенду та бекенду.

Обсяг текстової та графічної документації

Пояснювальна записка до роботи обсягом 80-100 сторінок друкованого тексту формату А4. Обсяги текстової інформації можуть бути скориговані в процесі роботи за погодженням з керівником.

Передбачувана трудомісткість роботи – 500 люд-годин.

Плановий термін захисту роботи

Робота планується до захисту _____ 2020 р.

Виконавець роботи
студент групи КІ - 161

Довгопол Т. В.

Керівник роботи

Бичко В. А.

Дата отримання завдання

«__» _____ 2020 р.

РЕФЕРАТ

Дипломна робота, 79 с., 27 рис., 1 табл., 19 джерел.

Інформаційно-комп'ютерна система обробки замовлень та аналізу попиту споживачів представлена у вигляді веб сайту (інтернет-магазину) як SPA додатку. Сайт планується бути кросбраузерним та адаптованим під мобільні пристрої. Буде використано такий стек технологій: HTML5, CSS3, SCSS, Bootstrap4, JavaScript, jQuery, PHP, RedBeanPHP, JSON, MySQL, Webpack та сторонні плагіни. Під час розробки, збереження версій проекту буде надано SCV GIT та GitHub.

Метою даної роботи є:

- Закріплення знань за допомогою практики у використанні актуальних інструментів для розробки сайтів.
- Розробка інтернет-магазину та збір даних користувачів для статистики.
- Поєднати знання у сфері дизайну, фронтенду та бекенду.

В останній час при створенні нового сайту все більшим вибором становиться односторінковий сайт-додаток, де динамікою відображення займається JavaScript з JavaScript-компонентами, які можуть бути повторно використані. Так як фактично сторінка HTML тільки одна, і це заощаджує час, який був би використаний для з'єднання з сервером та завантаженням наступного HTML файлу, роутинг, реалізований також через JavaScript, допомагає користувачу не бути обмеженим у своїх діях.

THE ABSTRACT

Diplom work, 79 pp., 27 fig., 1 tab., 19 sources.

Information and computer system for order processing and analysis of consumer demand is presented in the form of a website (online store) as a SPA application. The site is planned to be cross-browser and adapted for mobile devices. The following technology stack will be used: HTML5, CSS3, SCSS, Bootstrap4, JavaScript, jQuery, PHP, RedBeanPHP, JSON, MySQL, Webpack and third-party plug-ins. During development, saving versions of the project will be provided to SCV GIT and GitHub.

The purpose of this work is:

- Consolidation of knowledge through practice in the use of relevant tools for site development.
- Development of an online store and collection of user data for statistics.
- Combine knowledge in the field of design, frontend and backend.

Recently, when creating a new site, more and more choice is a one-page application site, where the dynamics of the display is JavaScript with JavaScript-components that can be reused. Because there is actually only one HTML page, which saves time that would be used to connect to the server and load the next HTML file, routing, also implemented through JavaScript, helps the user not to be limited in their actions.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 АНАЛІЗ ЗАДАЧІ СТВОРЕННЯ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ТА АНАЛІЗУ ПОПИТУ СПОЖИВАЧІВ	
1.1 Аналіз предметної області	9
1.1.1 Поняття і сутність Інтернет-магазину, переваги його використання.....	9
1.1.2 Аналіз уже існуючих магазинів на дану тематик.....	12
1.2 Аналіз вимог до інтерфейсу системи.....	13
1.3 Постановка задачі на розробку ІКС.....	14
1.3.1 Функціональні вимоги до системами.....	16
1.3.2 Не функціональні вимоги до системи.....	16
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ТА АНАЛІЗУ ПОПИТУ СПОЖИВАЧІВ... 17	
2.1 Вибір технічних засобів	17
2.1.1 Фронтенд	17
2.1.1.1 Гіпертекстова мова HTML.....	17
2.1.1.2 Каскадні таблиці стилів CSS	24
2.1.1.3 Графічний редактор PhotoShop	28
2.1.1.4 Мова сценаріїв JavaScript.....	31
2.1.1.5 Бібліотека Bootstrap.....	35
2.1.2 Бекенд.....	36
2.1.2.1 MySQL	36
2.1.2.2 Скриптова мова програмування PHP	37
2.1.2.3 JSON.....	43
2.1.2.4 Бібліотека RedBean	45
2.2 Розробка інтерфейсу.....	46

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ТА АНАЛІЗУ ПОПИТУ СПОЖИВАЧІВ.....	50
3.1 Підготовка бекенду, ініціалізація бази даних	50
3.2 Опис роботи Web-сайту	56
3.3 Принцип роботи ІКСОЗТАПС.....	57
3.4 Реалізація роутеру.....	62
3.5 Реалізація компонентів.....	65
3.6 Реалізація класу Store.....	68
3.7 Реалізація класу Backend.....	69
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	77
ВИСНОВКИ	79

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

Таблиця 1.1 – Перелік термінів

Термін	Значення
API	Application Programming Interface
JSON	JavaScript Object Notation
SCV	System Control Version
SPA	Single Page Application
СУБД	Система управління базою даних
sass	Software as a service
ІКСОЗТАПС	Інформаційно-комп'ютерна система обробки замовлень та аналізу попиту споживачів
ІКС	Інформаційно-комп'ютерна система
UI	User Interface
UX	User Experience
ОП	Оперативна пам'ять

ВСТУП

Кожен користувач комп'ютеру або смартфона користується сайтами у повсякденному житті для покупок товарів або перегляду інформації. В останній час сайти стали розділяти на шість основних типів:

- Landing Site
- Корпоративні або багатосторінкові сайти
- Інтернет-магазини
- Новини та пошукові сайти
- Інформаційні портали
- Веб-системи або saas (software as a service)

Не так давно повноцінно вийшли на арену навіть онлайн ігри, як окремий вид сайтів.

Але і у вищеперелічених видах сайтів трапилися різкі зміни. Зокрема, концепції багатосторінкових сайтів було замінено односторінковими SPA-додатками, які мають певні переваги у виді зручності розробки проекту та заощадження ресурсів користувача і його часу.

Також, в останній час було перетворено індустрію продажу, з метою скорочення витрат для підприємств на оренду та персонал, та зручності знаходження потрібного товару, порівняння цін та заощадження часу клієнта. Так і з'явилися інтернет-магазини.

1 АНАЛІЗ ЗАДАЧІ СТВОРЕННЯ СТВОРЕННЯ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ТА АНАЛІЗУ ПОПИТУ СПОЖИВАЧІВ

1.1 Аналіз предметної області

1.1.1 Поняття і сутність Інтернет-магазину, переваги його використання

Багато часу минуло з тих пір, як з'явилася «Всесвітня павутина». З часом дуже багато чого змінювалося: технічні можливості Мережі розширилися, виріс попит, Мережа перестала бути виключно для наукових або військових робіт. На сьогоднішній момент Інтернет являється величезним сховищем, в якому кожен по своєму бажанню може знайти потрібний текст, музику, навчальні матеріали, а з деякого часу і просто купити все, що необхідно - від продуктів, до програмного забезпечення. Все більше з'являється комерційних сайтів, основною метою яких є просування товарів і послуг за допомогою Інтернет.

Ось уже кілька років в Україні успішно розвивається електронна комерція. Як і в звичайній комерції, в електронній комерції можна виділити два напрямки: міжфірмове ділове партнерство і роздрібна торгівля. Перший напрямок розвивається досить успішно, так як цей спосіб є більш зручнішим та дешевшим. Другий напрямок, на відміну від першого лише починає розвиватися в нашій країні в порівнянні з аналогічним напрямком в світі. Проте, уже зараз можна сказати, що через деякий час конкуренція в роздрібній торгівлі через Інтернет стане настільки ж напруженою, як і в звичайній роздрібній торгівлі.

Однак, що ж являє собою Інтернет-магазин, як він створюється, в чому його переваги і в чому недоліки? Перш за все, Інтернет-магазин - це сайт, що містить докладний каталог товарів з описом і зображенням. Основна

відмінність від звичайного інтернет-каталогу полягає в тому, що товари, представлені в інтернет-магазині можна не тільки побачити, а й замовити, не виходячи з дому та не встаючи з місця.

Існує кілька типів інтернет-магазинів:

- магазин, в якому торгують товарами вузьконаправленої категорії;
- магазин, що торгує певним видом товарів;

Для всіх інтернет-магазинів властивий певний обов'язковий набір елементів, таких як:

Спеціалізований каталог з підрозділами, в яких представлені всі наявні в наявності товари. Зовнішній вигляд каталогу може бути різним: список, дерево або вкладені списки меню.

Система реєстрації користувача, яка створює для кожного нового клієнта його власну «кошик», в яку можна «покласти» обраний товар і згодом замовити.

Система оплати товару: покупцеві пропонується використовувати різні способи оплати - кредитні карти, електронні гроші, оплата готівкою (кур'єру або при отриманні на пошті).

Система доставки товару: тут теж широкий вибір можливостей: пересилання по електронній пошті (програмне забезпечення), доставка кур'єрською службою, звичайна пошта.

Однак, незважаючи на загальні риси, Інтернет-магазини все ж таки відрізняються один від одного. Власник кожного магазину прагне зробити свій сайт максимально зручним для відвідувача, удосконалюючи систему замовлення та способи переходу від одного розділу до іншого. Як і в звичайному магазині, в Інтернет-магазині можуть влаштовуватися розпродажі та знижки. Головна відмінність Інтернет-магазину від звичайного магазину полягає не тільки в можливості купити щось, не виходячи з дому чи офісу, а також можливість заощадити кошти. Через це виходить так, що покупка в Інтернет-магазині стає краще ніж в звичайному.

Для створення Інтернет-магазину не потрібно купувати або орендувати приміщення під магазин, ремонтувати і оформляти його, наймати штат продавців і охорону - а значить знижуються початкові витрати, а з ними і ціна товару. Тепер навіть з урахуванням доставки товар буде коштувати дешевше тільки тому, що клієнту не доведеться платити ту частину ціни, за допомогою якої продавець намагається відшкодувати витрати за щомісячне утримання магазину і штату співробітників.

Інтернет магазин має такі переваги:

- швидка орієнтація в асортименті;
- можна розглянути товар «з усіх боків»;
- порівняння характеристик товару, ціни, зовнішнього вигляду з іншими товарами;
- перегляд інформації про знижки та акції;
- можна розрахувати точну вартість замовлення;
- оформлення замовлення online;
- оформлення доставки на будинок;
- ведення переписки;
- економія коштів;

Аудиторія Інтернет-магазину не обмежується територією прилеглого району або міста, при відповідному обслуговуванні розмір аудиторії не обмежиться навіть країною.

Незважаючи на технічні складності, можемо зробити висновок, що створити Інтернет-магазин простіше і дешевше. До того ж «онлайн» магазин буде хорошим доповненням і рекламою уже існуючої діяльності. А головне - покупцями стануть ті, у кого з якихось причин немає можливості або часу вийти в звичайний магазин, та й ті щасливці, кому подобається робити покупки, не встаючи з місця. Таких людей стає все більше і більше.

1.1.2 Аналіз уже існуючих магазинів на дану тематику

Дуже складно підібрати точні кольори з 16 777 216 можливих, які будуть гармонічними між собою. Тому подивимося на вже існуючі популярні інтернет-магазини і підберемо стиль, який сподобається.

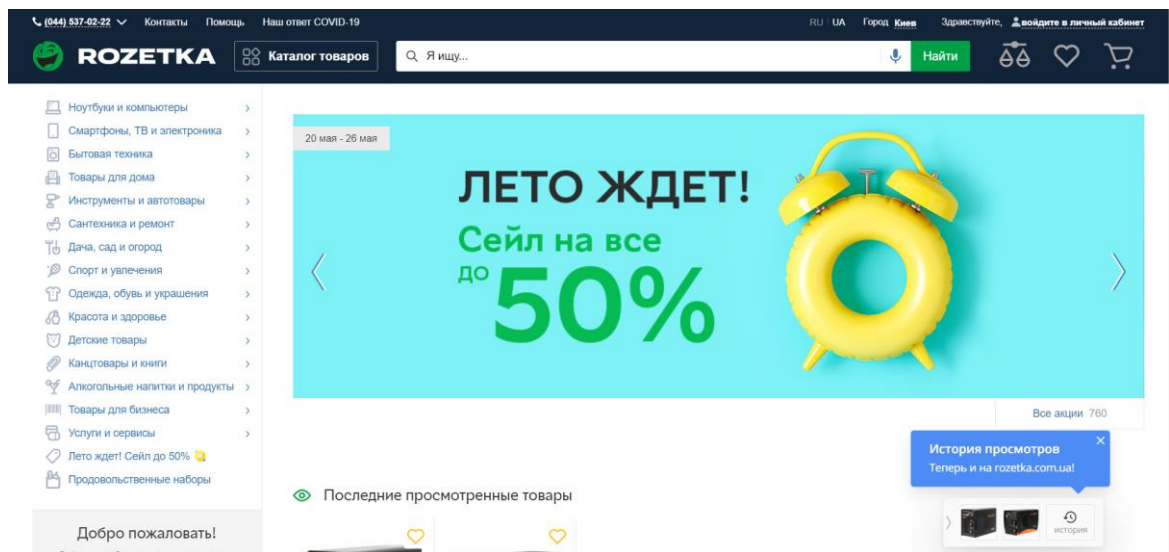


Рисунок 1.1.2.1 – Головна сторінка сайту «Розетки»

За допомогою програми ColorMania візьмемо зелений та темно-синій кольори. Це #00A046 та #122538 в форматі Нех відповідно.

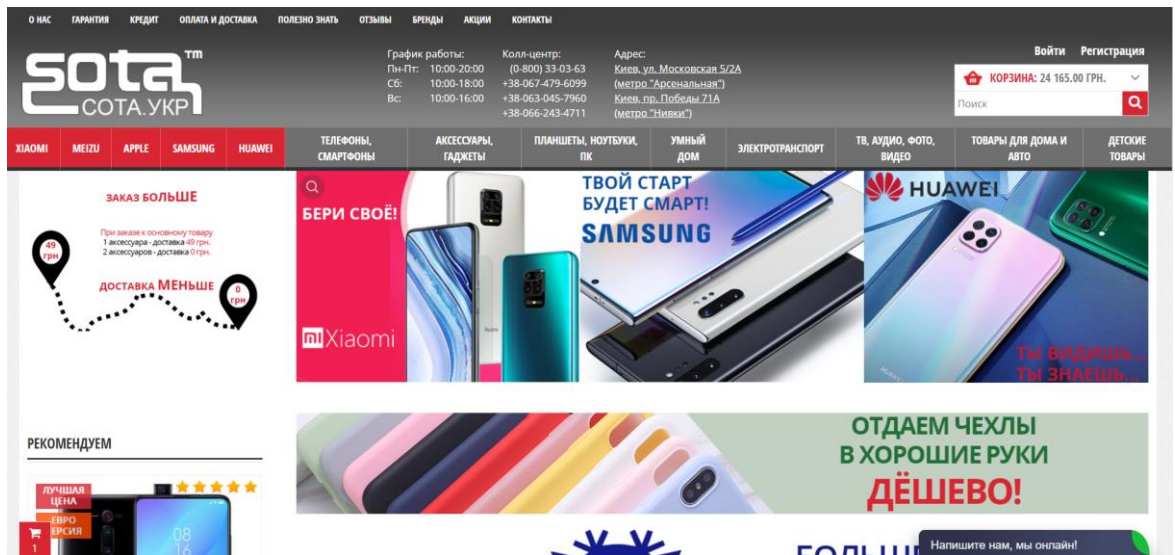


Рисунок 1.1.2.2 – Головна сторінка сайту «Сота.укр»

Дизайн добре зроблений у Розетки, а багатий функціонал має Сота. Сота скоріш за все використовує бутстрап під копотом. Розетка має не тільки гармонічні кольори, але і яскраву анімацію та ладери при AJAX запросах.

Обидва сайти мають багато сторінок, які підгружаються з беку, і це займає деякий час. Скоріш за все не використовується компонентних підхід, так як розширення в Хромі не показує наявності React у сайтів. Ми також не будемо використовувати бібліотеку React, але сами напишемо функціональні компоненти, які зможуть бути перевикористані. Роутинг у сайтів стандартний, реалізований через тег `<a>` з відповідним до потреб атрибутом `href`. У проєкті ІКСОЗТАПС використаємо інструмент `addEventListener` ('popstate') який буде брати хеш у стрічці домену за допомогою `location.state`. Зміну стеїту реалізуємо через `history.pushState()`.

1.2 Аналіз вимог до інтерфейсу системи

Користувачами системи є люди різної вікової категорії, статі, статусу, можливостей та рівня комп'ютерного досвіду.

Зважаючи на ці умови робимо висновок, що інтерфейс повинен бути стриманим, без зайвих деталей і простим у використанні. Текст не має бути

маленьким, а середнього розміру, мова інтерфейсу – роїська. З кожної сторінки сайту повинен бути доступ до меню.

Інтерфейс покупця та адміністратора відрізнятиметься можливими діями.

1.3 Постановка задачі на розробку ІКС

Сформулюємо основні вимоги до системи та її функціональності. Система повинна виконувати такі функції:

- Додавати користувача;
- Перегляд товарного каталогу;
- Вибір товарів;
- Перегляд інформації про середовище існування;
- Додавання до кошика;
- Оформлення замовлення;
- Відправлення повідомлення із замовленням;

Додатком будуть користуватися:

- Адміністратор – має необмежений прямий доступ до бази через систему управління базами даних і до вихідного коду додатку з метою виправлення можливих помилок , усунення несправностей та покращення роботи системи, а також замовлення нового товару, інвентаризація та формування звіту продаж.

Діаграма варіантів використання для адміністратора представлена на рисунку 1.3.1

- Покупець - перегляд наявних у каталозі асортиментів та їх замовлення.

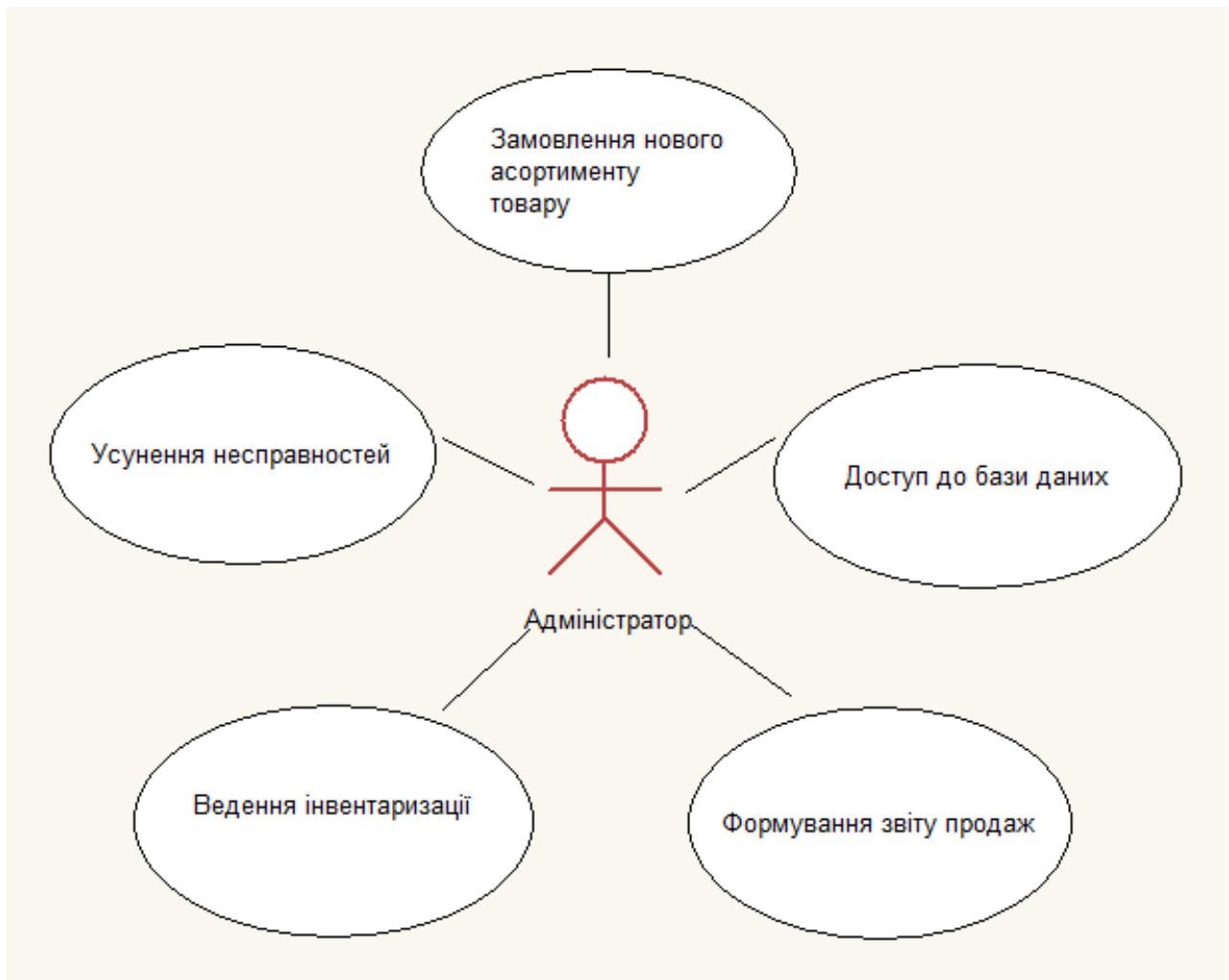


Рисунок 1.3.1 - Діаграма варіантів використання для адміністратора

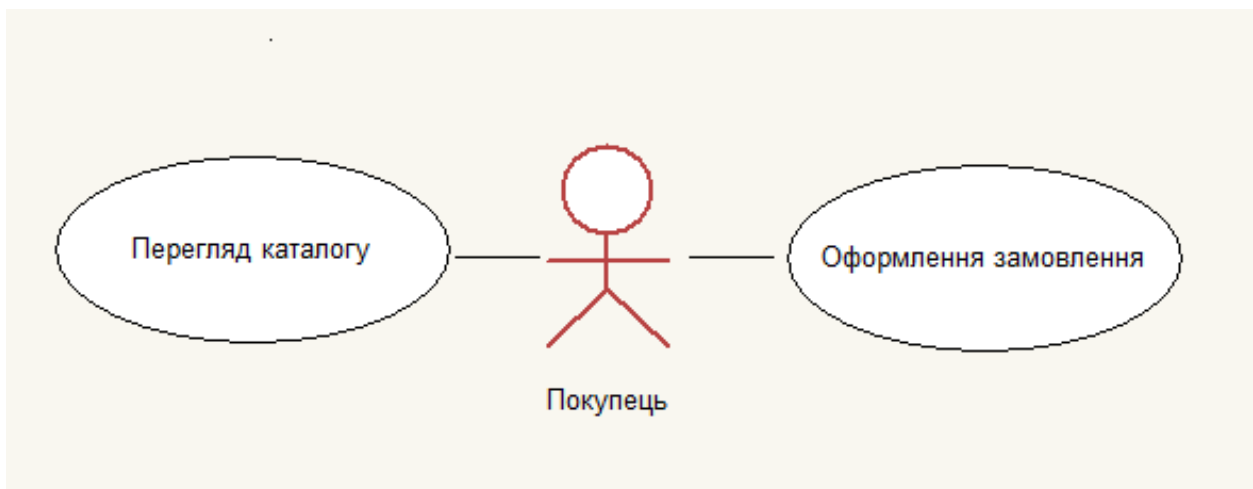


Рисунок 1.3.2 - Діаграма варіантів використання для покупця

1.3.1 Функціональні вимоги до системи

З аналізу поставленого завдання можна виділити такі функціональні вимоги:

- Можливість перегляду товарного каталогу та вибір товарів;
- Реєстрація покупця;
- Сучасний інтерфейс;
- Вибір форми оплати та доставки товару;
- Відправлення повідомлення з оформленим замовленням;

1.3.2 Не функціональні вимоги до системи

- Логічне структурне та функціональне розміщення компонентів на сторінці додатку;

- Перегляд товарів;
- Система має бути швидкодіючою, працювати з мінімальними затримками;

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ТА АНАЛІЗУ ПОПИТУ СПОЖИВАЧІВ

2.1 Вибір технічних засобів

2.1.1 Фронтенд

2.1.1.1 Гіпертекстова мова HTML

HTML (англ. HyperText Markup Language — мова розмітки гіпертексту) — це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет.

Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.

Структура HTML-коду:

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
</head>  
  
<body>  
  
<header class="header">  
  
<div class="header-menu-wrap">  
  
<ul>  
  
<li></li>  
  
<li></li>
```

```
</ul>  
</div>  
</header>  
<main class="main">  
</main>  
<footer class="footer">  
</footer>  
</body>  
</html>
```

Теги. Типи тегів. Значення атрибутів тегів:

<div> - блок

<body> - тіло

 - маркерований список

 - елемент маркерованого списку

<a> - посилання

 - фото

<p> - абзац

<button> - кнопка

class - атрибут, що дає змогу працювати більш конкретно з цим тегом в подальшому

href - посилання на елемент для тегу <a> src - посилання на елемент для тегу

alt - підпис зображення, якщо картинка не завантажилась placeholder - атрибут для підказки перед введенням даних до кнопки

Робота з текстом. Посилання:

```
<p><a href="#">ГАРАНТИЯ</a></p>
```

```
<p><a href="#">ОПЛАТА И ДОСТАВКА</a></p>
```

```
<p><a class="filter-discount" href="#">АКЦИИ</a></p>
```

```
<p><a class="feedback-button" href="#">ОТЗЫВЫ</a></p>
```

```
<p class="page-footer px-2 news-footer">ПОДПИСАТЬСЯ НА НОВОСТИ  
ОБ АКЦИЯХ</p>
```

```
<p class="page-footer mr-3">ТЕЛЕФОН/ЕМАИ</p>
```

Робота з зображеннями. Списки:

```
<ul class="nav mr-auto d-flex flex-column">
```

```
<li class="nav-item">
```

```
<a class="nav-link active border-bottom-line" href="#">ЛИЧНЫЙ  
КАБИНЕТ</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link border-bottom-line" href="#">КОРЗИНА</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link border-bottom-line" href="#">ЖЕЛАНИЕ</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link border-bottom-line" href="#">ГАРАНТИЯ</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link border-bottom-line" href="#">ОПЛАТА И
ДОСТАВКА</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link border-bottom-line" href="#">КОНТАКТЫ</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link border-bottom-line" href="#">АКЦИИ</a>
```

```
</li>
```

```
</ul>
```

```

```

```

```

```

```

Таблиці:

```
<table class="table table-goods">
```

```
<tbody><tr class="table-goods-th">
```

```
<td class="padding-img-photo">Фото</td>
```

```
<td>Товар</td>
```

```
<td>Кол-во</td>
```

```
<td>Цена</td>
```

```
<td>Итого</td>
```

```
</tr><tr>
```

```
<td class="pl-0">
```

```

<div class="img-photo">
  
</div>
</td>
<td>Xiaomi Redmi Note 8 Pro</td>
<td>
  <div class="d-flex" data-id="8">
    
    <input class="amount" value="1" placeholder="1">
    
  </div>
</td>
<td>6 500 ГРН</td>
<td class="wrap-price-for-all"><span class="price-for-all-8">6 500</span>
ГРН</td>
</tr>
</tbody></table>

```










Фото	Товар	Кол-во	Цена	Итог
	Xiaomi Redmi Note 8 Pro	 <input type="text" value="1"/> 	6 500 ГРН	6 500 ГРН
	Xiaomi Redmi Note 8 Pro	 <input type="text" value="1"/> 	6 500 ГРН	6 500 ГРН
	Mi band 4	 <input type="text" value="1"/> 	200 ГРН	200 ГРН

Рисунок 2.1.1.1.1 – Приклад таблиці

HTML5:

```

<header class="header">
<div class="row main-navigation-row">
<div class="header-menu-wrap">
<div class="header-chose-menu col-6 ProximaNovaExCn-Regular">
<ul class="nav ml-3 pt-3">
<li class="nav-item">
<a class="nav-link px-2 border-bottom-line" href="#">ГАРАНТИЯ</a>
</li>
<li class="nav-item">
<a class="nav-link px-2 border-bottom-line" href="#">ОПЛАТА И
ДОСТАВКА</a>
</li>
<li class="nav-item">
<a class="nav-link px-2 border-bottom-line" href="#">КОНТАКТЫ</a>
</li>
<li class="nav-item">
<a class="filter-discount nav-link px-2 border-bottom-line"
href="#">АКЦИИ</a>
</li>
<li class="nav-item">
<a class="feedback-button nav-link px-2 border-bottom-line"
href="#">ОТЗЫВЫ</a>
</li>
</ul>

```

```

</header>

<main class="main">

</main>

<footer class="footer">

<div class="footer-1 row d-flex justify-content-between DINPro-Medium pt-3 mr-0">

<div class="col-5 d-flex text-white">

<div class="col-6 ml-4">

<p class="page-footer">НАШИ КОНТАКТЫ</p>

<hr class="hr-footer-1">

</footer>

```



Рисунок 2.1.1.1.2 – Приклад хедеру

ВСЕ ТОВАРЫ



Рисунок 2.1.1.1.3 – Приклад main



Рисунок 2.1.1.1.4 – Приклад футеру

2.1.1.2 Каскадні таблиці стилів CSS

CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних.

CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript.

Базовий синтаксис CSS. Значення стильових властивостей.

```

селектор {
свойство: значение;

}

```

```

.header .header-input { padding-right: 23px; margin-top: 10px; margin-right: -
10px; color: white;

```

```

width: 149px; height: 23px;

```

```

border-radius: 10px; border: 1px solid #ffffff; background: none; outline:
none;

```

```

opacity: 0.65; transition: 0.3s;

```

```

}

```

```

.header .header-input:focus { opacity: 1;

```

```

width: 250px;

```

```

border: 1px solid #80BDFE;

```



```
box-shadow: 0 0 0 3px #0D3B6A;
}
```



2.1.1.2 – Приклад інпуту

Класи. Селектори.

```
img[src="img/Эллипс.png"] { position: relative;
top: 8px; right: 4px; display: none;
}
.input-footer {
transition: box-shadow 0.3s;
}
```

Селектори атрибутів. Універсальний селектор.

```
.header img[alt="search"] { margin-left: 10px; width: 22px;
height: 22px; margin-top: 10px; cursor: pointer;
transition: transform .3s;
}
*, *:after, *:before {
-webkit-box-sizing: border-box; box-sizing: border-box;
}
```

Псевдокласи. Псевдоелементи.

```
.header .header-input:focus { opacity: 1;
width: 250px;
```

```

border: 1px solid #80BDFF;

box-shadow: 0 0 0 3px #0D3B6A;

}

#icon:before { content: ";

display: inline-block; width: 14px;

height: 2px;

background: var(--cross); transition: all var(--time) ease; position: absolute;

margin-left: -6px; transform: rotate(180deg); border-radius: 50px;

}

html::-webkit-scrollbar { width: 5px;

height: 5px;

background-color: black;

}

```

Групування. Наслідування. Каскадування.

```

.header a.border-bottom-line:hover,

.header a.border-bottom-line:hover:after,

.header .nav-link.active:after { color: #00a046;

width: 100%;

}

.nav-link, .footer a, .registration-hover { transition: color 0.3s;

}

```

Блочна модель. Блочні, рядкові, рядково-блочні елементи.

```

.section-good-name .old-price::after { content: ";

position: relative; top: -27px; display: block; height: 20px;

```

```

background: url("../img/through.svg");
}

.basketMenu img[alt="cancel"] { cursor: pointer;
width: 12px;
display: inline-block;
}

#icon:after { content: "";
display: inline-block; width: 2px;
height: 14px; background: var(--cross);
transition: all var(--time) ease; position: absolute;
margin-top: -6px; transform: rotate(360deg); border-radius: 50px;
}

```

Потік документа. Плаваючі елементи. Позиціонування елементів.

```

.sign-in-form > img[src="img/cancel.svg"], .sign-up-form >
img[src="img/cancel.svg"] {
position: relative; top: -37px;
left: 103%; width: 24px; cursor: pointer;
}

.alert-window { position: fixed; bottom: 0;
right: 0; padding: 10px; width: 250px;
word-break: break-word; border-radius: 5px; cursor: default;
}

.basket-wrap { position: absolute; z-index: 100; right: 110px; width: 350px;
}

```

2.1.1.3 Графічний редактор PhotoShop

Adobe Photoshop — графічний редактор, розроблений і поширюваний фірмою Adobe Systems. Цей продукт є лідером ринку в галузі комерційних засобів редагування растрових зображень і найвідомішим продуктом фірми Adobe. Часто цю програму називають просто Photoshop (Фотошоп). У наш час Photoshop доступний на платформах Mac OS X/Mac OS і Microsoft Windows

Підготовка зображень для сайту

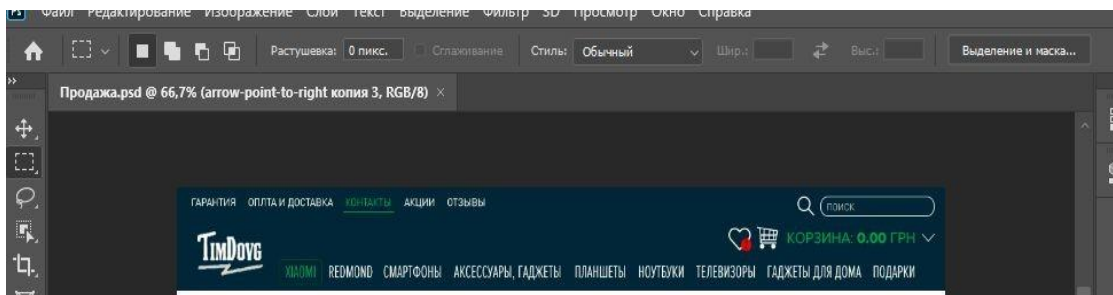


Рисунок 2.1.1.3.1 – Створення хедеру в PhotoShop

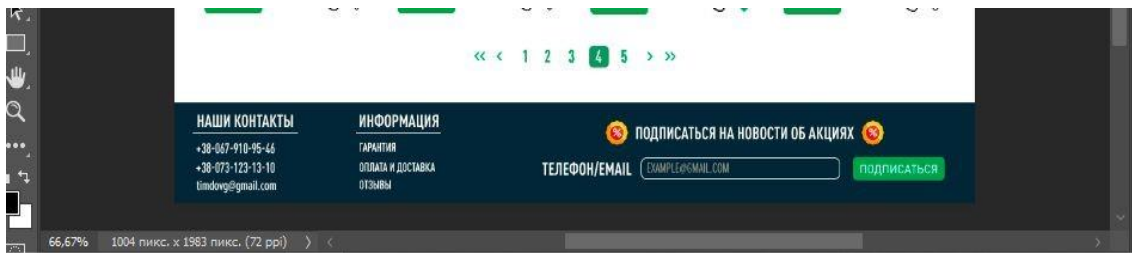




Рисунок 2.1.1.3.2 – Створення футеру для сайту в програмі PhotoShop

ГАРАНТИЯ ОПЛАТА И ДОСТАВКА **КОНТАКТЫ** АКЦИИ ОТЗЫВЫ

TImDovG
XIAOMI
REDMOND
СМАРТФОНЫ
АКСЕССУАРЫ, ГАДЖЕТЫ
ПЛАНШЕТЫ
НОУТБУКИ
ТЕЛЕВИЗОРЫ
ГАДЖЕТЫ ДЛЯ ДОМА
ПОДАРКИ

НАУШНИКИ С МИКРОФОНОМ APPLE AIRPODS MMEF2

★★★★★

Товар в наличии на складе

Цвет

3 600 ГРН

КУПИТЬ В ОДИН КЛИК
В КОРЗИНУ

ОБРАТНЫЙ ЗВОНОК

ДОБАВИТЬ В ИЗБРАННОЕ

ХАРАКТЕРИСТИКИ

<p>Процессор</p> <p>Режим разговора</p> <p>Интерфейсы</p> <p>Вес, г</p> <p>Материал</p> <p>Цвет</p> <p>Другие функции</p> <p>Совместимость</p> <p>Комплектация</p>	<p>Apple W1</p> <p>До 2 ч</p> <p>Bluetooth</p> <p>4 (каждый)</p> <p>Пластик</p> <p>Белый</p> <p>Встроенный в наушник микрофон, Регулировка громкости, Контроль звонков</p> <p>Apple</p> <p>Наушники, Чехол с зарядным устройством, Кабель Lightning to USB</p>
--	--

НАШИ КОНТАКТЫ

+38-067-910-95-46

+38-073-123-13-10

timdovg@gmail.com

ИНФОРМАЦИЯ

ГАРАНТИЯ

ОПЛАТА И ДОСТАВКА

ОТЗЫВЫ

ПОДПИСАТЬСЯ НА НОВОСТИ ОБ АКЦИЯХ

ТЕЛЕФОН/EMAIL

Рисунок 2.1.1.3.3 – Макет сторінки з товаром

Верстання головної веб-сторінки із шаблону psd

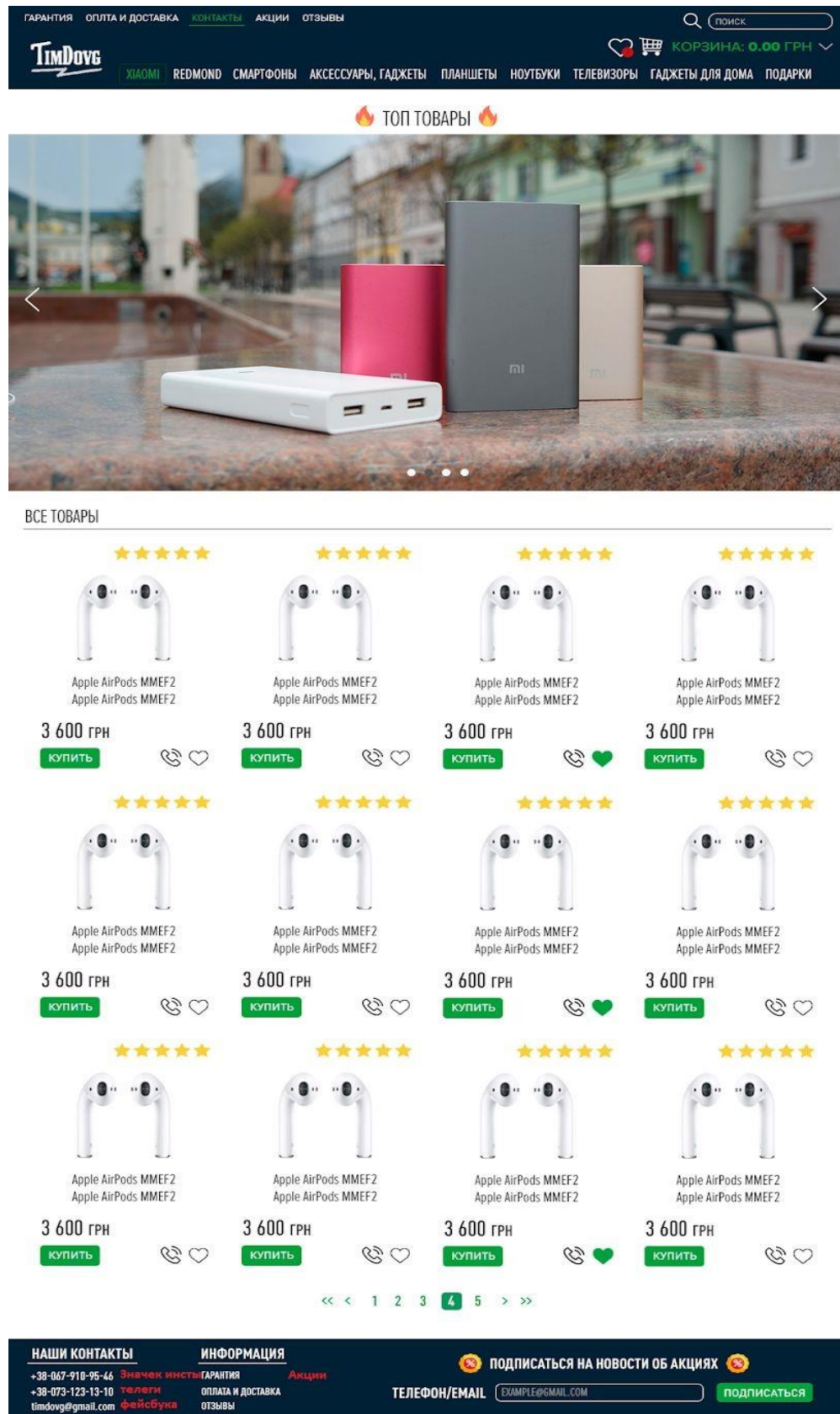


Рисунок 2.1.1.3.4 – Головна сторінка сайту

2.1.1.4 Мова сценаріїв JavaScript

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

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

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

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

Масиви. Властивості і методи рядкових об'єктів. Функції.

```

document.querySelectorAll('.cart .amount').forEach(inputQuantity => {
inputQuantity.addEventListener('keyup', e => {
  const numbers = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
  'Backspace', 'ArrowUp', 'ArrowDown'];
  const currentValue = inputQuantity.value.trim(); Backend.getWishes(email)
  .then(wishes => {
  let goodsFromWishes = [];

  wishes.forEach(wish => goodsFromWishes.push(store.goods[wish]));
store.wishes = goodsFromWishes;
  main.innerHTML = renderHeaderForGoods('СПИСОК ЖЕЛАНИЙ',
store.wishes);
  renderPartOfGoodsByPage(0,18,1, store.wishes);
  })
;
}
$('.good-image').imagezoomsl({
  zoomrange: [1, 4],
  zoomstart: 2, innerzoom: true, magnifierborder: "none"
});
let newPrise = "";
summaryPrise = summaryPrise.toString().split("").reverse().join(""); for (let i =
0; i < summaryPrise.length; i++) {
  if (i % 3 === 0 && i !== 0) { newPrise += ' ' + summaryPrise[i];
  } else {
  newPrise += summaryPrise[i];
  }
}
}

```



```
return newPrise.split("").reverse().join("");
};
```

Бібліотека jQuery

jQuery — популярна JavaScript-бібліотека з відкритим кодом. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресігом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день.

Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

```
if (document.querySelector('.alert-window')) {
document.querySelector('.alert-window').remove();
}
```

```
const alertWindow = document.createElement('div');
```

```
alertWindow.classList.add('alert-window'); alertWindow.style.color = color;
alertWindow.innerHTML = `

```

```

    const timer = setTimeout(() => { alertWindow.style.opacity = '0.9';
window.clearTimeout(timer);
    }, 0);
    const timer2 = setTimeout(() => { alertWindow.remove();
window.clearTimeout(timer2);
    }, time);

    document.querySelector('.alert-window >
img[src="img/cancel.svg"]').addEventListener('click', () => {
    alertWindow.remove();
    });
}

export default function back(backHistory = true) { if
(document.querySelector('.form-wrap')) {
    document.querySelector('.form-wrap form').setAttribute('style', 'transform:
scale(0)');
    document.querySelector('.form-wrap form').addEventListener('transitionend',
() => {
        document.querySelector('.form-wrap').remove();
document.querySelector('.header').classList.remove('blur');
document.querySelector('.main').classList.remove('blur');
document.querySelector('.footer').classList.remove('blur'); if (backHistory)
history.back();
    });
}
}

```

Функції та методи числових об'єктів

```

    constructor: f Number() toExponential: f toExponential() toFixed: f
toFixed() toLocaleString: f toLocaleString() toPrecision: f toPrecision() toString:
f toString()

```

```
summaryPrise = summaryPrise.toString().split("").reverse().join("");
```

```

for (let i = 0; i < summaryPrise.length; i++) { if (i % 3 === 0 && i !== 0) {
newPrise += ' ' + summaryPrise[i];
} else {
newPrise += summaryPrise[i];
}
}
}

```

```
return newPrise.split("").reverse().join("");
```

```
};
```

2.1.1.5 Бібліотека Bootstrap

Bootstrap — це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків.

Bootstrap — це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері. Репозиторій із цим фреймворком є одним із найпопулярніших на GitHub. Серед інших, його використовують NASA і MSNBC.

```
<div class="col-6 ml-4 pb-3 text-black"></div>
```

```
col-6 - flex: 0 0 50%;
```

```
ml-4 - margin-left: 1.5rem;
```

```
pb-3 - padding-bottom: 1rem;
text-black - color: black;
```

2.1.2 Бекенд

2.1.2.1 MySQL

MySQL — вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

```
CREATE TABLE shop (
  article INT(4) UNSIGNED ZEROFILL DEFAULT '0000' NOT NULL,
  dealer CHAR(20)          DEFAULT '' NOT NULL,
  price DOUBLE(16,2)      DEFAULT '0.00' NOT NULL,
  PRIMARY KEY(article, dealer));
```

```
INSERT INTO shop VALUES
(1,'A',3.45),(1,'B',3.99),(2,'A',10.99),(3,'B',1.45),(3,'C',1.69),
(3,'D',1.25),(4,'D',19.95);
```

```
+-----+-----+-----+-----+-----+
| ID | NAME  | AGE | ADDRESS | SALARY |
+-----+-----+-----+-----+-----+
| 1 | Maxim | 35 | Moscow  | 21000.00 |
```

```
| 2 | AndreyEx | 38 | Krasnodar | 55500.00 |
| 3 | Oleg      | 33 | Rostov    | 34000.00 |
| 4 | Masha     | 35 | Moscow    | 31500.00 |
| 5 | Ruslan    | 34 | Omsk      | 43000.00 |
| 6 | Dima      | 32 | SP        | 45000.00 |
| 7 | Roma      | 34 | SP        | 10000.00 |
+----+-----+-----+-----+-----+
```

2.1.2.2 Скриптова мова програмування PHP

Базові поняття PHP. Основні конструкції PHP.

```
<?PHP
if ('application/json' == $_SERVER['CONTENT_TYPE'] && 'POST' ==
$_SERVER['REQUEST_METHOD'])
{
$data = json_decode(file_get_contents('php://input') , true);
}
require "db.php";
$user = R::findOne('users', "email = ?", array(
$data['email']
));
if ($user)
{
$user->cart_praise = $data['newPrise']; R::store($user);
}
else
{
$user = R::findOne('unnamedusers', "email = ?", array(
$data['email']
```

```
));
```

```
$user->cart_price = $data['newPrice']; R::store($user);
```

```
}
```

```
?>
```

```
<?PHP
```

```
if ('application/json' == $_SERVER['CONTENT_TYPE'] && 'POST' ==
```

```
$_SERVER['REQUEST_METHOD'])
```

```
{
```

```
$data = json_decode(file_get_contents('php://input') , true);
```

```
}
```

```
require "db.php"; class ObjectError
```

```
{
```

```
var $error;
```

```
}
```

```
class Answer
```

```
{
```

```
var $email; var $token;
```

```
}
```

```
$errors = array();
```

```
$user = R::findOne('users', "email = ?", array(
```

```
$data['email']
```

```
));
```

```
if ($user)
```

```
{
```

```
// email найден
```

```
if (password_verify($data['password'], $user->password))
```

```
{
$answer = new Answer;
$answer->email = $data['email'];
$answer->token = $user->token;

echo json_encode($answer);
}
else
{
$errors[] = 'Неверно введен пароль!';

}
}
else
{
$errors[] = 'Пользователь с данным Email не найден!';
}

if (!empty($errors))
{
$errorShow = new ObjectError;
$errorShow->error = array_shift($errors); echo json_encode($errorShow);
}
?>
```

Работа с массивами в PHP.

```
$errors = array();
```

```
if (R::count('users', "email = ?", array(
$data['email']
```

```

)) > 0)
{
$errors[] = 'Пользователь с таким Email уже существует!';
}

if (empty($errors))
{
// регистрация
$token = password_hash("soCool", PASSWORD_DEFAULT);

$user = R::dispense('users');
$user->email = $data['email'];
$user->password = password_hash($data['password'],
PASSWORD_DEFAULT);
$user->token = $token; R::store($user);
echo json_encode($token);
}
else
{
$errorShow = new ObjectError;

$errorShow->error = array_shift($errors); echo json_encode($errorShow);
}
?>

{
$user = R::findOne('unnamedusers', "email = ?", array(
$data['email']
));

```



```
$wishes = explode("??", $user->wish); array_pop($wishes);
```

```
echo json_encode($wishes);
```

```
}
```

```
?>
```

Основні функції PHP.

```
function update($user)
```

```
{
```

```
if ($user)
```

```
{
```

```
$user->cart = "";
```

```
$user->cart_price = ""; R::store($user);
```

```
}
```

```
}
```

```
function insertData($user, $data)
```

```
{
```

```
$price = (int)$user->cart_price - $data['price'];
```

```
$cart = explode("??", $user->cart); array_pop($cart);
```

```
$key = array_search($data['id'], $cart); array_splice($cart, $key, 1);
```

```
$cart = implode("??", $cart);
```

```
if (strlen($cart) > 0)
```

```
{
```

```
$cart .= "??";
```

```
}
```

```

$user->cart_pride = $pride;
$user->cart = $cart; R::store($user);
}

```

Передача змінних.

```

$data = json_decode(file_get_contents('php://input') , true);
$statistic = R::findOne('statistic', "id_product = ?", array(
    $data['id']
));
$user = R::findOne('users', "email = ?", array(
    $data['email']
));
if ($data['oldEmail']) {
    $user = R::findOne('unnamedusers', "email = ?", array(
        $data['oldEmail']
    ));
    $cart = $user->cart;
    $wish = $user->wish;
    $cart_pride = $user->cart_pride;

    $userNew = R::findOne('users', "email = ?", array(
        $data['newEmail']
    ));

    $userNew->cart = $cart;
    $userNew->wish = $wish;
    $userNew->cart_pride = $cart_pride; R::store($userNew);
}
?>

```

Підключення блоків та файлів PHP.

```
<?php
require "db.php";
?>

<?php
require "libs/rb.php";
R::setup( 'mysql:host=localhost;dbname=xiaomishop', 'root', 'root' );
// автозаповнення R::freeze(true);
?>
```

З'єднання з БД із PHP. Вибірка даних з БД MySQL

```
$user = R::findOne('subscribers', "email = ?", array(
$data['email'] R::store($user); R::store($phone); R::store($statistic);
$user = R::dispense('subscribers');
if (R::count('users', "email = ?", array(
$data['email']
)) > 0)
```

```
<?php
require "libs/rb.php";
R::setup( 'mysql:host=localhost;dbname=xiaomishop', 'root', '123' );
// автозаповнення R::freeze(true);
?>
```

2.1.2.3 JSON

JSON (англ. JavaScript Object Notation, укр. запис об'єктів JavaScript, вимовляється джейсон) — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат

використовується переважно для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією).

Розробив і популяризував формат Дуглас Крокфорд.

JSON знайшов своє головне призначення в написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

```
[
  {
    "stars": 5,
    "discount": "800",
    "name": "Apple Airdots original", "prise": "1 000",
    "url": "apple-airdots",
    "category": "accessories-and-gadgets", "characteristics": [
      ["Процессор", "Apple W1"],
      ["Режим разговора", "До 2 ч"], ["Интерфейсы", "Bluetooth"], ["Вес, г", "4
(каждый)"],
      ["Материал", "Пластик"],
      ["Цвет", "Белый"],
      ["Другие функции", "Встроенный в наушник микрофон,
регулировка громкости, контроль звонков"], ["Совместимость",
"Apple"],
      ["Комплектация", "Наушники, чехол с зарядным устройством, кабель
Lightning to USB"]
    ],
  },
]
```

```

    "img": [ "img/goodsImages/Airpods/1.png",
"img/goodsImages/Airpods/2.png", "img/goodsImages/Airpods/3.png",
"img/goodsImages/Airpods/4.png", "img/goodsImages/Airpods/5.png"
    ],
    "id": 0
  }
]

```

2.1.2.4 Бібліотека RedBean

RedBeanPHP - це незалежне, безкоштовне ліцензійне програмне забезпечення з відкритим кодом, з ліцензією BSD, з відкритим вихідним кодом, програмне забезпечення для відображення з відкритим кодом.

```

public function getColumns( $table )
{
    $table = $this->esc( $table );

    $columnsRaw = $this->adapter->get( "SHOW COLUMNS FROM
    $table" );

    $columns = array();
    foreach ( $columnsRaw as $r ) {
        $columns[$r['Field']] = $r['Type'];
    }

    return $columns;
}

```

2.2 Розробка інтерфейсу

Bootstrap - вільний набір інструментів для створення сайтів і веб-додатків. Включає в себе HTML- і CSS-шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення.

До недоліків можна віднести бідну колірну гамму стандартного набору іконок. До переваг - хорошу реалізацію grid-сітки для масштабування веб-сторінки, створення адаптивного дизайну.

Основні інструменти Bootstrap:

Сітки - заздалегідь задані розміри колонок, які можна відразу ж використовувати, наприклад ширина колонки 140 px відноситься до класу .span2 (.col-md-2 в третій версії фреймворка), який можна використовувати в CSS-описі документа.

Шаблони - фіксований або гумовий шаблон документа.

Типографіка - опису шрифтів, визначення деяких класів для шрифтів, таких як код, цитати і т. П.

Медіа - представляє деякий спосіб упорядкування зображень та відео.

Таблиці - кошти оформлення таблиць, аж до додавання функціональності сортування.

Форми - класи для оформлення форм і деяких подій, що відбуваються з ними.

Навігація - класи оформлення для табів, вкладок, сторінковий, меню і панелі інструментів.

Алерт - оформлення діалогових вікон, підказок і спливаючих вікон.

Адаптивна верстка буде створена за допомогою медіа запитів в CSS.

Медіа-запит складається з ключового слова, що описує тип пристрою (необов'язковий параметр) і вирази, котрий перевіряє характеристики даного пристрою. З усіх характеристик найчастіше перевіряється ширина пристрою width. Медіа-запит є логічним виразом, яке повертає істину або брехня.

Медіа-запити можуть бути додані наступними способами:

1) За допомогою HTML:

```
<Link rel = "stylesheet" media = "screen and (color)" href = "example.css">
```

2) За допомогою правила `@import` всередині елемента `<style>` або зовнішньої таблиці стилів:

```
@import url (color.css) screen and (color);
```

3) Безпосередньо в коді сторінки:

```
<Style>
```

```
@media (max-width: 600px) {
```

```
  #sidebar {display: none;}
}
```

```
</ Style>
```

4) Всередині таблиці стилів `style.css`:

```
@media (max-width: 600px) {
```

```
  #sidebar {display: none;}
}
```

Таблиця стилів, прикріплена через тег `<link>`, буде завантажуватися разом з документом, навіть якщо її медіа-запит поверне брехня.

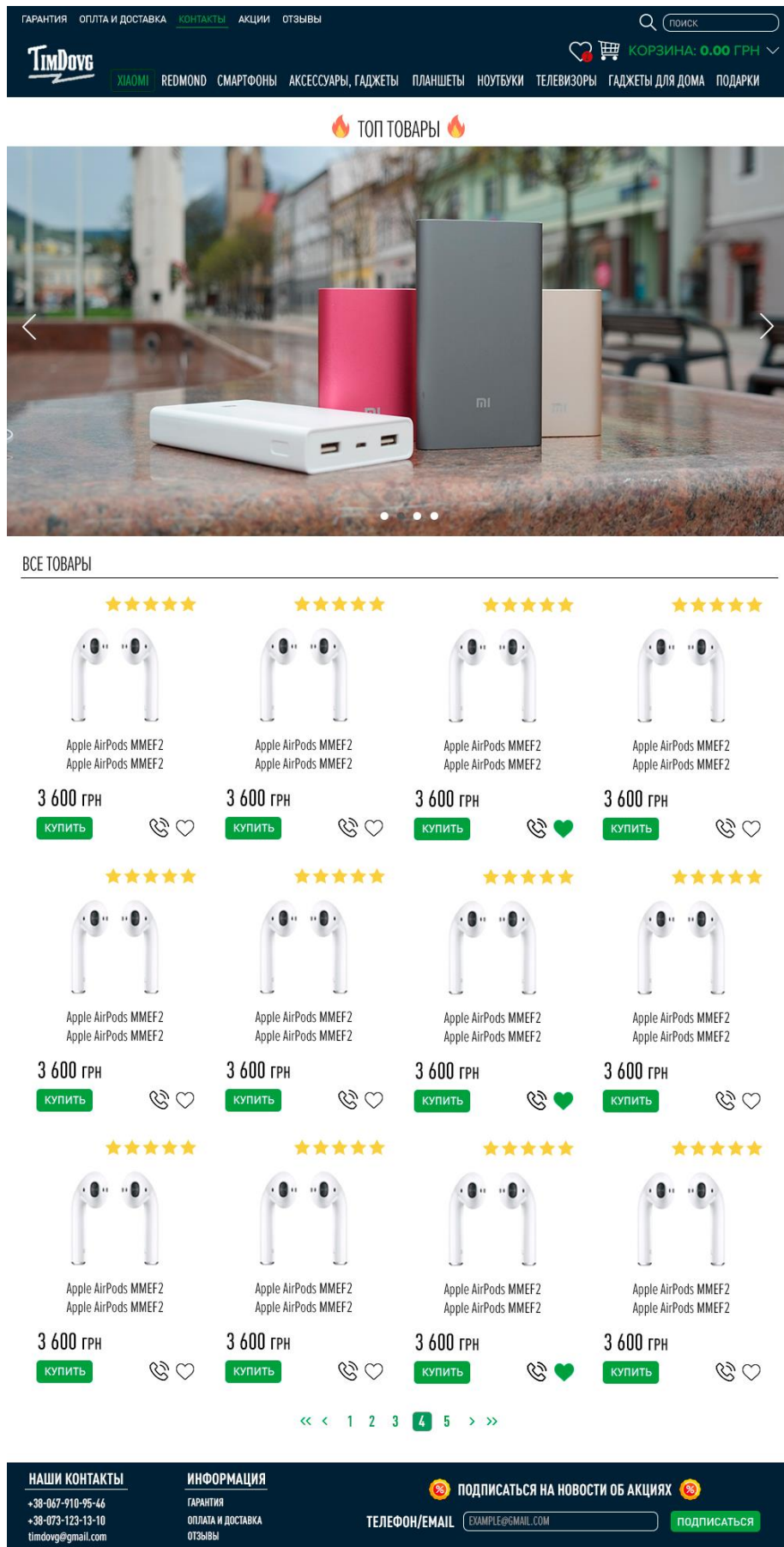


Рисунок 2.2.1 – Цілісний макет для головної сторінки

ГАРАНТИЯ ОПЛАТА И ДОСТАВКА КОНТАКТЫ АКЦИИ ОТЗЫВЫ

ХИАОМИ REDMOND СМАРТФОНЫ АКСЕССУАРЫ, ГАДЖЕТЫ ПЛАНШЕТЫ НОУТБУКИ ТЕЛЕВИЗОРЫ ГАДЖЕТЫ ДЛЯ ДОМА ПОДАРКИ

КОРЗИНА: 0.00 ГРН

НАУШНИКИ С МИКРОФОНОМ APPLE AIRPODS MMEF2

Товар в наличии на складе

Цвет

3 600 ГРН

КУПИТЬ В ОДИН КЛИК

В КОРЗИНУ

ОБРАТНЫЙ ЗВОНОК

ДОБАВИТЬ В ИЗБРАННОЕ

ХАРАКТЕРИСТИКИ

Процессор	Apple W1
Режим разговора	До 2 ч
Интерфейсы	Bluetooth
Вес, г	4 (каждый)
Материал	Пластик
Цвет	Белый
Другие функции	Встроенный в наушник микрофон, Регулировка громкости, Контроль звонков
Совместимость	Apple
Комплектация	Наушники, Чехол с зарядным устройством, Кабель Lightning to USB

НАШИ КОНТАКТЫ

+38-067-910-95-46
+38-073-123-13-10
timdovg@gmail.com

ИНФОРМАЦИЯ

ГАРАНТИЯ
ОПЛАТА И ДОСТАВКА
ОТЗЫВЫ

ПОДПИСАТЬСЯ НА НОВОСТИ ОБ АКЦИЯХ

ТЕЛЕФОН/EMAIL
ПОДПИСАТЬСЯ

Рисунок 2.2.1 – Цілісний макет для сторінки товару

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ТА АНАЛІЗУ ПОПИТУ СПОЖИВАЧІВ

3.1 Підготовка бекенду, ініціалізація бази даних

Для запуску локального веб-серверу буде використовуватись OpenServer.

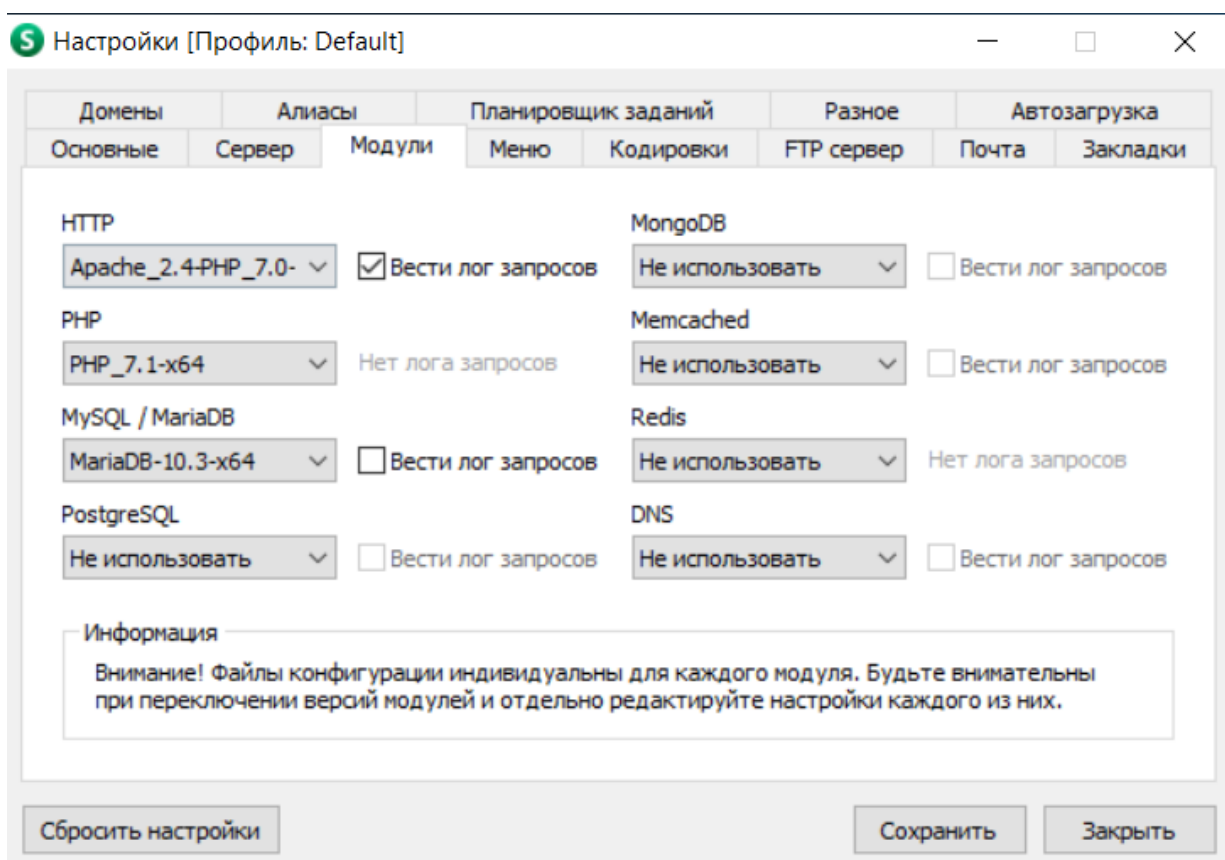


Рисунок 3.1.1 – Налаштування модулів для підняття веб-серверу

Драйвер `mysql` надає процедурний інтерфейс і призначене для використання з MySQL версії 4.1.3 або більш ранніми версіями. Його також можна використовувати і з більш свіжими версіями СУБД, але в цьому випадку багато можливостей сервера, що з'явилися після версії 4.1.3, будуть недоступні програмісту.

Драйвер `mysqli`, або як його ще називають поліпшене (improved) MySQL розширення, було розроблено, щоб дати можливість програмістам в повній мірі скористатися функціоналом MySQL сервера версій 4.1.3 і вище. Розширення `mysqli` включається в поставку PHP версій 5 і вище. Саме тому вибір зупинився на ньому.

MySQLi має ряд переваг і удосконалень у порівнянні з mysql, які полягають в наступному:

- Об'єктно-орієнтований інтерфейс
- Підтримка підготовлених запитів
- Підтримка мульти запитів
- Підтримка транзакцій
- Покращені можливості налагодження
- Підтримка вбудованого сервера

Драйвер PDO MYSQL розташовується між самим PDO і сервером MySQL. Програміст викликає функції інтерфейсу PDO API, а PDO в свою чергу використовує драйвер PDO MYSQL для обміну даними і командами з сервером MySQL. Драйвер PDO MYSQL збирається за допомогою фреймворка розширень PHP, його вихідний код розташований в директорії ext / pdo_mysql. Він не надає API-інтерфейс програмісту PHP.

Посередником між MySQL та PHP буде бібліотека RedBeanPHP. Якщо потраплятиме новий запит до таблиці, яка ще не створена, вона буде створена самостійно бібліотекою, та типи встановлюються згідно надісланих даних. Скористаємось цим, та не створимо БД “xiaomishop”.

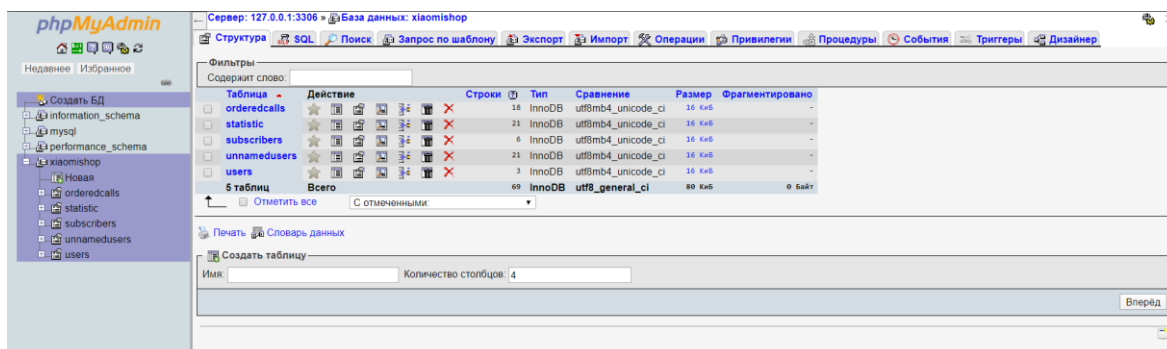


Рисунок 3.1.2 – Створення БД

Для даних, які потребують шифрування будемо використовувати password_hash:

```
password_hash($data['password'], PASSWORD_DEFAULT);
```

Для декодування:

```
password_verify($data['password'], $user->password)
```

Приклад реєстрування користувача:

```
<?PHP
if ('application/json' == $_SERVER['CONTENT_TYPE'] && 'POST' ==
$_SERVER['REQUEST_METHOD'])
{
    $data = json_decode(file_get_contents('php://input') , true);
}
require "db.php";
class ObjectError
```

```

{
  var $error;
}

$errors = array();

if (R::count('users', "email = ?", array(
  $data['email']
)) > 0)
{
  $errors[] = 'Пользователь с таким Email уже существует!';
}

if (empty($errors))
{
  // регистрация
  $token = password_hash("soCool", PASSWORD_DEFAULT);

  $user = R::dispense('users');
  $user->email = $data['email'];
  $user->password = password_hash($data['password'],
PASSWORD_DEFAULT);
  $user->token = $token;
  R::store($user);
  echo json_encode($token);
}
else
{
  $errorShow = new ObjectError;
  $errorShow->error = array_shift($errors);
  echo json_encode($errorShow);
}
?>

```

Для даних, які не потребують ніякого захисту (дані для відображення інформації) будемо використовувати формат JSON. Дані у десятків тисяч стрічок буду важити дуже мало, так як формат даних – це фактично об’єкт, який буде перетворено при спілкування з мовами у дуже довгу строку. Це будуть дані о товарах: характеристика, найменування товару, ціна та інше.

Приклад JSON файлу:

```

[
  {
    "stars": 5,
    "discount": "800",
    "name": "Apple Airdots original",

```

```

    "prise": "1 000",
    "url": "apple-airdots",
    "category": "accessories-and-gadgets",
    "characteristics": [
      ["Процессор", "Apple W1"],
      ["Режим разговора", "До 2 ч"],
      ["Интерфейсы", "Bluetooth"],
      ["Вес, г", "4 (каждый)"],
      ["Материал", "Пластик"],
      ["Цвет", "Белый"],
      ["Другие функции", "Встроенный в наушник микрофон,
регулировка громкости, контроль звонков"],
      ["Совместимость", "Apple"],
      ["Комплектация", "Наушники, чехол с зарядным устройством,
кабель Lightning to USB"]
    ],
    "img": [
      "img/goodsImages/Airpods/1.png",
      "img/goodsImages/Airpods/2.png",
      "img/goodsImages/Airpods/3.png",
      "img/goodsImages/Airpods/4.png",
      "img/goodsImages/Airpods/5.png"
    ],
    "id": 0
  },
  {
    "stars": 4,
    "discount": "200",
    "name": "Mi band 4",
    "prise": "700",
    "url": "mi-band-4",
    "category": "accessories-and-gadgets",
    "characteristics": [
      ["Разъемы", "3 шт."]
    ],
    "img": [
      "img/goodsImages/Mi Band 4/1.jpg",
      "src2",
      "src3",
      "src4",
      "src5",
      "src6"
    ],
    "colors": ["red", "blue", "gray"],
    "id": 1
  }

```

```

},

{
  "name": "Xiaomi Redmi Note 8 Pro",
  "prise": "6 500",
  "url": "xiaomi-redmi-note-8-pro",
  "category": "smartphones",
  "characteristics": [
    ["Кamera", "24 mpx"]
  ],
  "img": [
    "img/goodsImages/Xiaomi Redmi Note 8 Pro/1.jpg",
    "src2",
    "src3",
    "src4",
    "src5",
    "src6"
  ],
  "colors": ["red", "blue", "gray"],
  "id": 2
},

{
  "name": "Xiaomi Redmi Note 8 Pro",
  "prise": "6 500",
  "url": "xiaomi-redmi-note-8-pro",
  "category": "smartphones",
  "characteristics": [
    ["Кamera", "24 mpx"]
  ],
  "img": [
    "img/goodsImages/Xiaomi Redmi Note 8 Pro/1.jpg",
    "src2",
    "src3",
    "src4",
    "src5",
    "src6"
  ],
  "colors": ["red", "blue", "gray"],
  "id": 3
},

{
  "name": "Xiaomi Redmi Note 8 Pro",
  "prise": "6 500",

```

```
"url": "xiaomi-redmi-note-8-pro",
"category": "smartphones",
"characteristics": [
  ["Камера", "24 mpx"]
],
"img": [
  "img/goodsImages/Xiaomi Redmi Note 8 Pro/1.jpg",
  "src2",
  "src3",
  "src4",
  "src5",
  "src6"
],
"colors": ["red", "blue", "gray"],
"id": 4
},

{
  "name": "Xiaomi Redmi Note 8 Pro",
  "prise": "6 500",
  "url": "xiaomi-redmi-note-8-pro",
  "category": "smartphones",
  "characteristics": [
    ["Камера", "24 mpx"]
  ],
  "img": [
    "img/goodsImages/Xiaomi Redmi Note 8 Pro/1.jpg",
    "src2",
    "src3",
    "src4",
    "src5",
    "src6"
  ],
  "colors": ["red", "blue", "gray"],
  "id": 5
},

{
  "name": "Xiaomi Redmi Note 8 Pro",
  "prise": "6 500",
  "url": "xiaomi-redmi-note-8-pro",
  "category": "smartphones",
  "characteristics": [
    ["Камера", "24 mpx"]
  ],
```

```

"img": [
  "img/goodsImages/Xiaomi Redmi Note 8 Pro/1.jpg",
  "src2",
  "src3",
  "src4",
  "src5",
  "src6"
],
"colors": ["red", "blue", "gray"],
"id": 6
},
{
  "name": "Xiaomi Redmi Note 8 Pro",
  "prise": "6 500",
  "url": "xiaomi-redmi-note-8-pro",
  "category": "smartphones",
  "characteristics": [
    ["Камера", "24 мрх"]
  ],
  "img": [
    "img/goodsImages/Xiaomi Redmi Note 8 Pro/1.jpg",
    "src2",
    "src3",
    "src4",
    "src5",
    "src6"
  ],
  "colors": ["red", "blue", "gray"],
  "id": 7
}
]

```

Замокаємо рисунки для заощадження часу та створимо декілька сот однакових товарів. Вийшло 40 тис. стрічок.

3.2 Опис роботи Web-сайту

1. Для того, щоб реально функціонував Інтернет-магазин, необхідно звернутись до служб, які надають відповідні послуги в Інтернет, де можна запустити власний Інтернет-магазин, скачати програму Open Server. А можна зробити локальний сервер на базі OpenCart.

2. Для встановлення програми необхідно:

- a. Скопіювати папку shop.local в каталог програми
- b. C:\OpenServer\domains ;
- c. Запустити файл openserver.exe на виконання;

- d. Завантажити браузер та в URL-адресі ввести www.xiaomishop;
- e. По закінченню перегляду Інтернет-магазину запустити на виконання вимкнення локального сервера.

3.3 Принцип роботи ІКСОЗТАПС

При завантаженні Web-сайту інтернет-магазину «TimDovg» перед користувачем відкривається головна сторінка.

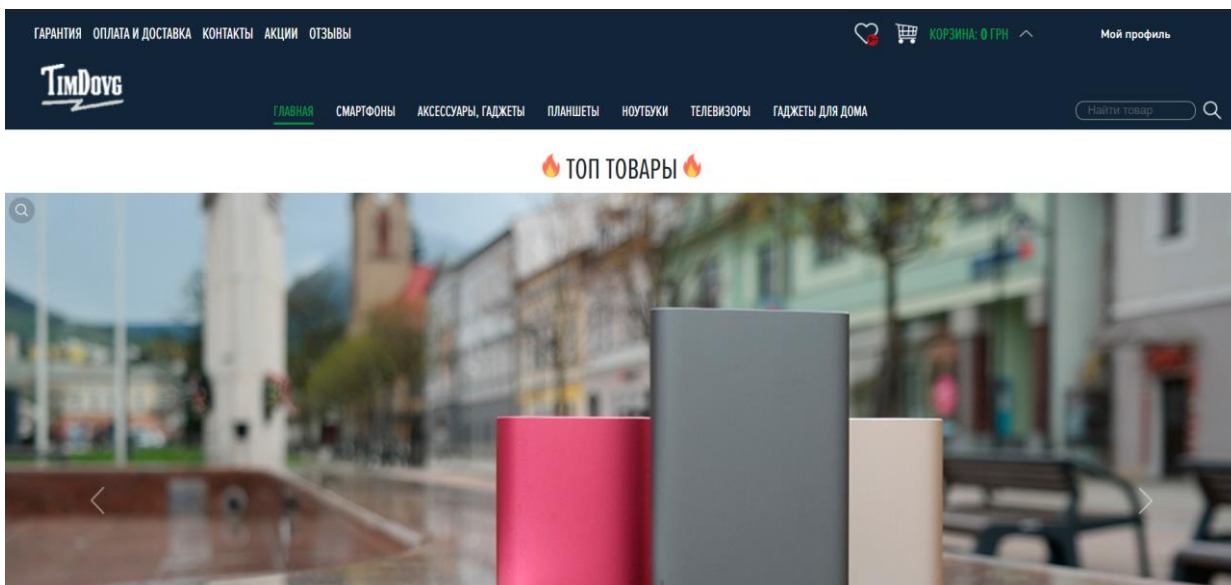


Рисунок 3.3.1 – Головна сторінка сайту

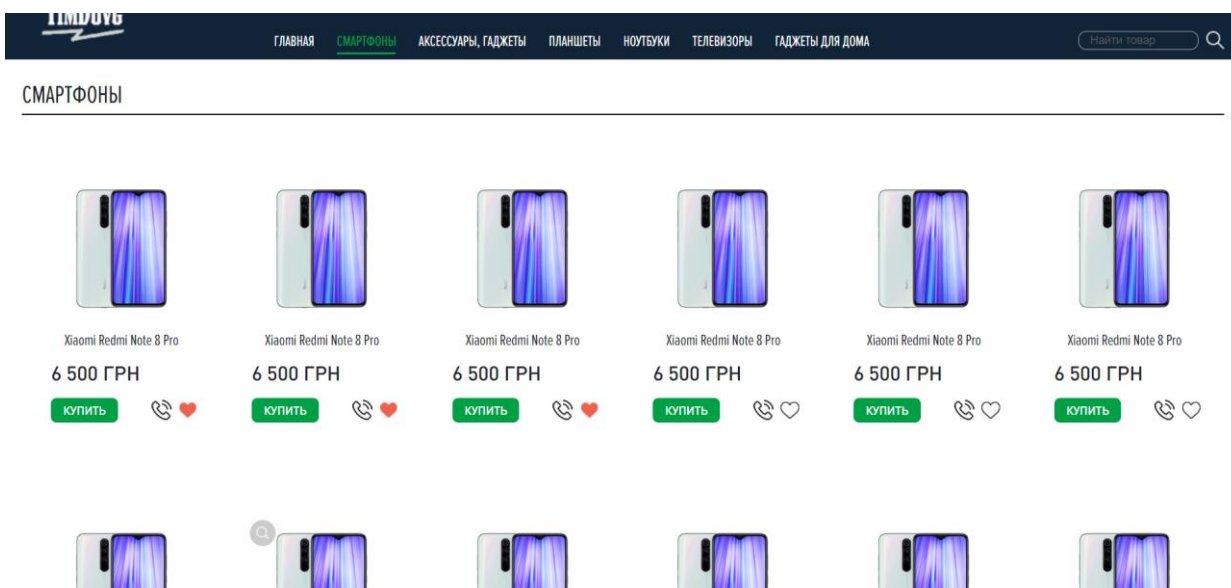


Рисунок 3.3.2 – Фільтрація товарів по категорії «Смартфони»

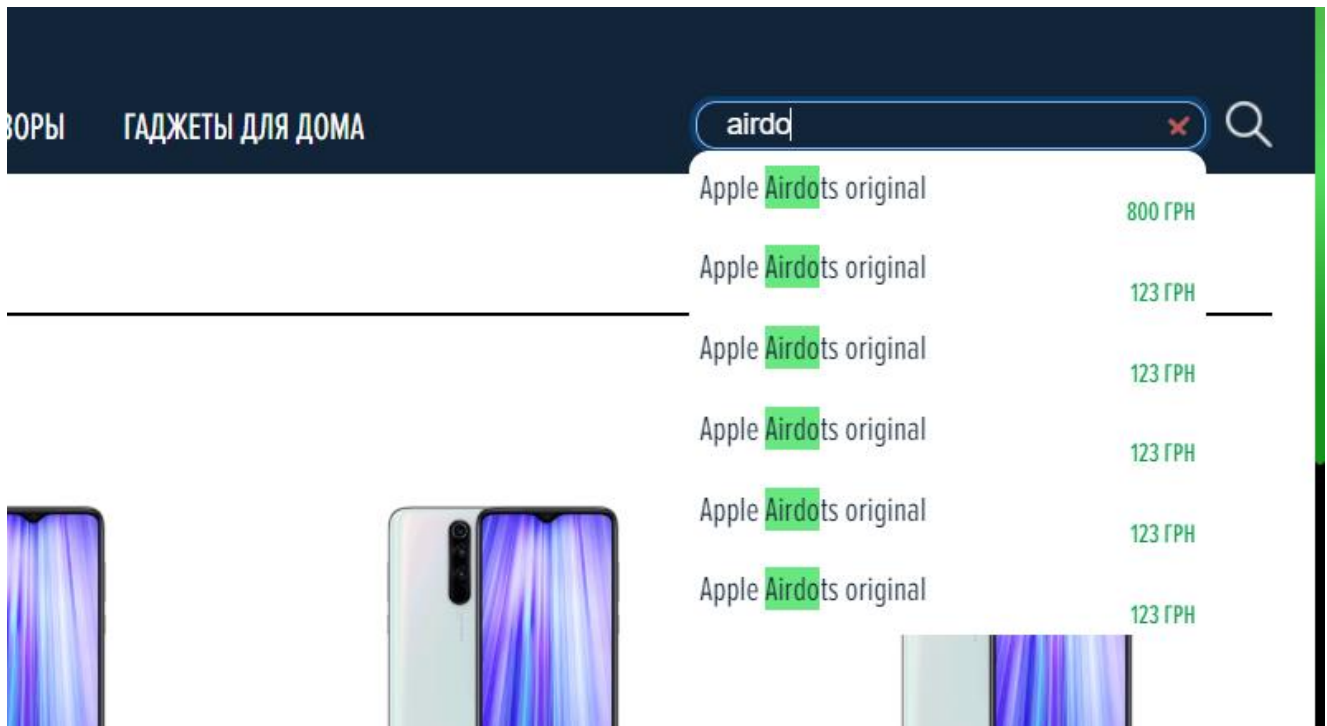


Рисунок 3.3.3 – AJAX запит, пошук і відображення даних згідно введених даних користувача

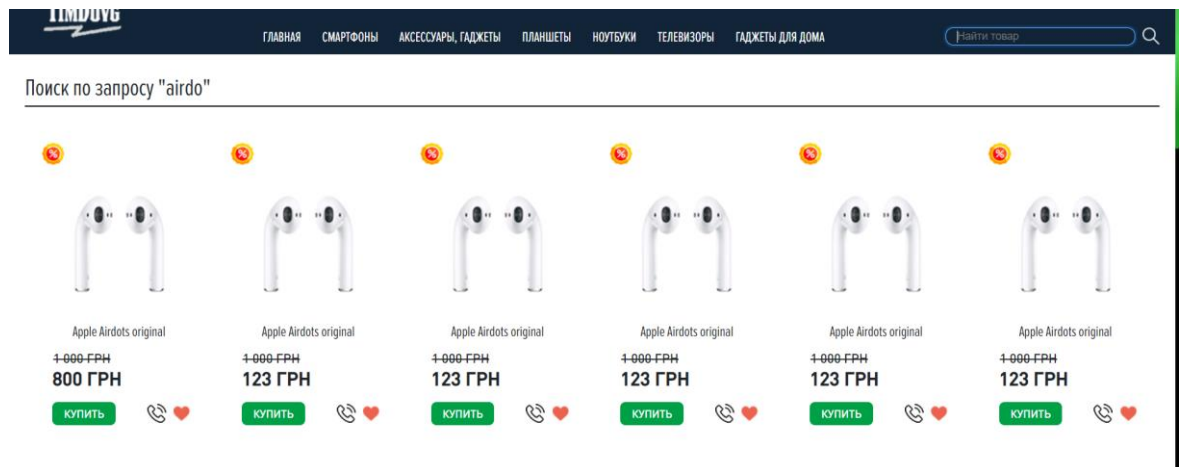


Рисунок 3.3.4 – Відображення сторінки пошуку згідно запиту користувача



Рисунок 3.3.5 – Відображення сторінки відгуків

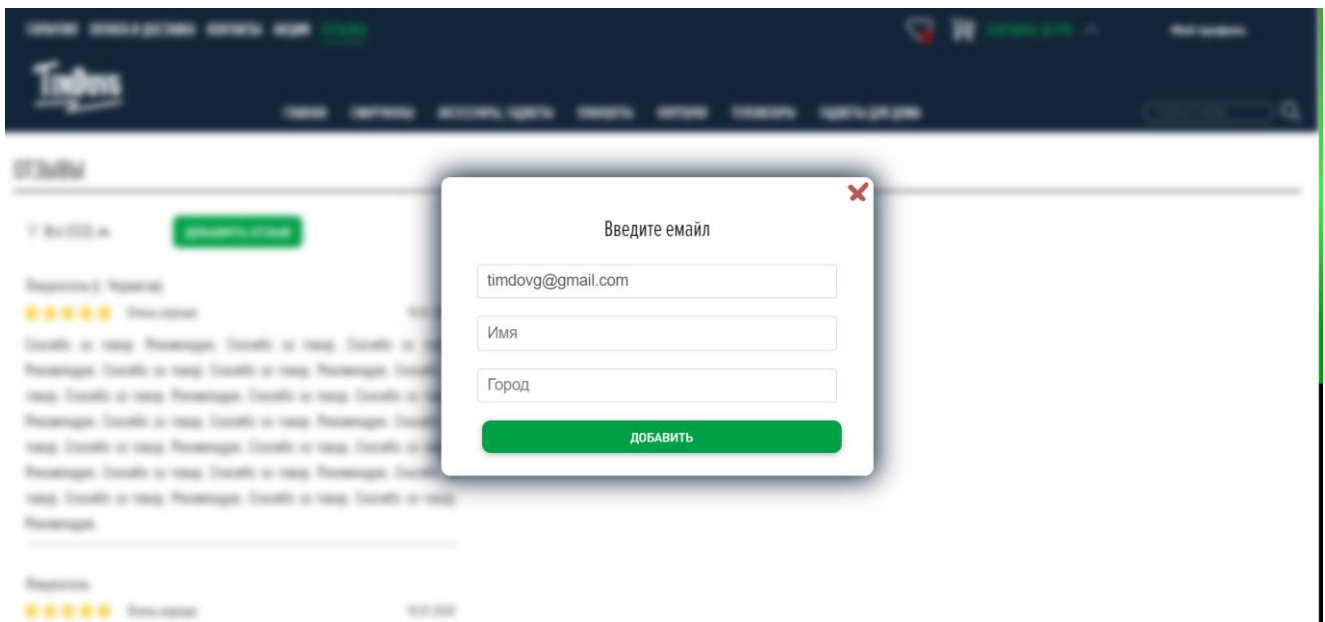


Рисунок 3.3.6 – Вікно для написання відгуку

ГАРАНТИЯ ОПЛАТА И ДОСТАВКА КОНТАКТЫ АКЦИИ ОТЗЫВЫ

КОРЗИНА: 0 ГРН
Мой профиль

TimDovg

[ГЛАВНАЯ](#) [СМАРТФОНЫ](#) [АКСЕССУАРЫ, ГАДЖЕТЫ](#) [ПЛАНШЕТЫ](#) [НОУТБУКИ](#) [ТЕЛЕВИЗОРЫ](#) [ГАДЖЕТЫ ДЛЯ ДОМА](#)

Найти товар

APPLE AIRDOTS ORIGINAL

Товар в наличии на складе

~~1 000 ГРН~~

800 ГРН

[КУПИТЬ](#)

Обратный звонок

Добавить в избранное

ХАРАКТЕРИТИКИ

Процессор	Apple W1
Режим разговора	До 2 ч
Интерфейсы	Bluetooth
Вес, г	4 (каждый)
Материал	Пластик
Цвет	Белый
Другие функции	Встроенный в наушник микрофон, регулировка громкости, контроль звонков
Совместимость	Apple
Комплектация	Наушники, чехол с зарядным устройством, кабель Lightning to USB

НАШИ КОНТАКТЫ

+380-67-910-95-46
+380-73-123-13-10
timdovg@gmail.com

ИНФОРМАЦИЯ

ГАРАНТИЯ
ОПЛАТА И ДОСТАВКА
АКЦИИ
ОТЗЫВЫ

ПОДПИСАТЬСЯ НА НОВОСТИ ОБ АКЦИЯХ

ТЕЛЕФОН/EMAIL [ПОДПИСАТЬСЯ](#)

Рисунок 3.3.7 – Сторінка товару

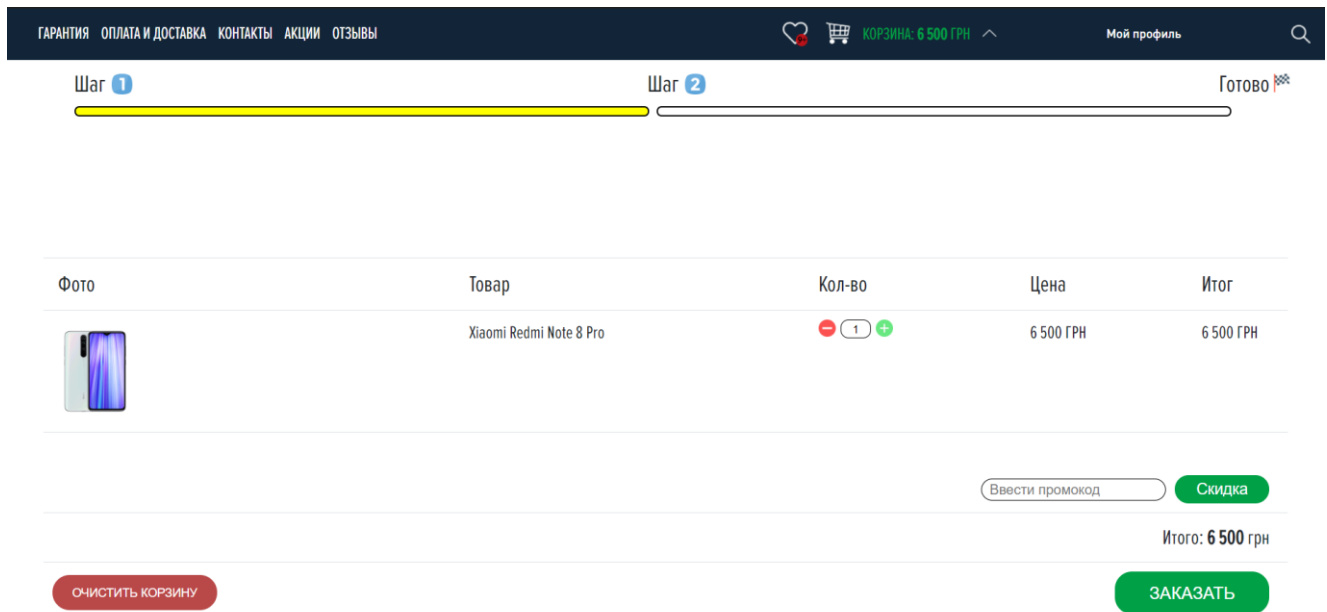


Рисунок 3.3.8 – Сторінка корзина

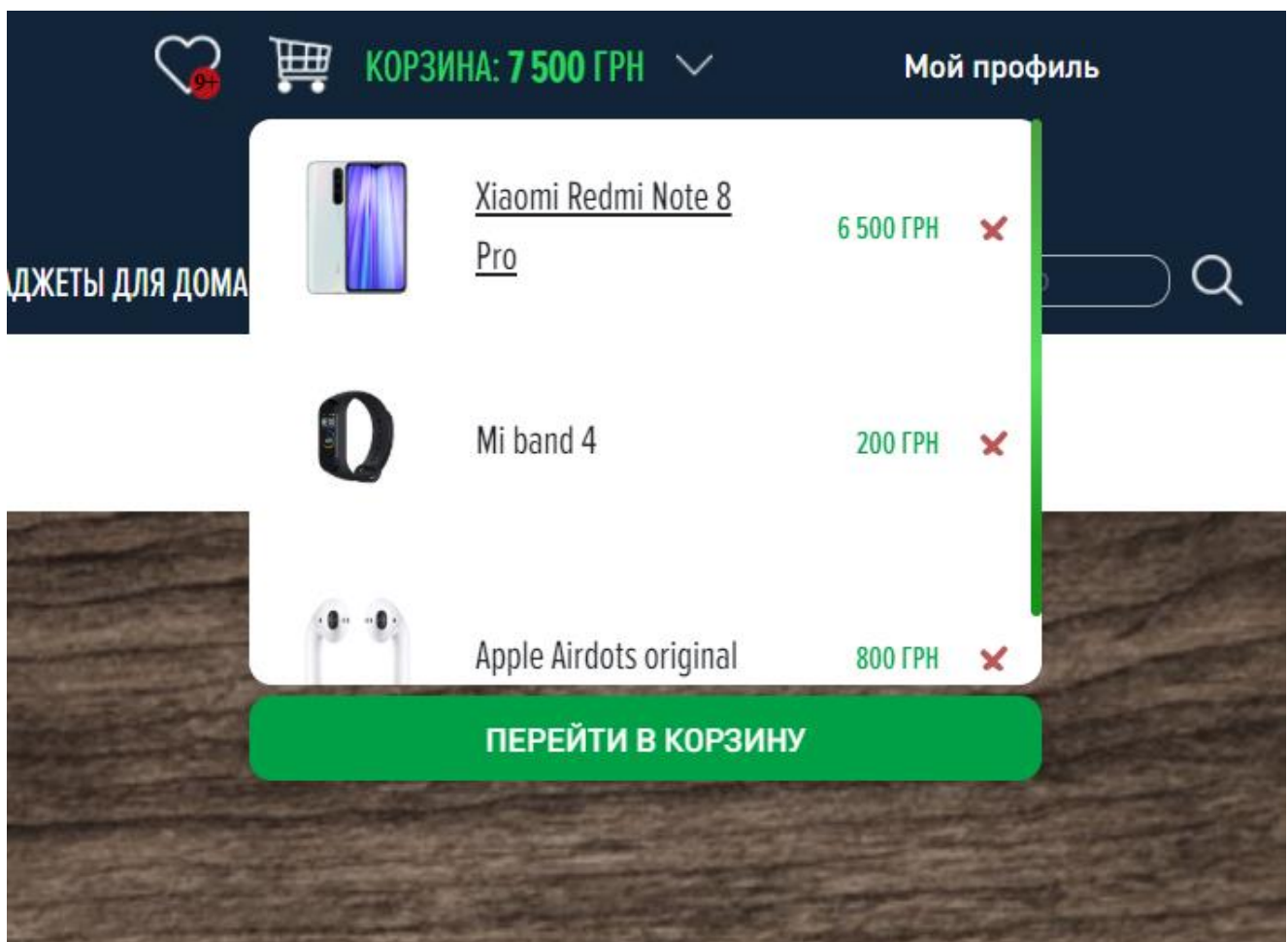


Рисунок 3.3.9 – Вікно корзины

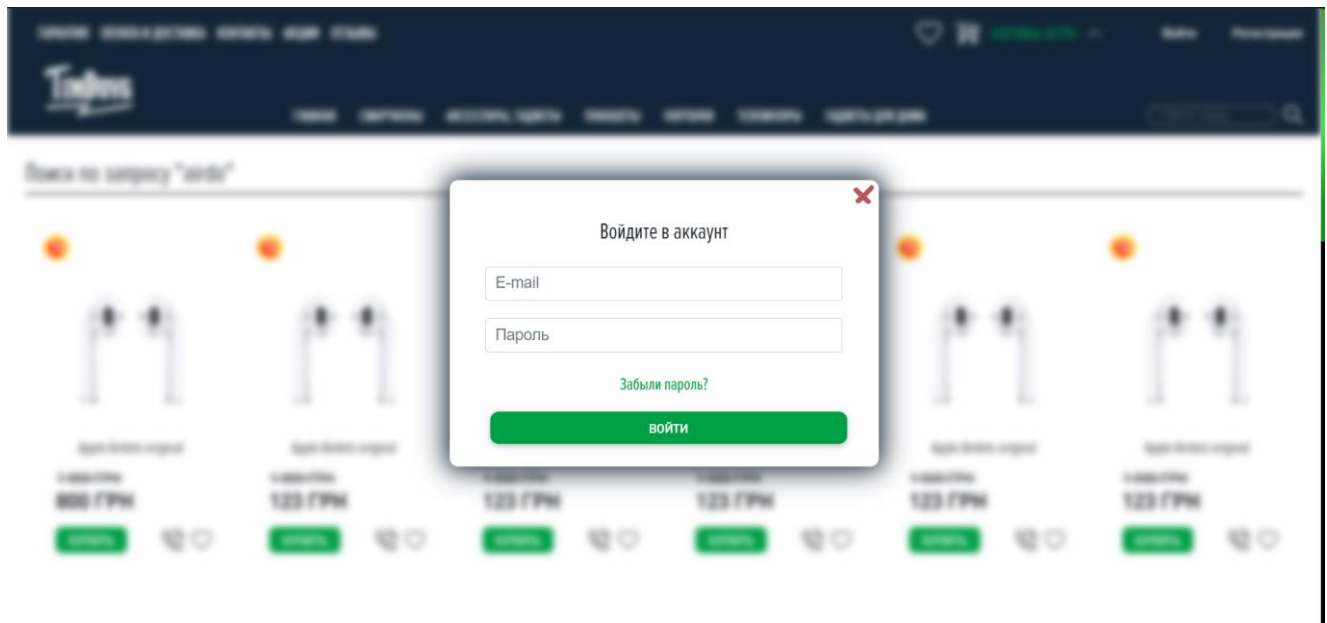


Рисунок 3.3.10 – Вікно входу в акаунт

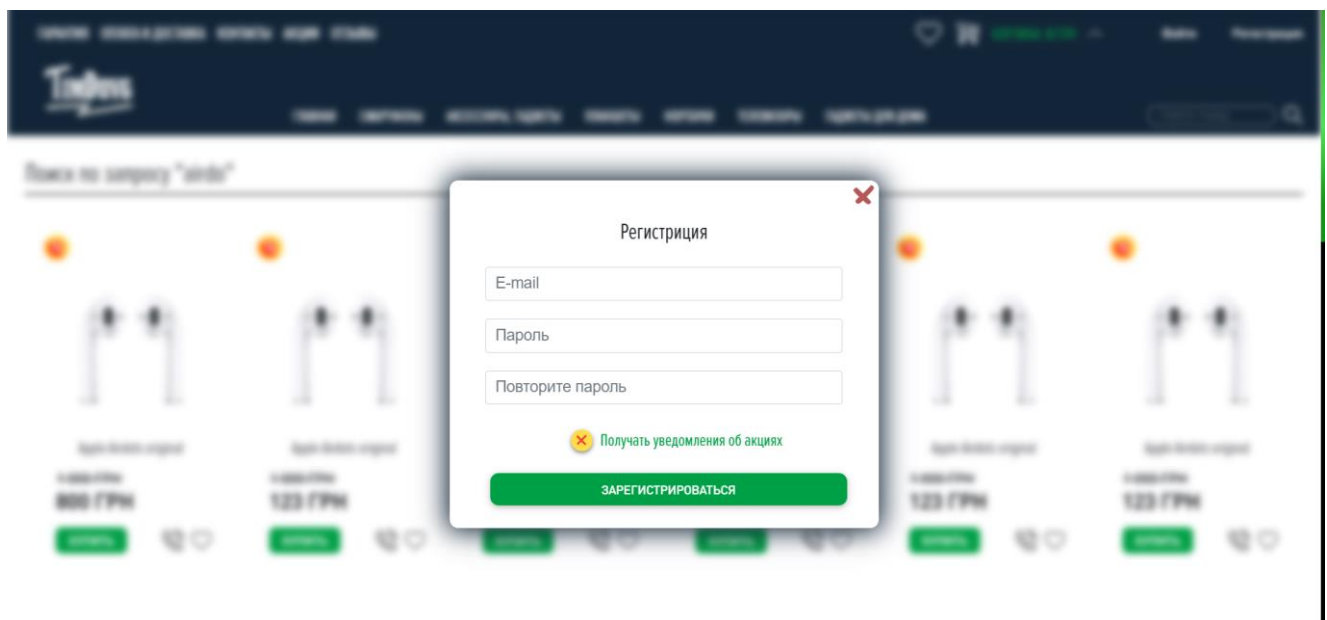


Рисунок 3.3.11 – Вікно реєстрації акаунту

3.4 Реалізація роутеру

Для того, щоб відбувалося пересування по «сторінкам» напишемо функцію-роутер:

```
import cleanBubbleWindow from "./cleanBubbleWindow.js";
import store from "./store.js";
```

```

import renderHeaderForGoods from "./headerForGoods.js";
import renderPartOfGoodsByPage from "./renderPartOfGoodsByPage.js";
import renderPageOfGood from "./renderPageOfGood.js";

export default function routerFunction() {
  if (location.hash === "") {
    cleanBubbleWindow();
    document.querySelector('.filter-main').dispatchEvent(new Event('click'));
  } else if (location.hash === '#sign-up') {
    cleanBubbleWindow();
    document.querySelector('#sign-up-button').dispatchEvent(new
Event('click'));
  } else if (location.hash === '#sign-in') {
    cleanBubbleWindow();
    document.querySelector('#sign-in-button').dispatchEvent(new
Event('click'));
  } else if (location.hash === '#wishes') {
    cleanBubbleWindow();
    document.querySelector('.wishes').dispatchEvent(new Event('click'));
  } else if (location.hash === '#accessories-and-gadgets') {
    cleanBubbleWindow();
    document.querySelector('.filter-accessories-and-
gadgets').dispatchEvent(new Event('click'));
  } else if (location.hash === '#smartphones') {
    cleanBubbleWindow();
    document.querySelector('.filter-smartphones').dispatchEvent(new
Event('click'));
  } else if (location.hash === '#tablets') {
    cleanBubbleWindow();
    document.querySelector('.filter-tablets').dispatchEvent(new Event('click'));
  }
}

```

```

} else if (location.hash === '#laptops') {
  cleanBubbleWindow();
  document.querySelector('.filter-laptops').dispatchEvent(new Event('click'));
} else if (location.hash === '#television-set') {
  cleanBubbleWindow();
  document.querySelector('.filter-television-set').dispatchEvent(new
Event('click'));
} else if (location.hash === '#home-gadgets') {
  cleanBubbleWindow();
  document.querySelector('.filter-home-gadgets').dispatchEvent(new
Event('click'));
} else if (location.hash === '#discounts') {
  cleanBubbleWindow();
  document.querySelector('.filter-discount').dispatchEvent(new
Event('click'));
} else if (/^#search/.test(location.hash)) {
  cleanBubbleWindow();
  let main = document.querySelector('.main');
  let value = location.hash.split('&')[1];
  let header = value.replace('-', ' ');
  let valueArr = value.split('-');
  let answer = store.goods.filter(good => {
    for (let value of valueArr) {
      let re = new RegExp(value, 'i');

      if (!re.test(good.name)) return false;
    }

    return true;
  });
}

```



```

    main.innerHTML = renderHeaderForGoods( Поиск по запросу
    "${header}" , answer);
    renderPartOfGoodsByPage(0,18,1, answer);
    document.querySelectorAll('.ProximaNovaExCn-Regular .nav .nav-
link').forEach(item =>
        item.classList.remove('active'));
    } else if (location.hash === '#feedback') {
        cleanBubbleWindow();
        document.querySelector('.feedback-button').dispatchEvent(new
Event('click'));
    } else if (location.hash === '#cart') {
        cleanBubbleWindow();
        document.querySelector('.cart-area').dispatchEvent(new Event('click'));
    } else {
        cleanBubbleWindow();
        // вырисовка товара по ID
        const goodId = location.hash.split('?')[1];

        renderPageOfGood(null, goodId);
    }
}

```

3.5 Реалізація компонентів

Вікно нотифікації:

```

export default function alertWindow(message, color = `#B55858`, time =
3000) {
    if (document.querySelector('.alert-window')) {
        document.querySelector('.alert-window').remove();
    }
}

```

```

    }

    const alertWindow = document.createElement('div');

    alertWindow.classList.add('alert-window');
    alertWindow.style.color = color;
    alertWindow.innerHTML = `
        <div class="message">${ message}</div>`;
    document.body.append(alertWindow);

    const timer = setTimeout(() => {
        alertWindow.style.opacity = '0.9';
        window.clearTimeout(timer);
    }, 0);

    const timer2 = setTimeout(() => {
        alertWindow.remove();
        window.clearTimeout(timer2);
    }, time);

    document.querySelector('.alert-window >
img[src="img/cancel.svg"]').addEventListener('click', () => {
        alertWindow.remove();
    });
}

```

Сторінки:

```

export default function renderPages(page, numberOfGoods) {
    [].slice.call(document.querySelectorAll('li.page-item'), 3, -
2).forEach(item => item.remove());
}

```

```

let pagination = document.querySelector('.pagination');
let numberOfAllPages = Math.ceil(numberOfGoods / 18);
let newPages = `- 

```

```

    </li>
    <li class="forward-to-finish page-item"><a class="page-
link">&gt;&gt;</a></li>`;
    pagination.innerHTML = newPages;
}

```

3.6 Реалізація класу Store

У скриптах синхронно завантажуються чотири скрипти:

```

<script defer src="js/jquery.min.js"></script>
<script defer src="js/zoomsl-3.0.min.js"></script>
<script defer src="js/materialize.min.js"></script>
<script defer type="module" src="js/store.js"></script>

```

Тобто, три – це бібліотеки, один – написаних власноруч.

У класі Store робиться асинхронний запит до описання товарів та їхній запис на сторону клієнта:

```

import Backend from "./backend.js";
import index from "./index.js";
class Store {
  constructor() {
  }
}
let store = new Store();
Backend.getGoods()
  .then(goods => {
    store.goods = goods;
  })
  .then(() => {
    index();
  })

```

```
;
export default store;
```

Далі іде запуск модуля `index.js`, який відобразить товари.

3.7 Реалізація класу `Backend`

Клас для роботи с бекендом:

```
export default class Backend {
  static signUp(userInfo) {
    return fetch('php/sign-up.php', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(userInfo),
    })
    .then(response => response.json())
    .then(response => {
      if (!response.error) {
        return response;
      }
      else {
        throw response.error;
      }
    })
  }

  static signIn(userInfo) {
```

```
return fetch('php/sign-in.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify(userInfo),
})
.then(response => response.json())
.then(response => {
  if (!response.error) {
    return response;
  }
  else {
    throw response.error;
  }
})
)
```

```
static addToCart(id, email) {
  return fetch('php/add-to-cart.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({id, email}),
  })
  .then(response => response.json())
  .then(response => {
    if (!response.error) {
```

```

        return response;
    }
    else {
        throw response.error;
    }
}
)
.catch(console.log)
;
}

static addToWish(id, email) {
    return fetch('php/add-to-wish.php', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({id, email}),
    })
    .then(response => response.json())
    .then(email => {
        if (email !== 'alreadyRegistered') {
            localStorage.setItem('unnamedUser', email);
        }
    })
    .catch(console.log)
    ;
}

static removeFormWish(id, email) {

```

```
return fetch('php/remove-from-wish.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({id, email}),
})
.catch(console.log)
;
}

static addStatisticBuying(id) {
  return fetch('php/add-statistic-buying.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({id}),
  })
  .catch(console.log)
  ;
}

static addStatisticClicking(id) {
  return fetch('php/add-statistic-clicking.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({id}),
  })
}
```



```
    })  
    .catch(console.log)  
    ;  
  }  
  
  static getWishes(email) {  
    return fetch('php/get-wishes.php', {  
      method: 'POST',  
      headers: {  
        'Content-Type': 'application/json',  
      },  
      body: JSON.stringify({email}),  
    })  
    .then(response => response.json())  
    .catch(console.log)  
    ;  
  }  
  
  static subscribeUser(email) {  
    return fetch('php/subscribe-user.php', {  
      method: 'POST',  
      headers: {  
        'Content-Type': 'application/json',  
      },  
      body: JSON.stringify({email}),  
    })  
    .then(response => response.json())  
    .catch(console.log)  
    ;  
  }  
}
```

```
static addCartPrise(info) {  
    return fetch('php/add-cart-prise.php', {  
        method: 'POST',  
        headers: {  
            'Content-Type': 'application/json',  
        },  
        body: JSON.stringify(info),  
    })  
    .catch(console.log)  
    ;  
}
```

```
static getCartPrise(info) {  
    return fetch('php/get-cart-prise.php', {  
        method: 'POST',  
        headers: {  
            'Content-Type': 'application/json',  
        },  
        body: JSON.stringify(info),  
    })  
    .then(response => response.json())  
    .catch(console.log)  
    ;  
}
```

```
static getGoods() {  
    return fetch('json/goods.json', {  
        method: 'GET'  
    })  
}
```

```
.then(response => response.json())
.catch(console.log)
;
}

static copyOldDataForNewUser(oldEmail, newEmail) {
  return fetch('php/copy-old-data-for-new-user.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({oldEmail, newEmail}),
  })
  .catch(console.log)
;
}

static toOrderCall(phone) {
  return fetch('php/to-order-call.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({phone}),
  })
  .catch(console.log)
;
}

static getGoodsIdFromCart(email) {
  return fetch('php/get-goods-id-from-cart.php', {
    method: 'POST',
```

```
headers: {
  'Content-Type': 'application/json',
},
body: JSON.stringify({email}),
})
.then(response => response.json())
.catch(console.log)
;
}
static removeFromCart(id, email, prise) {
  return fetch('php/remove-from-cart.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({id, email, prise}),
  })
  .catch(console.log)
}
static cleanUpCart(email) {
  return fetch('php/clean-up-cart.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({email}),
  })
  .catch(console.log)
;
}}
```

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дейт К. Введение в системы баз данных: Пер. с англ.-К.: М.: СПб:Изд. Дом "Вильямс", 1999. -848с., ил.
2. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж. Приемы объектноориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил.
3. <http://htmlbook.ru>
4. <https://hostingkartinok.com/process.php>
5. <http://postgresql.ru.net/manual/ddl-inherit.html>
6. <https://istarik.ru/blog/programmirovanie/27.html>
7. <https://uk.wikipedia.org/wiki/Интернет-магазин>
8. [https://uk.wikipedia.org/wiki/Java Database Connectivity](https://uk.wikipedia.org/wiki/Java_Database_Connectivity)
9. Стив Суэринг, Тим Конверс, Джойс Парк PHP и MySQL:Изд. «Диалектрика», 2010. -354с., ил.
10. <https://myrusakov.ru/html-osnovy.html>
11. Web Database Application with PHP and MySQL, 2nd Edition By David Lane, Hugh E. Williams. © O'Reilly, May 2004. ISBN: 0-596-00543-1.
12. Материал из Википедии - свободной энциклопедии о системах управления сайтом. <http://ru.wikipedia.org/wiki/CMS>

13. Каратыгин С., Тихонов А., Долголаптев В. Базы данных: простейшие средства обработки информации, электронные таблицы, системы управления базами данных: В 2 т. - М.: АБР, 1995.

14. <https://learn.javascript.ru/>

15. <https://git-scm.com/book/ru/v2>

16. <https://www.youtube.com/>

17. <https://htmlacademy.ru/>

18. <https://habr.com/>

19. <https://tproger.ru/>

ВИСНОВКИ

В ході виконання дипломної роботи було створено інтернет-магазин для продажу електронних товарів. Але движок сайту дуже зручний та може бути адаптований протягом короткого часу в інтернет-магазин майже будь-якої тематики. Такі переваги доступні через функціональну парадигму програмування та компонентний підхід.

Інтернет-магазини є необхідними в наш час, так як вони мають багато переваг як зі сторони компанії, так і зі сторони клієнту: економія часу підприємства на обробку заказів та консультацію, швидке обслуговування клієнтів, можливість охопити більший спектр клієнтів не зважаючи на їхню геолокацію. Також, є і переваги для клієнтів, які по факту є і перевагами для підприємства: швидкий доступ до товарів, швидка консультація і обширний опис товару, можливість швидко їх порівняти і зробити замовлення дуже швидко.

Але є і великий недолік інтернет-магазинів, це шахрайство. Тому багато часу було витрачено на інтерфейс користувача, щоб надати максимальну оцінку UI та UX. Це надає клієнту впевненість у тому, що компанія серйозна, та сайт не закриється вже після першого ордеру.

За швидкість роботи сайту відповідає гарно написаний роутер з лише однією сторінкою HTML. Багато асинхронних запитів до беку виконується під час рендеренгу сторінки, що теж заощаджує час користувача. Так як з самого старту клієнт одержує майже всю необхідну інформацію о товарах, він не гae час на зайві запити до беку; а самі дані о товарах представлені у виді JSON, тому майже не займають місця в ОП та завантажуються дуже швидко.

При створенні інтернет-магазину були використаний такий стек технологій: HTML5, CSS3, SCSS, Bootstrap4, JavaScript, jQuery, PHP, RedBeanPHP, JSON, MySQL, Webpack та сторонні плагіни. Під час розробки, збереження версій проекту буде надано SCV GIT та GitHub.

Враховуючи вище сказане, можна зробити висновок, що всі поставленні завдання в дипломній роботі виконані та одержано гарний результат.

Довідка про відсутність плагіату у випускній кваліфікаційній роботі

Міністерство освіти і науки України
Чернігівський національний технологічний університет
Кафедра інформаційних та комп'ютерних систем

ДОВІДКА
про відсутність плагіату у випускній кваліфікаційній роботі

За результатами перевірки випускної кваліфікаційної роботи здобувача вищої освіти

Довгопола Тимофія Вікторовича
(прізвище, ім'я, по батькові)

на тему:

«Інформаційно-комп'ютерна система обробки замовлень та аналізу попиту споживачів»

в роботі не виявлено порушень академічної доброчесності.

Здобувач вищої освіти гарантує, що написана випускна кваліфікаційна робота повністю оригінальна. Якщо здобувач вищої освіти використав роботу і/або слова інших авторів, це має бути відповідно процитоване або вказане посилання на оригінальну роботу.

Керівник ВКР

(підпис)

(ПІБ)