

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Ігор ШЕЛЕХОВ  
(підпис)

\_\_\_\_\_ 202 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня бакалавр**

зі спеціальності 122 - Комп'ютерні науки,  
освітньо-наукової програми «Інформатика»  
на тему: «Інформаційна система управління продажами товарів “GoldStar”»  
здобувача групи ІН-06-2 – Маладики Давіда Олександровича

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело.

\_\_\_\_\_ Давід МАЛАДИКА  
(підпис)

Керівник, доцент,  
кандидат фізико-математичних наук,  
доцент

\_\_\_\_\_ Галина ОЛЕКСІЄНКО

(підпис)

**Суми – 2024**

**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

## ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

### на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»  
здобувача групи ІН-06-2 Маладики Давіда Олександровича

1. Тема роботи: «Інформаційна система управління продажами товарів «GoldStar»  
затверджую наказом по СумДУ від 22 квітня 2024 року № 0414-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 29 травня 2024 року

3. Вхідні дані до кваліфікаційної роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
1) Аналітичний огляд предметної області, аналіз аналогів та постановка задачі. 2) Огляд технологій для розробки інформаційної системи продажу товарів. 3) Розробка інформаційної системи управління продажами товарів «GoldStar». 4) Аналіз результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «  »    202   р.

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналітичний огляд предметної області, аналіз аналогів та постановка задачі</i>	06.05.24- 12.05.24	
2	<i>Аналіз технологій для створення інформаційних систем</i>	13.05.24- 15.05.24	
3	<i>Розробка інформаційної системи продажу товарів</i>	16.05.24-	

		22.05.24	
4	<i>Тестування інформаційної системи продажу товарів</i>	23.05.24- 27.05.24	
5	<i>Аналіз результатів, оформлення пояснювальної записки</i>	28.05.25- 31.05.24	

Здобувач вищої освіти

\_\_\_\_\_

(підпис)

Керівник

\_\_\_\_\_

(підпис)

## АНОТАЦІЯ

**Записка:** 61 сторінко, 11 рисунків, 3 додатки, 12 використаних джерел.

**Обґрунтування актуальності теми роботи** — тема кваліфікаційної роботи є важливою та актуальною, оскільки спрямована на вирішення практичної проблеми продажу товарів «GoldStar».

**Об'єкт дослідження** — Інформаційна система продажу товарів.

**Предмет дослідження** — Процеси керування продажами товарів магазину.

**Мета роботи** — розробка інформаційної системи продажу товарів, яка дозволяє переглядати товар, його характеристики. Функціонал з реєстрації\авторизації користувача, додавання товару в кошик товарів.

**Методи дослідження** — аналіз існуючих інформаційних систем, моделювання системи, оцінку інструментів розробки, проектування бази даних і розробку програмної частини.

**Результати** — досліджена актуальність проєкту, проведено аналіз інформаційних систем з продажу товарів та аналогів. Сформовано вимоги до проєкту. Визначено задачі до роботи. Обрано інструменти та технології для розробки ІС. Розроблено базу даних, клієнтську та серверну частину. Проведено тестування програмного продукту.

ІНФОРМАЦІЙНА СИСТЕМА, ПРОДАЖ ТОВАРІВ, HTML, CSS,  
JAVASCRIPT, MYSQL, PHP

## ЗМІСТ

ВСТУП.....	6
1. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
1.1 Актуальність ІС продажу товарів	
1.2 Принципи розробки інформаційної системи	
1.3 Аналіз аналогів	
1.4 Вимоги до проєкту	
1.5 Постановка задачі	
2. ВИБІР МЕТОДІВ РОЗВ’ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	13
2.1 Інструменти для розробки ІС	
2.1.1 Інтегровані середовища розробки (IDE).....	13
2.1.2 Технології для веброзробки клієнтської частини.....	14
2.1.3 Технології для веброзробки серверної частини.....	17
3. ПРОГРАМНА РЕАЛІЗАЦІЯ.....	20
3.1 Проектування БД	
3.2 Розробка клієнтської частини	
3.3 Розробка серверної частини	
3.4 Тестування	
ВИСНОВКИ.....	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	31
ДОДАТОК А.....	32
ДОДАТОК Б.....	42
ДОДАТОК В.....	47

## ВСТУП

**Актуальність.** Завдяки ІС з продажу товарів, компанії можуть краще розуміти потреби та поведінку своїх клієнтів. Системи оперують інформацією про покупки, вподобання, що дозволяє персоналізувати пропозиції та підвищити рівень задоволення клієнтів. Також надають матеріали для збору та аналізу даних про продажі. Це дозволяє бізнесам прогнозувати майбутні тенденції, ідентифікувати можливості для зростання та швидко реагувати на зміни на ринку. Аналітичні можливості таких систем є ключовим елементом для прийняття обґрунтованих управлінських рішень.

**Об'єкт дослідження.** Інформаційна система продажу товарів.

**Предмет дослідження.** Процеси керування продажами товарів магазину.

**Гіпотеза.** Розробка інформаційної системи продажу товарів «GoldStar» покращити впізнаваність бренду та збільшить обсяг аудиторії.

**Новизна.** Система представляє собою суміш сучасного дизайну, зручного інтерфейсу та надійного функціоналу.

**Структура.** Дана робота складається зі вступу, аналітичного огляду, постановки задачі, вибір методу розв'язання, опису ПЗ ІС, висновків, списку використаних джерел та додатків.

# 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Актуальність ІС продажу товарів

Актуальність можна поділити на декілька пунктів:

– Підвищення конкуренції на ринку: в сучасних умовах високої конкуренції на ринку роздрібної торгівлі підприємства шукають ефективні способи для оптимізації своїх бізнес-процесів і підвищення конкурентоспроможності. Інформаційні системи управління продажами стають необхідним інструментом для досягнення цих цілей.

– Автоматизація бізнес-процесів: автоматизація процесів продажу дозволяє значно скоротити час на обробку замовлень, управління запасами та аналітику ринку. Це забезпечує більшу точність і швидкість виконання операцій, що важливо для задоволення потреб клієнтів та підтримання високого рівня сервісу.

– Аналітика та прогнозування: сучасні інформаційні системи забезпечують доступ до потужних аналітичних інструментів, які допомагають прогнозувати попит, аналізувати продажі і визначати тренди. Це дає можливість приймати обґрунтовані рішення на основі реальних даних, що покращує управління запасами і знижує витрати.

– Інтеграція з іншими системами: інформаційні системи управління продажами можуть бути інтегровані з іншими корпоративними системами, такими як CRM (управління взаємодією з клієнтами) та ERP (планування ресурсів підприємства). Це забезпечує комплексний підхід до управління бізнесом і сприяє підвищенню ефективності всіх бізнес-процесів.

– Підвищення якості обслуговування клієнтів: завдяки автоматизації і зручним інструментам для управління продажами, такі системи допомагають покращити якість обслуговування клієнтів, забезпечуючи швидке і точне виконання замовлень, персоналізовані пропозиції і оперативну підтримку.

– Глобальні тренди цифрової трансформації: в епоху цифрової трансформації використання сучасних інформаційних систем стає необхідністю для збереження конкурентних переваг. Компанії, що впроваджують такі системи, можуть швидше адаптуватися до змін на ринку і пропонувати інноваційні продукти та послуги.

– Ефективне управління запасами та логістикою: Інформаційні системи управління продажами дозволяють ефективніше управляти запасами і логістичними процесами, зменшуючи витрати на зберігання та транспорт, а також мінімізуючи ризики дефіциту або надлишку товарів.

Таким чином, актуальність обумовлена необхідністю підвищення ефективності бізнес-процесів, забезпечення високого рівня обслуговування клієнтів, адаптації до ринкових змін та підтримки конкурентоспроможності підприємств в умовах цифрової економіки.

## **1.2 Принципи розробки інформаційної системи**

Інформаційна система (ІС) - це комплекс взаємопов'язаних компонентів, які призначені для збирання, зберігання, оброблення, передачі та використання інформації з метою досягнення певних цілей в організації або компанії.

Інформаційна система продажу товарів (ІСПТ) - це комплекс програмних та апаратних засобів, які призначені для автоматизації процесів управління продажами товарів або послуг. Основна мета ІСПТ - оптимізувати роботу торговельного підприємства, забезпечуючи швидке та ефективне оброблення замовлень, контроль за складом товарів, ведення обліку, аналіз та прогнозування продажів, взаємодію з клієнтами тощо.[1]

Зазвичай, інформаційні системи продажу товарів включають в себе такі компоненти:

– **ПО для точок продажу (POS-системи):** Це програмне забезпечення, що використовується для реєстрації та оплати товарів або послуг на касі. POS-системи можуть включати функції збору даних, оброблення платежів, видачі чеків ,тощо.



– **Модулі управління запасами:** Ці модулі дозволяють вести облік товарів на складі, контролювати їх рух, визначати потреби в поповненні запасів та оптимізувати їх розподіл.

– **Клієнтські бази даних:** Інформаційна система може містити базу даних про клієнтів, що дозволяє збирати та аналізувати інформацію про покупців, їхній історії покупок, уподобання та контактні дані.

– **Аналітичні засоби:** ІСПТ можуть мати вбудовані аналітичні засоби, які дозволяють аналізувати дані про продажі, робити прогнози, визначати ефективність рекламних кампаній та стратегій продажів.

– **Інтеграція з іншими системами:** ІСПТ можуть бути інтегровані з іншими інформаційними системами підприємства, такими як бухгалтерська система, система управління відносинами з клієнтами (CRM), система управління виробництвом тощо.

Саме час перейти до відповіді на головне питання, а саме головні принципи у розробці ІС для управління продажами товарів, з точки зору веб-розробника:

– **Відповідність потребам користувачів:** Перш за все, важливо зрозуміти потреби та очікування користувачів системи управління продажами. Веб-розробник повинен вивчити особливості бізнесу клієнта, його процеси продажів та внутрішні потреби для ефективного управління.

– **Зручний інтерфейс користувача:** Важливо створити інтуїтивно зрозумілий та зручний інтерфейс, щоб користувачі могли легко взаємодіяти з системою. Це може включати зрозумілі меню, інтуїтивні елементи керування та швидкий доступ до важливої інформації.

– **Надійність та безпека:** Веб-розробник повинен забезпечити надійність та безпеку інформаційної системи. Це означає використання захисту даних, шифрування та інших заходів безпеки для запобігання несанкціонованому доступу та втраті даних.

– **Інтеграція з іншими системами:** Інформаційна система управління продажами може потребувати інтеграції з іншими системами, такими як система бухгалтерського обліку, складська система або CRM-система. Веб-розробник повинен забезпечити зручну інтеграцію та обмін даними між цими системами.

– **Масштабованість та гнучкість:** Інформаційна система повинна бути масштабованою та гнучкою, щоб забезпечити зміни в потребах бізнесу та можливість розширення функціональності в майбутньому.

– **Підтримка та супровід:** Після розгортання системи важливо забезпечити її підтримку та супровід. Веб-розробник повинен бути готовий вирішувати будь-які проблеми та надавати технічну підтримку користувачам системи.

### 1.3 Аналіз аналогів

Вебсайт «Puma»[2] - це про стиль та якість. Німецький бренд завоював репутацію надійного і досвідченого виробника завдяки постійному вдосконаленню технологій. Компанія була створена ще в 1948 році, а сьогодні має магазини в десятках країн.

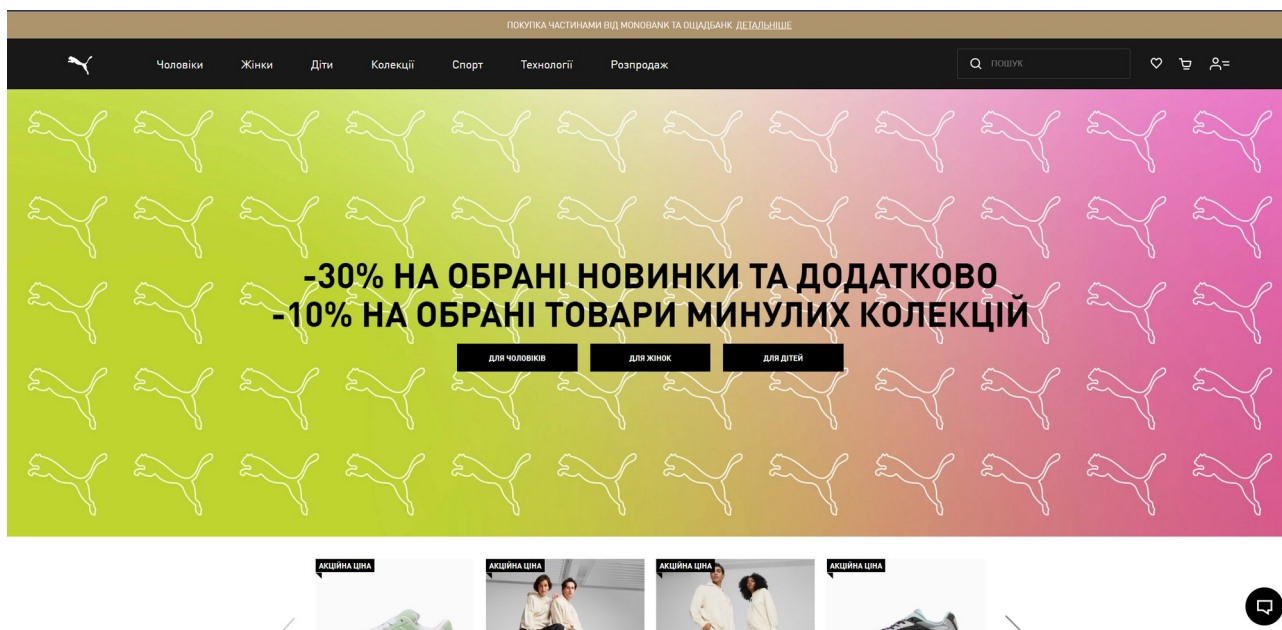


Рисунок 1.1 – Головна сторінка «Puma»[2]

## Переваги:

- Великий вибір товару.
- Красивий та сучасний дизайн.
- Зручний фільтр пошуку та сортування товару.
- Особистий кабінет, можливість додати товар у обране та купити товар.

## Недоліки:

- Довге завантаження вебсайту.
- Складна навігація по сторінкам.

Вебсайт «Intertop»[3] - це відома мережа магазинів взуття та аксесуарів. Вона пропонує широкий асортимент взуття для різних цільових аудиторій, включаючи спортивне взуття, повсякденне взуття, туфлі, сандалі та багато іншого. Інтертоп має свої магазини в різних країнах і часто відомий своїми акціями та знижками.

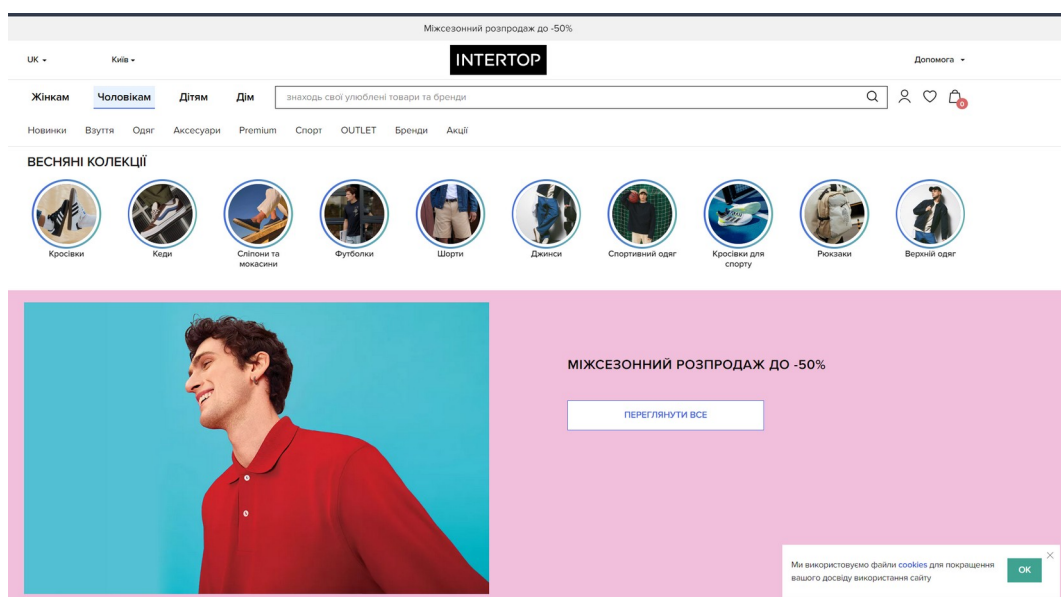


Рисунок 1.2 – Головна сторінка «Intertop»[3]

## Переваги:

- Великий вибір товару.
- Красивий та сучасний дизайн.

- Особистий кабінет, можливість додати товар у обране та купити товар.

- Зручний та сучасний дизайн.

Недоліки:

- Складна навігація по сторінкам.

У підсумку, можна констатувати наступне, для великої кількості товару дуже добре підходять дуже великі вебсайти, з окремими сторінками для кожної категорії товару, розподіл товару по його характеристикам, наприклад: за розміром, кольором, призначенням, тощо. Такі ІС мають свої недоліки, при чому вони спільні у всіх випадках – це час завантаження та складна система навігації по вебсайту через їх розміри.

#### **1.4 Вимоги до проєкту**

Враховуючи результати аналізу конкурентів, та того, що на вебсайті буде представлено лінійку однієї марки товару (до 10 одиниць), наведене ТЗ для теми: «Інформаційна система управління продажами товарів «GoldStar». Вебсайт має бути лендінгом з однієї сторінки, яка адаптується під розмір пристрою. Товари розміщені на слайдері по центру вебсторінки, перегляд здійснюється через перемикання/скрол.

Має бути особистий кабінет для реєстрації/входу користувачів, які зможуть додавати товари в кошик після авторизації. Кошик та можливість додати товар доступні лише авторизованим користувачам.

Вебсайт має включати прелоадер і анімаційні сценарії для блоків.

Дизайн простий, з фоном, що змінюється відносно активного товару. Вся інформація зберігається в БД. Функціонал може бути розширений (наприклад, додання адмін-панелі чи можливості оплати).

#### **1.5 Постановка задачі**

В результаті роботи необхідно реалізувати наступні задачі:

- Вибір технологій для розробки клієнтської частини.

- Вибір технологій для розробки серверної технології.
- Вибір СКБД.
- Розробка концептуальної моделі та проектування БД.
- Написання інструкцій для взаємодії БД та інтерфейсу користувача.

## 2. ВИБІР МЕТОДІВ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

### 2.1 Інструменти для розробки ІС

Інструменти для розробки ІС можуть бути дуже різноманітними і залежать від конкретних потреб та вимог проекту. Важливо, щоб інструменти спирались на ТЗ майбутнього продукту. Нижче наведені загальні, базові, інструменти, які розбиті по категоріям.

#### 2.1.1 Інтегровані середовища розробки (IDE)

Visual Studio Code - це безкоштовний і відкритий текстовий редактор, розроблений і підтримуваний компанією Microsoft. Він став дуже популярним серед розробників програмного забезпечення завдяки своїй легкості використання, багатству функцій та розширюваності.[4]

Для розробки вебсайту найкраще підходить VS Code, через свої переваги:

- Кросплатформенність: Visual Studio Code доступний для операційних систем Windows, macOS і Linux, що дозволяє користувачам розробляти на будь-якій платформі.
- Легкий та швидкий: VS Code відзначається високою швидкістю завантаження та реакції. Він запускається швидко і споживає мало системних ресурсів.
- Розширюваність: Однією з основних переваг VS Code є можливість розширювати його за допомогою різноманітних розширень. Розширення дозволяють додавати нові функції, підтримку мов програмування, інструменти розробки та інше.
- Інтеграція з Git: VS Code має вбудовану підтримку Git, що полегшує роботу з версійним контролем. Користувачі можуть виконувати операції Git, такі як коміти, гілки, злиття тощо безпосередньо з редактора.
- Підтримка багатьох мов програмування: VS Code підтримує багато мов програмування і має розширення для підсвічування синтаксису, автодоповнення коду та інші функції для різних мов.

- Відладка: Редактор має вбудовану підтримку відладки для різних мов і платформ. Ви можете встановлювати точки зупинки, виконувати код по кроці і вивчати значення змінних.
- Термінал: VS Code має вбудований термінал, який дозволяє виконувати команди безпосередньо в редакторі, не виходячи з нього.
- Робочі простори: Ви можете створювати робочі простори, які містять набір проектів та налаштувань для зручного управління розробкою.
- Розробка веб-додатків: VS Code підтримує розробку веб-додатків і має розширення для роботи з HTML, CSS, JavaScript та іншими веб-технологіями.
- Спільнота і підтримка: Visual Studio Code має активну спільноту користувачів та багато документації. Користувачі можуть звертатися за допомогою на форумах та GitHub.

Visual Studio Code - це потужний та універсальний текстовий редактор, який використовується для розробки різних видів програмного забезпечення, від веб-додатків до серверних застосунків. Саме через ці переваги його було обрано для використання в даній роботі.

### **2.1.2 Технології для веброзробки клієнтської частини**

HTML/CSS/JavaScript: Основні мови для створення вебсторінок та взаємодії з користувачем. Розглянемо кожну технологію окремо, їх особливості та їх інтеграцію при роботі з обраним середовищем розробки.

1. HTML (Hypertext Markup Language) - це мова розмітки, що використовується для створення структури веб-сторінок. Вона визначає скелет документа і включає елементи, які визначають заголовки, параграфи, посилання, зображення, таблиці та інші елементи контенту.[5]

Основні концепції HTML включають в себе:

- Теги: HTML використовує теги для визначення різних елементів на сторінці. Теги зазвичай мають вигляд ``<назва_тегу>...</назва_тегу>``, де

`<назва\_тегу>` - це відкриваючий тег, а ``</назва_тегу>`` - закриваючий тег. Наприклад, ``<p>`` - тег для параграфа.

– Атрибути: Теги можуть мати атрибути, які надають додаткову інформацію про елемент. Наприклад, ``href`` - атрибут для посилання, що вказує URL адресу.

– Структура: HTML-документи зазвичай складаються з декількох основних елементів, таких як ``<!DOCTYPE html>``, ``<html>``, ``<head>``, та ``<body>``, які визначають тип документа, заголовок, метадані та вміст сторінки відповідно.

– Елементи контенту: HTML надає різні елементи контенту для вставки тексту, зображень, відео, аудіо та іншого контенту на сторінці.

HTML є основою для будь-якого веб-сайту і використовується разом з CSS (Cascading Style Sheets) для стилізації вигляду сторінок та JavaScript для надання їм динаміки та інтерактивності.

2. CSS (Cascading Style Sheets) - це мова стилів, що використовується для визначення вигляду та форматування елементів на веб-сторінках, написаних за допомогою HTML або інших мов розмітки. CSS дозволяє розділити зовнішній вигляд веб-сайту від його структури та контенту, що робить код більш організованим та підтримуваним.[6]

Основні концепції CSS включають в себе:

– Селектори: CSS використовує селектори для вибору елементів HTML, які потребують стилізації. Селектори можуть бути елементарними (наприклад, ``p`` для параграфів), класами (наприклад, ``.class``), ідентифікаторами (наприклад, ``#id``) або іншими атрибутами.

– Властивості і значення: CSS використовує властивості для визначення стилів, таких як кольори, розміри шрифтів, відступи, рамки і так далі. Кожна властивість має значення, яке вказує, яким чином ця властивість буде застосовуватися до вибраних елементів.



– Каскадування і наслідуванність: CSS використовує концепцію каскадування, що дозволяє властивостям елементів з однаковими селекторами перевизначати один одного. Також використовується спадковість, коли стилі, визначені для батьківських елементів, можуть успадковуватися дочірніми елементами.

– Медіазапити: CSS дозволяє визначати стилі, які будуть застосовуватися в залежності від характеристик пристрою, на якому відображається веб-сторінка, таких як розмір екрану, орієнтація та інші.

CSS дуже потужний інструмент, який дозволяє розробникам з легкістю керувати виглядом та макетом веб-сторінок, забезпечуючи при цьому високий рівень контролю та гнучкість.

3. JavaScript (JS) - це мова програмування, яка використовується для надання динамічної функціональності веб-сторінкам. Вона дозволяє розробникам створювати інтерактивні елементи на сторінках, здійснювати обробку подій, взаємодію з користувачем та змінювати вміст сторінок в реальному часі без перезавантаження.[7]

Основні концепції JavaScript включають в себе:

– 1. Змінні та типи даних: JavaScript дозволяє використовувати змінні для збереження даних різних типів, таких як числа, рядки, масиви, об'єкти тощо.

– 2. Оператори і вирази: JavaScript має широкий набір операторів, таких як арифметичні, порівняння, логічні тощо, для виконання різноманітних операцій над даними.

– 3. Функції: Функції в JavaScript використовуються для організації коду та виконання певних дій. Вони можуть бути оголошеними, виразами, анонімними та іншими.

– 4. Події і обробники подій: JavaScript дозволяє взаємодіяти з різними подіями, які виникають на сторінці, такими як клік миші, натискання

клавіші, завантаження сторінки тощо. Ви можете призначити обробники подій для реагування на ці події та виконання певних дій.

– 5. Об'єкти і методи: JavaScript базується на об'єктно-орієнтованому підході, де об'єкти представляють собою сутності з властивостями і методами.

– 6. DOM (Document Object Model): JavaScript може взаємодіяти з HTML і CSS через DOM, який представляє структуру документа як дерево об'єктів. Це дозволяє змінювати вміст сторінки, стилізувати елементи та багато іншого.[8]

JavaScript широко використовується для створення веб-додатків, веб-сайтів, ігор та багатьох інших веб-проектів.

Технології описані в даному пункті – це базові технології, які використовуються майже на кожному сучасному вебсайті. Виключенням може слугувати мова програмування JavaScript, адже її використовують, якщо хочуть надати вебсайту динамічності, створити сценарії анімації або інші скрипти.

Для реалізації проєкту усі технології описані вище використовуються. Використовуються саме вони через переваги та особливості описані вище та через свою відносну простоту та високу надійність а також велику кількість інформації в мережі Інтернет.

### **2.1.3 Технології для веброзробки серверної частини**

1. MySQL - це відкрита система керування базами даних (СКБД), яка використовується для зберігання та управління реляційними даними. Вона є однією з найпопулярніших СКБД у світі і використовується для різноманітних цілей, включаючи веб-розробку, бізнес-додатки, аналітику даних та інше. [9]

Основні характеристики MySQL включають в себе:

– Реляційна модель даних: MySQL використовує реляційну модель даних, де дані зберігаються у вигляді таблиць, які взаємно пов'язані за допомогою ключів. Це дозволяє легко організовувати та управляти великими обсягами даних.

– Мова запитів SQL: MySQL використовує мову структурованих запитів (SQL) для взаємодії з базою даних. SQL дозволяє виконувати різноманітні операції, такі як вставка, оновлення, видалення та вибірка даних з бази даних.

– Масштабованість і продуктивність: MySQL підтримує високу масштабованість і продуктивність, що робить його ідеальним вибором для веб-проектів з великою кількістю користувачів та обсягом даних.

– 4. Багатопоточність: MySQL може обробляти багато запитів одночасно, що дозволяє забезпечити швидкий доступ до даних для багатьох користувачів одночасно.

– 5. Підтримка транзакцій: MySQL підтримує транзакції, що дозволяють групувати кілька операцій у логічні одиниці та забезпечувати атомарність, консистентність, ізоляцію та довговічність (ACID властивості).

MySQL є вільною та відкритою системою, що означає, що вона доступна безкоштовно для використання та розповсюдження, а також має відкритий вихідний код, що дозволяє розробникам налаштовувати її під свої потреби. Вона підтримується широким спільнотою розробників і має велику кількість документації та ресурсів для вивчення.[7]

2. PHP (Hypertext Preprocessor) - це серверна мова програмування, яка використовується для розробки динамічних веб-сайтів і веб-додатків. Вона широко використовується в індустрії веб-розробки через свою простоту в освоєнні та потужність.[10]

Основні характеристики PHP включають в себе:

– Серверна мова програмування: PHP виконується на стороні сервера, що означає, що код PHP обробляється на веб-сервері перед відправленням вмісту сторінки клієнтам. Це дозволяє генерувати HTML, керувати базами даних, обробляти форми і багато іншого до того, як сторінка відправиться на клієнтську сторону.

– Динамічна генерація вмісту: PHP дозволяє вбудовувати свій код безпосередньо в HTML сторінки, що дозволяє створювати динамічний контент на основі різних умов та даних.

– 3. Підтримка різних типів баз даних: PHP легко інтегрується з різними СКБД, такими як MySQL, PostgreSQL, SQLite, Oracle та іншими, що дозволяє розробникам працювати з даними веб-додатків.

– 4. Широке співробітництво і спільнота: PHP має велику спільноту розробників та багато ресурсів для вивчення, включаючи документацію, форуми, онлайн-курси та бібліотеки коду.

– 5. Безкоштовність і відкритий код: PHP є вільною та відкритою мовою програмування, що означає, що вона доступна для використання безкоштовно та має відкритий вихідний код, що дозволяє розробникам адаптувати її під свої потреби.

PHP є дуже популярним вибором для розробки веб-додатків різного рівня складності, від простих веб-сайтів до складних корпоративних систем. Вивчення PHP може бути важливим кроком для тих, хто хоче стати веб-розробником або розширити свої навички в цій сфері.

Для розробки серверної частини були обрані дані технології через свої переваги та особливості. СУБД MySQL як і мова програмування PHP – є одними з найпопулярніших для створення backend-у вебсайтів. Для них написана офіційна документація та присутньо багато інформації в мережі Інтернет.

### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

#### 3.1 Проектування БД

В даному пункті роботи описана програмна реалізація серверної частини майбутнього проєкту, а саме: процес створення БД та функціонал для роботи з нею(додавання\видалення товару, реєстрація\авторизація користувача).

Першим кроком у реалізації backend частини проєкту – це створення БД на основі вимог до проєкту. Виходячи з ТЗ, база даних має складатись з декількох таблиць, а саме: «користувачі», «товари» та «корзина товарів». На рисунку 3.1 зображена концептуальна модель БД «Gold Star»

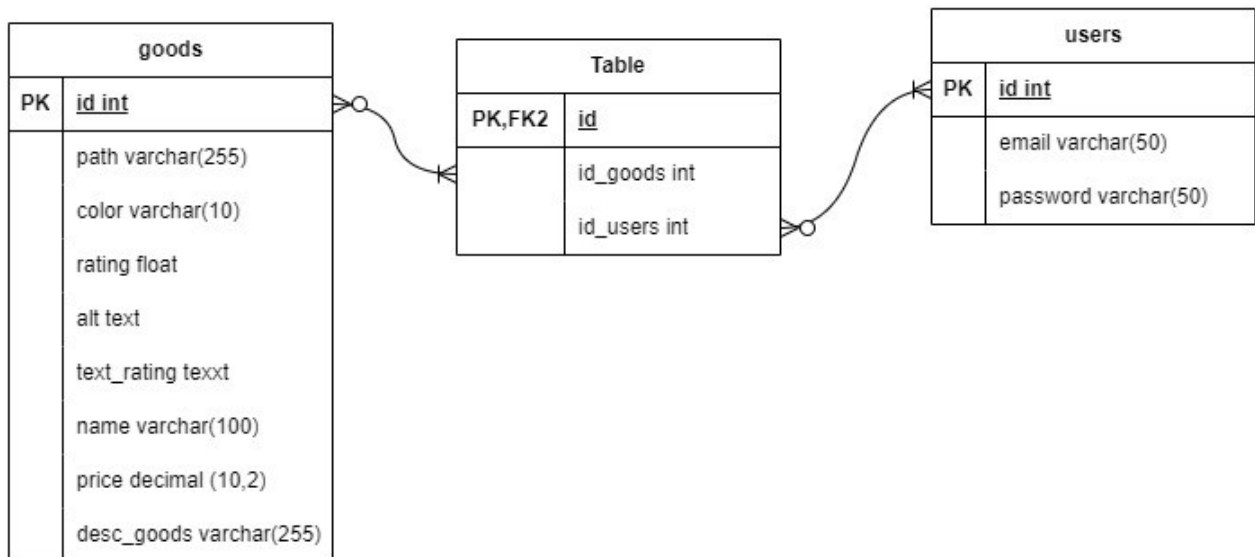


Рисунок 3.1 - концептуальна модель БД «Gold Star»

Таблиця «goods» - описує сутність товару, складається з наступних колонок:

Id – номер запису.

Path – шлях до зображення.

Color – колір товару\фону сторінки.

Rating – рейтинг товару.

Alt – альтернативний підпис для відповідного атрибуту тега зображення

Text\_rating – короткий опис рейтинг\характеристика.

Name – назва товару.

Price – ціна товару.

Desc\_goods – опис товару.

Таблиця «users» - описує сутність користувача, складається з наступних колонок:

Id – номер запису.

Email – електронна адреса користувача.

Password – пароль користувача.

Таблиця «shopping\_cart» - описує сутність корзини товарів, складається з наступних колонок:

Id – номер запису.

Id\_user – номер користувача, який додав товар до корзини.

Id\_goods – номер товару, який був доданий.

Зв'язок між таблицями «users» та «goods» - багато до багатьох, адже один користувач може додати багато товарів до корзини, в той самий час один і той же товар може бути у декількох користувачів.

Проаналізувавши сутність, використовувані в моделі ІС, перейдемо до реалізації структури БД. Для цього представимо імена необхідних таблиць, атрибутів, типів, їх призначення та обмеження (табл. 3.1)

Таблиця 3.1 Структура БД

Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
goods	id	Номер товару	INT	PK	Не пустий
	path	Шлях до зображення	VARCAHR		Не пустий
	color	Колір товару	VARCAHR		Не пустий
	rating	Рейтинг товару	FLOAT		Не пустий
	alt	Альтернативний підпис для зображення	VARCAHR		
	Taxt_rating	Текст рейтингу	TEXT		Не пустий
	name	Назва товару	VARCAHR		Не пустий
	price	Ціна товару	DECIMAL		Не пустий
	Desc_goods	Опис товару	VARCAHR		Не пустий
Shopping_car t	id	Номер запису	INT	PK	Не пустий
	Id_goods	Номер товару	INT	FK	Не пустий
	Id_users	Номер користувача	INT	FK	Не пустий
users	id	Номер користувача	INT	PK	Не пустий
	email	Електронна адреса	VARCAHR		Не пустий
	password	Пароль	VARCAHR		Не пустий

### 3.2 Розробка клієнтської частини

При розробці клієнтської частини, основними труднощами виступили дві вимоги: лендінг та, при цьому, до десяти один товару. На цій основі було розроблено дизайн вебсайту(рис. 3.2)

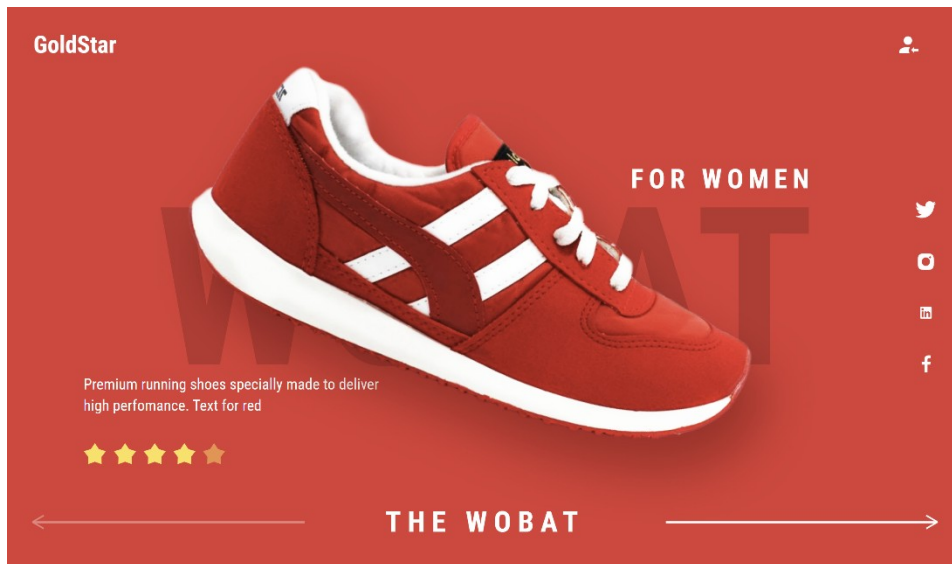


Рисунок 3.2 - Вебсайт «Gold Star»

Основні фішки проєкту:

– Розмір вебсайту. Розмір залежить від пристрою з якого відкривається вебсайт. При цьому на ньому немає сролу. На рисунку 3.3 зображено вигляд проєкту на мобільному пристрої.

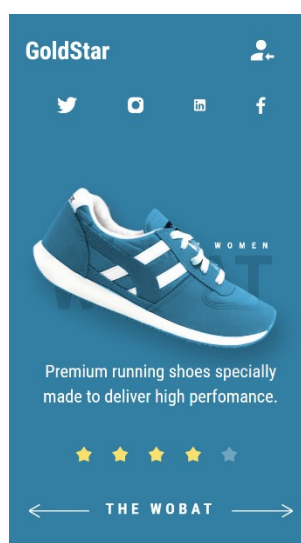


Рисунок 3.2 - Вебсайт «Gold Star»



– Перегляд товару. Відбувається за рахунок перелистування слайдеру, на якому і зображені одиниці товару. При цьому слайдер має вертикальне перемикання. Перемикання відбуваються за рахунок відповідних елементів керування – кнопок(рис. 3.4) або за рахунок свайпу.



Рисунок 3.4 – Елементи керування слайдером

– Сценарій анімації та попереднє завантаження. При завантаженні вебсайту, користувача зустрічає так званий прелоадер(рис. 3.5). Після завантаження всього контенту текст прелоадеру переміщується на своє місце(рис. 3.4) в блоці керування.

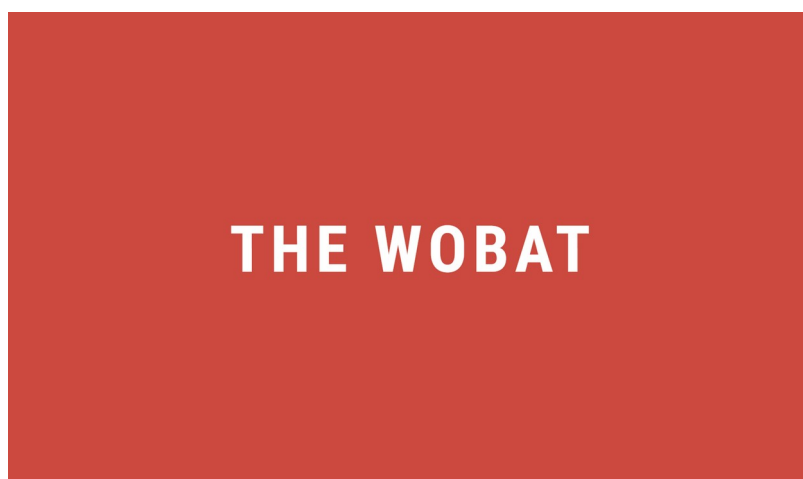


Рисунок 3.5 – Прелоадер вебсайту «Gold Star»

Через деякий час після попередньої анімації з'являються наступні блоки(рис. 3.6)

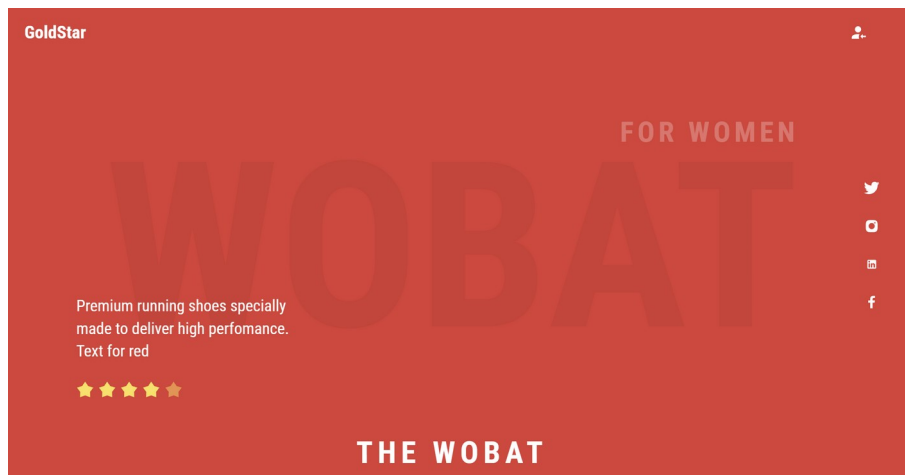


Рисунок 3.6 – Сценарій анімації вебсайту «Gold Star»

Після чого з'являється сам товар. Дане рішення дозволяє прикути увагу та зацікавити користувача.

– Колір. У кожного товару присутній свій унікальний колір, при перейманні слайдів змінюється фоновий колір сторінки та модальних вікно(реєстрації\авторизації та кошика)(рис. 3.7). Зміна кольору відбувається з певною плавністю.

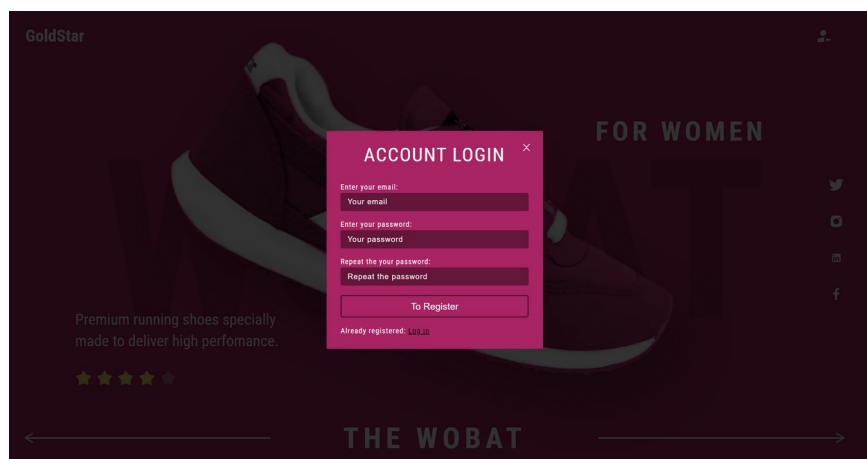


Рисунок 3.7 – Вікно реєстрація користувача

– Кошик. Після входу в особистий кабінет, з'являється можливість додавати товар в кошик(рис. 3.8)

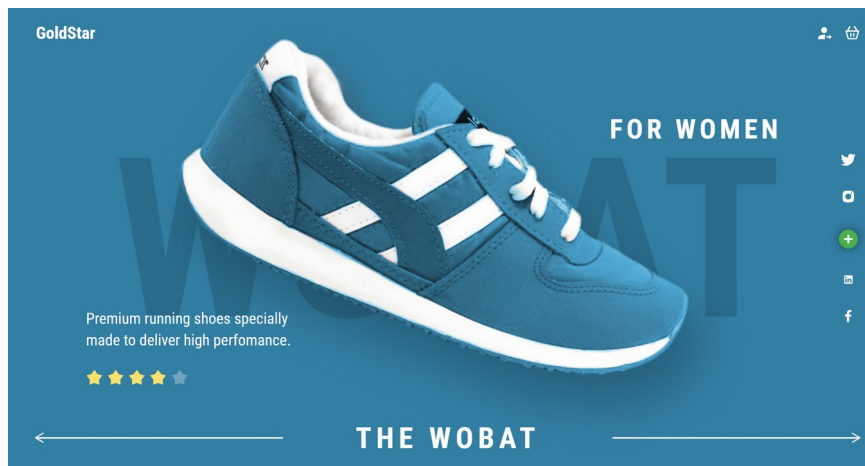


Рисунок 3.8 – Вебсайт «Gold Star»

Після реєстрації, змінюється інтерфейс користувача, з'являється кнопка додати товар у кошик, кнопка вийти з особистого кабінету та кошик товарів. При кліку на кнопку додати товар, він додається в кошик(рис. 3.9)



Рисунок 3.9 – Кошик товарів

### 3.3 Розробка серверної частини

Для файлів з логікою формату «.php», у проєкті була створена окрема папка з назвою «php» та папка «include» для файлу, який містить код для підключення до БД «gold\_star», код файлу «connect.php», з даної папки:

```
<?php

$mysql = new mysqli("localhost", "root", "", "gold_star");

if (!$mysql)
```

```
die('Database connecting error!');
```

```
$mysql->query("SET NAMES 'utf8'");
```

З цього прикладу, можна побачити як відбувається під'єднання до БД в мові програмування PHP. Обробку можливої помилки при під'єднанні до БД. Запит до БД з встановленням кодування. Також ,як можна зрозуміти з цього прикладу, в даному проєкті весь код PHP написаний у вигляді окремих файлів, які роблять певні дії, наприклад файл «reg\_user.php» відповідає за реєстрацію користувача а файл «getting\_goods.php» відповідає за отримання усіх товарів з БД. Усі файли наведені у додатках.

Даний підхід, коли окрема логіка реалізована в окремих файлах а не класах, через розмір вебсайту. Вебсайт представляє собою лендінг, тобто вебсайт який складається тільки з однієї частини.

Усі файли з папки «php» повертають масив даних, наприклад, якщо користувач намагається додати товар до кошика, який вже додано:

```
echo json_encode(array('success' => false, 'message' => 'Помилка:  
Товар вже доданий.'));
```

З цього прикладу видно як відбувається передача даних між серверною частиною та клієнтською, а саме за допомогою fetch-запитів.

Для того, щоб мати можливість працювати в БД у файлах папки «php» потрібно підключитись до файлу «connect.php», підключення відбувається наступним чином:

```
require_once './include/connect.php';
```

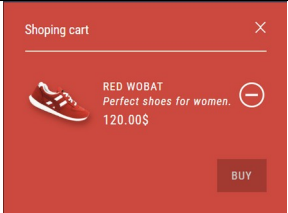
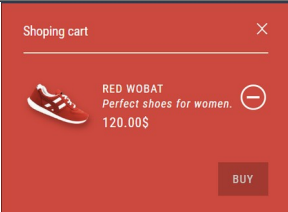
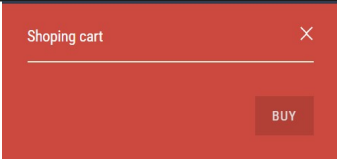

В результаті, була розроблена серверна частина для проєкту «Gold Star», вона складається з БД та семи файлів формату «.php», які відповідають за підключення до БД та функції вебсайту, такі як авторизація\реєстрація додаванню товару, тощо.

### 3.4 Тестування

Таблиця 3.1 - Тестування вебзастосунку

№	Умова, що тестується	Очікуваний результат	Фактичний результат	1/0
1	Реєстрація без ведення даних	Повідомлення про помилку		1
2	Введення не коректного email	Повідомлення про помилку		1
3	Введення паролю менше 8-символів	Повідомлення про помилку		1
4	Введення некоректного паролю	Повідомлення про помилку		1
5	Введення різних паролів	Повідомлення про помилку		1
6	Реєстрація користувачів з однаковими email	Повідомлення про помилку		1
7	Введення коректних даних	Успішна реєстрація користувача		1

Таблиця 3.1(Продовження) - Тестування вебзастосунку

8	Додавання товару	Товар успішно доданий		1
9	Додавання двох однакових товарів	Товар не доданий		1
10	Видалення товару з кошика	Товар видалений		1
11	Вихід з особистого кабінету	Користувач вийшов з особистого кабінету		1

## ВИСНОВКИ

Під час виконання дипломного проєкту було досліджено актуальність ІС продажу товару. Праведно аналіз принципів розробки ІС, виявлені типові характеристики та типовий функціонал даних вебсистем. Для отримання більш точної картини було проведено аналіз аналогів, у результаті були виявлені типові переваги та недоліки подібних ІС. Сформовані задачі до даної роботи.

Було проаналізовано технології та інструменти для розробки вебдодатків, у результаті були обрані певні їхні представники для розробки ІС. Для написання коду був обраний текстовий редактор від компанії Microsoft – VS Code. Як технології для клієнтської частини – HTML, CSS, JavaScript. Для серверної частини – мова програмування PHP, як СКБД – MySQL.

Під час програмної реалізації, першим кроком було розроблено базу даних, виходячи з вимог до проєкту. Після розробки БД, почалась розробка клієнтської частини, в результаті було отримано верстка проєкту та певний функціонал ІС. Для завершення проєкту було розроблена серверна частина, для взаємодії БД та ІС. В кінці проведено тестування програмного продукту.

В результаті дипломної роботи було отримано програмний продукт – ІС продажу товару «GoldStar», який повністю відповідає вимогам.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Traditional and Digital Catalogues [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bbpress.co.uk/news/pros-and-cons-of-traditional-and-digital-catalogues>
2. Puma [Електронний ресурс] - Режим доступу до ресурсу: <https://ua.puma.com/>
3. Intertop[Електронний ресурс] - Режим доступу до ресурсу: <https://intertop.ua/>
4. VS Code [Електронний ресурс] – <https://code.visualstudio.com/>
5. What Is HTML? Hypertext Markup Language Basics Explained [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hostinger.com/tutorials/what-is-html>
6. Mdn web docs [Електронний ресурс] - Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/CSS>
7. JavaScript.info [Електронний ресурс] - Режим доступу до ресурсу: <https://uk.javascript.info/>
8. Wikipedia [Електронний ресурс] - Режим доступу до ресурсу: <http://surl.li/aypb>
9. Youtube [Електронний ресурс] - Режим доступу до ресурсу: <https://www.youtube.com/>
10. The Good and the Bad of Vue.js Framework Programming [Електронний ресурс] – Режим доступу до ресурсу: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>
11. PHP Manual [Електронний ресурс] - Режим доступу до ресурсу: <https://www.php.net/manual/en/index.php>
12. What Is A User Interface, And What Are The Elements That Comprise one? [Електронний ресурс] – Режим доступу до ресурсу: <https://careerfoundry.com/en/blog/ui-design/what-is-a-user-interface/>



## ДОДАТОК А

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <link
      rel="shortcut icon"
      href="./img/misc/shopping_cart.svg"
      type="image/x-icon"
    />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin />
    <link
      href="https://fonts.googleapis.com/css2?
family=Roboto+Condensed:ital,wght@0,100..900;1,100..900&display=swap"
      rel="stylesheet"
    />
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/swiper@11/swiper-
bundle.min.css"
    />
    <link rel="stylesheet" href="css/style.css" />
    <title>GoldStar</title>
  </head>

  <body>
    <div class="wrapper">
      <header class="header">
        <div class="container">
          <div class="header__body">
            <div class="header__logo _anim-items left
_first">GoldStar</div>
```

```

<div class="header__buttons _anim-items right _first">
  <div class="buttons__element">
    <svg
      xmlns="http://www.w3.org/2000/svg"
      id="mdi-account-arrow-left"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      class="element__acc login _active"
    >
      <path
        d="M17 18H21V16H17V14L14 17L17 20V18M11 4C8.8 4
7 5.8 7 8S8.8 12 11 12 15 10.2 15 8 13.2 4 11 4M11 14C6.6 14 3 15.8 3
18V20H12.5C12.2 19.2 12 18.4 12 17.5C12 16.3 12.3 15.2 12.9 14.1C12.3
14.1 11.7 14 11 14"
        fill="white"
        class="element__color"
      />
    </svg>
    <svg
      xmlns="http://www.w3.org/2000/svg"
      id="mdi-account-arrow-right"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      class="element__acc logout"
    >
      <path
        d="M18 16H14V18H18V20L21 17L18 14V16M11 4C8.8 4
7 5.8 7 8S8.8 12 11 12 15 10.2 15 8 13.2 4 11 4M11 14C6.6 14 3 15.8 3
18V20H12.5C12.2 19.2 12 18.4 12 17.5C12 16.3 12.3 15.2 12.9 14.1C12.3
14.1 11.7 14 11 14"
        fill="white"
        class="element__color"
      />
    </svg>
  </div>
  <div class="buttons__element shopping_cart-image">

```

```

        </div>
    </div>
</div>
</div>
</header>
<main class="main _anim-wobat">
    <div class="container">
        <div class="main__product">
            <div class="swiper _anim-items _third">
                <div class="swiper-wrapper"></div>
            </div>
            <div class="main__background _anim-items
_second">WOBAT</div>
            <div class="main__rating _anim-items left _first">
                <div class="rating__text">
                    Premium running shoes specially made to deliver
high performance.
                </div>
                <div class="rating__stars">
                    <div class="stars__item">
                        <svg
                            width="35"
                            height="32"
                            viewBox="0 0 35 32"
                            fill="none"
                            xmlns="http://www.w3.org/2000/svg"
                        >
                            <path
                                d="M17.5515 0L23.7414 9.03181L34.2439
12.1278L27.5669 20.8057L27.868 31.7509L17.5515 28.0824L7.23498
31.7509L7.53601 20.8057L0.85903 12.1278L11.3616 9.03181L17.5515 0Z"
                                fill="#F8E16E"
                                class="item__color"
                            />
                        </svg>
                    </div>
                    <div class="stars__item">
                        <svg

```

```

        width="35"
        height="32"
        viewBox="0 0 35 32"
        fill="none"
        xmlns="http://www.w3.org/2000/svg"
    >
    <path
        d="M17.5515 0L23.7414 9.03181L34.2439
12.1278L27.5669 20.8057L27.868 31.7509L17.5515 28.0824L7.23498
31.7509L7.53601 20.8057L0.85903 12.1278L11.3616 9.03181L17.5515 0Z"
        fill="#F8E16E"
        class="item__color"
    />
</svg>
</div>
<div class="stars__item">
    <svg
        width="35"
        height="32"
        viewBox="0 0 35 32"
        fill="none"
        xmlns="http://www.w3.org/2000/svg"
    >
    <path
        d="M17.5515 0L23.7414 9.03181L34.2439
12.1278L27.5669 20.8057L27.868 31.7509L17.5515 28.0824L7.23498
31.7509L7.53601 20.8057L0.85903 12.1278L11.3616 9.03181L17.5515 0Z"
        fill="#F8E16E"
        class="item__color"
    />
</svg>
</div>
<div class="stars__item">
    <svg
        width="35"
        height="32"
        viewBox="0 0 35 32"
        fill="none"

```

```

        xmlns="http://www.w3.org/2000/svg"
    >
        <path
            d="M17.5515 0L23.7414 9.03181L34.2439
12.1278L27.5669 20.8057L27.868 31.7509L17.5515 28.0824L7.23498
31.7509L7.53601 20.8057L0.85903 12.1278L11.3616 9.03181L17.5515 0Z"
            fill="#F8E16E "
            class="item__color"
        />
    </svg>
</div>
<div class="stars__item">
    <svg
        width="35"
        height="32"
        viewBox="0 0 35 32"
        fill="none"
        xmlns="http://www.w3.org/2000/svg"
    >
        <path
            d="M17.5515 0L23.7414 9.03181L34.2439
12.1278L27.5669 20.8057L27.868 31.7509L17.5515 28.0824L7.23498
31.7509L7.53601 20.8057L0.85903 12.1278L11.3616 9.03181L17.5515 0Z"
            fill="#FFF"
            class="item__color"
        />
    </svg>
</div>
</div>
<div class="main__social _anim-items right _first">
    <a
        href="https://twitter.com"
        class="social__link"
        target="_blank"
    >
        <svg
            width="32"

```

```

        height="26"
        viewBox="0 0 32 26"
        fill="none"
        xmlns="http://www.w3.org/2000/svg"
    >
        <path

            fill="#FFFDFD"
            class="link__item"
        />
    </svg>
</a>
<a
    href="https://www.instagram.com"
    class="social__link"
    target="_blank"
>
    <svg
        width="25"
        height="26"
        viewBox="0 0 25 26"
        fill="none"
        xmlns="http://www.w3.org/2000/svg"
    >
        <path

            fill="#FFFDFD"
            class="link__item"
        />
    </svg>
</a>
<a href="#" class="social__link social__add">
    </a>
<a
    href="https://ua.linkedin.com/"
    class="social__link"
    target="_blank"

```

```

><svg
  xmlns="http://www.w3.org/2000/svg"
  id="mdi-linkedin"
  width="24"
  height="24"
  viewBox="0 0 24 24"
>
  <path
    d="M19 3A2 2 0 0 1 21 5V19A2 2 0 0 1 19 21H5A2
2 0 0 1 3 19V5A2 2 0 0 1 5 3H19M18.5 18.5V13.2A3.26 3.26 0 0 0 15.24
9.94C14.39 9.94 13.4 10.46 12.92 11.24V10.13H10.13V18.5H12.92V13.57C12.92
12.8 13.54 12.17 14.31 12.17A1.4 1.4 0 0 1 15.71 13.57V18.5H18.5M6.88
8.56A1.68 1.68 0 0 0 8.56 6.88C8.56 5.95 7.81 5.19 6.88 5.19A1.69 1.69 0
0 0 5.19 6.88C5.19 7.81 5.95 8.56 6.88 8.56M8.27
18.5V10.13H5.5V18.5H8.27Z"
    fill="#FFFDFD"
    class="link__item"
  />
</svg>
</a>
<a
  href="https://www.facebook.com"
  class="social__link"
  target="_blank"
>
  <svg
    width="15"
    height="26"
    viewBox="0 0 15 26"
    fill="none"
    xmlns="http://www.w3.org/2000/svg"
  >
    <path
      d="M4.8704
25.9999H9.71847V14.2058H13.7723L14.3845 9.61747H9.71847V6.67703C9.71847
5.35221 10.099 4.44746 12.0515 4.44746H14.55V0.327621C14.1198 0.279152
12.6472 0.149902 10.9264 0.149902C7.31925 0.149902 4.8704 2.29868 4.8704
6.22465V9.61747H0.799999V14.2058H4.8704V25.9999Z"

```

```

        fill="#FFDFDF"
        class="link__item"
    />
</svg>
</a>
</div>
</div>
</div>
</main>
<footer class="footer _anim-items _first">
    <div class="container">
        <div class="footer__body">
            <div class="swiper-button-prev"></div>
            <div class="swiper-button-next"></div>
        </div>
    </div>
</footer>
<div id="shoping" class="popup-shoping">
    <div class="shoping__fild"></div>
    <div class="shoping__body">
        <div class="shoping__content">
            <div class="shoping__close"></div>
            <div class="shoping__title">Shoping cart</div>
            <div class="shoping__items">
            </div>
            <div class="shoping__button"><span>BUY</span></div>
        </div>
    </div>
</div>
<div id="popup" class="popup">
    <div class="popup__fild"></div>
    <div class="popup__body">
        <div class="popup__content">
            <div class="popup__close"></div>
            <div class="popup__title">Account Login</div>
            <form action="#" class="popup__form reg _active"
method="post">
                <label for="email">Enter your email:</label>

```



```

<input type="text" name="login" placeholder="Your
email" />

<label for="pass">Enter your password:</label>
<input type="password" name="pass" placeholder="Your
password" />

<label for="rep_pass">Repeat the your
password:</label>

<input
  type="password"
  name="rep_pass"
  placeholder="Repeat the password"
/>

<div class="form__message">Error</div>

<button type="submit">To Register</button>
<div class="form__switch">
  Already registered: <a href="#">Log in</a>
</div>
</form>

<form action="#" class="popup__form login"
method="post">

  <label for="email">Enter your email:</label>
  <input type="text" name="login" placeholder="Your
email" />

  <label for="pass">Enter your password:</label>
  <input type="password" name="pass" placeholder="Your
password" />

  <div class="form__message">Error</div>

  <button type="submit">Log In</button>
  <div class="form__switch">
    I am not registered: <a href="#">Registration</a>
  </div>

```

```
        </form>
    </div>
</div>
</div>
</div>
</body>
```

```
<script src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-
bundle.min.js"></script>
<script src="./js/popup.js" type="module"></script>
<script src="./js/shoping.js" type="module"></script>
<script src="./js/animation.js" type="module"></script>
<script src="./js/script.js" type="module"></script>
</html>
```

## ДОДАТОК Б

### Файл «shopping\_card\_add.php»

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

    if(isset($data['idUser']) && isset($data['idGoods'])){
        $id_user = $data['idUser'];
        $id_goods = $data['idGoods'];

        require_once './include/connect.php';

        $result = $mysql->query("SELECT * FROM `shopping_cart` WHERE
`id_goods` = '$id_goods' AND `id_user` = '$id_user'");

        if($result->num_rows === 0){
            $mysql->query("INSERT INTO `shopping_cart` (`id`, `id_goods`,
`id_user`) VALUES (NULL, '$id_goods', '$id_user')");
            $mysql->close();
            echo json_encode(array('success' => true, 'message' => 'Товар
успішно доданий!'));
        }else{
            $mysql->close();
            echo json_encode(array('success' => false, 'message' =>
'Помилка: Товар вже доданий.'));
        }

        } else {
            echo json_encode(array('success' => false, 'message' =>
'Помилка: Дані не були передані коректно.'));
        }
    }
```

### Файл «reg\_user.php»

```
<?php
function valid($email, $pass, $repeat_pass) {
    if (filter_var($email, FILTER_VALIDATE_EMAIL) && $pass ===
$repeat_pass) {
        return true;
    }
```

```

    } else {
        return false;
    }
}

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

if(isset($data['email']) && isset($data['password']) &&
isset($data['repeatPassword'])) {
    $email = $data['email'];
    $password = bin2hex($data['password']);
    $repeat_password = bin2hex($data['repeatPassword']);

    require_once './include/connect.php';

    if(valid($email, $password, $repeat_password)){
        $result = $mysql->query("SELECT `id` FROM `users` WHERE `email`
= '$email'");

        if($result->num_rows === 0){
            $mysql->query("INSERT INTO `users` (`id`, `email`,
`password`) VALUES (NULL, '$email', '$password')");
        }else{
            echo json_encode(array('success' => false, 'message' =>
'Даний користувач вже зареєстрован!'));
        }

        $result = $mysql->query("SELECT `id` FROM `users` WHERE `email`
= '$email'");

        while($row = $result->fetch_assoc()){
            $id_user = $row["id"];
        }

    }else{
        echo json_encode(array('success' => false, 'message' => 'Дані
не пройшли валідацію.'));
    }
}

```

```

    $mysql->close();

    echo json_encode(array('success' => true, 'message' =>
'Користувача успішно зареєстровано.', 'id' => $id_user));
    } else {
        echo json_encode(array('success' => false, 'message' => 'Помилка:
Дані не були передані коректно.'));
    }

```

### Файл «log\_user.php»

```

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

if(isset($data['email']) && isset($data['password'])) {
    $email = $data['email'];
    $password = bin2hex($data['password']);

    require_once './include/connect.php';

    $result = $mysql->query("SELECT * FROM `users` WHERE `email` =
'$email' AND `password` = '$password'");

    $mysql->close();

    if($result->num_rows === 1){
        while($row = $result->fetch_assoc()){
            $id_user = $row["id"];
        }
        echo json_encode(array('success' => true, 'message' =>
'Користувача успішно увійшов.', 'id' => $id_user));
    }
    else{
        echo json_encode(array('success' => false, 'message' =>
'Помилка: такого користувача не було знайдено.'));
    }

} else {
    echo json_encode(array('success' => false, 'message' => 'Помилка:
Не повні данні.'));
}

```

```
}
```

### Файл «getting\_goods.php»

```
<?php
require_once './include/connect.php';

$result = $mysql->query("SELECT * FROM `goods`");
$data = array();

while($row = $result->fetch_assoc()){
    array_push($data, $row);
}

$mysql->close();

header('Content-Type: application/json');
```

```
echo json_encode($data);
```

### Файл «getting\_goods\_for\_shopping\_cart.php»

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

    if(isset($data['idUser'])){
        $id_user = $data['idUser'];
        $data = array();

        require_once './include/connect.php';

        $result = $mysql->query("SELECT g.`id`, g.`name`,
g.`desc_goods`, g.`price`, g.`path` FROM `shopping_cart` sc JOIN `goods`
g ON sc.`id_goods` = g.`id` WHERE sc.`id_user` = '$id_user'");

        while($row = $result->fetch_assoc()){
            array_push($data, $row);
        }

        $mysql->close();

        header('Content-Type: application/json');
```

```

        echo json_encode($data);
    }else {
        echo json_encode(array('success' => false, 'message' =>
'Помилка: Дані не були передані коректно.'));
    }

```

### Файл «delete\_item\_on\_shopping\_cart.php»

```

<?php

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

if(isset($data['idUser']) && isset($data['idGoods'])) {
    $id_user = $data['idUser'];
    $id_goods = $data['idGoods'];

    require_once './include/connect.php';

    $mysql->query("DELETE FROM `shopping_cart` WHERE `id_goods` =
'$id_goods' AND `id_user` = '$id_user'");

    $mysql->close();

    echo json_encode(array('success' => true, 'message' => 'Товар
успішно видалено.', 'id'=> $id_goods));
}else {
    echo json_encode(array('success' => false, 'message' => 'Помилка:
Дані не були передані коректно.'));
}

```

## ДОДАТОК В

### Файл «animation.js»

```
const animItemsFirst = document.querySelectorAll("_anim-
items._first");
const animItemsSecond = document.querySelectorAll("_anim-
items._second");
const animItemsThird = document.querySelectorAll("_anim-
items._third");

const wobatPreload = document.querySelector("_anim-wobat");
export const animDuration = 1000;
const animDurationWobat =
  parseFloat(
    window
      .getComputedStyle(wobatPreload, "::after")
      .getPropertyValue("transition-duration")
      .replace("s", "")
  ) * 1000;

export function animStart() {
  wobatPreload.classList.add("_active");
  setTimeout(animOnLoadFirst, animDurationWobat);
}
function animOnLoadFirst() {
  animItemsFirst.forEach((item) => {
    item.classList.add("_active");
  });
  setTimeout(animOnLoadSecond, animDuration);
}
function animOnLoadSecond() {
  animItemsSecond.forEach((item) => {
    item.classList.add("_active");
  });
  setTimeout(animOnLoadThird, animDuration);
}
function animOnLoadThird() {
```



```

    animItemsThird.forEach((item) => {
        item.classList.add("_active");
    });
}

```

### Файл «popup.js»

```

import { shoppingCart } from "./shopping.js";

const forms = document.querySelectorAll(".popup__form");
const openButton = document.querySelector(".login");
const closeButton = document.querySelector(".popup__close");
const field = document.querySelector(".popup__field");
const switchButtons = document.querySelectorAll(".form__switch a");
const buttonSendReq = document.querySelector(".popup__form.reg
button");
const buttonSendLog = document.querySelector(".popup__form.login
button");
const shoppingImage = document.querySelector(".shopping_cart-image");
const loginImage = document.querySelector(".element__acc.login");
const logoutImage = document.querySelector(".element__acc.logout");

function openOrCloseForm() {
    const popup = document.querySelector(".popup");

    popup.classList.toggle("_open");
}

switchButtons.forEach((button) => {
    button.addEventListener("click", (e) => {
        e.preventDefault();

        forms.forEach((form, index) => {
            form.classList.toggle("_active");
        });
    });
});

openButton.addEventListener("click", openOrCloseForm);
closeButton.addEventListener("click", openOrCloseForm);

```

```

field.addEventListener("click", openOrCloseForm);

const validateEmail = (email) => {
  if (email.value === "") {
    viewErrorMessage("Заповніть поле email.");
    email.classList.add("_error");
    return false;
  }

  if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email.value)) {
    viewErrorMessage("Не коректно введенний email.");
    email.classList.add("_error");
    return false;
  }

  return true;
};

const validatePassword = (password) => {
  if (password.value.length < 8) {
    viewErrorMessage("Маленька довжина паролю.");
    password.classList.add("_error");
    return false;
  }

  if (/[@#$$%^&*(),.?"':{}|<>]/.test(password.value)) {
    viewErrorMessage("Не коректно введенний пароль.");
    password.classList.add("_error");
    return false;
  }

  if (!/[A-ZА-ЯиIiİİ]/.test(password.value.charAt(0))) {
    viewErrorMessage("Перша літера паролю маленька.");
    password.classList.add("_error");
    return false;
  }

  return true;
};

```

```

function validateRepPassword(repeatPassword, password) {
  if (repeatPassword.value.length < 8) {
    viewErrorMessage("Маленька довжина повторного паролю.");
    repeatPassword.classList.add("_error");
    return false;
  }

  if (/[@#$$%^&*(),.?"':{}|<>]/.test(repeatPassword.value)) {
    viewErrorMessage("Не коректно введений повторний пароль.");
    repeatPassword.classList.add("_error");
    return false;
  }

  if (!/[A-Za-zÀàIïİï]/.test(repeatPassword.value.charAt(0))) {
    viewErrorMessage("Перша літера повторного паролю маленька.");
    repeatPassword.classList.add("_error");
    return false;
  }

  if (password.value !== repeatPassword.value) {
    viewErrorMessage("Паролі не співпадають.");
    repeatPassword.classList.add("_error");
    return false;
  }

  return true;
}

buttonSendReq.addEventListener("click", async (event) => {
  event.preventDefault();
  closeErrorMessage();
  resetInput();
  const email = document.querySelector(".popup__form.reg
input[name='login']");
  const password = document.querySelector(
    ".popup__form.reg input[name='pass']"
  );
  const repeatPassword = document.querySelector(
    ".popup__form.reg input[name='rep_pass']"
  );

```

```

);
const error =
  validateEmail(email) &&
  validatePassword(password) &&
  validateRepPassword(repeatPassword, password);

if (error) {
  const dataUser = {
    email: email.value,
    password: password.value,
    repeatPassword: repeatPassword.value,
  };

  const requestOptions = {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(dataUser),
  };
  viewErrorMessage("Реєстрація...");

  await fetch("../php/reg_user.php", requestOptions)
    .then((response) => response.json())
    .then((data) => {
      visibleShopping();
      localStorage.setItem("userId", data.id);
      viewErrorMessage("Ви успішно зареєстровані.");
      shoppingCart();
      switchImage();
    })
    .catch((error) => {
      console.error("Помилка:", error);
      viewErrorMessage("Помилка при реєстрації.");
    });
}
});
buttonSendLog.addEventListener("click", async (event) => {

```

```

event.preventDefault();
closeErrorMessage();
resetInput();
const email = document.querySelector(
  ".popup__form.login input[name='login']"
);
const password = document.querySelector(
  ".popup__form.login input[name='pass']"
);
const error = validateEmail(email) && validatePassword(password);

if (error) {
  const dataUser = {
    email: email.value,
    password: password.value,
  };
  const requestOptions = {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(dataUser),
  };
  viewErrorMessage("Вхід до акаунту...");

  await fetch("../php/log_user.php", requestOptions)
    .then((response) => response.json())
    .then((data) => {
      if (data.success) {
        visibleShopping();
        localStorage.setItem("userId", data.id);
        viewErrorMessage("Ви успішно увійшли до кабінету.");
        switchImage();
        shoppingCart();
        openOrCloseForm();
      } else {
        viewErrorMessage("Помилка при вході.");
      }
    })
}

```

```

    })
    .catch((error) => {
        console.error("Помилка:", error);
        viewErrorMessage("Помилка при вході.");
    });
}
});
function viewErrorMessage(message) {
    const messageField = document.querySelector(".form__message");
    messageField.classList.add("_active");
    messageField.textContent = message;
}
function closeErrorMessage() {
document.querySelector(".form__message").classList.remove("_active");
}
function resetInput() {
    const input = document.querySelectorAll(".popup__form input");
    input.forEach((e) => {
        e.classList.remove("_error");
    });
}
function visibleShopping() {
    shopingImage.innerHTML = `
<svg
        width="31"
        height="30"
        viewBox="0 0 31 30"
        fill="none"
        xmlns="http://www.w3.org/2000/svg"
        class="shopping"
    >
        <path
            fill-rule="evenodd"
            clip-rule="evenodd"

            class="element__color"
            fill="white"

```

```

        />
    </svg>
    `;
}
if (localStorage.getItem("userId")) {
    visibleShopping();
    shopingCart();
}

function switchImage() {
    loginImage.classList.toggle("_active");
    logoutImage.classList.toggle("_active");
}

const logoutButton =
document.querySelector(".element__acc.logout");

logoutButton.addEventListener("click", () => {
    const addGoodsButton = document.querySelector(".social__add");
    addGoodsButton.classList.remove("_active");
    localStorage.removeItem("userId");
    shopingImage.innerHTML = "";
    switchImage();
});

if (localStorage.getItem("userId")) {
    switchImage();
}

```

### Файл «script.js»

```

import { animStart, animDuration } from "./animation.js";

const textsRating = [];
const colors = [];

function addSlide(src, color, rating, id) {
    const swiperWrapper = document.querySelector(".swiper-wrapper");
    const div = document.createElement("div");
    const img = document.createElement("img");

```

```

div.classList.add("swiper-slide");

img.id = id;
img.src = `./img/product/${src}`;
img.setAttribute("color", color);
img.setAttribute("rating", rating);

div.appendChild(img);

swiperWrapper.appendChild(div);
}

(async () => {
  await fetch("../php/getting_goods.php")
    .then((response) => {
      if (!response.ok) {
        throw new Error("Network response was not ok");
      }
      return response.json();
    })
    .then((data) => {
      data.forEach((e) => {
        setTimeout(animStart, animDuration);
        textsRating.push(e["text_rating"]);
        addSlide(e["path"], e["color"], e["rating"], e["id"]);
      });
    })
    .catch((error) => {
      console.error("There has been a problem with fetch
operation:", error);
    });

  const idSlide = document.querySelectorAll(".swiper-slide img");
  localStorage.setItem("activeSlider", idSlide[0].id);

  const slideColors = document.querySelectorAll("img[color]");
  slideColors.forEach((e) => {

```



```

        colors.push(e.getAttribute("color"));
    });
    rating(0);
}) ();

const swiper = new Swiper(".swiper", {
  autoplay: {
    delay: 10000,
    disableOnInteraction: false,
  },
  direction: "vertical",
  mousewheel: true,
  navigation: {
    nextEl: ".swiper-button-next",
    prevEl: ".swiper-button-prev",
  },
  speed: 1000,
  keyboard: {
    enabled: true,
  },
  on: {
    slideChange: function () {
      const idSlide = document.querySelectorAll(".swiper-slide
img");

      localStorage.setItem("activeSlider",
idSlide[this.activeIndex].id);

      if (document.querySelector(".popup._open")) {
        swiper.autoplay = false;
      } else {
        swiper.autoplay = true;
      }

      const blockForChangeColor = [
        document.querySelector(".wrapper"),
        document.querySelector(".popup__content"),
        document.querySelector(".shoping__button span"),

```

```

        document.querySelector(".shoping__content"),
    ];

    blockForChangeColor.forEach((e) => {
        e.style.backgroundColor = colors[this.activeIndex];
    });

    rating(this.activeIndex);
},
},
});

function rating(numberOfSlide) {
    const ratingText = document.querySelector(".rating__text");
    ratingText.textContent = textsRating[numberOfSlide];

    const slideRatings = document.querySelectorAll("img[rating]");
    const stars = document.querySelectorAll(".item__color");
    const ratings = [];
    let colorIndex = 1;

    slideRatings.forEach((e) => {
        ratings.push(e.getAttribute("rating"));
    });

    stars.forEach((e) => {
        e.style.fill = "#ffffff4b";
    });

    stars.forEach((e, i) => {
        if (i < ratings[numberOfSlide]) {
            e.style.fill = `rgba(248,225,110, 1)`;
        }
    });
    if (ratings[numberOfSlide] % 1 > 0) {
        colorIndex = colorIndex * (ratings[numberOfSlide] % 1);
        stars[
            Math.floor(ratings[numberOfSlide])

```

```

    ].style.fill = `rgba(248,225,110, ${colorIndex})`;
  }
}

```

### Файл «shoping.js»

```

export function shopingCart() {
  const buttonOpen = document.querySelector(".shoping");
  const buttonClose = document.querySelector(".shoping__close");
  const fieldClose = document.querySelector(".shoping__field");
  const buttonGoodsAdd = document.querySelector(".social__add");

  buttonGoodsAdd.classList.add("_active");
  updateShoppingCart();

  buttonOpen.addEventListener("click", openOrCloseForm);
  buttonClose.addEventListener("click", openOrCloseForm);
  fieldClose.addEventListener("click", openOrCloseForm);
  buttonGoodsAdd.addEventListener("click", async (event) => {
    event.preventDefault();
    if (
      localStorage.getItem("userId") &&
      localStorage.getItem("activeSlider")
    ) {
      const dataReq = {
        idUser: localStorage.getItem("userId"),
        idGoods: localStorage.getItem("activeSlider"),
      };
      const requestOptions = {
        method: "POST",
        headers: {
          "Content-Type": "application/json",
        },
        body: JSON.stringify(dataReq),
      };
      await fetch("../php/shopping_card_add.php", requestOptions)
        .then((response) => {
          return response.json();
        })
    }
  });
}

```

```

        .then((data) => {
            updateShoppingCart();
        })
        .catch((error) => {
            console.error("Помилка:", error);
        });
    } else {
        alert("Відсутній користувач або товар!");
    }
});

function openOrCloseForm() {
    const popup = document.querySelector(".popup-shopping");

    popup.classList.toggle("_open");
}

async function updateShoppingCart() {
    const dataReq = {
        idUser: localStorage.getItem("userId"),
    };

    const requestOptions = {
        method: "POST",
        headers: {
            "Content-Type": "application/json",
        },
        body: JSON.stringify(dataReq),
    };

    await fetch("../php/getting_goods_for_shopping_cart.php",
requestOptions)
        .then((response) => {
            return response.json();
        })
        .then((data) => {
            const shopingItems =
document.querySelector(".shopping__items");
            shopingItems.innerHTML = ``;
            if (data.length) {

```

```

        data.forEach((element) => {
            shoppingItems.innerHTML += `
            <div class="items__item" >
            
            <div class="item__desc">
                <div class="desc__title">${element.name}</div>
                <div
class="desc__subtitle">${element.desc_goods}</div>
                <div
class="desc__price"><span>${element.price}</span>${</div>
                </div>
                <div class="item__remove remove__goods" id="$
{element.id}"></div>
            </div>
            `;
        });
    }

    removeGoods();
})
.catch((error) => {
    console.error("Помилка:", error);
});
}

function removeGoods() {
    const buttonGoodsRemove =
document.querySelectorAll(".remove__goods");
    buttonGoodsRemove.forEach((element) =>
        element.addEventListener("click", async () => {
            const dataReq = {
                idUser: localStorage.getItem("userId"),
                idGoods: element.id,
            };
            const requestOptions = {
                method: "POST",
                headers: {
                    "Content-Type": "application/json",
                },
            },

```

```
        body: JSON.stringify(dataReq),
    };

    await fetch("../php/delete_item_on_shopping_cart.php",
requestOptions)
        .then((response) => {
            return response.json();
        })
        .then((data) => {
            console.log(data);
            updateShoppingCart();
        })
        .catch((error) => {
            console.error("Помилка:", error);
        });
    })
);
}
}
```