

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента Войченко Валентини Вікторівни
(ПІБ)

академічної групи 122-19-3
(шифр)

спеціальності 122 Комп'ютерні науки
(код і назва спеціальності)

освітньої програми Комп'ютерні науки
(назва освітньої програми)

на тему: Розробка інтернет-магазину
з продажу ювелірних виробів та аксесуарів
з використанням JavaScript

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Гуліна І.Г.			
розділів:				
спеціальний	доц. Гуліна І.Г.			
економічний	проф. Вагонова О.Г.			
Рецензент				
Нормоконтролер	доц. Гуліна І.Г.			

Дніпро
2023

РЕФЕРАТ

Пояснювальна записка: 61 с., 6 рис., 3 дод., 20 джерел.

Об'єкт розробки: сайт з продажу ювелірних виробів та аксесуарів.

Мета кваліфікаційної роботи: створення сайту з продажу комплектуючих для комп'ютерів, використовуючи мову програмування JavaScript.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточняється постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформи для розробки, виконано проектування і розробка програми, описана робота програми, алгоритм і структура її функціонування, а також виклик та завантаження програми, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення полягає у створенні web-сайту для реалізації продажу ювелірних виробів та аксесуарів.

Актуальність даного продукту визначається великим попитом на подібні розробки, що реалізує продаж товарів.

Список ключових слів: САЙТ, JavaScript, ReactJS, КОМП'ЮТЕР, ІНТЕРФЕЙС, АЛГОРИТМ, ПРОЕКТУВАННЯ, НАВІГАЦІЯ, WEB, РОЗРОБКА.

ABSTRACT

Explanatory note: 61 pp., 6 fig, 3 appendix, 20 sources.

Object of development: site for the sale of jewelry and accessories.

The purpose of the qualification work: to create a site for the sale jewelry and accessories using the JavaScript programming languages.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the task.

In the first section the subject branch is analyzed, the urgency of the task and the purpose of development are determined, the statement of the task is formulated, the requirements to the software implementation, technologies and software are specified.

The second section analyzes the available solutions, selected platforms for development, designed and developed the program, describes the program, algorithm and structure of its operation, as well as calling and loading the program, determines the input and output data, describes the parameters of hardware.

The economic section determines the complexity of the developed information system, calculates the cost of work to create a program and calculates the time for its creation.

The practical meaning is to create a web site for the sale of jewelry and accessories.

The relevance of this product is determined by the great demand for similar developments, which realizes the sale of goods.

List of keywords: SITE, JavaScript, ReactJS, COMPUTER, INTERFACE, ALGORITHM, DESIGN, NAVIGATION, WEB, DEVELOPMENT.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ЗМІСТ	5
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1	10
АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	10
1.1. Загальні відомості з предметної галузі.....	10
1.2. Призначення розробки та галузь застосування	11
1.3. Підстава для розробки.....	12
1.4. Постановка завдання	13
1.5. Ви моги до програми або програмного виробу	13
1.5.1. Вимоги до функціональних характеристик.....	13
1.5.2. Вимоги до інформаційної безпеки	14
1.5.3. Вимоги до складу та параметрів технічних засобів	14
1.5.4. Вимоги до інформаційної та програмної сумісності	14
РОЗДІЛ 2	15
ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	15
2.1. Функціональне призначення програми	15
2.2. Опис застосованих математичних методів	15
2.3. Опис використаних технологій та мов програмування.....	15
2.4. Опис структури системи та алгоритмів її функціонування	17
2.5. Обґрунтування та організація вхідних та вихідних даних програми	27
2.6. Опис розробленого програмного продукту	27
2.6.1. Використані технічні засоби	27
2.6.2. Використані програмні засоби.....	28
2.6.3. Виклик та завантаження програми	28
2.6.4. Опис інтерфейсу користувача	Ошибка! Закладка не определена.
РОЗДІЛ 3	30

ЕКОНОМІЧНИЙ РОЗДІЛ	30
3.1. Визначення трудомісткості розробки програмного забезпечення.....	30
3.2. Розрахунок витрат на створення програми.....	34
ВИСНОВКИ	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	37
ДОДАТОК А	38
ЛІСТИНГ ПРОГРАМИ.....	38
ДОДАТОК Б.....	60
ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ	60
ДОДАТОК В	61
ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ	61

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

ЕОМ – електронно-обчислювальна машина;

ПЗ – програмне забезпечення;

ПК – портативний комп'ютер.

ВСТУП

Темою даної кваліфікаційної роботи є створення сайту з продажу ювелірних виробів та аксесуарів, використовуючи мову програмування JavaScript.

Метою даної кваліфікаційної роботи є вивчення інструментальних засобів програми WSCode, вивчення та визначення різних методів створення WEB-сторінок.

У 1994 році Джефф Бізос перший отримав думку про створення інтернет-магазину. Ця ідея була заснована на роздумах про можливість людей, які спілкуються за допомогою інтернету, можуть захотіти замовляти товари та послуги онлайн. Після роздумів про можливі товари, що могли б вільно продаватись, тобто товарів які було легко безпечно відправляти поштою, у 1995 році був створений перший інтернет-магазин Amazon, який все ще є одним з найбільш активним інтернет-магазином.

На даний момент інтернет-магазини стали важливою частиною нашого життя. За 2022 рік, згідно з даними дослідження, 78% українців купують онлайн зі смартфона, 40% із них – через мобільні додатки інтернет-магазинів. Також за цей рік було встановлено, що найпопулярнішими категоріями товарів які замовляють є одяг, електронна техніка та аксесуари для цієї техніки.

У сучасному світі покупки через інтернет-магазини стають з кожним роком легше. Інтерфейс з роками ставав все простішим, але зрозумілим для звичайного користувача.

Велику роль розвитку інтернет-магазинів відіграв розвиток можливостей оплати за допомогою карток. Якщо в далекому минулому для оплати товару потрібно було заповнювати форму оплати вводячи усю інформацію своєї банківської картки, то зараз можна використовувати оплату через банківські додатки або використовувати банківську картку підключену до GOOGLE гаманця, що допомагає користувачам бути впевненим у безпеці оплати даного інтернет-магазину.

Також найважливішою причиною такої популярності інтернет-магазинів – це неймовірна економія часу. Здійснюючи покупки в інтернеті ви зберігаєте свій час, сили, маєте більшу різноманітність доступних товарів чим у фізичному магазині. Усе що вам потрібно зробити – це оформити своє замовлення, передати інформацію куди вам мають відправити товар та оплатити.

Усі ці переваги використання магазинів у мережі змушує нас використовувати їх все більше з кожним роком, тому розвиток ніколи не буде зупинятись.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

Інтернет-магазин - це електронна торгівельна платформа, що дозволяє споживачам придбати товари або послуги через веб-браузер. Клієнти мають можливість знайти бажаний товар, відвідавши веб-сайт роздрібного продавця безпосередньо або шляхом пошуку серед інших постачальників. Продажі можуть відбуватися через різноманітні канали, такі як невеликий магазин у місцевості, великий роздрібний продавець, електронна комерція або навіть приватні особи, що продають товари через посередницькі платформи. Веб-сайт інтернет-магазину має складну структуру з основним меню та підменю. Основне меню містить найважливішу інформацію про категорії товарів та акційні пропозиції. Один з ефективних способів визначити, які сторінки включити в панель навігації, а які можна пропустити або використати як частину списку.

Приклад успішно структурованого веб-сайту можна помітити на ресурсах багатьох спортивних брендів. На кожному з цих сайтів присутня дуже проста система навігації, незважаючи на велику кількість різноманітних товарів, що пропонуються. Цей простий підхід має свою логіку. Коли ви шукаєте шорти, вам не потрібно починати з загальної категорії "шорти", де представлені товари для чоловіків і жінок. Замість цього, на веб-сайтах світових спортивних брендів ви можете просто прокрутити мишею до відповідної основної категорії, і далі варіанти стають все більш конкретними. Таким чином, основна категорія стає напівклієнтом, що допомагає швидко і зручно знайти потрібний товар.

Структура інтернет-магазину розгорнута від простого до складного принципу, де кожна категорія включає ще більше підкатегорій. Важливою складовою добре організованого сервісу є зручний пошук за категоріями, який повинен бути доступний як на головній сторінці, так і на рівні підкатегорій.

Запуск інтернет-магазину охоплює такі етапи, як маркетингові заходи для просування, оцінка якості товарів, залучення інтересу клієнтів та налагодження процесу обробки замовлень.

Аби відкрити інтернет-магазин, необхідно зробити такі кроки:

При виборі товару який ви збираєтесь реалізовувати за допомогою інтернет-магазину потрібно оцінити місцевий та глобальний попит. Місцевий попит частіше стає важливішим критерієм, адже покупці воліють купляти товари у місцевого продавця, що зменшує очікування доставки товару та зменшує оплату за доставку продукту.

Також вам потрібно обрати свою бізнес модель для реалізації свого товару.

Ім'я домену має відповідати вашим продуктам і бути легко запам'ятовуватися для вашої цільової аудиторії.

Створіть базу даних та WEB-сторінку магазину.

Налаштуйте платіжний шлюз та захист сайту. Для підключення безпечної платіжної програми ви можете використовувати вже існуючі спеціальні сервіси.

Для зручності клієнтів, інтернет-магазин повинен надавати різні способи оплати, такі як кредитні та дебетові картки, онлайн-гаманці, інтернет-банкінг та післяплата. Для забезпечення безпеки вашого сайту та підвищення довіри клієнтів, важливо мати сертифікат SSL. В наші дні Google рекомендує встановлювати сертифікат SSL для кожного веб-сайту.

1.2. Призначення розробки та галузь застосування

Інтернет надає можливість охопити всі куточки світу, що дозволяє знаходити клієнтів і надавати послуги практично будь-де на планеті. Загалом, населення Землі налічує 8 мільярдів людей, а кількість інтернет-користувачів станом на 2022 рік складає понад 5 мільярдів. Ці великі числа повинні розвіяти ваші страхи щодо конкуренції, оскільки у такому масштабі якісний продукт обов'язково знайде свою аудиторію. Робота в Інтернеті дозволяє вам самостійно

визначати час та тривалість своєї роботи. Ви можете працювати 4 години на день або 20, якщо прагнете заробляти більше та прискорювати свій прогрес. Однак, ми настійно не рекомендуємо другий варіант, оскільки він негативно впливає на ваше здоров'я. У Інтернет-бізнесі набагато легше знаходити партнерів та укладати угоди, оскільки фізична присутність не є обов'язковою. Багато людей, які відчують, що не можуть реалізувати свій потенціал на найманій роботі, задумуються про розпочаток підприємницької діяльності. Мрія стати бізнес-коучем, психологом або реалізувати ідеї створення веб-сайтів стає реальністю завдяки появі Інтернету в нашому житті. Тут кожен може знайти свою нішу та свою аудиторію, а кожен споживач є власним контентом.

Тому інтернет-магазин стає невід'ємною частиною будь-якого підприємства в сучасному світі.

1.3. Підстава для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується з наказом ректора.

Таким чином підставами для розробки (виконанням кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка»;
- завдання на кваліфікаційну роботу на тему «Розробка інтернет-магазину з продажу ювелірних виробів та аксесуарів з використанням JavaScript».

1.4. Постановка завдання

Завданням даної роботи є створення сайту з продажу ювелірних виробів та аксесуарів за допомогою мови JavaScript. Основними характеристиками розробки повинні бути:

- Зручна навігація по сайту
- Створення розмітки та дизайну.

Поставлена задача може бути досягнута при виконанні наступних вимог:

- вивчення предметної області завдання;
- проведення порівняльної характеристики можливостей аналогічних продуктів;
- вибір платформи розробки;
- написання програмного коду.

Кінцевим результатом має бути WEB-сторінка зі зручною навігацією, можливістю вибрати, отримати інформацію та купити товар.

1.5. Вимоги до програми або програмного виробу

В результаті роботи ми повинні отримати інтернет-магазин з продажу ювелірних виробів та аксесуарів, написаний за допомогою мови JavaScript, на основі NodeJS і з використанням фреймворку ReactJS.

1.5.1. Вимоги до функціональних характеристик

Кінцевий результат має відповідати наступним функціональним критеріям:

- інтуїтивно зрозумілий інтерфейс користувача;
- можливість оперувати товарами в корзині;
- система фільтрації товарів.

1.5.2. Вимоги до інформаційної безпеки

Для забезпечення безпеки сайту необхідно зробити такі кроки:

- Використання протоколу HTTPS;
- Вхід адміністратора тільки через SSH;
- Захист від DDOS – атак;
- Захист від XSS - атак;
- Захист від SQL - ін'єкцій.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для коректної роботи сайту вам потрібні наступні параметри:

- WEB-браузер;
- IE > 11;
- ОЗУ > 1 ГБ

У React 18, використовуються типи колекцій Map та Set. Однак, в деяких старих пристроях та браузерах (таких як IE < 1), ці типи колекцій відсутні. Також існують інші браузери, де вони мають реалізацію, яка може відрізнитися (наприклад, IE 11).

1.5.4. Вимоги до інформаційної та програмної сумісності

Для використання системи не потрібно встановлювати додаткове програмне забезпечення на комп'ютери користувачів. Вони просто можуть переглядати сторінки сайту у своєму веб-браузері.

Для розробки було використано основні мови програмування які були вивчення під час навчання (HTML, CSS, JavaScript).

Додаток було написано в VSCode.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення програми

Результатом даної кваліфікаційної роботи має бути інтернет-магазин з продажу ювелірних виробів та аксесуарів.

Основний функціонал додатка полягає в можливості клієнта знайти, вибрати, купити товар, використовуючи систему фільтрації та пошук.

Для кращої і якісної роботи присутня налаштована корзина, аби користувач міг оперувати вибраним товаром.

2.2. Опис застосованих математичних методів

При створенні інтернет-магазину не було використано складних математичних методів.

2.3. Опис використаних технологій та мов програмування

Дворівнева архітектура веб-сайту охоплює розміщення інформації у двох різних місцях: на рівні представлення (інтерфейсу) на стороні клієнта та на рівні даних на стороні сервера. Цей поділ на компоненти, розташовані окремо, є основою дворівневої архітектури. Зазвичай, клієнтська програма запускається на комп'ютері користувача з метою збору даних та передачі їх на сервер бази даних. Це сприяє взаємодії та координації між двома рівнями.

Переваги дворівневої архітектури:

- Зменшення навантаження на сервер та клієнтські машини.
- Зниження мережевого трафіку та підвищення ефективності обробки інформації завдяки оптимізації та буферизації введення-виводу.
- Забезпечення захисту даних за допомогою систем управління базами даних (СУБД), що дозволяє блокувати небажані дії користувачів.

- Сервер реалізує контроль транзакцій та може блокувати спроби одночасної зміни ідентичних записів.

Недоліки дворівневої архітектури:

- Потреба у повторному використанні програмного коду та запитів може призвести до збільшення ресурсів, оскільки бізнес-логіка функціональної обробки та подання даних може бути однаковою для кількох клієнтських програм.
- Проблеми з контролем версій програмного забезпечення та розповсюдженням нових версій виникають через те, що клієнт бачить більшу частину логіки програми.
- Дворівнева модель не масштабується на велику кількість

Для даної кваліфікаційної роботи було розроблено front-end частину сайту. З мов програмування основною використовувалась JavaScript з використанням Node.js та React.js.

2.4. Опис структури системи та алгоритмів її функціонування

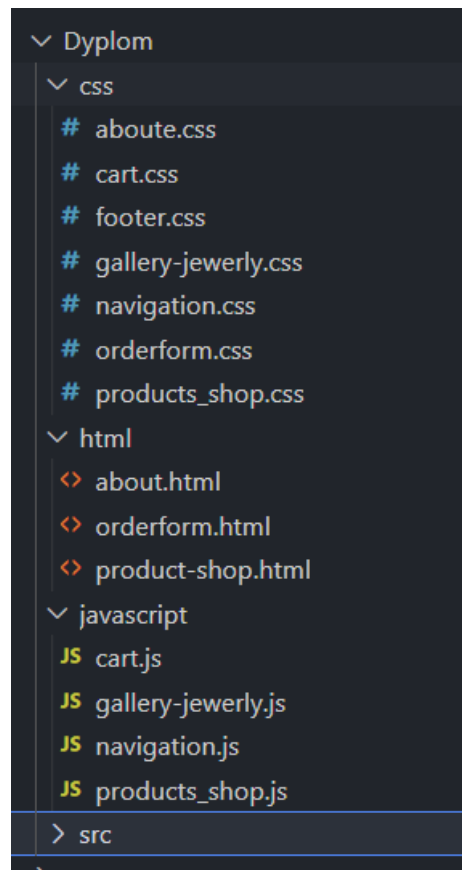


Рис.2.1. Структура програми у додатку VSCode

Кожен користувач інтернету шукає той самий сайт який буде легким для розуміння та приємним на вигляд. В мережі існує сотні інтернет-магазинів, і жоден клієнт не буде терпіти незручності, щоб скористатись саме вашим магазином. Скоріше клієнт закриє вкладку та повернеться до пошуків «того самого» інтернет магазину.

Щоб такого не сталося потрібно правильно розробити структуру свого сайту. Структура це «скелет» порталу, він відіграє важливу роль зручності використання. Ви точно зустрічали ті самі сайти які не могли спокійно використовувати. Як приклад можна уявити звичайний текст. Текст завжди складається з абзаців які розміщені по змісту, такий текст читати легше та приємніше. В той же час якщо вам доведеться читати текст без цього розподілу за змістом ви не зможете досить довго утримувати свою увагу.

Структура сайту – це логічний розподіл порталу на блоки за змістом. Початковий скелет кожного сайту складається з heder-y (секція в якій зазвичай знаходиться логотип та головна навігація по сайту), секцій де зберігається головний контент та footer-y (секція яка зазвичай має невелику кількість інформації, такі як номер телефону для зв'язку чи електрона пошта).

Ми можемо подивитись структуру мого інтернет-магазину. Ми бачемо що тут присутні усі, три зазначені вище, компоненти.Також там присутнє сортування товару за категорією(рис.2.2.)

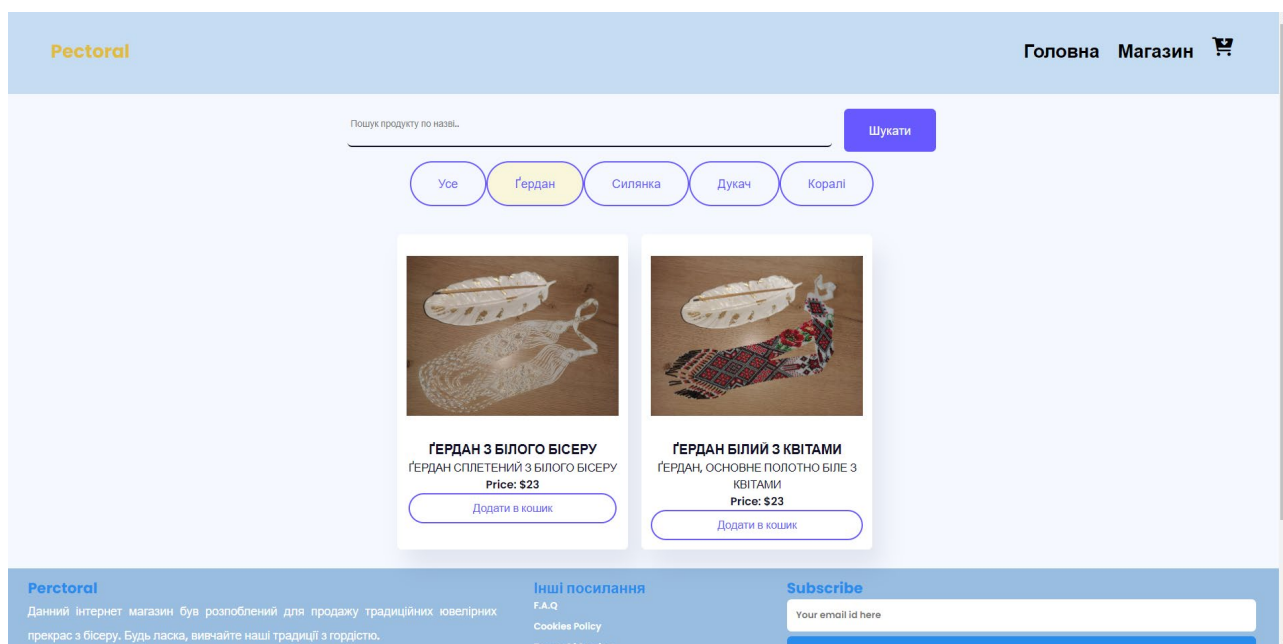


Рис.2.2. Сортування товару за категорією

Також було реалізовано пошук товару по назві(рис.2.3.)

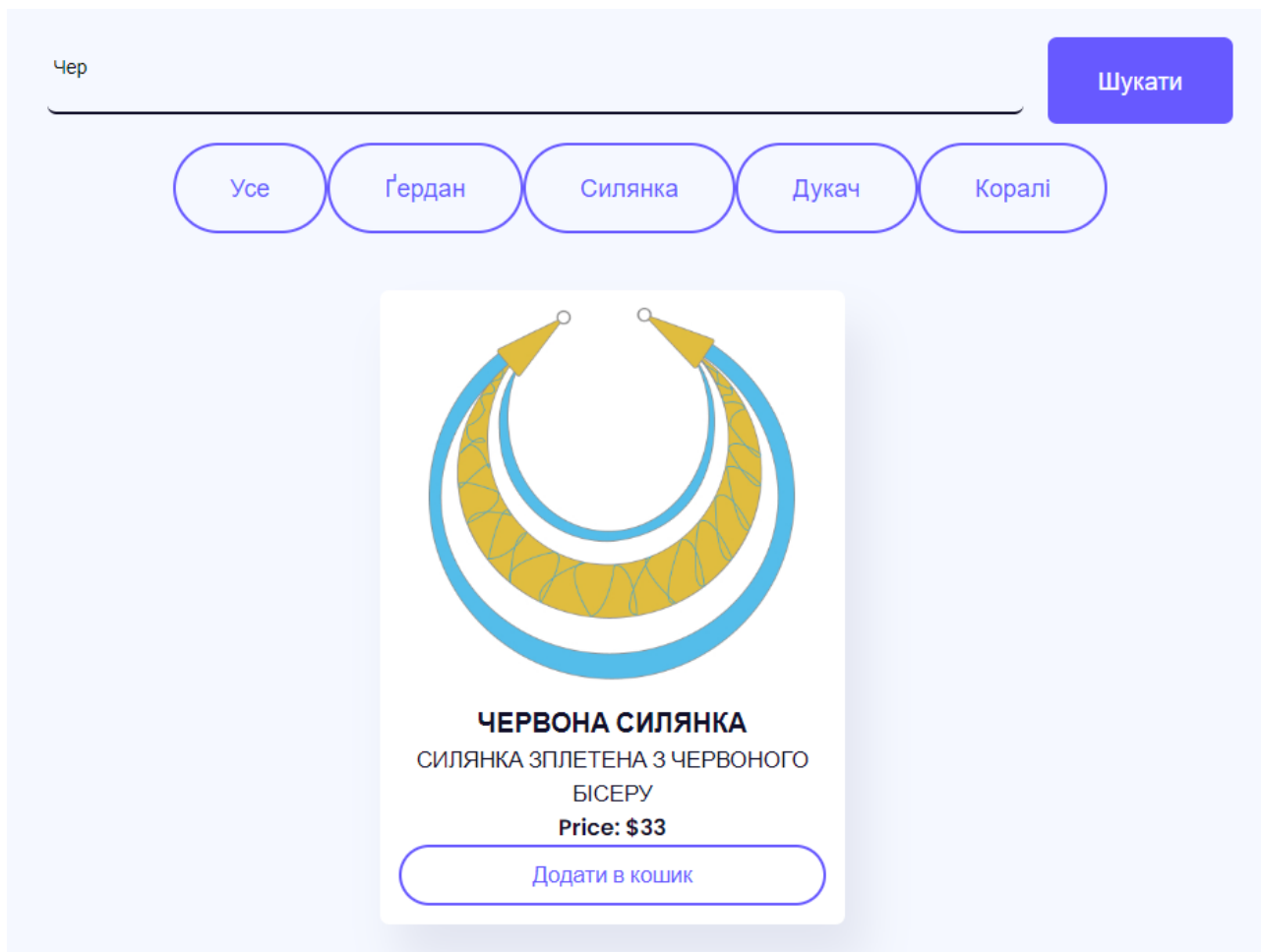


Рис.2.3. Пошук товару за назвою.

Для реалізації даних можливостей використовувалась мова програмування JavaScript.

Для початку була створена невелика база даних з інформацією для створення карточки товару. До цих даних відносяться назва(за допомогою якої буде потім реалізована можливість пошуку по назві), опис товару, зображення товару, ціна товару.

Після створення бази даних використовується цикл for() для виведення усіх даних у вигляді карточок товару. Код має такий вигляд:

```
for (let i of products.data) {  
  //Create Card  
  let card = document.createElement("div");  
  //Card should have category and should stay hidden initially  
  card.setAttribute('data-id',i.id)  
  card.classList.add("card", i.category, "hide");
```

```

//img tag
let image = document.createElement("img");
image.classList.add("img")
image.setAttribute("src", i.img);
card.appendChild(image);
//container
let container = document.createElement("p");
container.classList.add("container");
//product name
let name = document.createElement("h3");
name.classList.add("product-name");
name.innerText = i.productName.toUpperCase();
container.appendChild(name);
let detile = document.createElement("span");
detile.classList.add("product-detile");
detile.innerText = i.productDetile.toUpperCase();
container.appendChild(detile);
//price
let price = document.createElement("span");
price.classList.add("price")
price.innerText = "Price: $" + i.price;
container.appendChild(price);
let add_btn = document.createElement("span");
add_btn.classList.add("button_add");
add_btn.innerText = "Додати в кошик";
container.appendChild(add_btn);
card.appendChild(container);
document.getElementById("products").appendChild(card);
}

```

Після цього була створена функція для можливості провести фільтрацію товару по критеріям. Іншими словами коли ви обираєте критерій, то товари які не відносяться до цієї категорії будуть скриті для очей клієнта. Код програми:

```

//parameter passed from button (Parameter same as category)
function filterProduct(value) {
  //Button class code
  let buttons = document.querySelectorAll(".button-value");
  buttons.forEach((button) => {
    //check if value equals innerText
    if (value.toUpperCase() === button.innerText.toUpperCase()) {
      button.classList.add("active");
    } else {
      button.classList.remove("active");
    }
  });

  //select all cards
  let elements = document.querySelectorAll(".card");
  //loop through all cards
  elements.forEach((element) => {
    //display all cards on 'all' button click
    if (value === "all") {
      element.classList.remove("hide");
    } else {
      //Check if element contains category class
      if (element.classList.contains(value)) {
        //display element based on category
        element.classList.remove("hide");
      } else {
        //hide other elements
        element.classList.add("hide");
      }
    }
  });
}

```

Таким чином було створено можливість сортувати товар по категоріям використовуючи кнопки.

Після цього було реалізовано можливість шукати товар про його назві. Для цієї можливості використовувалась не функція а метод `document.getElementById()`. Даний код виглядає так:

```
//Search button click
document.getElementById("search").addEventListener("click", () => {
  //initializations
  let searchInput = document.getElementById("search-input").value;
  let elements = document.querySelectorAll(".product-name");
  let cards = document.querySelectorAll(".card");

  //loop through all elements
  elements.forEach((element, index) => {
    //check if text includes the search value
    if (element.innerText.includes(searchInput.toUpperCase())) {
      //display matching card
      cards[index].classList.remove("hide");
    } else {
      //hide others
      cards[index].classList.add("hide");
    }
  });
});
```

Окрім цього створений алгоритм додавання товару до кошику. Для цього був створений інший JavaScript файл. Була створена функція яка додає ваш товар у кошик. Код програми:

```
const cartIcon = document.querySelector('.fa-cart-arrow-down')
const wholeCartWindow = document.querySelector('.whole-cart-window')
wholeCartWindow.inWindow = 0
const addToCartBtns = document.querySelectorAll('.button_add')
addToCartBtns.forEach( (btn)=>{
  btn.addEventListener('click', addItemFunction)
} )
```

```

function addItemFunction(e){
  const id = e.target.parentNode.parentNode.getAttribute("data-id")
  const img = e.target.parentNode.parentNode.querySelector("img").src
  const name = e.target.parentElement.children[0].textContent
  const desc = e.target.parentElement.children[1].textContent
  let price = e.target.parentElement.children[2].textContent
  price = price.replace("Price: $", "")
  const item = new CartItem(name, desc, img, price)
  LocalCart.addItemToLocalCart(id, item)
  console.log(price)
}

```

Після додавання товару у кошик спрацьовує функція яка відповідає за оновлення вмісту кошика. Данна функція виглядає так:

```

function updateCartUI(){
  const cartWrapper = document.querySelector('.cart-wrapper')
  cartWrapper.innerHTML=""
  const items = LocalCart.getLocalCartItems()
  if(items === null) return
  let count = 0
  let total = 0
  for(const [key, value] of items.entries()){
    const cartItem = document.createElement('div')
    cartItem.classList.add('cart-item')
    let price = value.price*value.quantity
    price = Math.round(price*100)/100
    count+=1
    total += price
    total = Math.round(total*100)/100
    cartItem.innerHTML =
    `
    
      <div class="details">
        <h3>\${value.name}</h3>
        <p>\${value.desc}

```

```

        <span class="quantity">Кількість: ${value.quantity}</span>
        <span class="price">Вартість: $ ${price}</span>
    </p>
</div>
<div class="cancel"><i class="fas fa-window-close"></i></div>
,
cartItem.lastElementChild.addEventListener('click', ()=>{
    LocalCart.removeItemFromCart(key)
})
cartWrapper.append(cartItem)
}

if(count > 0){
    cartIcon.classList.add('non-empty')
    let root = document.querySelector(':root')
    root.style.setProperty('--after-content', ` "${count}"`)
    const subtotal = document.querySelector('.subtotal')

    subtotal.innerHTML = `Кінцева сума: $$${total}`
}
else
    cartIcon.classList.remove('non-empty')
}

```

В той же час створюється клас для маніпуляцій контентом кошика, адже користувач може передумати купляти саме цей продукт. Це виглядає так:

```

class LocalCart{
    static key = "cartItems"
    static getLocalCartItems(){
        let cartMap = new Map()
        const cart = localStorage.getItem(LocalCart.key)
        if(cart===null || cart.length===0) return cartMap
        return new Map(Object.entries(JSON.parse(cart)))
    }
}

```



```

}

static addItemToLocalCart(id, item){
  let cart = LocalCart.getLocalCartItems()
  if(cart.has(id)){
    let mapItem = cart.get(id)
    mapItem.quantity +=1
    cart.set(id, mapItem)
  }
  else
    cart.set(id, item)
  localStorage.setItem(LocalCart.key, JSON.stringify(Object.fromEntries(cart)))
  updateCartUI()
}

static removeItemFromCart(id){
  let cart = LocalCart.getLocalCartItems()
  if(cart.has(id)){
    let mapItem = cart.get(id)
    if(mapItem.quantity>1)
    {
      mapItem.quantity -=1
      cart.set(id, mapItem)
    }
    else
      cart.delete(id)
  }
  if (cart.length===0)
    localStorage.clear()
  else
    localStorage.setItem(LocalCart.key,
JSON.stringify(Object.fromEntries(cart)))

```

```
updateCartUI()
}
}
```

Усе це головні компоненти інтернет-магазину які виконують усі потрібні функції для покупця.

Також для зацікавлення створена сторінка «Про нас» основний контент котрої має мету зацікавити користувача товаром магазину. Головною частиною є галерея з старими фотографіями які демонструють подібний товар(рис.2.10)

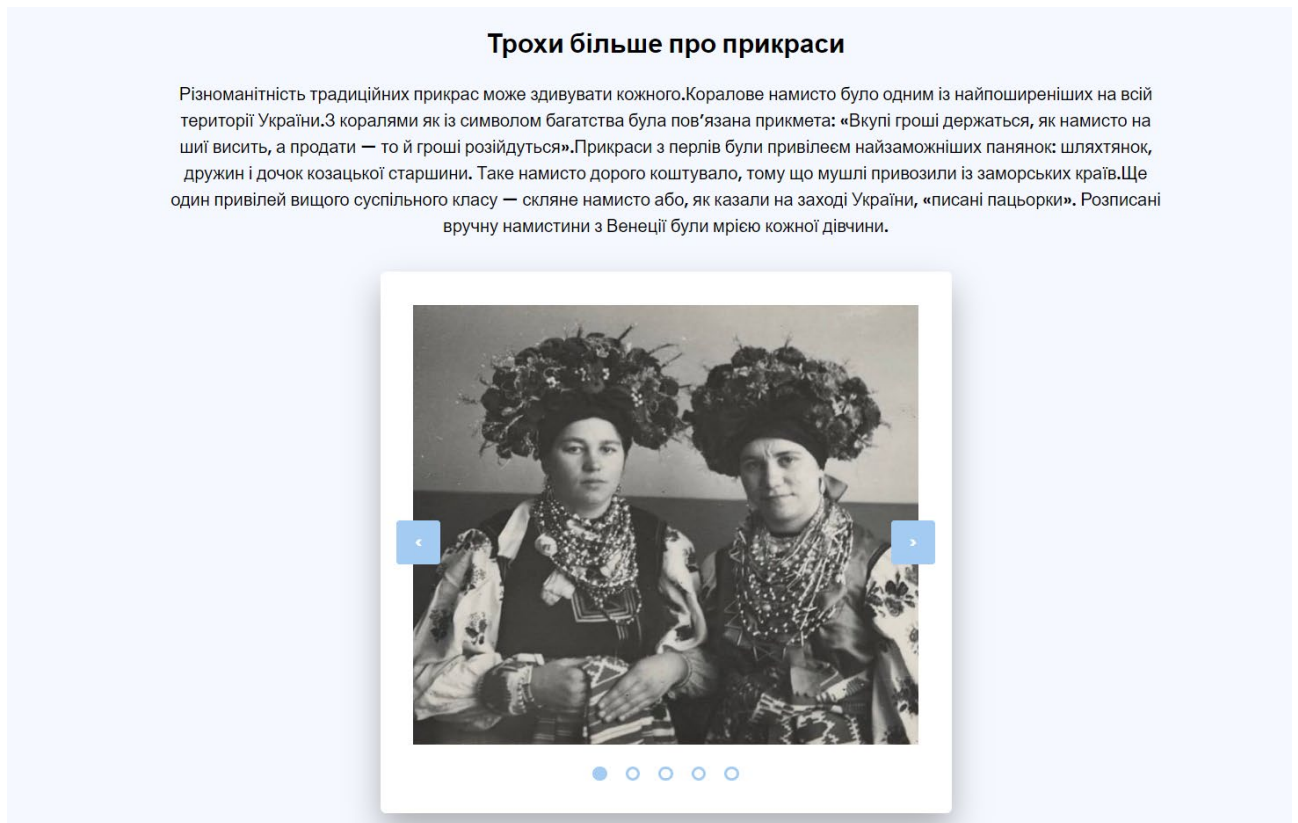


Рис.2.4. Галерея с фотографіями

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Інтерфейс сайту простий і не потребує для користування додаткового обладнання.

Механізм введення представляє собою матеріальний пристрій для прямої взаємодії користувача з сайтом. Тобто, цими механізмами є стандартні периферійні пристрої: клавіатура, миша, сенсорний дисплей телефону тощо.

Механізм виведення – візуальний. Користувач одразу після виконання якоїсь дії (нажаття на кнопку, вибір фільтрів тощо) отримує інформацію. Головні функції виводу даних – відображення інформації про товари, їх сортування, оперування їми в корзині, можливість придбати.

2.6. Опис розробленого програмного продукту

Сайт має сторінку з товарами, сторінку «Головна» а також форму замовлення. Можливість сортувати товари за критеріями та пошук по назві. Також є можливість оперування товарів у корзині.

Для самого сайту треба створити сервіси, що відносяться до back-end, після чого усе це потрібно розмістити, тобто потрібно побудувати інфраструктуру.

2.6.1. Використані технічні засоби

При розробці програми була використана персональна ЕОМ з наступними характеристиками:

- Процесор Intel Core i5-11400H;
- Відеопроцесор Nvidia GeForce GTX 1650;
- Оперативна пам'ять 32 ГБ DDR4.

Тестування проводилося на таких пристроях:

- ПК на Windows 10;

2.6.2. Використані програмні засоби

Додаток було написано в VSCode - інтегрованому середовищі розробки для JavaScript, HTML та CSS від компанії Microsoft.

2.6.3. Виклик та завантаження програми

Користувачу необхідно відкрити WEB-сторінку інтернет-магазину у браузері.

Інтерфейс є стандартним та зрозумілим для використання. Після відкриття користувач попадає на сторінку з товаром. Тут ми бачимо логотип який є назвою «Pectoral», посилання на сторінку «Головна», посилання на магазин , іконка кошика.

Як головний контент сторінки ми побачимо поле для пошуку товару по назві, кнопки для сортування товару за категоріями, а також карточки з товарами(рис.2.11.)

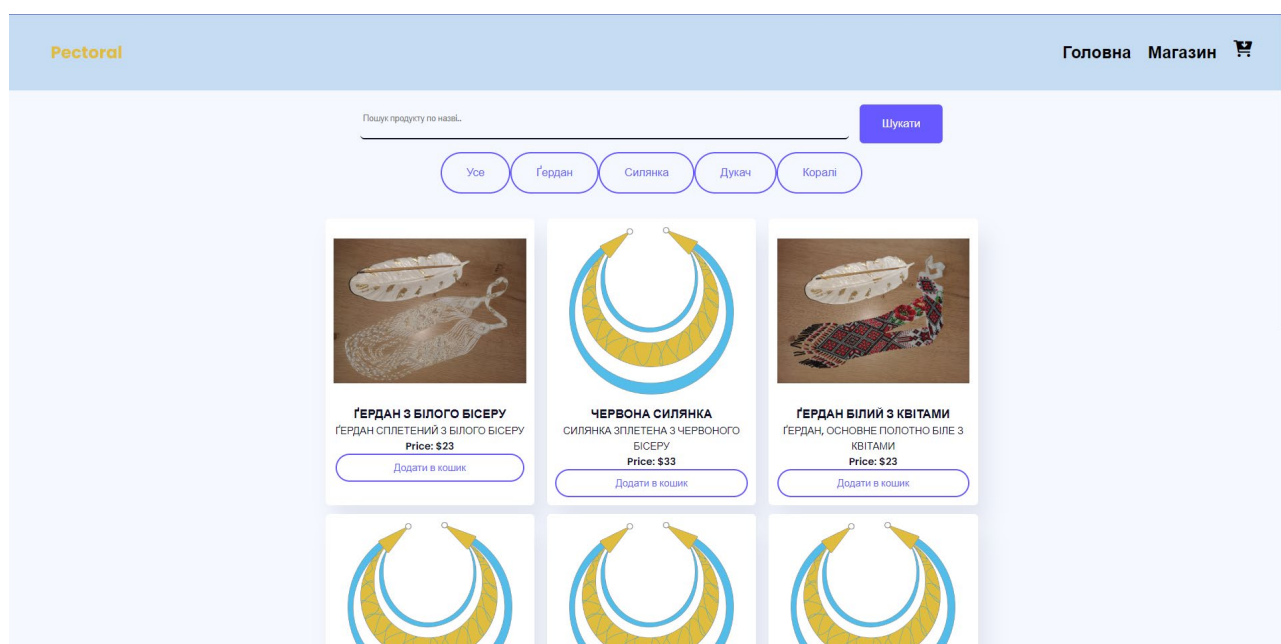


Рис.2.5. Головна сторінка магазину

Якщо нажати на посилання «Головна» відкриється сторінка з невеликою кількістю інформації а також галерея(рис.2.12)

Pectoral Головна Магазин

Пектораль

В минулому нашійні прикраси жінки були ознакою заможності сім'ї. Було багато різних прикрас та вони були не дешевими. Вважалося що якщо щось станеться з домо то продавши нашійні прикраси жінки можна було купити нову хату. Ми віримо щов сучасному світі все ще є місце для наших традиційних прикрас.

[До магазину](#)

Рис.2.6. Головна сторінка(А)

Трохи більше про прикраси

Різноманітність традиційних прикрас може здивувати кожного. Коралове намисто було одним із найпоширеніших на всій території України. З коралями як із символом багатства була пов'язана прикмета: «Вкупі гроші держаться, як намисто на шії висить, а продати — то й гроші розійдуться». Прикраси з перлів були привілеєм найзаможніших панянок: шляхтянок, дружин і дочок козацької старшини. Таке намисто дорого коштувало, тому що мушлі привозили із заморських країв. Ще один привілей вищого суспільного класу — скляне намисто або, як казали на заході України, «писані пацьорки». Розписані вручну намистини з Венеції були мрією кожної дівчини.

[<](#) [>](#)

● ○ ○ ○ ○

Рис.2.6. Головна сторінка(Б)

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Визначення трудомісткості розробки програмного забезпечення

Вихідні дані:

1. передбачуване число операторів програми – 120;
2. коефіцієнт складності програми – 1,3;
3. коефіцієнт корекції програми в ході її розробки – 0,1;
4. годинна заробітна плата full stuck developer – 190 грн/год (за версією сайту WorkUA) - <https://www.work.ua/salary-Front-end+developer/>;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 0,8;
7. вартість машино-години ЕОМ – 12 грн/год.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{омл} + t_{\partial} \text{ , людино-годин, (3.1)}$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n -витрати праці на програмування по готовій блок-схемі;

$t_{омл}$ -витрати праці на налагодження програми на ЕОМ;

t_{∂} - витрати праці на підготовку документації (приймається 40).

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється. Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де q - передбачуване число операторів (120);

C - коефіцієнт складності програми (1,3);

p - коефіцієнт корекції програми в ході її розробки (0,01).

$$Q = 120 \cdot 1.3 \cdot (1 + 0.01) = 157,56$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot K}, \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі (1,2);

K – коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності (0,8);

$$t_u = \frac{157,56 \cdot 1,2}{85 \cdot 0,8} \approx 2,7805, \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20..25) \cdot K}, \quad (3.4)$$

$$t_a = \frac{157,56}{20 \cdot 0,8} = 9,8475, \text{ людино-годин}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20..25) \cdot K}, \quad (3.5)$$

$$t_n = \frac{157,56}{25 \cdot 0,8} = 7,878, \text{ людино-годин}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4..5) \cdot K}, \quad (3.6)$$

$$t_{\text{отл}} = \frac{157,56}{5 \cdot 0,8} = 39,39, \text{ людино-годин,}$$

– за умови комплексного налагодження завдання:

$$t_{\text{отл}}^k = 1,2 \cdot t_{\text{отл}}, \quad (3.7)$$

$$t_{\text{отл}}^k = 1,2 \cdot 39,39 = 47,268, \text{ людино-годин}$$

Витрати праці на підготовку документації:

$$t_{\text{д}} = t_{\text{дп}} + t_{\text{до}}, \quad (3.8)$$

де t_{op} – трудомісткість підготовки матеріалів і рукопису;

$$t_{op} = \frac{Q}{(15..20) \cdot K}, \quad (3.9)$$

$$t_{op} = \frac{157,56}{20 \cdot 0,8} = 9,8475, \text{ людино-годин}$$

де $t_{до}$ – трудомісткість редагування, печатки й оформлення документації;

$$t_{до} = 0,75 \cdot t_{op}, \quad (3.10)$$

$$t_{до} = 0,75 \cdot 9,8475 = 7,386, \text{ людино-годин}$$

$$t_{\partial} = 9,8475 + 7,385625 = 17,233, \text{ людино-годин}$$

Отримаємо трудомісткість розробки програмного забезпечення:

$$t = 40 + 2,7805 + 9,8475 + 7,878 + 39,39 + 17,233 \approx 117,129, \text{ людино-годин,}$$

У результаті ми розрахували, що в загальній складності необхідно 117,129 людино-годин для розробки даного програмного забезпечення.

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ K_{no} включають витрати на заробітну плату виконавця програми $Z_{зп}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{no} = Z_{зп} + Z_{МВ}, \quad (3.11)$$

$Z_{зп}$ – заробітна плата виконавців, яка визначається за формулою:

$$Z_{зп} = t \cdot C_{ПР}, \quad (3.12)$$

де t – загальна трудомісткість, людино-годин;

$C_{ПР}$ – середня годинна заробітна плата програміста, грн/година.

З урахуванням того, що середня годинна зарплата front-end developer-а становить 190 грн/год, то отримаємо:

$$Z_{зп} = 117,129 \cdot 190 = 22255,51, \text{ грн.}$$

Вартість машинного часу $Z_{МВ}$, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{МЧ}, \text{ грн.} \quad (3.13)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год;

$C_{МЧ}$ – вартість машино-години ЕОМ, грн/год.

$$З_{MB} = 39,39 \cdot 12 = 472,68 \text{ ,грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{no} = 22255,51 + 472,68 = 22728,19 \text{ , грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс} \quad (3.14)$$

де B_k – число виконавців;

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні

$F_p = 176$ годин).

Витрати на створення програмного продукту:

$$T = \frac{117,129}{1 \cdot 176} = 0,6655 \text{ , міс.}$$

Висновки. Інтернет-магазин має вартість 22728,19 грн. Ймовірний очікуваний час розробки – 0,6655 місяці при стандартному 40-годинному робочому тижні і 178-годинному робочому місяці. Цей термін пов'язаний з кількістю операторів і включає в себе час для дослідження та розробку алгоритму розв'язання задачі, розробку дизайну і створення документації. На розробку інтернет-магазину буде витрачено 120 людино-годин.

ВИСНОВКИ

Мета бакалаврської кваліфікаційної роботи полягає в розробці веб-магазину, що продає ювелірні вироби та аксесуари, з використанням мови JavaScript. В ході написання цієї роботи виявлено, що інтернет-магазин є невід'ємною складовою будь-якої компанії і його створення є необхідним кроком для розвитку бізнесу.

Сайт має додатковий функціонал для спрощення користуванням, а саме:

- пошук товарів по назві;
- фільтрація товарів;

Актуальність поставленої задачі обумовлюється великим попитом на такі програмні продукти. Створення інтернет-магазину сприятиме зміцненню позицій вашої компанії на традиційних ринках і відкриттю нових можливостей. За допомогою інтернет-магазину ви зможете збільшити обсяги продажів і, відповідно, збільшити оборот та отримати більший прибуток.

Особливість створення front-end частини інтернет-магазину, якщо ви підприємець/початківець, ви можете з мінімальними витратами організувати свій власний бізнес, а точніше як саме ви бажаєте його бачити, а інтернет-магазин допоможе збільшити продажі та відкрити нові ринки збуту товарів.

В «Економічному розділі» визначено трудомісткість розробки програмного забезпечення (117,129 чол-год), підраховані витрати на створення програмного забезпечення (22728,19 грн.) і гаданий період розробки (0,6655 міс.).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://w3schoolsua.github.io/html/index.html#gsc.tab=0>
2. <https://w3schoolsua.github.io/css/index.html#gsc.tab=0>
3. <https://w3schoolsua.github.io/js/index.html#gsc.tab=0>
4. Web-технології та Web-дизайн : застосування мови HTML для створення електронних ресурсів Бородкіна І.Л.
5. React.js.Швидкий старт – Стоян Стефанов (2017)
6. JavaScript and JQuery: Interactive Front-End Web Development – John Duckett (2013)
7. Learning JavaScript Design Patterns - Addie Osmani (2012)
8. <https://nodejs.dev/en/learn/>
9. https://www.w3schools.com/nodejs/nodejs_intro.asp
10. <https://github.com/nodejs/node>
11. https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm
12. Node.js Design Patterns. Third Edition By – Mario Casciaro , Luciano Mammino
13. Coding HTML CSS JavaScript Made Easy. Web, Apps and Desktop. Flame Tree, – Adam Ceute & Frederic Johnson (2016)
14. <https://codeguida.com/post/1447>
15. <https://kitapp.pro/uk/rozrobka-dodatkov-na-node-js/>
16. <https://uk.javascript.info/>
17. <https://bool.dev/blog/detail/scope-chain-v-javascript>
18. <https://www.freecodecamp.org/ukrainian/news/posibnyk-z-masyvu-ob-yektiv-javascript-yak-stvoryuvaty-onovlyuvaty-ta-perehlyadaty-ob-yekty-za-dopomohoyu-metodiv-masyvu-js/>
19. You Don't Know JS: Scope & Closures – Kyle Simpson (2014)
20. Data Structures and Algorithms with JavaScript: Bringing classic computing approaches to the Web - Michael McMillan(2014)

ЛІСТИНГ ПРОГРАМИ

(данний код був завантажений на GitHub:
<https://github.com/ValyaVV/Dyplom>)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Shop-page</title>
    <!-- Font Awesome Icons -->
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css"
    />
    <!-- Google Fonts -->
    <link
      href="https://fonts.googleapis.com/css2?family=Poppins:wght@600;700&display=swap"
      rel="stylesheet"
    />
    <!-- Stylesheet -->
    <link rel="stylesheet" href="../css/navigation.css" />
    <link rel="stylesheet" href="../css/products_shop.css" />
    <link rel="stylesheet" href="../css/cart.css" />
    <link rel="stylesheet" href="../css/footer.css" />
  </head>
  <body>
    <header>
      <nav>
        <a href="#home" id="logo">Pectoral</a>
        <i class="fas fa-bars" id="ham-menu"></i>
        <ul id="nav-bar">
          <li>
```

```

    <a href="about.html">Головна</a>
</li>
<li>
    <a href="#services">Магазин</a>
</li>
<li>
    <div class="cart-icon">
        <i class="fas fa-cart-arrow-down fa-2x"></i>
        <div class="whole-cart-window hide">
            <h2>Ваш кошик</h2>
            <div class="cart-wrapper">
                <div class="cart-item">
                    
                    <div class="details">
                        <h3>Щось</h3>
                        <p>щось
                            <span class="quantity">Кількість: 1</span>
                            <span class="price">Вартість: $49</span>
                        </p>
                    </div>
                    <div class="cancel"><i class="fas fa-window-close"></i></div>
                </div>
            </div>
            <div class="subtotal">Кінцева сума: $0.00</div>
            <div
                class="checkout"><a
                    class="order"
href="../html/orderform.html">Оплатити</a></div>
        </div>
    </div>
</li>
</ul>

</nav>
</header>
<div class="wrapper">
    <div id="search-container">

```

```

<input
  type="search"
  id="search-input"
  placeholder="Пошук продукту по назві.."
/>
<button id="search">Шукати</button>
</div>
<div id="buttons" class="buttons">
  <button class="button-value active" onclick="filterProduct('all')">Усе</button>
  <button class="button-value" onclick="filterProduct('Гердан')">
    Гердан
  </button>
  <button class="button-value" onclick="filterProduct('Силянка')">
    Силянка
  </button>
  <button class="button-value" onclick="filterProduct('Дукач')">
    Дукач
  </button>
  <button class="button-value" onclick="filterProduct('Коралі')">
    Коралі
  </button>
</div>
<section id="products">
</section>
</div>
<footer>
  <div class="row primary">
    <div class="column about">
      <h3 class="footer-h3">Perctoral</h3>
      <p class="footer-p">
        Даний інтернет магазин був розроблений для продажу традиційних ювелірних
        прекрас з бісеру. Будь ласка, вивчайте наші традиції з гордістю.
      </p>
    </div>
    <div class="column links">

```



```

<h3 class="footer-h3">Інші посилання</h3>
<ul class="footer-ul">
  <li class="footer-li">
    <a href="#faq">F.A.Q</a>
  </li>
  <li class="footer-li">
    <a href="#cookies-policy">Cookies Policy</a>
  </li>
  <li class="footer-li">
    <a href="#terms-of-services">Terms Of Service</a>
  </li>
  <li class="footer-li">
    <a href="#support">Support</a>
  </li>
  <li class="footer-li">
    <a href="#careers">Careers</a>
  </li>
</ul>
</div>
<div class="column subscribe">
  <h3 class="footer-h3">Subscribe</h3>
  <div>
    <input type="email" placeholder="Your email id here" />
    <button class="footer-btn">Subscribe</button>
  </div>
  <div class="social">
    <i class="fa-brands fa-facebook-square"></i>
    <i class="fa-brands fa-instagram-square"></i>
    <i class="fa-brands fa-twitter-square"></i>
  </div>
</div>
</div>
<div class="row secondary">
  <div>
    <p>

```

```

        <i class="fas fa-phone-alt"></i>
    </p>
    <p><a class="footer-tel" href="tel: +38096*****">+38096*****</a></p>
</div>
<div>
    <p><i class="fas fa-envelope"></i></p>
    <p><a class="mail" href="mailto:mail@domain.com">mail@domain.com</a></p>
</div>
<div>
    <p><i class="fas fa-map-marker-alt"></i></p>
    <p>1234 Pearl Street</p>
</div>
</div>
</footer>
<!-- Script -->
<script src="../javascript/navigation.js"></script>
<script src="../javascript/products_shop.js"></script>
<script src="../javascript/cart.js"></script>
</body>
</html>

```

```

let hamMenuIcon = document.getElementById("ham-menu");
let navBar = document.getElementById("nav-bar");
let navLinks = navBar.querySelectorAll("li");
hamMenuIcon.addEventListener("click", () => {
    navBar.classList.toggle("active");
    hamMenuIcon.classList.toggle("fa-times");
});
navLinks.forEach((navLinks) => {
    navLinks.addEventListener("click", () => {
        navBar.classList.remove("active");
        hamMenuIcon.classList.toggle("fa-times");
    });
});

```

```
function displaycart() {  
  let j = 0;  
  document.getElementById("count").innerHTML=cart.length;  
}
```

```
let products = {  
  data: [  
    {  
      id: '1',  
      productName: "Гердан з білого бісеру",  
      productDetile:"Гердан сплетений з білого бісеру",  
      category: "Гердан",  
      price: "23",  
      img: "../src/Гердан-білий.jpg",  
    },  
    {  
      id: '2',  
      productName: "Червона силянка",  
      productDetile:"Силянка зплетена з червоного бісеру",  
      category: "Силянка",  
      price: "33",  
      img: "../src/Пектораль.jpg",  
    },  
    {  
      id: '3',  
      productName: "Гердан білий з квітами",  
      productDetile:"Гердан, основне полотно біле з квітами",  
      category: "Гердан",  
      price: "23",  
      img: "../src/Гердан-білий-квіти.jpg",  
    },  
    {  
      id: '4',  
      productName: "Дукач",  
      productDetile:"Дукач на червоній стрічці",
```

```

    category: "Дукач",
    price: "14",
    img: "../src/Пектораль.jpg",
  },
  {
    id: '5',
    productName: "Чорна силянка",
    productDetile:"Силянка зплетена з чорного бісеру",
    category: "Силянка",
    price: "33",
    img: "../src/Пектораль.jpg",
  },
  {
    id: '6',
    productName: "Коралі",
    category: "Коралі",
    productDetile:"Червоні коралі",
    price: "24",
    img: "../src/Пектораль.jpg",
  },
  {
    id: '7',
    productName: "Дукач",
    productDetile:"Дукач на синій стрічці",
    category: "Дукач",
    price: "14",
    img: "../src/Пектораль.jpg",
  },
],
};

```

```

for (let i of products.data) {
  //Create Card
  let card = document.createElement("div");
  //Card should have category and should stay hidden initially

```

```

card.setAttribute('data-id',i.id)
card.classList.add("card", i.category, "hide");
//img tag
let image = document.createElement("img");
image.classList.add("img")
image.setAttribute("src", i.img);
card.appendChild(image);
//container
let container = document.createElement("p");
container.classList.add("container");
//product name
let name = document.createElement("h3");
name.classList.add("product-name");
name.innerText = i.productName.toUpperCase();
container.appendChild(name);
let detile = document.createElement("span");
detile.classList.add("product-detile");
detile.innerText = i.productDetile.toUpperCase();
container.appendChild(detile);
//price
let price = document.createElement("span");
price.classList.add("price")
price.innerText = "Price: $" + i.price;
container.appendChild(price);
let add_btn = document.createElement("span");
add_btn.classList.add("button_add");
add_btn.innerText = "Додати в кошик";
container.appendChild(add_btn);
card.appendChild(container);
document.getElementById("products").appendChild(card);
}

//parameter passed from button (Parameter same as category)
function filterProduct(value) {
//Button class code

```

```

let buttons = document.querySelectorAll(".button-value");
buttons.forEach((button) => {
  //check if value equals innerText
  if (value.toUpperCase() === button.innerText.toUpperCase()) {
    button.classList.add("active");
  } else {
    button.classList.remove("active");
  }
});

//select all cards
let elements = document.querySelectorAll(".card");
//loop through all cards
elements.forEach((element) => {
  //display all cards on 'all' button click
  if (value === "all") {
    element.classList.remove("hide");
  } else {
    //Check if element contains category class
    if (element.classList.contains(value)) {
      //display element based on category
      element.classList.remove("hide");
    } else {
      //hide other elements
      element.classList.add("hide");
    }
  }
});
}

//Search button click
document.getElementById("search").addEventListener("click", () => {
  //initializations
  let searchInput = document.getElementById("search-input").value;
  let elements = document.querySelectorAll(".product-name");

```

```

let cards = document.querySelectorAll(".card");

//loop through all elements
elements.forEach((element, index) => {
  //check if text includes the search value
  if (element.innerText.includes(searchInput.toUpperCase())) {
    //display matching card
    cards[index].classList.remove("hide");
  } else {
    //hide others
    cards[index].classList.add("hide");
  }
});
});

```

```

//Initially display all products
window.onload = () => {
  filterProduct("all");
};

```

```

class CartItem{
  constructor(name, desc, img, price){
    this.name = name
    this.desc = desc
    this.img=img
    this.price = price
    this.quantity = 1
  }
}

```

```

class LocalCart{
  static key = "cartItems"

  static getLocalCartItems(){

```

```

    let cartMap = new Map()
const cart = localStorage.getItem(LocalCart.key)
if(cart===null || cart.length===0) return cartMap
    return new Map(Object.entries(JSON.parse(cart)))
}

static addItemToLocalCart(id, item){
    let cart = LocalCart.getLocalCartItems()
    if(cart.has(id)){
        let mapItem = cart.get(id)
        mapItem.quantity +=1
        cart.set(id, mapItem)
    }
    else
        cart.set(id, item)
    localStorage.setItem(LocalCart.key, JSON.stringify(Object.fromEntries(cart)))
    updateCartUI()
}

static removeItemFromCart(id){
let cart = LocalCart.getLocalCartItems()
if(cart.has(id)){
    let mapItem = cart.get(id)
    if(mapItem.quantity>1)
    {
        mapItem.quantity -=1
        cart.set(id, mapItem)
    }
    else
        cart.delete(id)
}
if (cart.length===0)
localStorage.clear()
else

```



```

    localStorage.setItem(LocalCart.key, JSON.stringify(Object.fromEntries(cart)))
    updateCartUI()
  }
}

```

```

const cartIcon = document.querySelector('.fa-cart-arrow-down')
const wholeCartWindow = document.querySelector('.whole-cart-window')
wholeCartWindow.inWindow = 0
const addToCartBtns = document.querySelectorAll('.button_add')
addToCartBtns.forEach( (btn)=>{
  btn.addEventListener('click', addItemFunction)
} )

```

```

function addItemFunction(e){
  const id = e.target.parentNode.parentNode.getAttribute("data-id")
  const img = e.target.parentNode.parentNode.querySelector("img").src
  const name = e.target.parentElement.children[0].textContent
  const desc = e.target.parentElement.children[1].textContent
  let price = e.target.parentElement.children[2].textContent
  price = price.replace("Price: $", "")
  const item = new CartItem(name, desc, img, price)
  LocalCart.addItemToLocalCart(id, item)
  console.log(price)
}

```

```

cartIcon.addEventListener('mouseover', ()=>{
  if(wholeCartWindow.classList.contains('hide'))
  wholeCartWindow.classList.remove('hide')
})

```

```

cartIcon.addEventListener('mouseleave', ()=>{
  // if(wholeCartWindow.classList.contains('hide'))
  setTimeout( () =>{

```

```

    if(wholeCartWindow.inWindow===0){
        wholeCartWindow.classList.add('hide')
    }
},500 )

})

wholeCartWindow.addEventListener('mouseover', ()=>{
    wholeCartWindow.inWindow=1
})

wholeCartWindow.addEventListener('mouseleave', ()=>{
    wholeCartWindow.inWindow=0
    wholeCartWindow.classList.add('hide')
})

function updateCartUI(){
    const cartWrapper = document.querySelector('.cart-wrapper')
    cartWrapper.innerHTML=""
    const items = LocalCart.getLocalCartItems()
    if(items === null) return
    let count = 0
    let total = 0
    for(const [key, value] of items.entries()){
        const cartItem = document.createElement('div')
        cartItem.classList.add('cart-item')
        let price = value.price*value.quantity
        price = Math.round(price*100)/100
        count+=1
        total += price
        total = Math.round(total*100)/100
        cartItem.innerHTML =
        `
        

```

```

    <div class="details">
      <h3>${value.name}</h3>
      <p>${value.desc}
        <span class="quantity">Кількість: ${value.quantity}</span>
        <span class="price">Вартість: $ ${price}</span>
      </p>
    </div>
    <div class="cancel"><i class="fas fa-window-close"></i></div>
  ,
  cartItem.lastElementChild.addEventListener('click', ()=>{
    LocalCart.removeItemFromCart(key)
  })
  cartWrapper.append(cartItem)
}

if(count > 0){
  cartIcon.classList.add('non-empty')
  let root = document.querySelector(':root')
  root.style.setProperty('--after-content', ` "${count}"`)
  const subtotal = document.querySelector('.subtotal')
  subtotal.innerHTML = `Кінцева сума: $$ ${total}`
}
else
  cartIcon.classList.remove('non-empty')
}
document.addEventListener('DOMContentLoaded', ()=>{updateCartUI()})

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Home-about</title>
    <!-- Font Awesome Icons -->
    <link
      rel="stylesheet"

```

```

    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css"
  />
  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@600;700&display=swap"
    rel="stylesheet"
  />
  <!-- Stylesheet -->
  <link rel="stylesheet" href="../css/navigation.css" />
  <link rel="stylesheet" href="../css/aboute.css" />
  <link rel="stylesheet" href="../css/cart.css" />
  <link rel="stylesheet" href="../css/gallery-jewerly.css" />
  <link rel="stylesheet" href="../css/footer.css" />
</head>
<body>
  <header>
    <nav>
      <a href="#home" id="logo">Pectoral</a>
      <i class="fas fa-bars" id="ham-menu"></i>
      <ul id="nav-bar">
        <li>
          <a href="/html/about.html">Головна</a>
        </li>
        <li>
          <a href="/html/product-shop.html">Магазин</a>
        </li>
        <li>
          <div class="cart-icon">
            <i class="fas fa-cart-arrow-down fa-2x"></i>
            <div class="whole-cart-window hide">
              <h2>Ваш кошик</h2>
              <div class="cart-wrapper">
                <div class="cart-item">
                  

```

```

    <div class="details">
      <h3>Beige Short Skirt</h3>
      <p>Bottomwear
        <span class="quantity">Кількість: 1</span>
        <span class="price">Вартість: $49</span>
      </p>
    </div>
    <div class="cancel"><i class="fas fa-window-close"></i></div>
  </div>
</div>
<div class="subtotal">Кінцева сума: $0.00</div>
<div
      class="checkout"><a
      class="order"
href=" ../html/orderform.html">Оплатити</a></div>
  </div>
</div>
</li>
</ul>
</nav>
</header>
<section id="home">
  <section class="main-about">
    
    <div class="about-text">
      <H1>Пектораль</H1>
      <span>В минулому нашійні прикраси жінки були ознакою заможності сім'ї.
Було багато різних прикрас та вони були не дешевими. Вважалось що якщо щось станеться з
домо то продавши нашійні прикраси жінки можна було купити нову хату. Ми віримо щов
сучасному світі все ще є місце для наших традиційних прикрас. </span>
    </div>
  </section>
</div>
<div class="button-shop">
  <button
    class="button-value"><a
    href="/html/product-shop.html"
    class="to-
shop">До магазину</a></button>
</div>
</section>

```

```

<section class="jewerly">
  <div class="about-jewerly">
    <h2>Трохи більше про прикраси</h2>
    <p>
      Різноманітність традиційних прикрас може здивувати кожного. Коралове
      намисто було одним із найпоширеніших на всій території України. З коралами як із символом
      багатства була пов'язана прикмета: «Вкупі гроші держаться, як намисто на шії висить, а
      продати — то й гроші розійдуться». Прикраси з перлів були привілеєм найзаможніших
      панянок: шляхтянок, дружин і дочок козацької старшини. Таке намисто дорого коштувало,
      тому що мушлі привозили із заморських країв. Ще один привілей вищого суспільного класу
      — скляне намисто або, як казали на заході України, «писані пацьорки». Розписані вручну
      намистини з Венеції були мрією кожної дівчини.
    </p>
  </div>
  <div class="container">
    <div class="image-container">
      
      
      
      
      
    </div>
    <div class="dot-container">
      <button onclick = "dot(1)"></button>
      <button onclick = "dot(2)"></button>
      <button onclick = "dot(3)"></button>
      <button onclick = "dot(4)"></button>
      <button onclick = "dot(5)"></button>
    </div>
    <button id="prev" onclick="prev()"> &lt; </button>
    <button id="next" onclick="next()"> &gt; </button>
  </div>
</section>

```

```
</section>
```

```
<footer>
```

```
<div class="row primary">
```

```
<div class="column about">
```

```
<h3 class="footer-h3">Perctoral</h3>
```

```
<p class="footer-p">
```

Даний інтернет магазин був розроблений для продажу традиційних ювелірних прекрас з бісеру. Будь ласка, вивчайте наші традиції з гордістю.

```
</p>
```

```
</div>
```

```
<div class="column links">
```

```
<h3 class="footer-h3">Інші посилання</h3>
```

```
<ul class="footer-ul">
```

```
<li class="footer-li">
```

```
<a href="#faq">F.A.Q</a>
```

```
</li>
```

```
<li class="footer-li">
```

```
<a href="#cookies-policy">Cookies Policy</a>
```

```
</li>
```

```
<li class="footer-li">
```

```
<a href="#terms-of-services">Terms Of Service</a>
```

```
</li>
```

```
<li class="footer-li">
```

```
<a href="#support">Support</a>
```

```
</li>
```

```
<li class="footer-li">
```

```
<a href="#careers">Careers</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
<div class="column subscribe">
```

```
<h3 class="footer-h3">Subscribe</h3>
```

```
<div>
```

```
<input type="email" placeholder="Your email id here" />
```

```
<button class="footer-btn">Subscribe</button>
```

```
</div>
<div class="social">
  <i class="fa-brands fa-facebook-square"></i>
  <i class="fa-brands fa-instagram-square"></i>
  <i class="fa-brands fa-twitter-square"></i>
</div>
</div>
</div>
<div class="row secondary">
  <div>
    <p>
      <i class="fas fa-phone-alt"></i>
    </p>
    <p><a class="footer-tel" href="tel: +38096*****">+38096*****</a></p>
  </div>
  <div>
    <p><i class="fas fa-envelope"></i></p>
    <p><a class="mail" href="mailto:mail@domain.com">mail@domain.com</a></p>
  </div>
  <div>
    <p><i class="fas fa-map-marker-alt"></i></p>
    <p>1234 Pearl Street</p>
  </div>
</div>
</footer>
<!-- Script -->
<script src="../javascript/navigation.js"></script>
<script src="../javascript/cart.js"></script>
<script src="../javascript/gallery-jewerly.js"></script>
</body>
</html>
```

```
let i = 0; // current slide
let j = 5; // total slides
```



```

const dots = document.querySelectorAll(".dot-container button");
const images = document.querySelectorAll(".image-container img");

function next(){
    document.getElementById("content" + (i+1)).classList.remove("active");
    i = (j + i + 1) % j;
    document.getElementById("content" + (i+1)).classList.add("active");
    indicator(i+ 1);
}

function prev(){
    document.getElementById("content" + (i+1)).classList.remove("active");
    i = (j + i - 1) % j;
    document.getElementById("content" + (i+1)).classList.add("active");
    indicator(i+1);
}

function indicator(num){
    dots.forEach(function(dot){
        dot.style.backgroundColor = "transparent";
    });
    document.querySelector(".dot-container button:nth-child(" + num +
    ")").style.backgroundColor = "#8052ec";
}

function dot(index){
    images.forEach(function(image){
        image.classList.remove("active");
    });
    document.getElementById("content" + index).classList.add("active");
    i = index - 1;
    indicator(index);
}

<!DOCTYPE html>

```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Order form</title>
  <link rel="stylesheet" href="../css/orderform.css" />
</head>
<body>
  <div class="container">
    <div class="order-form" id="order-form">
      <h2>Заповніть форму замовлення</h2>
      <form class="form" >
        <input placeholder="Ім'я" name="name" type="text">
        <br>
        <input placeholder="Фамілія" name="secondname" type="text">
        <br>
        <input placeholder="Номер телефону" name="phone" type="tel">
        <br>
        <input placeholder="Email" name="email" type="email">
        <br>
        <input placeholder="Населений пункт" name="place" type="text">
        <br>
        <select class="select">
          <option>Нова пошта</option>
          <option>УкрПошта</option>
        </select>
      </form>
      <button class="order-btn" onclick="openPopup()">Відправити</button>
    </div>
    <div class="bg-popup" id="bg-popup">
      <div class="popup" id="popup">
        
        <h2>Дякуємо за замовлення!</h2>
        <p>З вами зв'яжуться для підтвердження замовлення!</p>
      </div>
    </div>
  </div>

```

```

        <button type="button" class="popup-btn" onclick="closePopup()"> <a
class="back" href="../html/product-shop.html">Ok</a></button>
    </div>
</div>

</div>
<script>
    let bgpopup = document.getElementById("bg-popup");
    let popup = document.getElementById("popup");
    let order = document.getElementById("order-form");

    function openPopup() {
        bgpopup.classList.add("open");
        popup.classList.add("open-popup");
        order.classList.add("after-popup")
    }
    function closePopup() {
        bgpopup.classList.remove("open");
        popup.classList.remove("open-popup");
        order.classList.remove("after-popup")
    }
</script>
</body>
</html>

```

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис файлу
Пояснювальні документи	
Кваліфікаційна_робота_Войченко.В.В.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна_робота_Войченко.В.В.pdf	Пояснювальна записка до кваліфікаційної роботи. Документ PDF.
Програма	
rectoral.rar	Архів. Містить коди програми
Презентація	
Презентація Войченко.ppt	Презентація кваліфікаційної роботи