

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-07

Дипломний проєкт

здобувачки освіти денної форми навчання

РП.07.22.000.ДП

***ХОМЕНКО ОКСАНИ
ВЛАДИСЛАВІВНИ***

м. Одеса
2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Інженерія програмного забезпечення»

Група: 4РП-07

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту на тему:

Розробка інтернет-магазину одягу для просування бренду

Проектний матеріал складається з пояснювальної записки на 86 сторінках та графічного (презентаційного) матеріалу на 11 аркушах (слайдах).

Дипломник _____ (Хоменко О. В.)

Керівник _____ (Жадан А. С.)

Консультанти:

з економічного розділу _____ (Іванченков В. С.)

з розділу охорони праці та техніки безпеки _____ (Чорновол Н. І.)

з нормоконтролю _____ (Петрашова В. І.)

старший консультант _____ (Кривченко Ю. В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю. В.)

Завідувач відділення _____ (Скорнякова О. В.)

Захист «25» 06 2024 р.

Протокол ДКК № 2

Оцінка ДКК 4 (добре) / 100б.

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І. В.

“ 15 ” 01 2024 року

ЗАВДАННЯ
на дипломний проєкт

Здобувачі освіти Хоменко Оксані Владиславівні

1. Тема проєкту Розробка інтернет-магазину одягу для просування бренду

Затверджена наказом по коледжу від “ 02 ” листопада 2023 р., наказ № 244-А2-ОД

2 Термін здачі закінченого проєкту 10.06.2024

3. Вихідні дані до проєкту _____

1. Застосувати фреймворк Bootstrap

2. Використовувати реляційну СУБД MySQL

3. Застосовувати Micro-framework PHP на базі архітектури MVC

4. Застосовувати принципи об'єктно-орієнтованого програмування

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз предметної області. 2. Технології та засоби розробки (проєктування).

3. Проєктування веб-дизайну. 4. Проєктування архітектури веб-застосунку.

5. Розробка веб-застосунку. 6. Тестування створеного веб-застосунку.

7. Економічний розрахунок. 8. Аспекти охорона праці та техніки безпеки

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Презентація Power Point – 11 слайдів

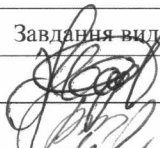
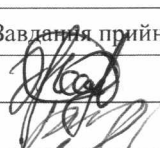
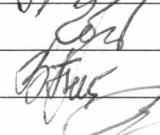
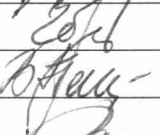
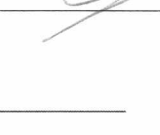

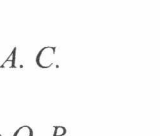

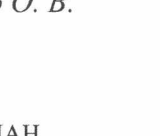
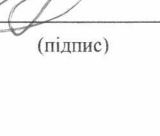
(актуальність, мета, практична цінність; Засоби розробки веб-застосунку;

Схема навігації веб-застосунку; Веб-дизайн веб-застосунку; Архітектура веб-застосунку

Процес розробки веб-застосунку; Схема бази даних веб-застосунку; Фрагмент коду;

Демонстрація інтернет-магазину)

6. Консультанти по проєкту, із зазначенням розділів проєкту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видано	Завдання прийняв
Основний розділ	Жадан А. С.		
Економічний розділ	Іванченков В. С.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

7. Дата видачі завдання

15.01.2024

Керівник

Жадан А. С.



(підпис)

Завдання прийняв до виконання

Хоменко О. В.



(підпис)

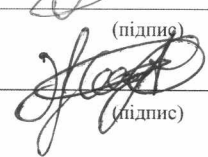
КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проєкту (роботи)	Термін виконання етапів дипломного проєкту (роботи)	Відмітка про виконання
1	Формування вступу	29.04.24	виконано
2	Аналіз предметної області	10.05.24	виконано
3	Підбір технічної літератури	19.05.24	виконано
4	Вибір технологій та засобів розробки (проектування)	20.05.24	виконано
5	Проектування веб-дизайну веб-застосунку	22.05.24	виконано
6	Проектування архітектури веб-застосунку	24.05.24	виконано
7	Реалізація веб-застосунку	27.05.24	виконано
8	Тестування створеного веб-застосунку	29.05.24	виконано
9	Оформлення пояснювальної записки	31.05.24	виконано
10	Оформлення графічної (презентаційної) частини	01.06.24	виконано
11	Економічний розрахунок	02.06.24	виконано
12	Опис охорони праці та техніки безпеки	09.06.24	виконано
13	Аналіз результатів проектування	13.06.24	виконано
14	Підготовка доповіді для захисту	16.06.24	виконано

Дипломник



Керівник



(підпис)

(підпис)

ЗМІСТ

ВСТУП	7
1 ОСНОВНИЙ РОЗДІЛ	9
1.1 Аналіз предметної області	9
1.1.1 Огляд існуючих рішень	9
1.1.2 Технології та засоби розробки	14
1.2 Проєктування застосунку	19
1.2.1 Проєктування дизайну застосунку	21
1.2.2 Проєктування архітектури застосунку	23
1.2.3 Проєктування бази даних	26
1.3 Реалізація застосунку	31
1.3.1 Створення моделей (Models)	31
1.3.2 Створення представлень (Views)	36
1.3.3 Створення контролерів (Controllers)	41
1.4 Тестування створеного веб-застосунку	44
1.4.1 Функціональний огляд головної сторінки	44
1.4.2 Функціональний огляд сторінки каталогу	46
1.4.3 Функціональний огляд сторінки корзини	49
1.4.4 Функціональний огляд сторінки контактів	49
1.4.5 Функціональний огляд сторінки відгуків	51
1.4.6 Функціональний огляд сторінки скарг та пропозицій	52
Рисунок 1.23. Сторінка Скарг та пропозицій застосунку	52
1.4.7 Функціональний огляд сторінки панелі Адміністратора	53
2 ЕКОНОМІЧНИЙ РОЗДІЛ	54
2.1 Резюме	54
2.2 Визначення трудомісткості розробки програмного забезпечення	54
2.3 Розрахунок ціни програмного продукту	58
3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	60
3.1 Вступ	60

3.2 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу	60
3.3 Гігієнічні вимоги до виробничого середовища	61
3.3.1 Вимоги до приміщення	61
3.3.2 Освітлення	62
3.3.3 Шум	62
3.3.4 Гігієнічні вимоги	63
3.3.5 Вентиляція та мікроклімат	63
3.3.6 Електробезпека	63
3.3.7 Робоче місце	64
3.4 Пожежна безпека	65
ВИСНОВКИ	67
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	68
ДОДАТОК А. Програмний код основної логіки веб-застосунку	69
ДОДАТОК Б. Слайди мультимедійної презентації	81

ВСТУП

Актуальність теми. Рано чи пізно, бізнес пов'язаний з торгівлею стикається з проблемою створення власного інтернет-магазину. Компанія хоче не лише залишатися на плаву серед своїх конкурентів, а й розвиватися та збільшувати свою популярність. Інтернет-магазин істотно зменшить витрати виробника, заощадивши на утриманні фізичного магазину, а також розширивши ринки збуту, так само як і можливість покупця – купувати будь-який товар у будь-який час. Веб-ресурс дозволяє компаніям подавати інформацію про товари чи послуги коротко і водночас повно. Розміщення актуальних новин та інформації на головній сторінці не будуть проблемою, а повідомлення про знижки, акції або зміни режиму роботи фізичних магазинів значно полегшить покупцям життя. Це надасть нашому інтернет-магазину перевагу перед звичайними.

Інтернет-магазини дозволяють розширити свою аудиторію далеко за межі фізичних магазинів, що є особливо важливим для брендів одягу. Вони не замінять традиційний спосіб придбання товару, а запропонують споживачу самому обирати комфортний для нього розвиток подій.

Мета і задачі розробки. Метою даної роботи є розробити інформаційну веб-систему для торговельного бізнесу. Зі сторони клієнта застосунок повинен містити функцію пошуку товару та його придбання. Зі сторони адміністратора – функцію додавання та прибирання товару, а також редагування вже існуючої інформації.

Методи розробки. При дослідженні основних проблем предметної області, аналізі аналогів та засобів розробки використано системний підхід. Для створення бази даних (БД) була використана реляційна СУБД MySQL. Розробка програмного продукту здійснювалася з використанням об'єктно-орієнтованого програмування (ООП) за допомогою мов PHP та JavaScript.

					<i>РП 07. 22 000. 00 ДП ПЗ</i>	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

Практичне значення одержаних результатів. Результатом розробки є інформаційна веб-система, яка може стати в нагоді власникам магазинів, надаючи їм можливість розширення їхнього бізнесу.

					<i>РП 07. 22 000. 00 ДП ПЗ</i>	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз предметної області

1.1.1 Огляд існуючих рішень

Розглянемо декілька інтернет-магазинів зі схожою тематикою.

Першим аналогом я обрала магазин – «ARGO».

На рис. 1.1 зображено головну веб-сторінку ARGO.



Рисунок 1.1. «ARGO»

Розглянемо переваги:

1. Стрічка переглянутих товарів: Наявність даної стрічки допомагає користувачу краще орієнтуватися у навігації сайту і не втрачати недавно переглянутий товар. Зі сторони сайту це допоможе відслідковувати вподобання користувача і надавати йому рекомендації.
2. Окрема вкладка з різноманітними акціями: Акція є чудовим шляхом для привернення уваги до товару та збільшує шанси продажу. На сайті акції зібрані в окремі категорії у зручному форматі.
3. Вибір кольору: До багатьох товарів, в наявності можна побачити велике різноманіття кольорів, що є дуже приємним бонусом, адже не багато

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 001. 00 ДП ПЗ

Арк.

9

магазинів можуть це запропонувати. Різноманіття кольорів в різних розмірах та варіанти вибору суттєво збільшують шанс продажу.

Розглянемо недоліки:

1. Оптимізація: Передивляючись цей інтернет-магазин я зіштовхнулася із проблемою довгого завантаження веб-сторінок. Для звичайного завантаження сторінки перегляду товару потрібно було очікувати принаймні 9 секунд, що не є дуже довгим терміном, але при перегляді великої кількості товару є досить помітною проблемою.
2. Застаріле оформлення сайту: Оформлення сайту є інтуїтивно зрозумілим, але на відміну від інших дизайн виглядає несучасно і просто. Білий фон швидко набридає очам, а маленькі і ніяк не виділені іконки, є складно помітними та ріжуть око.

Другим аналогом був обраний магазин – «Colin's» [1].

На рис. 1.2 зображено головну веб-сторінку «Colin's».

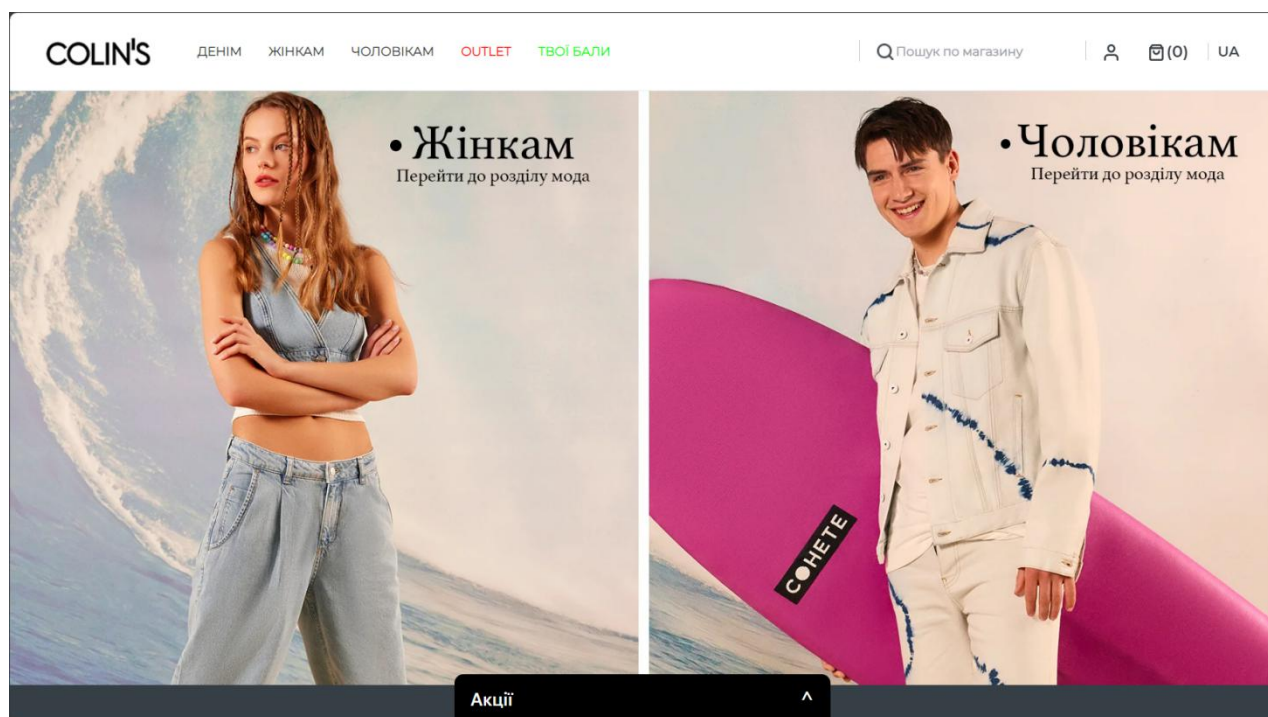


Рисунок 1.2. «Colin's»

Розглянемо переваги «Colin's»:

1. Велика кількість фотографій: Від кількості контенту на веб-сторінці часто залежить шанс продажу товару. Можливість розглянути товар з усіх боків допоможе користувачеві обрати потрібний йому варіант і не звертатися у підтримку за додатковими фото. На фотографіях можна буде розглянути як виглядає одяг на моделі і чи це саме те, що шукає користувач.
2. Збільшення фотографій: Приємним бонусом є можливість розглянути фото товару у збільшеному масштабі, при цьому фотографія не втрачає своєї якості і не розмивається. Це допомагає користувачеві розгледіти деталі одягу або малюнків на ньому.
3. Стрічка рекомендацій: У вікні товару, нижче, з'являється стрічка зі схожою тематикою або рекомендованими товарами. Якщо користувачеві сподобалося щось із каталогу, він зможе швидко знайти схожий продукт не витрачаючи на це багато часу.

Розглянемо недоліки:

1. Мала кількість розмірів: Знайти жіночий одяг розміру більший ніж 46 не вдасться, такого розміру неможливо знайти через фільтр, а серед чоловічих розмірів найбільшим виявився 56.
2. Малий опис товару: В описі товару, з корисного, значення буде мати лише колір, а про тип тканини взагалі ніде не написано і обрати його у фільтрі неможливо [2].

Розглянемо третій аналог – «LeBoutique».

На рис. 1.3 зображено головну веб-сторінку «LeBoutique».

					РП 07. 22 001. 00 ДП ПЗ	Арк.
						11
Ізм.	Лист	№ докум.	Підпис	Дата		

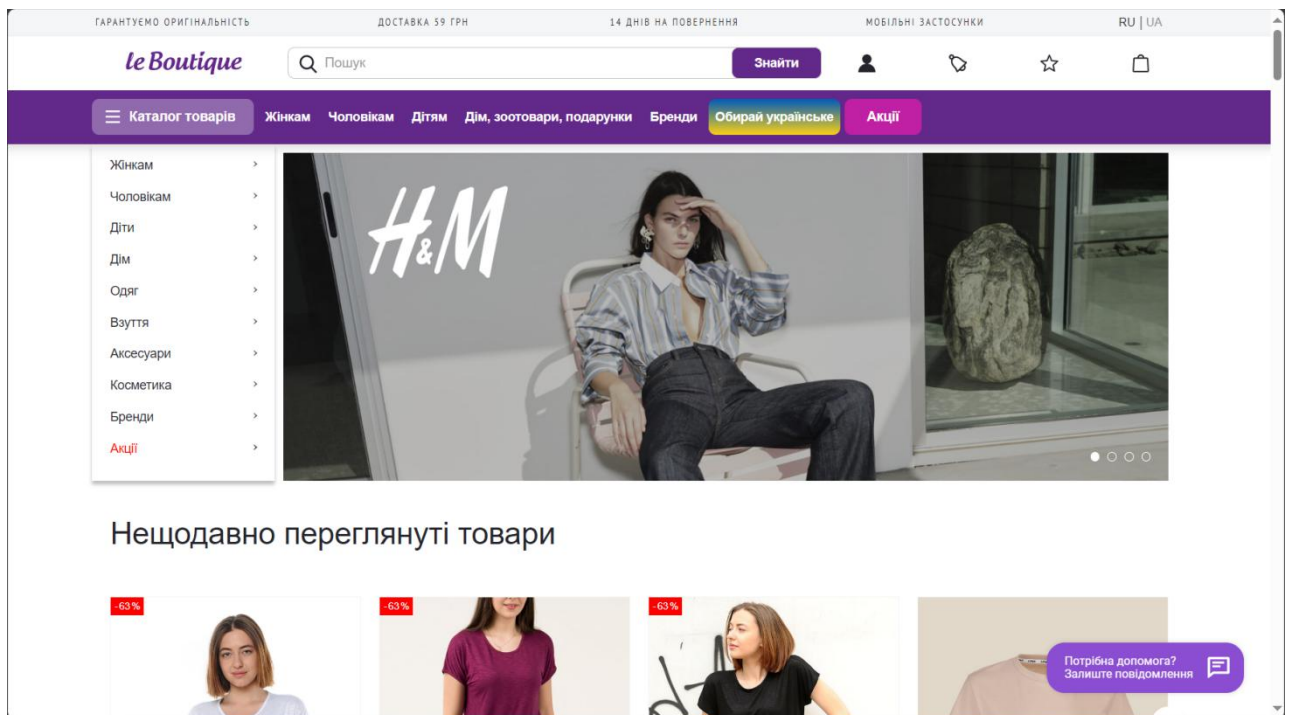


Рисунок 1.3. «LeBoutique»

Розглянемо переваги «LeBoutique»:

1. Великий вибір розмірів: Досить часто можна зіткнутися з проблемою відсутності потрібного розміру серед товару, який сподобався користувачеві. Даний сайт пропонує одяг від 42 розміру до 56 по жіночим міркам та від 44 розміру по 62 по чоловічим, хоча іноді можна знайти розміри ще більше. Звісно це розповсюджується не на весь товар, але на значну його частину.
2. Опис товару: До товару надається детальний опис, в якому може бути написаний склад, країна виробник і бренд виробник (якщо такий є). Нажаль, опис має не весь товар у каталозі, але значна його частина. Наявність детальної інформації позбавить користувача потреби звертатися до служби підтримки.
3. Окрема веб-сторінка українського виробництва: В даному інтернет-магазині є окрема сторінка для товару українських виробників, що, при бажанні користувача, допомагає відсортувати велику кількість іноземних брендів, що полегшує навігацію на веб-ресурсі і допомагає просуненню українського виробництва.

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 001. 00 ДП ПЗ

Арк.

12

Розглянемо недоліки:

1. Перенавантаження головної веб-сторінки акціями: Варто полистати головну сторінку трохи нижче, як почнеться нескінченний перелік акцій на групи товарів та брендів. Це створює дуже багато лишнього шуму в якому справжні вікна товару просто губляться, тоді як якість зображень цих акцій залишає бажати кращого.
2. Неактуальний інтерфейс: Листаючи головну сторінку іноді можна побачити вікна товару, якого вже немає в наявності. Цей товар можна додати у список бажаного з вказанням потрібного розміру, але при цьому, якщо Ви захочете відкрити свій список бажаного Вас просто перекине на головну сторінку сайту [3].

Розглянемо порівняльну таблицю аналогів.

Порівняння аналогів представлено на таблиці 1.1.

Таблиця 1.1. Порівняльна характеристика інтернет-магазинів одягу

Характеристика	ARGO	Colin's	LeBoutique
Актуальність інформації	так	ні	ні
Стрічка рекомендацій	ні	так	так
Великий вибір розмірів	ні	ні	так
Можливість вибору кольору	так	ні	так
Збільшення фотографії товару	ні	так	ні
Детальний опис товару	ні	ні	так

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 001. 00 ДП ПЗ

Арк.

13

Мобільний застосунок	ні	ні	так
Можливість безкоштовної доставки	так	так	ні

Можна зробити висновок, що у наведених аналогів немає великого вибору розмірів одягу та детального опису товару, тому я вважаю доцільною розробку розроблюваного застосунка.

1.1.2 Технології та засоби розробки

Для розробки графічної частини (UI/UX) інтерфейсу веб-застосунку були використані HTML, CSS та Bootstrap.

HTML є відповідальним за створення структури веб-сторінки.

CSS описує зовнішній вигляд веб-сторінок.

Bootstrap дозволяє легко створювати стильні та гнучкі веб-сайти за використанням HTML, CSS та JavaScript

JavaScript та бібліотека jQuery будуть використані для розробки клієнтської частини веб-застосунку (Front-End).

JavaScript надає динамічність та інтерактивність веб-сторінкам, дозволяючи реагувати на дії користувача в режимі реального часу.

jQuery спрощує роботу з JavaScript, пропонуючи легкі у використанні функції для маніпуляцій з елементами сторінки, обробки подій та виконання AJAX-запитів.

У розробці серверної частини (Back-end) веб-застосунку використовуються PHP, SQL, та MySQL.

PHP допомагає сайту генерувати динамічний контент і обробляти запити від користувачів.

SQL виступає як командний центр для баз даних, з можливістю запити, оновлення чи видалення інформації.

MySQL зберігає та керує всіма даними, забезпечуючи швидкий і зручний доступ до потрібної інформації.

На рис. 1.4. зображено обрані для розробки застосунку технології.

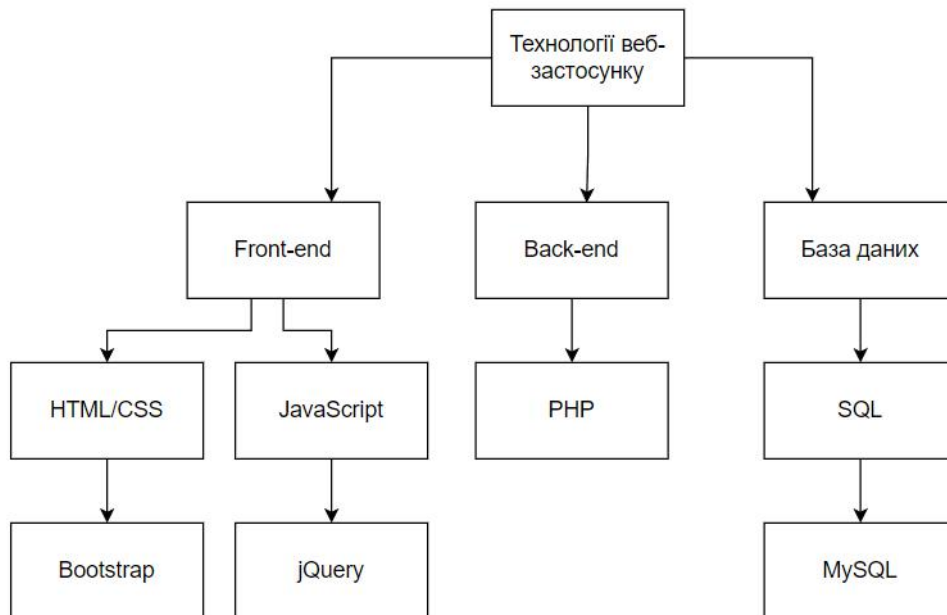


Рисунок 1.4. Схема «Технології веб-застосунку»

Дані технології активно займають ринок по всьому світу протягом багатьох років і як наслідок тримають найвищі позиції у конкурсі мов програмування.

Щоб розглянути цю статистику, було обрано рейтинг зображений на сайті «Stack Overflow»

Stack Overflow виступає як онлайн-спільнота для програмістів. Сервіс де можна задавати питання та знаходити на них відповіді від інших розробників. Це місце, де фахівці діляться своїми знаннями, вирішують проблеми та знаходять рішення. Там можна знайти безліч цікавої та корисної інформації.

Статистику популярності мов програмування з усього світу на 2023 рік було розглянуто на сервісі «Stack Overflow» [4].

На рис. 1.5 зображено топ технологій.

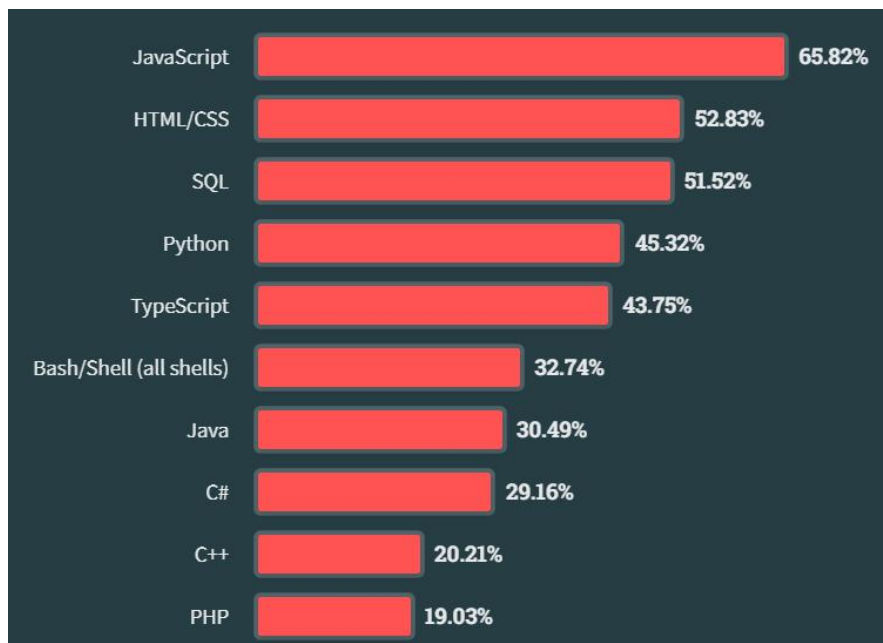


Рисунок 1.5. Статистика популярності мов програмування від сервісу «Stack Overflow»

Розглянемо більш детально обрану технологію для створення веб-системи – мову програмування PHP.

Переваги:

1. Документація. PHP добре документована. На офіційному сайті можна знайти детальну інформацію про всі можливості мови, а також приклади використання методів, функцій. PHP має активну спільноту, яка робить свій внесок у розвиток мови.
2. Багато готових рішень і фреймворків: Існує багато фреймворків (Laravel, Symfony) та CMS (WordPress, Drupal), які значно прискорюють розробку веб-застосунків.
3. Гнучкість. PHP – це вбудована мова, яка забезпечує надзвичайну гнучкість для розробки. Вона сумісна з більшістю існуючих платформ і веб-серверів.
4. Синтаксис. Синтаксис PHP інтуїтивно зрозумілий і гнучкий. Він належить до C-подібних мов і значною мірою запозичений з Java та Perl.
5. Вільне програмне забезпечення: PHP поширюється за спеціальною ліцензією і є безкоштовним.

Недоліки:

1. PHP має обмежені можливості для роботи з винятками, що робить його підтримку винятків слабкою. Це ускладнює обробку помилок і може призводити до менш надійного коду.
2. Проблеми з безпекою: Веб-додатки, написані на PHP, часто стикаються з проблемами безпеки.
3. PHP є спеціалізованою мовою програмування, призначеною для веб-розробки, що обмежує її застосування у таких сферах, як програмування автомобілів чи мікрохвильових печей.
4. PHP не є оптимальним вибором для розробки десктопних додатків, оскільки його основна спеціалізація полягає у створенні веб-додатків.
5. Великі проекти на PHP можуть бути складними в підтримці та масштабуванні, особливо якщо вони не були спочатку добре спроектовані.

Незважаючи на вказані недоліки, PHP було обрано для реалізації даного веб-застосунку з оглядом на його широке поширення та популярність у сфері веб-розробки. Ця мова залишається одним із найпоширеніших виборів для створення веб-сайтів завдяки своїй довговічності та стабільності [5].

Для порівняння з іншою мовою програмування було обрано Python, як одну з найпоширеніших та активно розвинутих мов.

Порівняння PHP та Python представлено на таблиці 1.2.

Таблиця 1.2. Порівняльна характеристика мов програмування

Характеристика	PHP	Python
Основне призначення	Веб-розробка	Веб-розробка, наука про дані, автоматизація
Синтаксис	Складніший, багато дужок	Простий та читабельний
Популярні фреймворки	Laravel, Symfony	Django, Flask

Швидкість виконання	Висока швидкість у веб-додатках	Зазвичай повільніше, але добре оптимізується
Легкість вивчення	Середня складність	Легка складність
Спільнота	Велика спільнота	Дуже велика та активна спільнота
Приклади великих проєктів	WordPress, Wikipedia	Instagram, YouTube
Інструменти та IDE	PhpStorm, NetBeans	PyCharm, VSCod
Розгортання	Легке розгортання на веб-серверах	Може вимагати налаштування середовища
Робота з базами даних	MySQL, PostgreSQL, SQLite через PDO чи MySQLi	MySQL, PostgreSQL, SQLite через SQLAlchemy, Django ORM
Типізація	Динамічна типізація	Динамічна типізація
Масштабованість	Добре підходить для малих і середніх проєктів	Підходить для проєктів будь-якого розміру
Безпека	Має вразливості, але їх можна уникнути при правильній конфігурації	Високий рівень безпеки завдяки сучасним фреймворкам
Документація	Відмінна, багато ресурсів та прикладів	Відмінна, багато ресурсів та прикладів
Підтримка сучасних технологій	Часто потребує додаткових бібліотек	Має вбудовану підтримку багатьох сучасних технологій
Середовище виконання	Переважно серверне	Серверне, десктопне, мобільне

Ізм.	Лист	№ докум.	Підпис	Дата

Модульність	Висока, з великою кількістю готових рішень	Висока, з багатою екосистемою модулів та пакетів
Підтримка багатозадачності	Обмежена, часто потребує сторонніх рішень	Вбудована підтримка багатозадачності, хороша робота з потоками
Використання в хмарних рішеннях	Добре підтримується	Висока підтримка, багато сервісів пропонують Python як основну мову

1.2 Проєктування застосунку

До застосунку, що розробляється, ставляться наступні вимоги:

1. **Наявність привабливого графічного інтерфейсу (GUI):** Графічний інтерфейс застосунку повинен створювати естетично приємний та візуально привабливий дизайн для своєї цільової аудиторії. Такий інтерфейс має відображати стиль і бренд магазину, використовуючи відповідні кольори, шрифти та зображення, щоб залучити та утримати увагу користувачів. При цьому інтерфейс не повинен бути перевищеним великою кількістю непотрібних кнопок і деталей, що лише створить візуальний шум і буде розріджувати увагу користувача.
2. **Наявність інтуїтивно зрозумілого досвіду користувача (UX):** Навігація у застосунку має бути легкою та логічною, а всі функції – доступними та зрозумілими навіть для новачків. Користувачі повинні мати змогу швидко знаходити потрібну інформацію, переглядати товари, додавати їх до кошика та здійснювати покупки без зайвих труднощів. Розташування кнопок повинне бути у знайомих місцях, задля уникнення плутанини.
3. **Наявність реєстрації та кабінету користувача:** Користувачі повинні мати можливість створювати облікові записи, де вони зможуть зберігати свої особисті дані, переглядати історію покупок, відстежувати статус

замовлень та отримувати персоналізовані пропозиції. Кабінет користувача повинен бути зручним у використанні та забезпечувати високий рівень безпеки даних.

4. Наявність панелі адміністратора: Панель дозволить власникам та менеджерам магазину ефективно керувати веб-сайтом. Адміністративна панель має включати функції для додавання та редагування товарів і каталогу в цілому, також управління замовленнями та налаштування різних аспектів сайту. Варто розглянути можливість відстежувати відвідуваність сайту та окремих його сторінок, це дозволить детально аналізувати коло інтересу своєї цільової аудиторії. Панель повинна бути зрозумілою та функціональною, забезпечуючи повний контроль над діяльністю інтернет-магазину.
 5. Наявність кошика: Для коректної роботи інтернет-магазину, кошик є важливим елементом, для реалізації. Кошик повинен забезпечувати користувачам можливість додавати товари та керувати ними. Користувачі мають бачити детальну інформацію про товари в кошику, включаючи назву, зображення, розмір, колір, ціну та кількість. Також важливо, щоб кошик дозволяв змінювати кількість товарів або видаляти їх з неї.
- На рис. 1.6 зображено схему навігації для розроблюваного сайту.

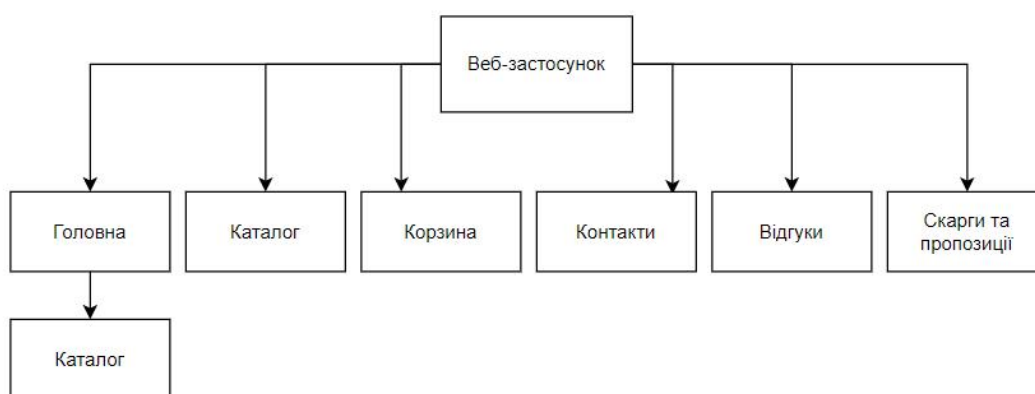


Рисунок 1.6. Схема навігації веб-застосунку

Ізм.	Лист	№ докум.	Підпис	Дата

1.2.1 Проєктування дизайну застосунку

При розробці застосунку важливим етапом є проєктування його дизайну. Почати можна з аналізу прототипів, тобто обрати веб-сайти зі схожою тематикою і виділити в них найкращі рішення для подальшого застосування у своєму проєкті. Невдалі рішення також варто запам'ятати, щоб не повторити цю помилку у своєму веб-застосунку. Прийняті рішення потрібно узгоджувати з замовником та дотримуватися його побажань.

Так як розробляється веб-сайт для інтернет-магазину, обов'язково будуть присутні специфічні елементи даної предметної області: Назва та логотип магазину, каталог товару та корзина в якій замовлення можна буде оформити. Сама веб-сторінка повинна мати панель навігації та кінцевий блок.

Зокрема, дизайн панелі навігації є важливою частиною застосунку, як візуальний елемент, що містить посилання на головні розділи або сторінки застосунку.

На рис. 1.7 зображено прототип дизайну навігаційного меню розроблюваного застосунку.

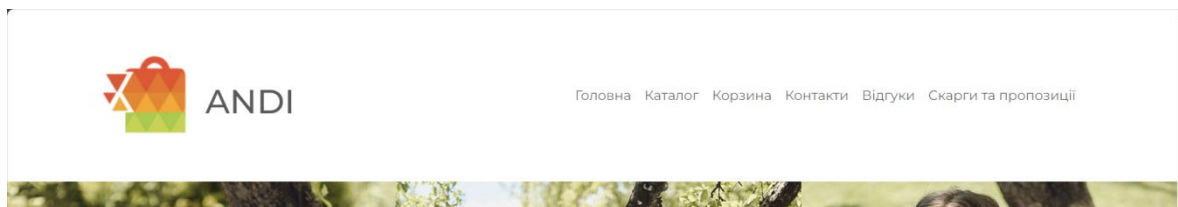


Рисунок 1.7. Навігаційна панель, розроблюваного інтернет-магазину

Розглянемо хедер головної сторінки, веб-застосунку.

У випадку розроблюваного веб-сайту хедер являє собою інтерактивне зображення, при натисненні на котре користувач потрапить на сторінку каталогу та окремих категорій товару, або на іншу вітку, обрану замовником/розробником.

На рис. 1.8 зображено хедер веб-застосунку.

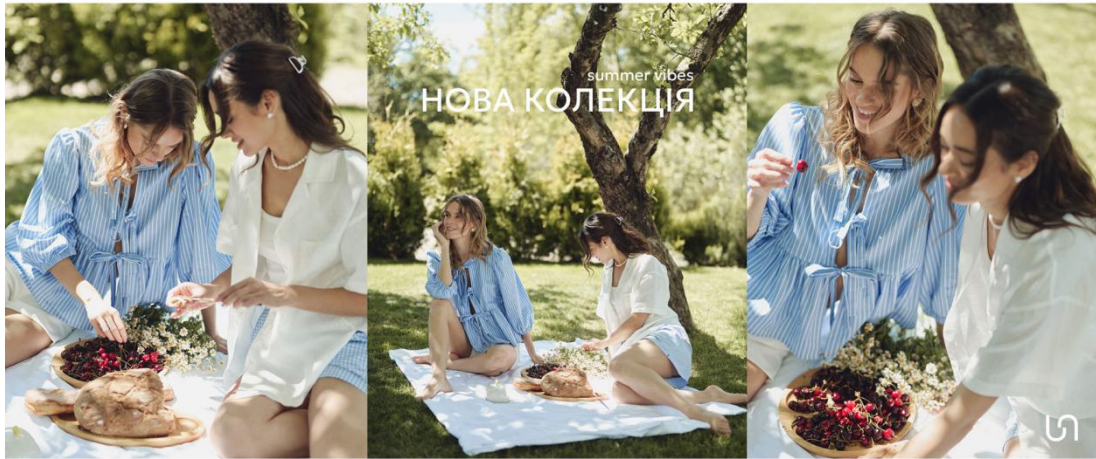


Рисунок 1.8. Дизайн хедеру веб-застосунку

Тепер розглянемо контентну частину веб-застосунку.

У контентній частині, у випадку розроблюваного застосунку, буде розміщено повідомлення від серверу. Це може бути як оголошення акції, так і сповіщення про зміни у розкладі роботи магазину.

На рис. 1.9 зображено контентну частину веб-застосунку.

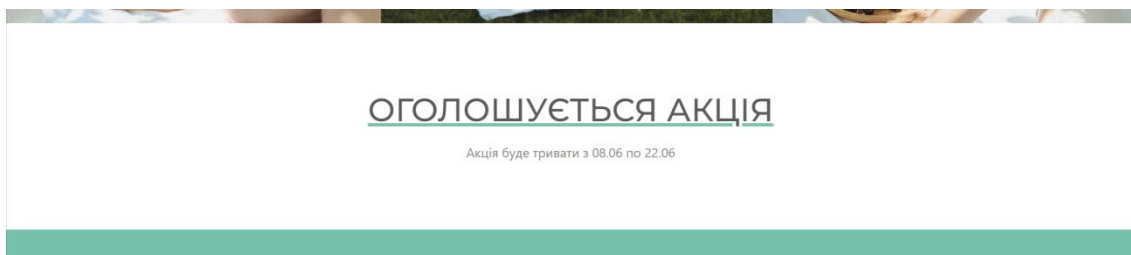


Рисунок 1.9. Дизайн контентної частини веб-застосунку

Розглянемо дизайн футеру веб-застосунку.

Футер представляє собою нижню (підвальну) частину веб-сторінки, у якій розміщено окрему навігацію по веб-сайту та додаткові посилання, наприклад контактні дані, соціальні мережі, відгуки і інші додаткові елементи [6].

На рис. 1.10 зображено футер веб-застосунку.



Рисунок 1.10. Дизайн футеру веб-застосунку

1.2.2 Проектування архітектури застосунку

При розробці архітектури застосунку важливо обрати відповідну модель, яка забезпечить ефективну організацію коду та полегшить його підтримку. Однією з найпопулярніших архітектурних моделей є модель "Model-View-Controller" (MVC). Ця модель забезпечує чітке розділення обов'язків між різними компонентами застосунку, що значно спрощує його розробку та підтримку.

У моделі MVC застосунок розділений на три основні компоненти: Model, View та Controller.

Model відповідає за управління даними застосунку. Він обробляє доменну логіку, взаємодіє з базою даних та відповідає за збереження і обробку даних. Всі зміни, що відбуваються з даними, обробляються в Model, яка також повідомляє View про необхідність оновлення відображення.

View займається представленням даних користувачу. Вона отримує дані з Model та відображає їх у відповідному форматі. View не містить бізнес-логіки і не взаємодіє з базою даних. Її єдина задача – забезпечити правильне та зручне відображення даних для користувача.

Controller діє як посередник між Model і View. Він отримує вхідні запити від користувача через View, обробляє їх за допомогою Model та визначає, яку View слід використовувати для відображення результату. Controller координує роботу всіх частин застосунку, забезпечуючи їх взаємодію [5].

На рис. 1.11 зображено схему архітектури веб-застосунку.

					РП 07. 22 001. 00 ДП ПЗ	Арк.
						23
Ізм.	Лист	№ докум.	Підпис	Дата		

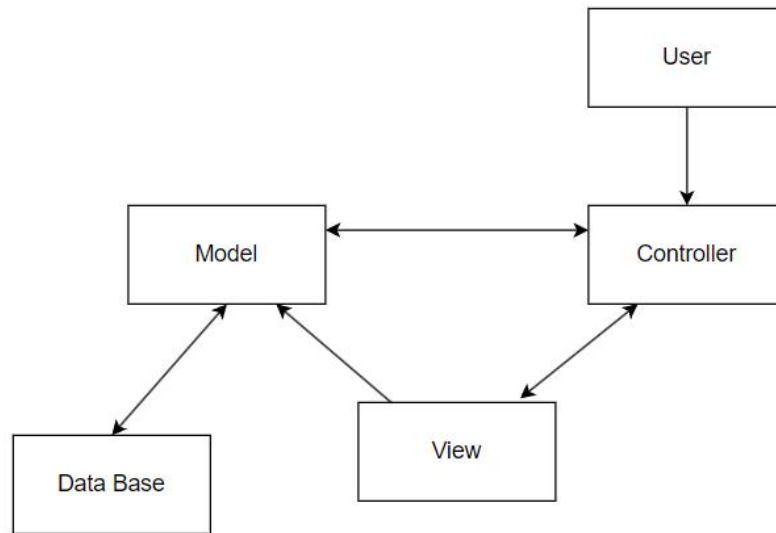


Рисунок 1.11. Схема архітектури веб-застосунку

Для реалізації веб-застосунку за даними параметрами використовується мікро-фреймворк PHP з наступною архітектурою.

На рис. 1.12 зображена файлова архітектура мікро-фреймворку PHP.

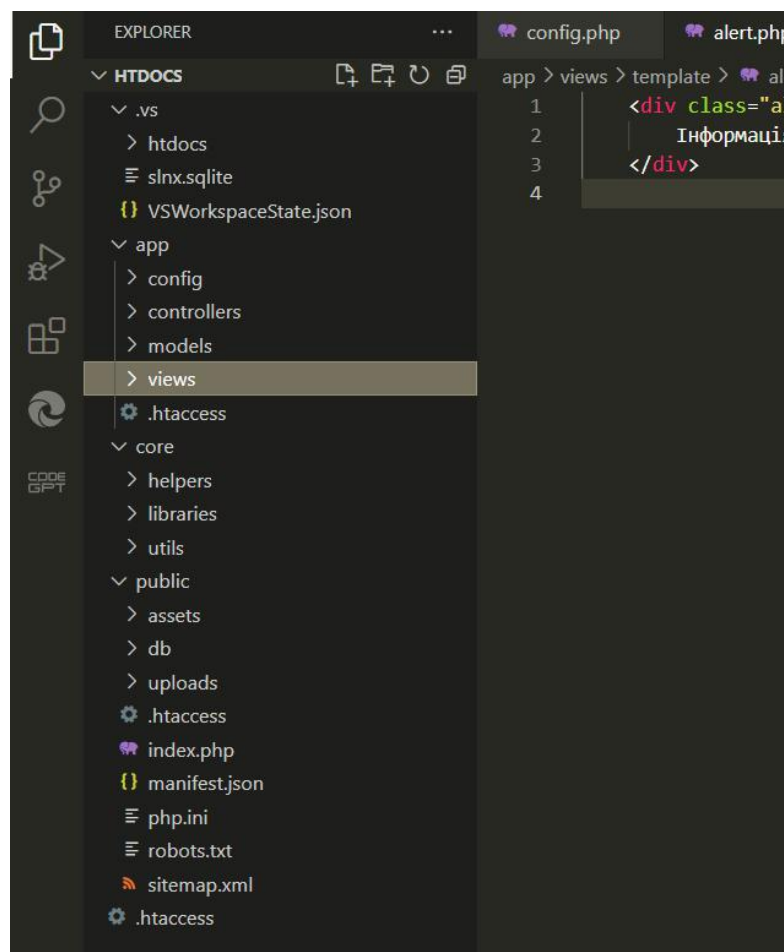


Рисунок 1.12. Архітектура мікро-фреймворку PHP.

Розглянемо файлову архітектуру мікро-фреймворку PHP детальніше:

1. `/app`: Ця папка містить основний застосунок. Вона містить логіку застосунку, включаючи моделі, контролери та представлення.
2. `/app/views`: У цій папці розташовані файли, які відповідають за представлення (View) в шаблонах MVC. Тут зазвичай знаходяться файли HTML, CSS та JavaScript, що відображають дані користувачам.
3. `/app/models`: Модель (Model) в MVC відповідає за взаємодію з базою даних та обробку даних. У цій папці зазвичай знаходяться файли, які представляють різні об'єкти даних та виконують логіку доступу до даних.
4. `/app/controllers`: Контролер (Controller) в MVC призначений для обробки запитів користувача та виклику відповідних методів моделі та представлення для відображення результатів. У цій папці зазвичай знаходяться файли, які містять класи контролерів.
5. `/app/config`: У цій папці зазвичай знаходяться файли конфігурації додатка, такі як файли налаштувань маршрутизації, бази даних, тощо.
6. `/core`: Ця папка містить основний код фреймворка. Вона може включати в себе допоміжні класи, функції та бібліотеки, необхідні для роботи фреймворка.
7. `/core/helpers`: У цій папці зазвичай знаходяться допоміжні функції або класи, які можуть бути використані для зручності розробки.
8. `/core/utils`: Ця папка містить допоміжні утиліти або класи, які надають корисні функції, які можуть бути використані в різних частинах фреймворка.
9. `/core/libraries`: Тут зазвичай знаходяться сторонні бібліотеки або компоненти, які використовуються в додатку.
10. `/public`: Ця папка містить ресурси, які доступні публічно через веб-сервер. Це може включати файли CSS, JavaScript, зображення тощо.
11. `/public/uploads`: У цій папці зазвичай зберігаються файли, які користувачі завантажують на сервер.

					<i>РП 07. 22 001. 00 ДП ПЗ</i>	Арк.
						25
Ізм.	Лист	№ докум.	Підпис	Дата		

12. ./public/assets: Тут зазвичай знаходяться статичні ресурси, такі як стилі, скрипти тощо, які використовуються в клієнтському коді.

13. ./public/db: Ця папка може містити файли резервних копій бази даних або інші файли, пов'язані з базою даних.

1.2.3 Проєктування бази даних

База даних являє собою систему, що зберігає пов'язані дані в таблицях. Програмне забезпечення, яке дозволяє створювати, редагувати та використовувати базу даних, називається системою управління базами даних. У базах даних таблиці складаються з записів, що містять іменовані поля з певними типами даних. При проєктуванні бази даних було обрано реляційну модель, де дані організовані у вигляді таблиць з рядками і стовпцями та зв'язками між таблицями.

Для створення схеми бази даних використано phpMyAdmin – веб-інструмент з відкритим кодом, призначений для адміністрування MySQL [7].

На рис. 1.13 зображено схему бази даних веб-застосунку.

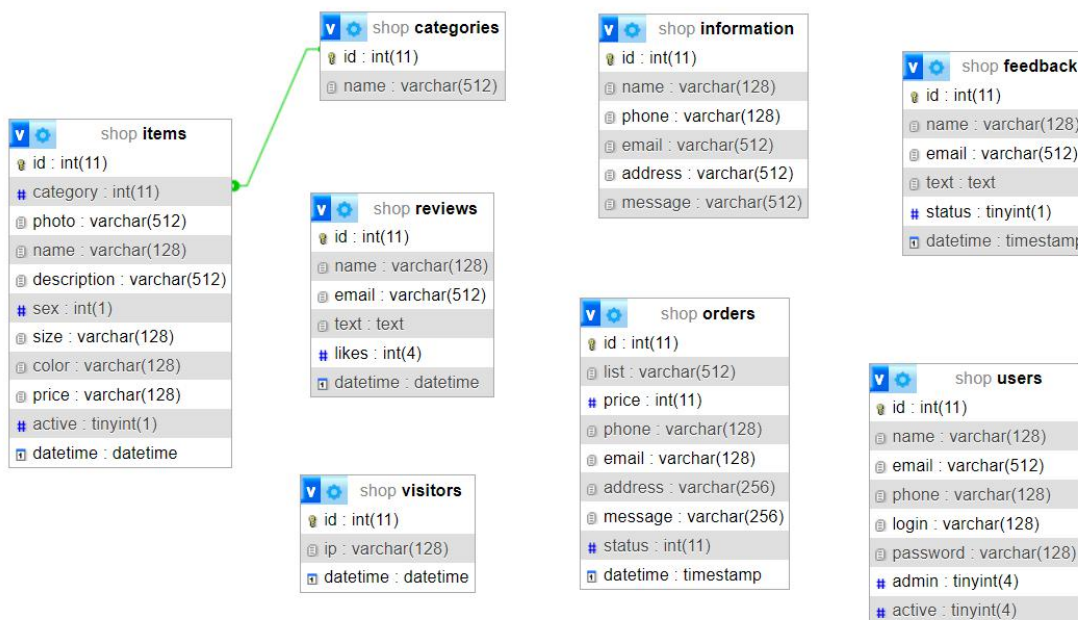


Рисунок 1.13. Схема бази даних веб-застосунку

Розглянемо структуру бази даних веб-застосунку.

Таблиця «items» містить інформацію про товар.

У табл. 1.3 наведено опис категорії «items».

Таблиця 1.3. Опис таблиці «items»

Поле	Тип	Опис
id	INT	Ідентифікатор товару.
category	INT	Ідентифікатор категорії товару (зв'язок з категоріями).
photo	VARCHAR	Шлях до зображення товару.
name	VARCHAR	Назва товару.
description	VARCHAR	Опис товару.
sex	INT	Стать товару.
size	VARCHAR	Розмір товару.
color	VARCHAR	Колір товару.
price	VARCHAR	Ціна товару.
active	TINYINT	Статус активності товару.
datetime	DATETIME	Дата та час додавання товару.

Таблиця “categories” містить інформацію про категорії товару.

У табл. 1.4 наведено опис категорії “categories”.

Таблиця 1.4. Опис таблиці “categories”

Поле	Тип	Опис
id	INT	Ідентифікатор категорії.
name	VARCHAR	Назва категорії.

Таблиця «reviews» містить інформацію про відгуки.

У табл. 1.5 наведено опис категорії «reviews».

Таблиця 1.5. Опис таблиці «reviews»

Поле	Тип	Опис
id	INT	Ідентифікатор відгуку.
name	VARCHAR	Ім'я автора відгуку.
email	VARCHAR	Email автора відгуку.
text	TEXT	Текст відгуку.
likes	INT	Кількість лайків для відгуку.
datetime	DATETIME	Дата та час створення відгуку.

Таблиця «visitors» містить інформацію про відвідувачів веб-системи.

У табл. 1.6 наведено опис категорії «visitors».

Таблиця 1.6 Опис таблиці «visitors»

Поле	Тип	Опис
id	INT	Ідентифікатор відвідувача.
ip	VARCHAR	IP-адреса відвідувача.
datetime	DATETIME	Дата та час візиту.

Таблиця «information» містить інформацію про інформацію у веб-системі.

У табл. 1.7 наведено опис категорії «information».

Таблиця 1.7. Опис таблиці «information»

Поле	Тип	Опис
id	INT	Ідентифікатор запису.
name	VARCHAR	Назва магазину.
phone	VARCHAR	Номер телефону магазину.
email	VARCHAR	Email магазину.

Продовження таблиці 1.7

address	VARCHAR	Адреса магазину.
message	VARCHAR	Сповіщення від магазину.

Таблиця «orders» містить інформацію про замовлення.

У табл. 1.8 наведено опис категорії «orders».

Таблиця 1.8. Опис таблиці «orders»

Поле	Тип	Опис
id	INT	Ідентифікатор замовлення.
list	VARCHAR	Список товару у замовленні.
price	INT	Ціна замовлення.
phone	VARCHAR	Номер телефону замовника.
email	VARCHAR	Email замовника.
adress	VARCHAR	Адреса замовника.
message	VARCHAR	Додаткове повідомлення від замовника.
status	INT	Статус замовлення.
datetime	TIMESTAMP	Дата та час створення замовлення.

Таблиця «feedback» містить інформацію про зворотній зв'язок.

У табл. 1.9 наведено опис категорії «feedback».

Таблиця 1.9. Опис таблиці «feedback»

Поле	Тип	Коментар
id	INT	Ідентифікатор відгуку.
name	VARCHAR	Ім'я автора відгуку.
email	VARCHAR	Email автора відгуку.

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 001. 00 ДП ПЗ

Арк.

29

text	TEXT	Текст відгуку.
status	TINYINT	Статус відгуку.
datetime	TIMESTAMP	Дата та час створення відгуку.

Таблиця «users» містить інформацію про користувачів у веб-системі.

У табл. 1.10 наведено опис категорії «users».

Таблиця 1.10. Опис таблиці «users»

Поле	Тип	Коментар
id	INT	Ідентифікатор користувача.
name	VARCHAR	Ім'я користувача.
email	VARCHAR	Email користувача.
phone	VARCHAR	Телефон користувача.
login	VARCHAR	Логін користувача.
password	VARCHAR	Пароль користувача.
admin	TINYINT	Прапорець адміністратора.
active	TINYINT	Статус активності облікового запису користувача.

Розглянемо список створених таблиць у phpMyAdmin.

На рис. 1.14 зображено список створених таблиць в phpMyAdmin.

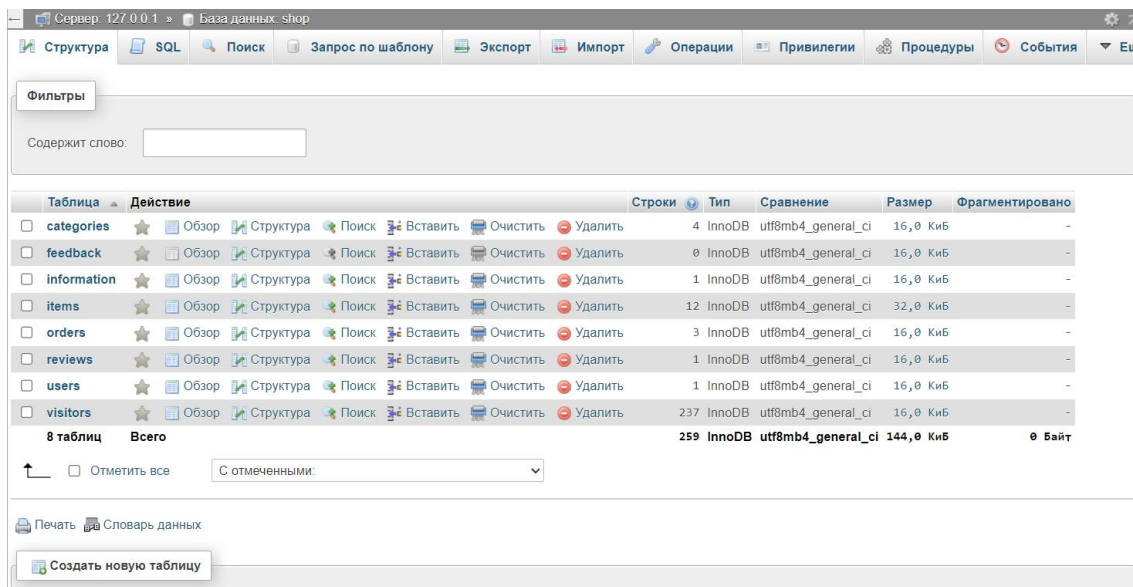


Рисунок 1.14. Список створених таблиц в phpMyAdmin

1.3 Реалізація застосунку

1.3.1 Створення моделей (Models)

У архітектурі MVC модель відповідає за управління даними та їх обробку. Вона забезпечує логіку програми, включаючи взаємодію з базою даних, якщо це потрібно. Модель не залежить від відображення (View) або контролера (Controller) і не має знати про інші компоненти програми. Основне завдання моделі – забезпечити стабільне представлення даних незалежно від того, як вони відображаються або як користувач з ними взаємодіє.

На рис. 1.15 представлено створені моделі веб-застосунку.

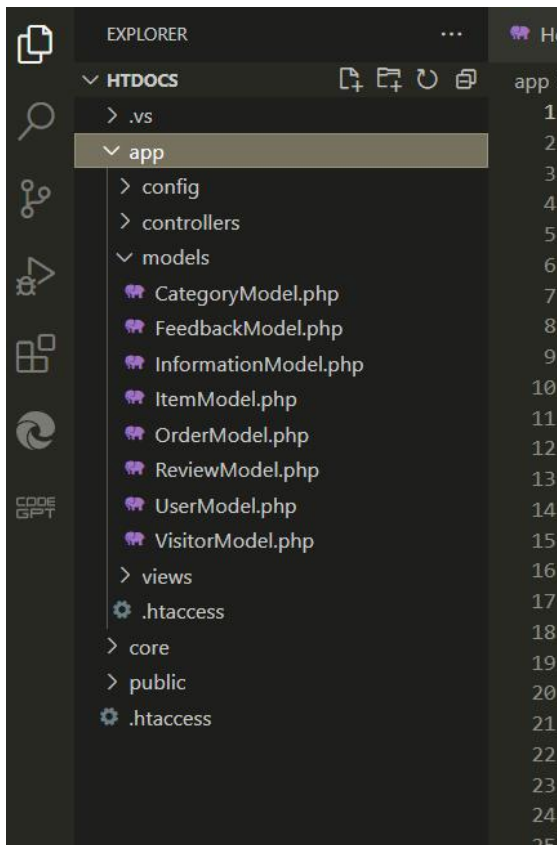


Рисунок 1.15. Створені моделі MVC

Розглянемо код основної моделі, що відповідає за товари.

Нижче наведено код моделі.

```
<?php
class ItemModel
{
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function selectAll()
    {
        // Query
        $this->db->query('SELECT * FROM items');

        // Execute
        $results = $this->db->resultSet();

        // Return result
        return $results;
    }

    public function selectById($id)
    {
        // Query
        $this->db->query('SELECT * FROM items WHERE id = :id');
```

```

// Bind value
$this->db->bind(':id', $id);

// Execute
$results = $this->db->single();

// Return result
return $results;
}

public function search($filters, $pageNumber = 1, $perPage = 10) {
// Base query
$query = 'SELECT * FROM items WHERE 1';

// Prepare an array to hold values to bind
$bindValue = [];

// Check each filter and add conditions to the query accordingly
if (is_array($filters) && !empty($filters['category'])) {
    $query .= ' AND category IN (' . implode(',', array_fill(0,
count($filters['category']), '?')) . ')';
    $bindValue = array_merge($bindValue, $filters['category']);
}

if (is_array($filters) && !empty($filters['name'])) {
    $query .= ' AND name LIKE ?';
    $bindValue[] = '%' . $filters['name'] . '%';
}

if (is_array($filters) && !empty($filters['description'])) {
    $query .= ' AND description LIKE ?';
    $bindValue[] = '%' . $filters['description'] . '%';
}

if (is_array($filters) && !empty($filters['sex'])) {
    $query .= ' AND sex IN (' . implode(',', array_fill(0,
count($filters['sex']), '?')) . ')';
    $bindValue = array_merge($bindValue, $filters['sex']);
}

if (!empty($filters['size'])) {
    $sizeConditions = [];
    foreach ($filters['size'] as $size) {
        $sizeConditions[] = 'size LIKE ?';
        $bindValue[] = '%' . $size . '%';
    }
    $query .= ' AND (' . implode(' OR ', $sizeConditions) . ')';
}

if (!empty($filters['color'])) {
    $colorConditions = [];
    foreach ($filters['color'] as $color) {
        $colorConditions[] = 'color LIKE ?';
        $bindValue[] = '%' . $color . '%';
    }
    $query .= ' AND (' . implode(' OR ', $colorConditions) . ')';
}

if (is_array($filters) && !empty($filters['price'])) {
    $query .= ' AND price <= ?'; // Adjusted for items with price less
than or equal to the specified price
}
}

```

					РП 07. 22 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		33

```

        $bindValue[] = $filters['price'];
    }

    // Pagination
    $offset = ($pageNumber - 1) * $perPage;
    $query .= ' LIMIT ' . $perPage . ' OFFSET ' . $offset;

    // Prepare the query
    $this->db->query($query);

    // Bind values
    foreach ($bindValue as $key => $value) {
        $this->db->bind($key + 1, $value); // PDO parameter index starts from
1    }

    // Execute
    $results = $this->db->resultSet();

    // Return result
    return $results;
}

public function insert($data)
{
    // Query
    $this->db->query('INSERT INTO items (category, photo, name, description,
sex, size, color, price, active)
VALUES(:category, :photo, :name, :description, :sex, :size, :color, :price, :active)');

    // Bind values
    $this->db->bind(':category', $data['category']);
    $this->db->bind(':photo', $data['photo']);
    $this->db->bind(':name', $data['name']);
    $this->db->bind(':description', $data['description']);
    $this->db->bind(':sex', $data['sex']);
    $this->db->bind(':size', $data['size']);
    $this->db->bind(':color', $data['color']);
    $this->db->bind(':price', $data['price']);
    $this->db->bind(':active', $data['active']);

    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}

public function update($data)
{
    // Query
    $this->db->query('UPDATE items SET category = :category, photo = :photo,
name = :name, description = :description, sex = :sex, size = :size, color = :color,
price = :price, active = :active WHERE id = :id');

    // Bind values

```

					РП 07. 22 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		34

```

$this->db->bind(':id', $data['id']);
$this->db->bind(':category', $data['category']);
$this->db->bind(':photo', $data['photo']);
$this->db->bind(':name', $data['name']);
$this->db->bind(':description', $data['description']);
$this->db->bind(':sex', $data['sex']);
$this->db->bind(':size', $data['size']);
$this->db->bind(':color', $data['color']);
$this->db->bind(':price', $data['price']);
$this->db->bind(':active', $data['active']);

// Execute
if($this->db->execute())
{
    return true;
}
else
{
    return false;
}
}

public function delete($data)
{
    // Query
    $this->db->query('DELETE FROM items WHERE id = :id');

    // Bind values
    $this->db->bind(':id', $data['id']);

    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}
}
?>

```

Цей PHP-код представляє основну модель, яка відповідає за роботу з товарами.

1. **construct**: Це конструктор класу `ItemModel`, який ініціалізує об'єкт бази даних при створенні нового об'єкта моделі.
2. **selectAll**: Цей метод виконує запит до бази даних для вибору всіх товарів і повертає результат.
3. **selectById**: Цей метод приймає ідентифікатор товару як параметр, виконує запит до бази даних для вибору товару за його ідентифікатором і повертає результат.

4. search: Цей метод приймає різні фільтри для пошуку товарів, такі як категорія, назва, опис, розмір, колір, ціна тощо. Він будує SQL-запит на основі цих фільтрів та параметрів пагінації та повертає результат.
5. insert: Цей метод виконує вставку нового товару в базу даних з даними, переданими як параметри.
6. update: Цей метод виконує оновлення існуючого товару в базі даних з даними, переданими як параметри.
7. delete: Цей метод виконує видалення товару з бази даних за його ідентифікатором.

1.3.2 Створення представлень (Views)

У архітектурі MVC представлення (View) відповідає за відображення даних користувачу та інтерактивну взаємодію з ним. Воно не містить логіки програми і не працює з даними напряму, а лише відображає їх, якщо отримує відповідні дані від контролера. Представлення може бути графічним інтерфейсом користувача, веб-сторінкою або будь-яким іншим способом відображення інформації для користувача. Воно сприймає введення від користувача і передає його контролеру для подальшої обробки.

На рис. 1.16 зображено представлення створюваного веб-застосунку

					РП 07. 22 001. 00 ДП ПЗ	Арк.
						36
Ізм.	Лист	№ докум.	Підпис	Дата		

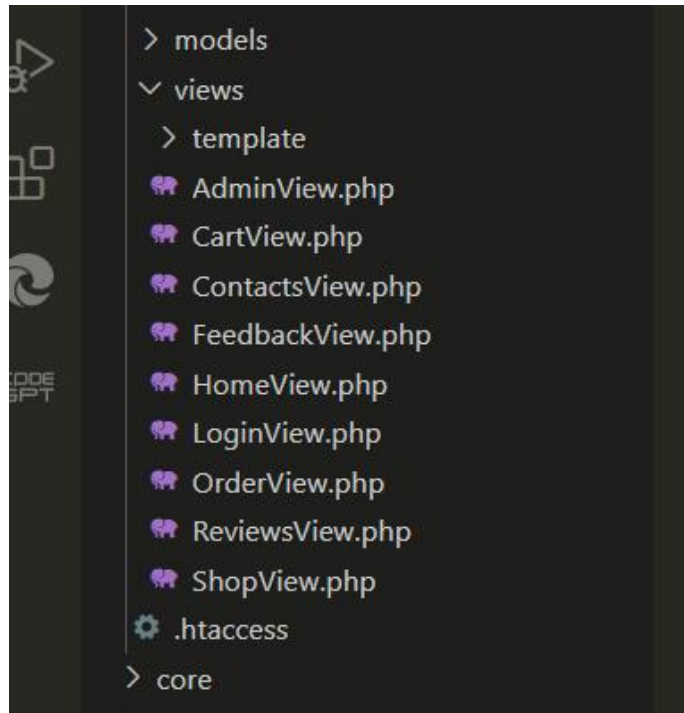


Рисунок 1.16. Представлення веб-застосунку

Розглянемо код представлення, що відповідає за каталог магазину.

Нижче наведено код представлення.

```

<!DOCTYPE html>
<html data-bs-theme="light" lang="en">

<head>
  <?php require_once APPROOT . '/views/template/meta.php'; ?>
  <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
</head>

<body class="bg-white">
  <?php require_once APPROOT . '/views/template/scripts.php'; ?>
  <?php require_once APPROOT . '/views/template/navmenu.php'; ?>
  <?php require_once APPROOT . '/views/template/header.php'; ?>

  <section class="p-5">
    <div class="container">
      <h1 class="text-center m-3">Фільтри</h1>
      <div class="row justify-content-center align-items-start m-3">
        <form method="GET" action="">
          <div class="mb-3">
            <label for="category">Категорія:</label><br>
            <?php foreach ($data['categories'] as $category) : ?>
              <input type="checkbox" name="category[]" value="<?php
echo $category->id; ?>" <?php if (isset($_GET['category']) && in_array($category-
>id, $_GET['category'])) echo 'checked'; ?>>
              <?php echo $category->name; ?><br>
            <?php endforeach; ?>
          </div>
          <div class="mb-3">
            <input type="text" class="form-control" name="name"
placeholder="Назва" value="<?php echo isset($_GET['name']) ? $_GET['name'] :
''; ?>">

```

```

        </div>
        <div class="mb-3">
            <input type="text" class="form-control" name="description"
placeholder="Опис" value="<?php echo isset($_GET['description']) ?
$_GET['description'] : ''; ?>">
        </div>
        <div class="mb-3">
            <label>Стать:</label><br>
            <input type="checkbox" name="sex[]" value="1" <?php if
(isset($_GET['sex']) && in_array('1', $_GET['sex'])) echo 'checked'; ?>>
Унісекс<br>
            <input type="checkbox" name="sex[]" value="2" <?php if
(isset($_GET['sex']) && in_array('2', $_GET['sex'])) echo 'checked'; ?>>
Жіночий<br>
            <input type="checkbox" name="sex[]" value="3" <?php if
(isset($_GET['sex']) && in_array('3', $_GET['sex'])) echo 'checked'; ?>>
Чоловічий<br>
        </div>
        <div class="mb-3">
            <label>Розмір:</label><br>
            <input type="checkbox" name="size[]" value="XS" <?php if
(isset($_GET['size']) && in_array('XS', $_GET['size'])) echo 'checked'; ?>> XS<br>
            <input type="checkbox" name="size[]" value="S" <?php if
(isset($_GET['size']) && in_array('S', $_GET['size'])) echo 'checked'; ?>> S<br>
            <input type="checkbox" name="size[]" value="M" <?php if
(isset($_GET['size']) && in_array('M', $_GET['size'])) echo 'checked'; ?>> M<br>
            <input type="checkbox" name="size[]" value="L" <?php if
(isset($_GET['size']) && in_array('L', $_GET['size'])) echo 'checked'; ?>> L<br>
            <input type="checkbox" name="size[]" value="XL" <?php if
(isset($_GET['size']) && in_array('XL', $_GET['size'])) echo 'checked'; ?>> XL<br>
            <input type="checkbox" name="size[]" value="XXL" <?php if
(isset($_GET['size']) && in_array('XXL', $_GET['size'])) echo 'checked'; ?>>
XXL<br>
        </div>
        <div class="mb-3">
            <label>Колір:</label><br>
            <input type="checkbox" name="color[]" value="червоний"
<?php if (isset($_GET['color']) && in_array('червоний', $_GET['color'])) echo
'checked'; ?>> червоний<br>
            <input type="checkbox" name="color[]" value="оранжевий"
<?php if (isset($_GET['color']) && in_array('оранжевий', $_GET['color'])) echo
'checked'; ?>> оранжевий<br>
            <input type="checkbox" name="color[]" value="жовтий" <?php
if (isset($_GET['color']) && in_array('жовтий', $_GET['color'])) echo
'checked'; ?>> жовтий<br>
            <input type="checkbox" name="color[]" value="зелений"
<?php if (isset($_GET['color']) && in_array('зелений', $_GET['color'])) echo
'checked'; ?>> зелений<br>
            <input type="checkbox" name="color[]" value="блакитний"
<?php if (isset($_GET['color']) && in_array('блакитний', $_GET['color'])) echo
'checked'; ?>> блакитний<br>
            <input type="checkbox" name="color[]" value="синій" <?php
if (isset($_GET['color']) && in_array('синій', $_GET['color'])) echo
'checked'; ?>> синій<br>
            <input type="checkbox" name="color[]" value="фіолетовий"
<?php if (isset($_GET['color']) && in_array('фіолетовий', $_GET['color'])) echo
'checked'; ?>> фіолетовий<br>
            <input type="checkbox" name="color[]" value="чорний" <?php
if (isset($_GET['color']) && in_array('чорний', $_GET['color'])) echo
'checked'; ?>> чорний<br>

```

					РП 07. 22 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		38

```

        <input type="checkbox" name="color[]" value="білий" <?php
if (isset($_GET['color']) && in_array('білий', $_GET['color'])) echo
'checked'; ?>> білий<br>
        <input type="checkbox" name="color[]" value="сірий" <?php
if (isset($_GET['color']) && in_array('сірий', $_GET['color'])) echo
'checked'; ?>> сірий<br>
        </div>
        <div class="mb-3">
            <input type="number" class="form-control" name="price"
placeholder="Ціна" value="<?php echo isset($_GET['price']) ? $_GET['price'] :
''; ?>">
        </div>
        <button type="submit" class="btn btn-
primary">Фільтрувати</button>
    </form>
</div>
<?php if (!empty($data['items'])): ?>
    <div class="row row-cols-3 justify-content-center align-items-
center">
        <?php foreach ($data['items'] as $item) : ?>
            <div class="col">
                <div class="card" data-bss-hover-animate="pulse">
                    
                    <div class="card-body">
                        <h1 class="card-title"><?php echo $item-
>name; ?></h1>
                        <p class="card-text"><?php echo $item-
>description; ?></p>
                        <p class="card-text">Стать:
                            <?php
switch ($item->sex) {
                                case 1:
                                    echo 'Унісекс';
                                    break;
                                case 2:
                                    echo 'Жіноча';
                                    break;
                                case 3:
                                    echo 'Чоловіча';
                                    break;
                            }
                            ?>
                        </p>
                        <p class="card-text">Кольори: <?php echo
$item->color; ?></p>
                        <p class="card-text">Розміри: <?php echo
$item->size; ?></p>
                        <h4 class="card-title"><?php echo $item-
>price; ?> грн.</h4>
                        <form method="post" action="<?php echo
URLROOT; ?>/cart/addToCart/<?php echo $item->id; ?>">
                            <input type="text" class="form-control mt-
3 mb-3" name="color" placeholder="Колір">
                            <input type="text" class="form-control mt-
3 mb-3" name="size" placeholder="Розмір">
                            <input type="number" class="form-control
mt-3 mb-3" name="quantity" placeholder="Кількість">
                            <button class="btn btn-primary mt-3 mb-3
w-100" type="submit">Додати в корзину</button>
                        </form>

```

					РП 07. 22 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		39

```

        </div>
    </div>
</div>
    <?php endforeach; ?>
</div>
    <?php else : ?>
        <div class="row row-cols-3 justify-content-center align-items-
center">
            <div class="col">
                <?php require APPROOT . '/views/template/alert.php'; ?>
            </div>
        </div>
    <?php endif; ?>
    <div class="row row-cols-3 justify-content-center align-items-center">
        <div class="col">
            <div class="pagination justify-content-center mt-4">
                <a href="?page=<?php echo $data['pageNumber'] - 1 .
buildQueryParams(array_merge($_GET, ['page' => null])); ?>" class="btn btn-
primary">Назад</a>
                <a href="?page=<?php echo $data['pageNumber'] + 1 .
buildQueryParams(array_merge($_GET, ['page' => null])); ?>" class="btn btn-
primary">Вперед</a>
            </div>
        </div>
    </div>
</div>
</section>

    <?php require_once APPROOT . '/views/template/footer.php'; ?>
</body>

</html>

```

Цей HTML-код представляє розмітку сторінки для відображення каталогу товарів і фільтрів для них.

1. Мета-теги: Включають мета-теги, такі як мова сторінки і посилання на стилі та скрипти.
2. Форма фільтрів: Форма, яка містить поля для фільтрації товарів за категорією, назвою, описом, статтю, розміром, коліром та ціною. Користувач може встановлювати багато варіантів фільтрів за допомогою прапорців.
3. Відображення товарів: Відображення товарів, які відповідають вибраним фільтрам. Для кожного товару відображається зображення, назва, опис, стаття, колір, розмір, ціна і форма для додавання товару в корзину.
4. Пагінація: Кнопки для переходу на попередню або наступну сторінку результатів. Пагінація використовується для розділення результатів на кілька сторінок для зручності навігації.

					<i>РП 07. 22 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		40

5. Навігаційне меню та заголовок: Включає навігаційне меню, заголовок сторінки та верхню панель з логотипом або назвою сайту.
6. Підвал: Включає в себе інформацію про авторські права, посилання на соціальні мережі та іншу додаткову інформацію.

Цей код організовує розмітку і взаємодію з користувачем, дозволяючи йому фільтрувати товари за різними критеріями та додавати їх до корзини для покупки.

1.3.3 Створення контролерів (Controllers)

Створимо класи-контролери, що зв'яжуть класи моделей з HTML-шаблонами графічного інтерфейсу користувача.

У архітектурі MVC контролер відповідає за обробку введення користувача, взаємодію з моделлю та оновлення відображення. Коли користувач взаємодіє з веб-сайтом або додатком, він часто надсилає запити серверу. Контролер приймає ці запити, аналізує їх, взаємодіє з відповідною моделлю для отримання необхідних даних або змін, і відправляє ці дані до відповідного представлення для відображення результатів користувачу.

На рис. 1.17 представлено створені контроллери веб-застосунку.

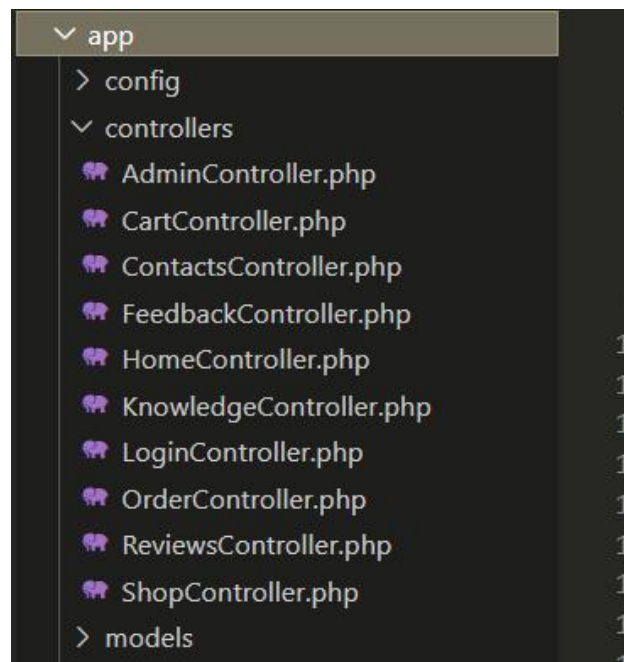


Рисунок 1.17. Створені контроллери веб-застосунку.

Розглянемо код контроллера, що відповідає за каталог товару.

Нижче наведено код контроллеру.

```
<?php
class ShopController extends Controller {
    private $informationModel;
    private $itemModel;
    private $categoryModel;

    public function __construct() {
        $this->informationModel = $this->model('Information');
        $this->itemModel = $this->model('Item');
        $this->categoryModel = $this->model('Category');
    }

    public function index()
    {
        // Initialize filters array with values from GET parameters
        $filters = [
            'category' => isset($_GET['category']) ? $_GET['category'] : null,
            'name' => isset($_GET['name']) ? $_GET['name'] : null,
            'description' => isset($_GET['description']) ? $_GET['description'] :
null,
            'sex' => isset($_GET['sex']) ? $_GET['sex'] : null,
            'size' => isset($_GET['size']) ? $_GET['size'] : null,
            'color' => isset($_GET['color']) ? $_GET['color'] : null,
            'price' => isset($_GET['price']) ? $_GET['price'] : null,
        ];

        // Pagination parameters
        $pageNumber = isset($_GET['page']) ? intval($_GET['page']) : 1;
        $perPage = 10; // Number of items per page

        // Fetch information by ID
        $information = $this->informationModel->selectById(1);

        // Fetch categories
        $categories = $this->categoryModel->selectAll();

        // Fetch items using filters and pagination
        $items = $this->itemModel->search($filters, $pageNumber, $perPage);

        // Prepare data for the view
        $data = [
            'information' => $information,
            'categories' => $categories,
            'items' => $items,
            'pageNumber' => $pageNumber, // Pass pageNumber to the view
            'perPage' => $perPage // Pass perPage to the view
        ];

        // Load the view with items by category
        $this->view('Shop', $data);

        // Display flash message if any
        flash('message');
    }
}
?>
```

					РП 07. 22 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		42

Цей PHP-код представляє контролер для сторінки каталогу товарів.

1. Конструктор: В конструкторі контролера відбувається ініціалізація моделей, необхідних для роботи контролера. Моделі включають модель для інформації про магазин (Information), модель для товарів (Item) та модель для категорій (Category).
2. Метод index: Цей метод відповідає за відображення сторінки каталогу товарів. Він обробляє фільтри та параметри пагінації, взаємодіє з моделями для отримання інформації про товари та категорії, а потім передає ці дані у вигляді до відображення.
3. Фільтри: Збираються значення фільтрів з параметрів GET-запиту і передаються в метод пошуку товарів моделі товарів.
4. Пагінація: Визначає параметри пагінації, такі як поточна сторінка та кількість товарів на сторінці.
5. Отримання інформації: Виконується запит до моделі Information для отримання загальної інформації про магазин та до моделі Category для отримання списку категорій.
6. Отримання товарів: Виконується запит до моделі Item для отримання товарів з врахуванням фільтрів та параметрів пагінації.
7. Підготовка даних: Зібрані дані передаються у вигляді разом з іншими необхідними даними, такими як інформація про магазин, категорії, номер сторінки та кількість товарів на сторінці.
8. Відображення сторінки: Викликається метод відображення (view), щоб відобразити сторінку каталогу разом із зібраними даними.
9. Повідомлення: Відображається повідомлення (якщо є) за допомогою функції flash.

Цей контролер відповідає за обробку запитів на відображення каталогу товарів та взаємодію з моделями та представленням для відображення необхідної інформації.

					<i>РП 07. 22 001. 00 ДП ПЗ</i>	Арк.
						43
Ізм.	Лист	№ докум.	Підпис	Дата		

1.4 Тестування створеного веб-застосунку

Реалізований веб-застосунок містить наступні веб-сторінки: «Головна», «Каталог», «Корзина», «Контакти», «Відгуки», «Скарги та пропозиції» та «Панель адміністратора».

Для коректної роботи сайту проводиться тестування кожної сторінки, на виявлення помилок та недоліків. Тестування веб-застосунку є критично важливим етапом розробки, оскільки забезпечує якість, стабільність і безпеку системи.

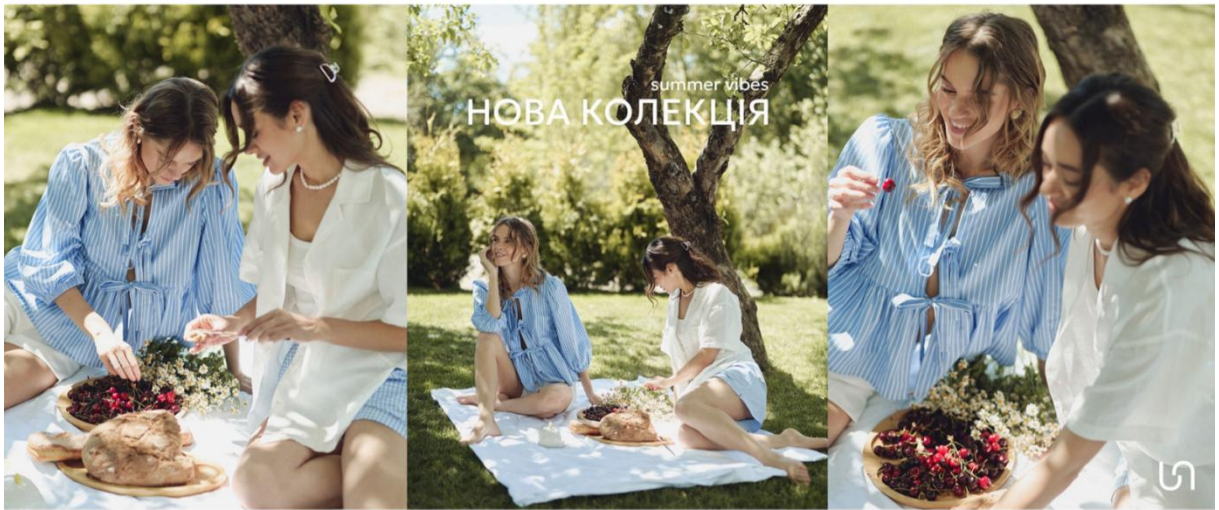
1.4.1 Функціональний огляд головної сторінки

Розглянемо головну сторінку застосунку.

Головна сторінка зустрічає користувача, її основна ціль – залучити відвідувачів та перетворити їх на покупців. Вона повинна представляти найпопулярніші або нові товари та їх категорії, акції та спеціальні пропозиції, що допомагає утримувати увагу користувачів.

На рис. 1.18 зображено головну сторінку застосунку.

					<i>РП 07. 22 001. 00 ДП ПЗ</i>	Арк.
						44
Ізм.	Лист	№ докум.	Підпис	Дата		



ОГОЛОШУЄТЬСЯ АКЦІЯ

Акція буде тривати з 08.06 по 22.06

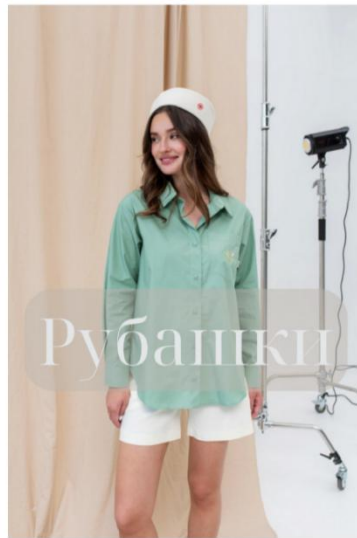


Рисунок 1.18. Головна сторінка застосунку

					РП 07. 22 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		45

1.4.2 Функціональний огляд сторінки каталогу

Розглянемо сторінку каталогу веб-застосунку.

Сторінка каталогу представляє асортименту товарів для потенційних покупців. Основним завданням цієї сторінки є забезпечення зручного та ефективного способу перегляду доступних товарів, орієнтація серед них та сприяння прийняттю рішення щодо покупки.

На рис. 1.19 зображено сторінку каталогу застосунку.

					<i>РП 07. 22 001. 00 ДП ПЗ</i>	Арк.
						46
Ізм.	Лист	№ докум.	Підпис	Дата		



Фільтри

Категорія:

- Штани
- Сукня
- Костюми
- Рубашки

Стать:

- Унісекс
- Жіночий
- Чоловічий

Назва

Опис

Розмір:

- XS
- S
- M
- L
- XL
- XXL

Колір:

- червоний
- оранжевий
- жовтий
- зелений
- блакитний
- синій
- фіолетовий
- чорний
- білий
- сірий

Ціна

Фільтрувати



Рисунок 1.19. Сторінка каталогу застосунку

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 001. 00 ДП ПЗ

На рис. 1.20 зображено сторінку каталогу застосунку.

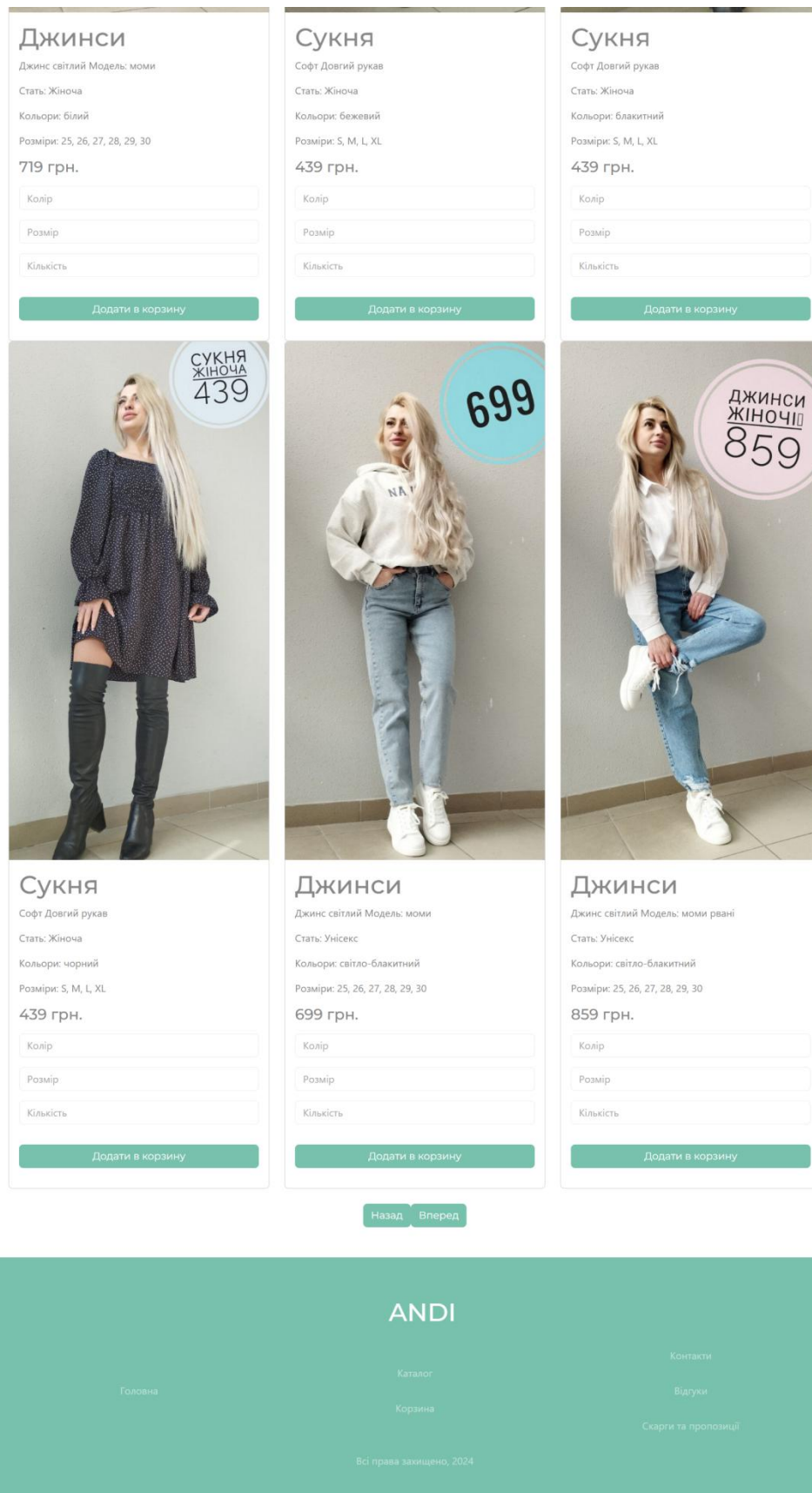


Рисунок 1.20. Сторінка каталогу застосунку

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 001. 00 ДП ПЗ

Арк.

48

1.4.3 Функціональний огляд сторінки корзини

Розглянемо сторінку корзини веб-застосунку.

Корзина надає користувачам переглядати та керувати товаром, який вони збираються придбати. Вона повинна забезпечувати зручний огляд усіх обраних товарів, дозволяючи користувачам змінювати кількість товарів, видаляти непотрібні позиції або продовжити покупки.

На рис. 1.21 зображено сторінку корзини застосунку.

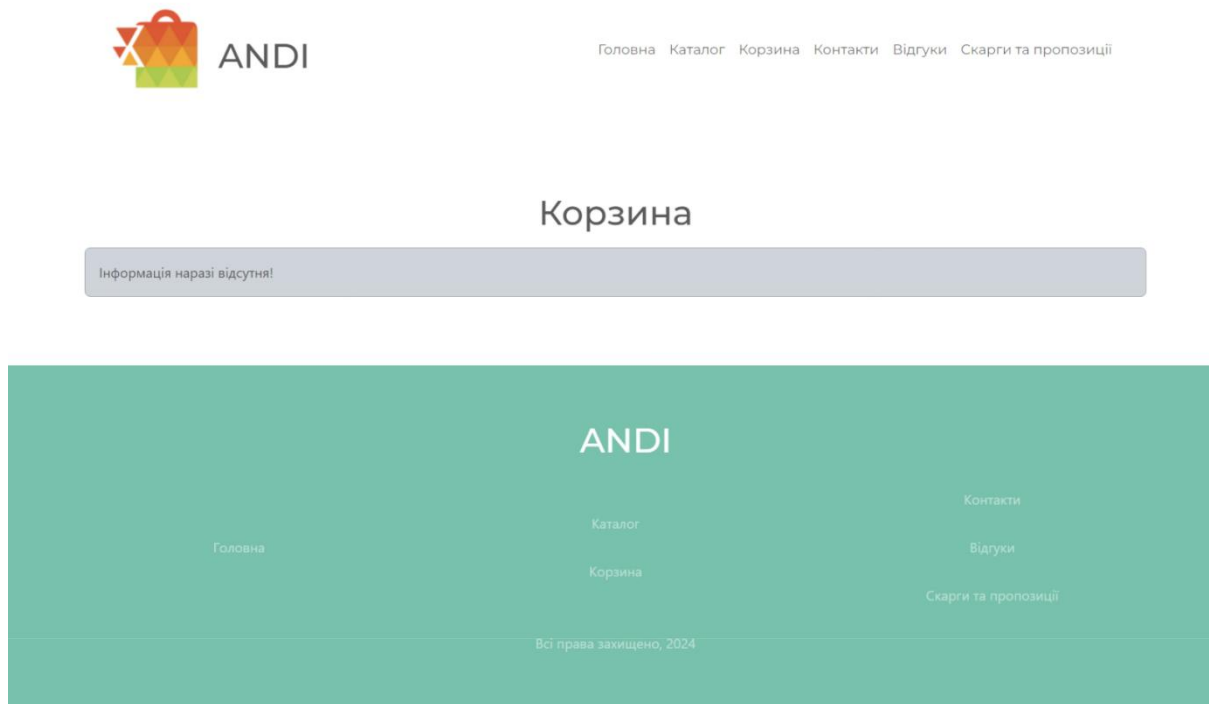


Рисунок 1.21. Сторінка корзини застосунку

1.4.4 Функціональний огляд сторінки контактів

Розглянемо сторінку контактів веб-застосунку.

Сторінка контактів надає користувачам контактну інформацію адміністрації інтернет-магазину. Там можна знайти робочий номер телефону та електронну пошту. Також там може бути написаний розклад роботи магазину та його фізична адреса.

На рис. 1.22 зображено сторінку контактів застосунку.

					<i>РП 07. 22 001. 00 ДП ПЗ</i>	Арк.
						49
Ізм.	Лист	№ докум.	Підпис	Дата		

Контакти

Адреса

м. Львів, вул. Виговського, 5

Телефон

+3800000000000

E-Mail

ANDI_odyag&vzutya@domain.com

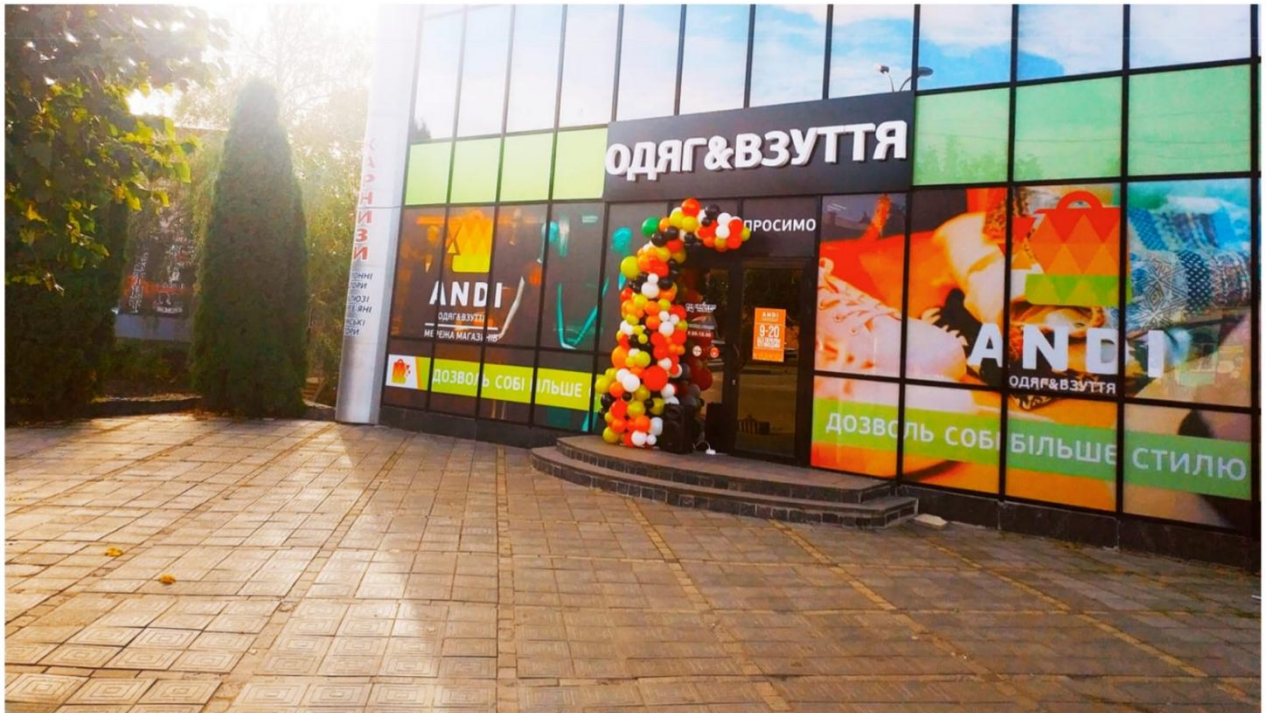


Рисунок 1.22. Сторінка контактів застосунку.

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 001. 00 ДП ПЗ

Арк.

50

1.4.5 Функціональний огляд сторінки відгуків

Розглянемо сторінку відгуків веб-застосунку.

Сторінка відгуків надає користувачам можливість залишити свої враження та надати рекомендації потенційним покупцям. Допомагає адміністрації отримувати зворотній зв'язок та покращувати свою роботу.

На рис. 1.23 зображено сторінку відгуків застосунку.

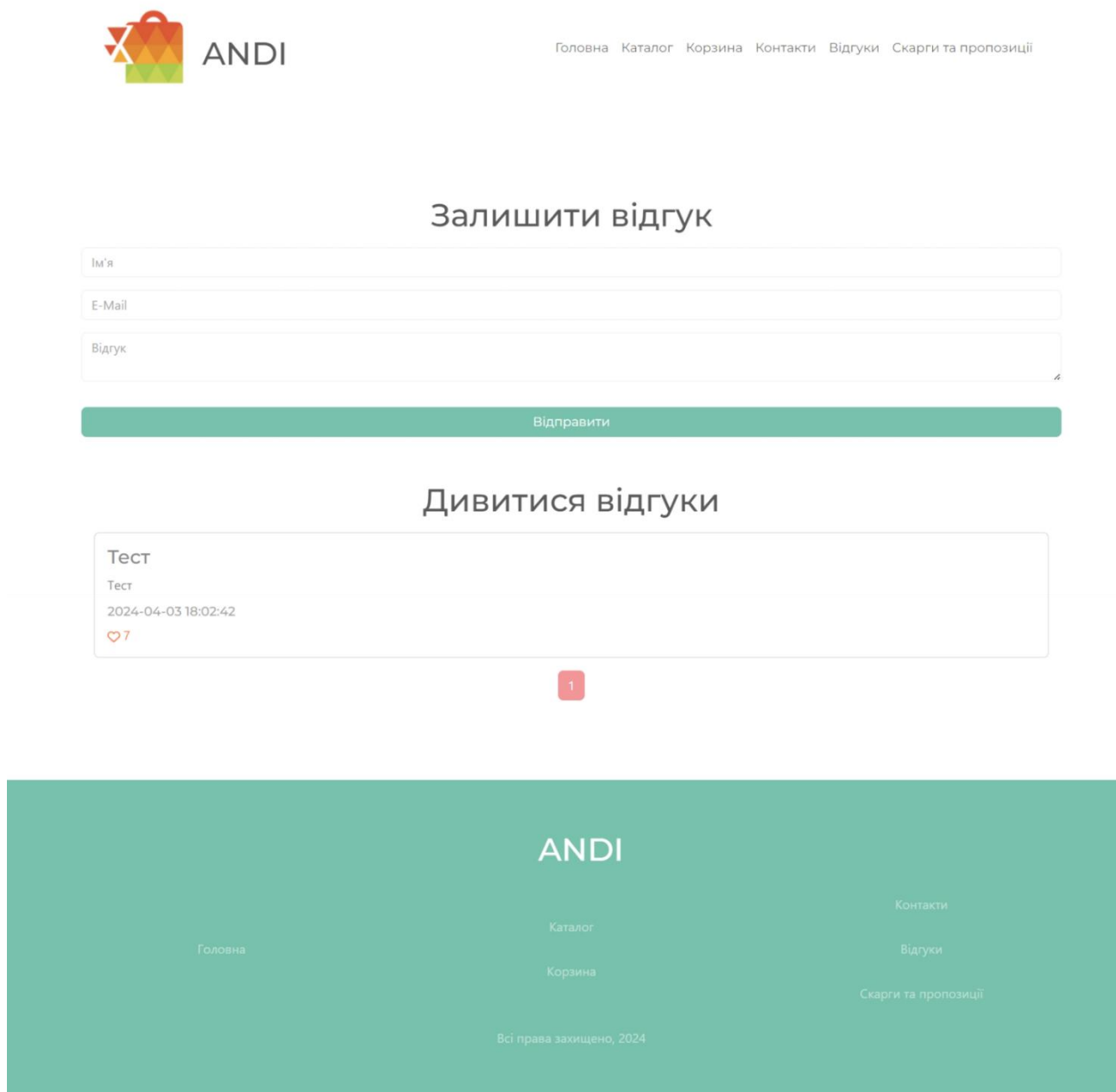


Рисунок 1.23. Сторінка відгуків застосунку

					РП 07. 22 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		51

1.4.6 Функціональний огляд сторінки скарг та пропозицій

Розглянемо сторінку скарг та пропозицій веб-застосунку.

Ціль сторінки "Скарги та пропозиції" в інтернет-магазині полягає в наданні клієнтам можливості висловити свої зауваження, скарги або пропозиції щодо продукції та обслуговування анонімно. Ця сторінка сприяє відкритому діалогу між покупцями і магазином, дозволяючи вчасно реагувати на проблеми та покращувати якість обслуговування.

На рис. 1.23 зображено сторінку скарг та пропозицій застосунку.

The screenshot shows the ANDI website's anonymous feedback form. At the top left is the ANDI logo, and at the top right is a navigation menu with links: Головна, Каталог, Корзина, Контакти, Відгуки, and Скарги та пропозиції. The main heading is "Анонімний зворотній зв'язок". Below it are three input fields: "Ім'я", "E-Mail", and "Відгук". A green "Відправити" button is positioned below the fields. At the bottom of the page, there is a green footer containing the ANDI logo, navigation links (Головна, Каталог, Корзина, Контакти, Відгуки, Скарги та пропозиції), and the text "Всі права захищено, 2024".

Рисунок 1.23. Сторінка Скарг та пропозицій застосунку

					<i>РП 07. 22 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		52

1.4.7 Функціональний огляд сторінки панелі Адміністратора

Розглянемо сторінку панелі Адміністратора веб-застосунку.

Ціль сторінки панелі адміністратора полягає в наданні адміністраторам і менеджерам інструментів для ефективного управління магазином. Ця сторінка дозволяє керувати асортиментом товарів, обробляти замовлення, переглядати звіти про продажі, оновлювати контент на сайті та виконувати інші адміністративні завдання.

На рис. 1.24 зображено сторінку панелі Адміністратора застосунку.

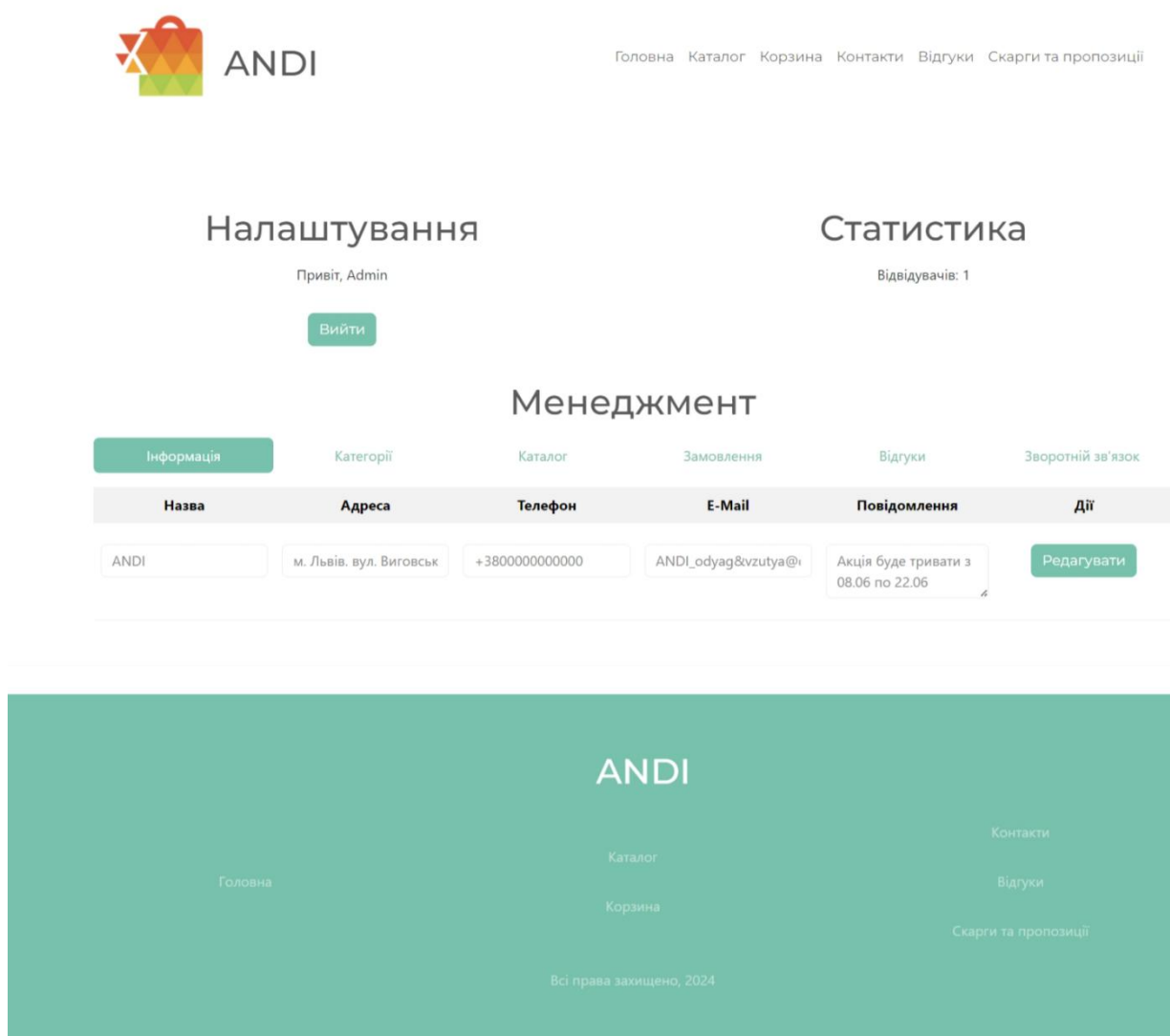


Рисунок 1.24. Сторінка панелі Адміністратора застосунку”

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

В дипломному проекті був розроблений веб-застосунок для просування бренду магазину-одягу. У процесі розробки були враховані різні аспекти економіки, зокрема аналіз ринку, поведінка споживачів і конкурентне середовище. Аналіз показав, що така система є досить важливою у питанні оптимізації процесів в інтернет-магазині одягу, оскільки це забезпечить йому впізнаваність серед конкурентів та ефективне управління. Застосування веб-систем дозволяє оптимізувати робочі процеси для працівників та забезпечити високий рівень обслуговування.

Застосунок дозволяє ефективно взаємодіяти з цільовою аудиторією, використовуючи персоналізовані рекламні кампанії та інтерактивний контент, що сприяє підвищенню продажів і зміцненню позицій бренду на ринку.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуою спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

У табл. 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

					РП 07. 22 002. 00 ДП ПЗ	Арк.
						54
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , усл. машинних командах
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл. 2.2.

Таблиця 2.2. Обсяг ПП

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306
6.00	330
7.00	357

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k = 0,7 \div 0,8$): $T_{ар} = 306 \times 0,8 = 244,8$ (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{ТЗ} = T^a p \leftarrow L_1 \leftarrow K_H \quad (2.1)$$

$$T_{ПП} = T^a p \leftarrow L_2 \leftarrow K_H \quad (2.2)$$

$$T_{РП} = T^a p \leftarrow L_3 \leftarrow K_H \leftarrow K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.3);

K_n – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4);

K_t – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5.).

Таблиця 2.3. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_n
А	Принципово нові ПО	1,75 – 1,2
Б	ПО – розвиток визначеног параметричного ряду	1,0 – 0,8
В	ПО маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПО типовими програмами, %	Значення K_t
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання:

$$T_{ТЗ} = T_a * L_1 * K_n = 214,2 * 0,12 * 0,7 = 17,99 \text{ (люд/годин)}$$

Трудомісткість розробки технічного проекту:

$$T_{ТП} = T_a * L_2 * K_n = 214,2 * 0,11 * 0,7 = 16,49 \text{ (люд/годин)}$$

Трудомісткість розробки робочого проекту:

$$T_{РП} = T_a * L_3 * K_n * K_T = 214,2 * 0,61 * 0,7 * 0,7 = 64,02 \text{ (люд/годин)}$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{ТЗ}=2$ (стр), розробка ТП $N_{ТП}=47$ (стр), розробка робочого проекту $N_{РП}=9$ (стр), пояснювальна записка відповідно $N_{ПЗ}=13$ (стр).

Розрахунок зведений у таблицю 2.6.

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин		
1.ТЗ	$T_{РТЗ}=17,99$	$T_{КК}=0,7*N_{ТЗ}=0,7*2=1,4$	$T_{НК}=0,15*N_{ТЗ}=0,15*2=0,30$
2.Розробка ТП	$T_{РТП}=16,49$	$T_{КК}=0,7*N_{ТП}=0,7*47=32,9$	$T_{НК}=0,15*N_{ТП}=0,15*47=7,05$
3.Розробка РП	$T_{РРП}=64,02$	$T_{КК}=0,7*N_{РП}=0,7*9=6,3$	$T_{НК}=0,15*N_{РП}=0,15*9=1,35$
4.Розробка ПЗ	$T_{ПЗ}=1,5* N_{ПЗ}=1,5*13=19,5$	$T_{КК}=0,7*N_{ТЗ}=0,7*13=9,1$	$T_{НК}=0,15*N_{ПЗ}=0,15*13=1,95$
Усього, в т.ч.:	178,35		
- на розробку	$T_p=118$		
- контроль керівника		$T_{КК}= 49,7$	
- нормоконтроль			$T_{НК}=10,65$

2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, загальні витрати на розробку ПП. Розрахунок основної заробітної плати виконавців приведений у табл. 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2024» встановлено мінімальну заробітну плату у місячному розмірі з 1 квітня 2024 року - 8000 гривень; мінімальну погодинну тарифну ставку – 46,00 грн.

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	118	48	5664
2.Контроль керівника	49,7	74,12	3683,76
3.Нормоконтроль	10,65	74,12	789,37
Усього	-	-	$30= 10137,13$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	80	4	320
Разом	-	-	-	$B_{mi}=320,0$
Транспортно– заготівельні витрати (10%)				$B_{tr_z} = 0,1 \times B_{m1} = 0,1 \times 320,0 = 32,0$
Усього				$B_m = B_{mi} + B_{tr_z} = 352,0$

Ізм.	Лист	№ докум.	Підпис	Дата

РП 07. 22 002. 00 ДП ПЗ

Арк.

58

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9.

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	352,0	V_m (див. табл. 2.8)
2. Основна заробітна плата	10137,13	Z_o (див. табл. 2.7)
3. Додаткова заробітна плата	1013,71	$Z_d = 0,15 \times Z_o = 10137,13 \times 0,1$
4. Відрахування до єдиного фонду соціального внеску	2453,18	$V_{с.с.в.} = 0,22 \times (Z_o + Z_d) = 0,22 \times (10137,13 + 1013,71)$
5. Накладні витрати	6082,27	$V_{нак.} = 0,6 \times Z_o = 0,6 \times 10137,13$
6. Повна собівартість	20038,29	$C_{пов} = V_m + Z_o + Z_d + V_{с.с.в.} + V_{нак.} = 352,0 + 10137,13 + 1013,71 + 2453,18 + 6082,27$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{п} * P) / 100 = (20038,29 * 10) / 100 = 2003,82 \text{ грн} \quad (2.4)$$

Де p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{п} + П = 20038,29 + 2003,82 = 22042,11 \text{ грн} \quad (2.5)$$

Податок на додану вартість визначаємо по наступній формулі:

$$ПДВ = 0,2 * Ц_o = 22042,11 * 0,2 = 4408,42 \text{ грн}; \quad (2.6)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + ПДВ = 22042,11 + 4408,42 = 26450,53 \text{ грн} \quad (2.7)$$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Вступ

Охорона праці – це система, що забезпечує збереження життя, здоров'я та працездатності працівників під час трудової діяльності. Вона включає в себе правові, соціально-економічні, організаційно-технічні, санітарно-гігієнічні, лікувально-профілактичні та інші заходи. Ця система спрямована на зниження ризиків травматизму, професійних захворювань та нещасних випадків на роботі. Важливою метою охорони праці є також підвищення продуктивності праці та зниження текучості кадрів.

Заходи охорони праці спрямовані на створення безпечного, комфортного та здорового робочого середовища. Дотримання цих заходів дозволяє знизити ризики професійних захворювань та нещасних випадків, забезпечуючи працівникам можливість зосередитися на своїх професійних обов'язках.

3.2 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу

У розробці інтернет-магазину одягу можуть виникати небезпечні та шкідливі чинники, що впливають на здоров'я та ефективність працівників. Визначення та аналіз цих чинників є ключовими для забезпечення безпечних умов праці та мінімізації ризиків. Основні небезпечні та шкідливі фактори, які варто враховувати:

Зорове стомлення та професійні захворювання зору: Тривала робота за комп'ютером може призводити до перенапруження очей, сухості, подразнення та погіршення зору. Недостатнє або неправильне освітлення лише посилює ці проблеми.

Опорно-рухові захворювання: Неправильна організація робочого місця, відсутність ергономічних меблів і тривале сидіння в незручній позі можуть спричинити болі в спині, шиї та зап'ястях, а також розвиток хронічних захворювань опорно-рухового апарату.

					РП 07. 22 003. 00 ДП ПЗ	Арк.
						60
Ізм.	Лист	№ докум.	Підпис	Дата		

Недостатня вентиляція та якість повітря: Погана вентиляція призводить до накопичення вуглекислого газу, пилу та інших шкідливих речовин у повітрі, що може спричинити головні болі, втому, алергічні реакції та зниження продуктивності праці.

Вплив шуму: Високий рівень шуму від офісної техніки та зовнішніх джерел негативно впливає на концентрацію, спричиняє стрес і може призвести до порушень слуху та психологічного дискомфорту.

Електромагнітне випромінювання та електробезпека: Електронне обладнання, таке як комп'ютери, монітори та інші пристрої, генерує електромагнітне випромінювання, яке може негативно впливати на здоров'я працівників. Несправне обладнання або неправильне його використання підвищує ризик електротравм та коротких замикань.

3.3 Гігієнічні вимоги до виробничого середовища

3.3.1 Вимоги до приміщення

Площа приміщення на одне робоче місце повинна складати не менше 6,0 квадратних метрів. Робочі місця мають бути розташовані на відстані не менше 1 метра від стіни з вікном і 1,4 метра від звичайної стіни. Відстань між бічними поверхнями комп'ютерів має бути не менше 1,2 метра, а відстань між тильною поверхнею одного комп'ютера і екраном іншого - не менше 2,5 метра.

Недозволено організовувати робочі місця в підвальних або цокольних приміщеннях будівель. Також забороняється використання полімерних матеріалів, які виділяють у повітря шкідливі хімічні речовини, таких як деревинно-стружкові плити, мийні шпалери, синтетичні рулонні матеріали та інші. Підлога повинна мати матове покриття і рівну, неслизьку поверхню з антистатичними властивостями.

Ці вимоги спрямовані на забезпечення комфортних та безпечних умов праці для працівників, зменшення ризику виникнення травм або професійних захворювань та підвищення їх продуктивності.

					РП 07. 22 003. 00 ДП ПЗ	Арк.
						61
Ізм.	Лист	№ докум.	Підпис	Дата		

3.3.2 Освітлення

Натуральне освітлення в офісних приміщеннях важливе для забезпечення комфортних умов праці та збереження зору працівників. Для цього встановлюється вимога до коефіцієнта природної освітленості (КПО), який має становити не менше 1,5%. Низький рівень природного освітлення може призвести до перенапруження зору під час тривалої роботи за комп'ютером або з документами.

Для регулювання рівня природного освітлення рекомендується використовувати жалюзі, які дозволяють контролювати вхід світла в приміщення. При розміщенні робочого місця з ПК необхідно уникати попадання прямого сонячного світла в очі працівників.

Щодо штучного освітлення, рекомендується використовувати систему загального рівномірного освітлення для забезпечення однакового розподілу світла в приміщенні. Використання світильників без розсіювачів або екрануючих сіток не допускається, оскільки це може спричинити нерівномірне освітлення та викликати відблиски на екранах комп'ютерів.

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300–500 люксів, що є оптимальним для зручної роботи і збереження зору працівників.

3.3.3 Шум

Максимально допустимий рівень шуму: Для офісних приміщень, де виконуються розумові роботи або роботи, що потребують зосередженості, рівень шуму не повинен перевищувати 50 дБА.

Захист від шуму: Використання звукопоглинальних матеріалів для стін, стель і підлог для зменшення відбиття звуку і зниження рівня шуму. Використання перегородок та екранів між робочими місцями для зменшення поширення шуму.

Організаційні заходи: Планування робочого простору з урахуванням потреб мінімізації шуму, зокрема розміщення робочих місць подалі від джерел шуму.

3.3.4 Гігієнічні вимоги

У приміщеннях, де здійснюється робота з комп'ютерами, необхідно щодня проводити вологе прибирання, щоб запобігти накопиченню пилу на підлозі та меблях. Це важливо для підтримання чистоти і здорового мікроклімату, а також для зменшення ризику виникнення алергічних реакцій у працівників.

3.3.5 Вентиляція та мікроклімат

Приміщення для роботи з персональними комп'ютерами повинні бути оснащені системами опалення, кондиціонування повітря або припливно-втяжною вентиляцією. В робочих зонах мають підтримуватися оптимальні параметри мікроклімату: температура повітря повинна бути в межах 22–25°C, відносна вологість — 40–60%, а швидкість руху повітря — не більше 0,1 м/с. Якщо зазначені параметри мікроклімату в офісних приміщеннях не дотримуються, робочий день працівників слід скоротити мінімум на 10%.

3.3.6 Електробезпека

Електробезпека є важливим аспектом охорони праці в будь-якій сфері, особливо в умовах інтенсивного використання комп'ютерної техніки та іншого електрообладнання. Дотримання правил електробезпеки допомагає запобігти нещасним випадкам та забезпечити безпечні умови праці для всіх співробітників.

Організація робочих місць: Усі робочі місця повинні бути обладнані справними електророзетками та подовжувачами, які відповідають нормативам. Розетки повинні бути розташовані таким чином, щоб уникнути пошкодження кабелів та можливості їх перегріву.

Використання обладнання: Використання тільки сертифікованих та перевірених електроприладів, що відповідають стандартам безпеки. Регулярне проведення технічного обслуговування та перевірки справності обладнання.

					РП 07. 22 003. 00 ДП ПЗ	Арк.
						63
Ізм.	Лист	№ докум.	Підпис	Дата		

Заземлення та ізоляція: Усі електроприлади повинні мати належне заземлення для запобігання ураженню електричним струмом, а ізоляція електропроводки та кабелів повинна бути без пошкоджень та відповідати встановленим нормам.

Запобігання перевантаженню мережі: Контроль за навантаженням на електромережу, уникнення підключення великої кількості пристроїв до одного розеткового блоку. Використання автоматичних вимикачів та запобіжників для запобігання перевантаженням та коротким замиканням.

3.3.7 Робоче місце

Забезпечення безпечних та здорових умов на робочих місцях є важливим аспектом охорони праці. Дотримання правил і вимог забезпечує зменшення ризику травматизму та покращує загальний стан здоров'я працівників. Основні аспекти, які слід враховувати щодо робочого місця, включають:

1. Ергономіка: Забезпечення оптимальних умов для працівників, таких як правильне розташування меблів, екранів комп'ютерів, клавіатури та миші, що підтримує зручне та безболісне положення тіла під час роботи.
2. Вентиляція: Забезпечення належної вентиляції, щоб уникнути накопичення шкідливих речовин у повітрі та підтримати свіжість та комфорт.
3. Освітлення: Забезпечення належного рівня освітлення на робочому місці для забезпечення комфортної та продуктивної роботи, а також для запобігання втомі та напруженню зору.
4. Безпека: Забезпечення наявності необхідного обладнання та інструкцій з безпеки для уникнення травм та нещасних випадків на робочому місці.
5. Чистота та порядок: Регулярне прибирання та підтримання чистоти на робочому місці для запобігання накопиченню бруду та бактерій, що можуть спричинити захворювання

3.4 Пожежна безпека

Пожежна безпека в офісному приміщенні є критично важливим аспектом забезпечення загальної безпеки працівників та майна. Перш за все, офіс має бути обладнаний системою пожежної сигналізації, яка оперативно реагує на задимлення або підвищення температури. Ця система повинна бути регулярно перевірена та обслуговувана для забезпечення її безперебійної роботи.

Важливим елементом пожежної безпеки є наявність вогнегасників у легкодоступних місцях. Кількість та тип вогнегасників повинні відповідати розмірам приміщення та можливим джерелам займання. Вуглекислотні вогнегасники є ефективними засобами боротьби з пожежею, вони гасять загорання за рахунок значного охолодження зони горіння струминою вуглекислоти CO₂, яка, випаровуючись, перетворюється на вуглекислий газ. Працівники офісу повинні бути навчені правилам користування вогнегасниками та іншими засобами пожежогасіння.

Евакуаційні шляхи та виходи повинні бути чітко позначені та вільні від будь-яких перешкод. На кожному поверсі офісного приміщення необхідно розмістити плани евакуації, які вказують маршрути виходу у разі пожежі. Ці плани мають бути легко доступні та зрозумілі для всіх працівників.

Особлива увага приділяється електробезпеці, оскільки електричні прилади та проводка можуть бути потенційними джерелами займання. Всі електроприлади повинні бути сертифіковані та регулярно перевірятися на справність. Перевантаження електромережі слід уникати, а використання подовжувачів має бути мінімізоване.

Паління в офісі повинно бути суворо заборонене або обмежене спеціально відведеними місцями поза будівлю. Це значно знижує ризик виникнення пожежі через недбало кинуті недопалки. Важливо також передбачити наявність засобів для первинного пожежогасіння, таких як пожежні крани або ковдри.

Навчання та тренування з пожежної безпеки є невід'ємною частиною профілактичних заходів. Регулярно проводити інструктажі та практичні заняття

					РП 07. 22 003. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		65

з евакуації, що допомагають працівникам діяти чітко і злагоджено у разі надзвичайної ситуації.

Окрім того, важливо забезпечити контроль за справністю всіх засобів пожежогасіння та сигналізації, проводячи регулярні перевірки та технічне обслуговування. Впровадження цих заходів допоможе створити безпечні умови роботи та мінімізувати ризики, пов'язані з пожежною небезпекою в офісі.

					<i>РП 07. 22 003. 00 ДП ПЗ</i>	Арк.
						66
Ізм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

У дипломному проєкті було створено інтернет-магазин одягу для просування бренду. Проєкт включає в себе всі необхідні компоненти для успішного функціонування онлайн-магазину. Зокрема, у створеному програмному продукті реалізований широкий функціонал, який дозволяє користувачам додавати товари до кошика, оформляти замовлення, а адміністраторам - додавати нові товари до каталогу за допомогою спеціальної панелі адміністратора.

При розробці програмного продукту використовувався комплексний інструментарій, який включає HTML для створення структури веб-сторінок, CSS для стилізації, Bootstrap для забезпечення адаптивності та зручності користування, JavaScript та jQuery для динамічної взаємодії з користувачем. Серверна частина була реалізована за допомогою PHP, а для роботи з базою даних використовувалися SQL та MySQL.

У пояснювальній записці розглянуті всі питання, передбачені технічним завданням на дипломне проєктування. Проведено детальний аналіз предметної області, що включає дослідження ринку одягу, аналіз конкурентів та виявлення потреб цільової аудиторії. Докладно описано всі технології та засоби, які використовувалися при створенні проєкту, з поясненням вибору кожного з них та їхніх переваг. Також у пояснювальній записці проведено розрахунок економічної ефективності створення та впровадження створеного проєкту. Зокрема, оцінено потенційні витрати на розробку та впровадження, прогнозований прибуток, період окупності та аналіз ризиків. Це дозволяє оцінити доцільність і перспективи проєкту з економічної точки зору. Окрім технічних та економічних аспектів, розглянуті також питання охорони праці, що є важливим аспектом при розробці та впровадженні будь-якого програмного продукту. Наведений перелік використаних джерел включає в себе літературу, інтернет-ресурси та нормативні документи, які були використані під час роботи над проєктом.

					<i>РП 07. 22 000. 00 ДП ПЗ</i>	Арк.
						67
Ізм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. ARGO [Веб-сайт]. URL: <https://argo.com.ua/ua/>.
2. LeBoutique [Веб-сайт]. URL: <https://leboutique.com/uk>.
3. Colin's [Веб-сайт]. URL: <https://www.colins.ua/>.
4. Stack Overflow [Веб-сайт]. URL: <https://survey.stackoverflow.co/2023/#technology>.
5. О. С. Бунке. Серверні WEB-технології: Навчальний посібник. – «КПІ ім. Ігоря Сікорського», 2023.
6. К. В. Двірничук, Д. О. Вацек. Веб-програмування та веб-дизайн: Навчальний посібник. – «Чернівецький національний університет імені Юрія Федьковича», 2022.
7. С. І. Доценко. Організація та системи керування базами даних: Навчальний посібник. – «Український державний університет залізничного транспорту», 2023.

					<i>РП 07. 22 000. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		68

ДОДАТОК А. Програмний код основної логіки веб-застосунку

```
// ItemModel.php

<<?php
class ItemModel
{
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function selectAll()
    {
        // Query
        $this->db->query('SELECT * FROM items');

        // Execute
        $results = $this->db->resultSet();

        // Return result
        return $results;
    }

    public function selectById($id)
    {
        // Query
        $this->db->query('SELECT * FROM items WHERE id = :id');

        // Bind value
        $this->db->bind(':id', $id);

        // Execute
        $results = $this->db->single();

        // Return result
        return $results;
    }

    public function search($filters, $pageNumber = 1, $perPage = 10) {
        // Base query
        $query = 'SELECT * FROM items WHERE 1';

        // Prepare an array to hold values to bind
        $bindValue = [];

        // Check each filter and add conditions to the query accordingly
        if (is_array($filters) && !empty($filters['category'])) {
            $query .= ' AND category IN (' . implode(',', array_fill(0,
count($filters['category']), '?')) . ')';
            $bindValue = array_merge($bindValue, $filters['category']);
        }

        if (is_array($filters) && !empty($filters['name'])) {
            $query .= ' AND name LIKE ?';
        }
    }
}
```

```

        $bindValue[] = '%' . $filters['name'] . '%';
    }

    if (is_array($filters) && !empty($filters['description'])) {
        $query .= ' AND description LIKE ?';
        $bindValue[] = '%' . $filters['description'] . '%';
    }

    if (is_array($filters) && !empty($filters['sex'])) {
        $query .= ' AND sex IN (' . implode(',', array_fill(0,
count($filters['sex'], '?')) . ')';
        $bindValue = array_merge($bindValue, $filters['sex']);
    }

    if (!empty($filters['size'])) {
        $sizeConditions = [];
        foreach ($filters['size'] as $size) {
            $sizeConditions[] = 'size LIKE ?';
            $bindValue[] = '%' . $size . '%';
        }
        $query .= ' AND (' . implode(' OR ', $sizeConditions) . ')';
    }

    if (!empty($filters['color'])) {
        $colorConditions = [];
        foreach ($filters['color'] as $color) {
            $colorConditions[] = 'color LIKE ?';
            $bindValue[] = '%' . $color . '%';
        }
        $query .= ' AND (' . implode(' OR ', $colorConditions) . ')';
    }

    if (is_array($filters) && !empty($filters['price'])) {
        $query .= ' AND price <= ?'; // Adjusted for items with price less than or
equal to the specified price
        $bindValue[] = $filters['price'];
    }

    // Pagination
    $offset = ($pageNumber - 1) * $perPage;
    $query .= ' LIMIT ' . $perPage . ' OFFSET ' . $offset;

    // Prepare the query
    $this->db->query($query);

    // Bind values
    foreach ($bindValue as $key => $value) {
        $this->db->bind($key + 1, $value); // PDO parameter index starts from 1
    }

    // Execute
    $results = $this->db->resultSet();

    // Return result
    return $results;
}

public function insert($data)
{
    // Query

```

```

        $this->db->query('INSERT INTO items (category, photo, name, description, sex,
size,                color,                price,                active)
VALUES(:category, :photo, :name, :description, :sex, :size, :color, :price, :active)');

    // Bind values
    $this->db->bind(':category', $data['category']);
    $this->db->bind(':photo', $data['photo']);
    $this->db->bind(':name', $data['name']);
    $this->db->bind(':description', $data['description']);
    $this->db->bind(':sex', $data['sex']);
    $this->db->bind(':size', $data['size']);
    $this->db->bind(':color', $data['color']);
    $this->db->bind(':price', $data['price']);
    $this->db->bind(':active', $data['active']);

    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}

public function update($data)
{
    // Query
    $this->db->query('UPDATE items SET category = :category, photo = :photo, name
= :name, description = :description, sex = :sex, size = :size, color = :color, price
= :price, active = :active WHERE id = :id');

    // Bind values
    $this->db->bind(':id', $data['id']);
    $this->db->bind(':category', $data['category']);
    $this->db->bind(':photo', $data['photo']);
    $this->db->bind(':name', $data['name']);
    $this->db->bind(':description', $data['description']);
    $this->db->bind(':sex', $data['sex']);
    $this->db->bind(':size', $data['size']);
    $this->db->bind(':color', $data['color']);
    $this->db->bind(':price', $data['price']);
    $this->db->bind(':active', $data['active']);

    // Execute
    if($this->db->execute())
    {
        return true;
    }
    else
    {
        return false;
    }
}

public function delete($data)
{
    // Query
    $this->db->query('DELETE FROM items WHERE id = :id');

    // Bind values

```

```

        $this->db->bind(':id', $data['id']);

        // Execute
        if($this->db->execute())
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
?>

// OrderModel.php

<?php
class OrderModel
{
    private $db;

    public function __construct()
    {
        $this->db = new Database;
    }

    public function selectAll()
    {
        // Query
        $this->db->query('SELECT * FROM orders');

        // Execute
        $results = $this->db->resultSet();

        // Return result
        return $results;
    }

    public function insert($data)
    {
        // Query
        $this->db->query('INSERT INTO orders (list, price, phone, email, address,
message) VALUES(:list, :price, :phone, :email, :address, :message)');

        // Bind values
        $this->db->bind(':list', $data['list']);
        $this->db->bind(':price', $data['price']);
        $this->db->bind(':phone', $data['phone']);
        $this->db->bind(':email', $data['email']);
        $this->db->bind(':address', $data['address']);
        $this->db->bind(':message', $data['message']);

        // Execute
        if($this->db->execute())
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

```

    }

    public function update($data)
    {
        // Query
        $this->db->query('UPDATE orders SET list = :list, price = :price, phone = :phone,
email = :email, address = :address, message = :message, status = :status WHERE id
= :id');

        // Bind values
        $this->db->bind(':id', $data['id']);
        $this->db->bind(':list', $data['list']);
        $this->db->bind(':price', $data['price']);
        $this->db->bind(':phone', $data['phone']);
        $this->db->bind(':email', $data['email']);
        $this->db->bind(':address', $data['address']);
        $this->db->bind(':message', $data['message']);
        $this->db->bind(':status', $data['status']);

        // Execute
        if($this->db->execute())
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public function delete($data)
    {
        // Query
        $this->db->query('DELETE FROM orders WHERE id = :id');

        // Bind values
        $this->db->bind(':id', $data['id']);

        // Execute
        if($this->db->execute())
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
?>

```

// ShopController.php

```

<?php
class ShopController extends Controller {
    private $informationModel;
    private $itemModel;
    private $categoryModel;

    public function __construct() {
        $this->informationModel = $this->model('Information');
        $this->itemModel = $this->model('Item');
    }
}

```

```

        $this->categoryModel = $this->model('Category');
    }

    public function index()
    {
        // Initialize filters array with values from GET parameters
        $filters = [
            'category' => isset($_GET['category']) ? $_GET['category'] : null,
            'name' => isset($_GET['name']) ? $_GET['name'] : null,
            'description' => isset($_GET['description']) ? $_GET['description'] : null,
            'sex' => isset($_GET['sex']) ? $_GET['sex'] : null,
            'size' => isset($_GET['size']) ? $_GET['size'] : null,
            'color' => isset($_GET['color']) ? $_GET['color'] : null,
            'price' => isset($_GET['price']) ? $_GET['price'] : null,
        ];

        // Pagination parameters
        $pageNumber = isset($_GET['page']) ? intval($_GET['page']) : 1;
        $perPage = 10; // Number of items per page

        // Fetch information by ID
        $information = $this->informationModel->selectById(1);

        // Fetch categories
        $categories = $this->categoryModel->selectAll();

        // Fetch items using filters and pagination
        $items = $this->itemModel->search($filters, $pageNumber, $perPage);

        // Prepare data for the view
        $data = [
            'information' => $information,
            'categories' => $categories,
            'items' => $items,
            'pageNumber' => $pageNumber, // Pass pageNumber to the view
            'perPage' => $perPage // Pass perPage to the view
        ];

        // Load the view with items by category
        $this->view('Shop', $data);

        // Display flash message if any
        flash('message');
    }
}
?>

```

```

<?php
class CartController extends Controller {
    private $itemModel;
    private $informationModel;

    public function __construct() {
        $this->itemModel = $this->model('Item');
        $this->informationModel = $this->model('Information');

        if(session_status() == PHP_SESSION_NONE) {
            session_start();
        }
    }
}

```

```

    }
}

public function index()
{
    // Fetch information
    $information = $this->informationModel->selectById(1);

    // Get cart items from session
    $cart = isset($_SESSION['cart']) ? $_SESSION['cart'] : [];

    // Prepare data to pass to the view
    $data = [
        'information' => $information,
        'cart' => $cart, // Add cart items to data array
    ];

    // Load the view with data
    $this->view('Cart', $data);

    // Display flash message (if any)
    flash('message');
}

public function addToCart($itemId) {
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $_POST = filter_input_array(INPUT_POST, FILTER_SANITIZE_FULL_SPECIAL_CHARS);

        // Validate and sanitize input data
        $color = trim($_POST['color']);
        $size = trim($_POST['size']);
        $quantity = filter_var($_POST['quantity'], FILTER_VALIDATE_INT);

        $item = $this->itemModel->selectById($itemId);

        // Initialize cart if it doesn't exist
        if (!isset($_SESSION['cart'])) {
            $_SESSION['cart'] = [];
        }

        // Check if the item already exists in the cart
        $found = false;
        foreach ($_SESSION['cart'] as &$cartItemInSession) {
            // Check if the item already exists based on ID, color, and size
            if ($cartItemInSession['id'] == $item->id &&
                $cartItemInSession['color'] == $color &&
                $cartItemInSession['size'] == $size) {
                // Update quantity
                $cartItemInSession['quantity'] += $quantity;
                // If quantity is greater than 1, multiply price by quantity
                if ($cartItemInSession['quantity'] > 1) {
                    $cartItemInSession['price'] *= $cartItemInSession['quantity'];
                }
                $found = true;
                break;
            }
        }

        if (!$found) {
            // Create new cart item
            $cartItem = [
                'id' => $item->id,

```

```

        'name' => $item->name,
        'price' => $item->price,
        'color' => $color,
        'size' => $size,
        'quantity' => $quantity
    ];
    // If quantity is greater than 1, multiply price by quantity
    if ($cartItem['quantity'] > 1) {
        $cartItem['price'] *= $cartItem['quantity'];
    }

    $_SESSION['cart'][] = $cartItem;
}

// Calculate total cart price
$totalPrice = 0;
foreach ($_SESSION['cart'] as $cartItem) {
    $totalPrice += $cartItem['price'];
}

// Store total cart price in session
$_SESSION['total'] = $totalPrice;

flash('message', 'Додано в корзину');
redirect('shop');
}
}

public function deleteItem($itemId) {
    // Ensure the item exists in the cart
    if (isset($_SESSION['cart'])) {
        // Loop through cart items
        foreach ($_SESSION['cart'] as $index => $cartItem) {
            // Check if the item ID matches
            if ($cartItem['id'] == $itemId) {
                // Remove the item from the cart
                unset($_SESSION['cart'][$index]);
                break;
            }
        }
        flash('message', 'Товар видалено з корзини');
        redirect('cart');
    }
}
}
?>

```

// ShopView.php

```

<!DOCTYPE html>
<html data-bs-theme="light" lang="en">

<head>
    <?php require_once APPROOT . '/views/template/meta.php'; ?>
    <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
</head>

<body class="bg-white">
    <?php require_once APPROOT . '/views/template/scripts.php'; ?>
    <?php require_once APPROOT . '/views/template/navmenu.php'; ?>
    <?php require_once APPROOT . '/views/template/header.php'; ?>

```

```

<section class="p-5">
  <div class="container">
    <h1 class="text-center m-3">Фільтри</h1>
    <div class="row justify-content-center align-items-start m-3">
      <form method="GET" action="">
        <div class="row">
          <div class="col-md-4 mb-3">
            <label for="category">Категорія:</label><br>
            <?php foreach ($data['categories'] as $category) : ?>
              <input type="checkbox" name="category[]" value="<?php
echo $category->id; ?>" <?php if (isset($_GET['category']) && in_array($category->id,
$_GET['category'])) echo 'checked'; ?>>
              <?php echo $category->name; ?><br>
            <?php endforeach; ?>
          </div>
          <div class="col-md-4 mb-3">
            <input type="text" class="form-control" name="name"
placeholder="Назва" value="<?php echo isset($_GET['name']) ? $_GET['name'] : ''; ?>"
          </div>
          <div class="col-md-4 mb-3">
            <input type="text" class="form-control" name="description"
placeholder="Опис" value="<?php echo isset($_GET['description']) ? $_GET['description'] :
''; ?>"
          </div>
          <div class="col-md-4 mb-3">
            <label>Стать:</label><br>
            <input type="checkbox" name="sex[]" value="1" <?php if
(isset($_GET['sex']) && in_array('1', $_GET['sex'])) echo 'checked'; ?>> Унісекс<br>
            <input type="checkbox" name="sex[]" value="2" <?php if
(isset($_GET['sex']) && in_array('2', $_GET['sex'])) echo 'checked'; ?>> Жіночий<br>
            <input type="checkbox" name="sex[]" value="3" <?php if
(isset($_GET['sex']) && in_array('3', $_GET['sex'])) echo 'checked'; ?>> Чоловічий<br>
          </div>
          <div class="col-md-4 mb-3">
            <label>Розмір:</label><br>
            <input type="checkbox" name="size[]" value="XS" <?php if
(isset($_GET['size']) && in_array('XS', $_GET['size'])) echo 'checked'; ?>> XS<br>
            <input type="checkbox" name="size[]" value="S" <?php if
(isset($_GET['size']) && in_array('S', $_GET['size'])) echo 'checked'; ?>> S<br>
            <input type="checkbox" name="size[]" value="M" <?php if
(isset($_GET['size']) && in_array('M', $_GET['size'])) echo 'checked'; ?>> M<br>
            <input type="checkbox" name="size[]" value="L" <?php if
(isset($_GET['size']) && in_array('L', $_GET['size'])) echo 'checked'; ?>> L<br>
            <input type="checkbox" name="size[]" value="XL" <?php if
(isset($_GET['size']) && in_array('XL', $_GET['size'])) echo 'checked'; ?>> XL<br>
            <input type="checkbox" name="size[]" value="XXL" <?php if
(isset($_GET['size']) && in_array('XXL', $_GET['size'])) echo 'checked'; ?>> XXL<br>
          </div>
          <div class="col-md-4 mb-3">
            <label>Колір:</label><br>
            <input type="checkbox" name="color[]" value="червоний" <?php
if (isset($_GET['color']) && in_array('червоний', $_GET['color'])) echo 'checked'; ?>>
червоний<br>
            <input type="checkbox" name="color[]" value="оранжевий"
<?php if (isset($_GET['color']) && in_array('оранжевий', $_GET['color'])) echo
'checked'; ?>> оранжевий<br>
            <input type="checkbox" name="color[]" value="жовтий" <?php
if (isset($_GET['color']) && in_array('жовтий', $_GET['color'])) echo 'checked'; ?>>
жовтий<br>
            <input type="checkbox" name="color[]" value="зелений" <?php
if (isset($_GET['color']) && in_array('зелений', $_GET['color'])) echo 'checked'; ?>>
зелений<br>

```

```

        <input type="checkbox" name="color[]" value="блакитний"
<?php if (isset($_GET['color']) && in_array('блакитний', $_GET['color'])) echo
'checked'; ?>> блакитний<br>
        <input type="checkbox" name="color[]" value="синій" <?php if
(isset($_GET['color']) && in_array('синій', $_GET['color'])) echo 'checked'; ?>>
синій<br>
        <input type="checkbox" name="color[]" value="фіолетовий"
<?php if (isset($_GET['color']) && in_array('фіолетовий', $_GET['color'])) echo
'checked'; ?>> фіолетовий<br>
        <input type="checkbox" name="color[]" value="чорний" <?php
if (isset($_GET['color']) && in_array('чорний', $_GET['color'])) echo 'checked'; ?>>
чорний<br>
        <input type="checkbox" name="color[]" value="білий" <?php if
(isset($_GET['color']) && in_array('білий', $_GET['color'])) echo 'checked'; ?>>
білий<br>
        <input type="checkbox" name="color[]" value="сірий" <?php if
(isset($_GET['color']) && in_array('сірий', $_GET['color'])) echo 'checked'; ?>>
сірий<br>
    </div>
    <div class="col-md-4 mb-3">
        <input type="number" class="form-control" name="price"
placeholder="Ціна" value="<?php echo isset($_GET['price']) ? $_GET['price'] : ''; ?>">
    </div>
    </div>
    <button type="submit" class="btn btn-primary">Фільтрувати</button>
</form>
</div>
<?php if (!empty($data['items'])): ?>
    <div class="row row-cols-3 justify-content-center align-items-center">
        <?php foreach ($data['items'] as $item) : ?>
            <div class="col">
                <div class="card" data-bss-hover-animate="pulse">
                    
                    <div class="card-body">
                        <h1 class="card-title"><?php echo $item-
>name; ?></h1>
                        <p class="card-text"><?php echo $item-
>description; ?></p>
                        <p class="card-text">Стать:
                            <?php
                                switch ($item->sex) {
                                    case 1:
                                        echo 'Унісекс';
                                        break;
                                    case 2:
                                        echo 'Жіноча';
                                        break;
                                    case 3:
                                        echo 'Чоловіча';
                                        break;
                                }
                            ?>
                        </p>
                        <p class="card-text">Кольори: <?php echo $item-
>color; ?></p>
                        <p class="card-text">Розміри: <?php echo $item-
>size; ?></p>
                        <h4 class="card-title"><?php echo $item->price; ?>
                        грн.</h4>
                        <form method="post" action="<?php echo
URLROOT; ?>/cart/addToCart/<?php echo $item->id; ?>">

```

```

        <input type="text" class="form-control mt-3 mb-3" name="color" placeholder="Колір">
        <input type="text" class="form-control mt-3 mb-3" name="size" placeholder="Розмір">
        <input type="number" class="form-control mt-3 mb-3" name="quantity" placeholder="Кількість">
        <button class="btn btn-primary mt-3 mb-3 w-100" type="submit">Додати в корзину</button>
    </form>
</div>
</div>
</div>
<?php endforeach; ?>
</div>
<?php else : ?>
    <div class="row row-cols-3 justify-content-center align-items-center">
        <div class="col">
            <?php require APPROOT . '/views/template/alert.php'; ?>
        </div>
    </div>
<?php endif; ?>
    <div class="row row-cols-3 justify-content-center align-items-center">
        <div class="col">
            <div class="pagination justify-content-center mt-4">
                <a href="?page=<?php echo $data['pageNumber'] - 1 . buildQueryParams(array_merge($_GET, ['page' => null])); ?>" class="btn btn-primary">Назад</a>
                <a href="?page=<?php echo $data['pageNumber'] + 1 . buildQueryParams(array_merge($_GET, ['page' => null])); ?>" class="btn btn-primary">Вперед</a>
            </div>
        </div>
    </div>
</div>
</section>

    <?php require_once APPROOT . '/views/template/footer.php'; ?>
</body>

</html>

// OrderView.php

<!DOCTYPE html>
<html data-bs-theme="light" lang="en">

<head>
    <?php require_once APPROOT . '/views/template/meta.php'; ?>
    <?php require_once APPROOT . '/views/template/stylesheets.php'; ?>
</head>

<body>
    <div class="container mt-5">
        <div class="card">
            <div class="card-header">
                <h2>Успішне замовлення</h2>
            </div>
            <div class="card-body">
                <p class="card-text"><strong>Деталі замовлення:</strong></p>
                <p class="card-text"><?php echo isset($data['list']) ? htmlspecialchars($data['list']) : 'Немає даних'; ?></p>
            </div>
        </div>
    </div>

```

```
        <p class="card-text"><strong>Загальна ціна:</strong> <?php echo
isset($data['price']) ? htmlspecialchars($data['price']) : 'Немає даних'; ?> грн</p>
        <p class="card-text"><strong>Телефон:</strong> <?php echo
isset($data['phone']) ? htmlspecialchars($data['phone']) : 'Немає даних'; ?></p>
        <p class="card-text"><strong>Email:</strong> <?php echo
isset($data['email']) ? htmlspecialchars($data['email']) : 'Немає даних'; ?></p>
        <p class="card-text"><strong>Адреса:</strong> <?php echo
isset($data['address']) ? htmlspecialchars($data['address']) : 'Немає даних'; ?></p>
        <p class="card-text"><strong>Повідомлення:</strong> <?php echo
isset($data['message']) ? htmlspecialchars($data['message']) : 'Немає даних'; ?></p>
    </div>
</div>
</div>
</body>

</html>
```

ДОДАТОК Б. Слайди мультимедійної презентації

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ
ФАХОВИЙ КОЛЕДЖ ОНТУ»

Розробка інтернет- магазину одягу для просування бренду

Хоменко Оксана

АКТУАЛЬНІСТЬ, МЕТА, ПРАКТИЧНА ЦІННІСТЬ

- Актуальність теми полягає в необхідності створення інтернет-магазину для розвитку та підвищення популярності торговельного бізнесу, що дозволяє зменшити витрати та розширити ринки збуту.
- Метою роботи є розробка веб-системи, яка забезпечує пошук та придбання товарів для клієнтів, а також додавання, видалення та редагування товарів для адміністратора.
- Практичне значення полягає в наданні власникам магазинів інструменту для розширення їхнього бізнесу.

ЗАСОБИ РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ

HTML – гіпертекстова мова розмітки веб-сторінки.

CSS – каскадні таблиці стилів.

Bootstrap – CSS-фреймворк.

JavaScript – мова програмування, надає динамічність та інтерактивність веб-сторінкам.

jQuery – JavaScript бібліотека.

PHP – мова програмування для роботи з базами даних.

SQL – Командний центр для бази даних.

MySQL – зберігає та керує всіма даними.

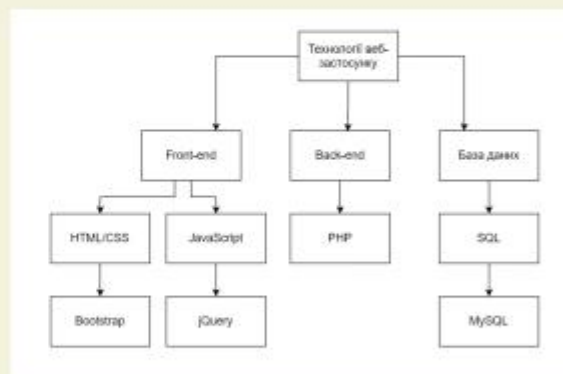
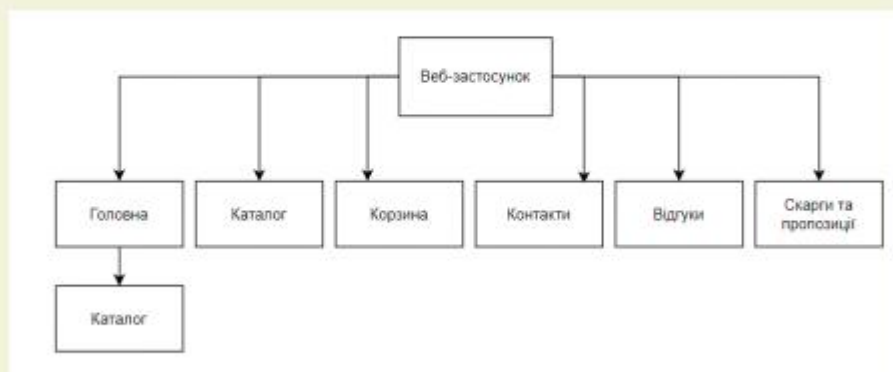
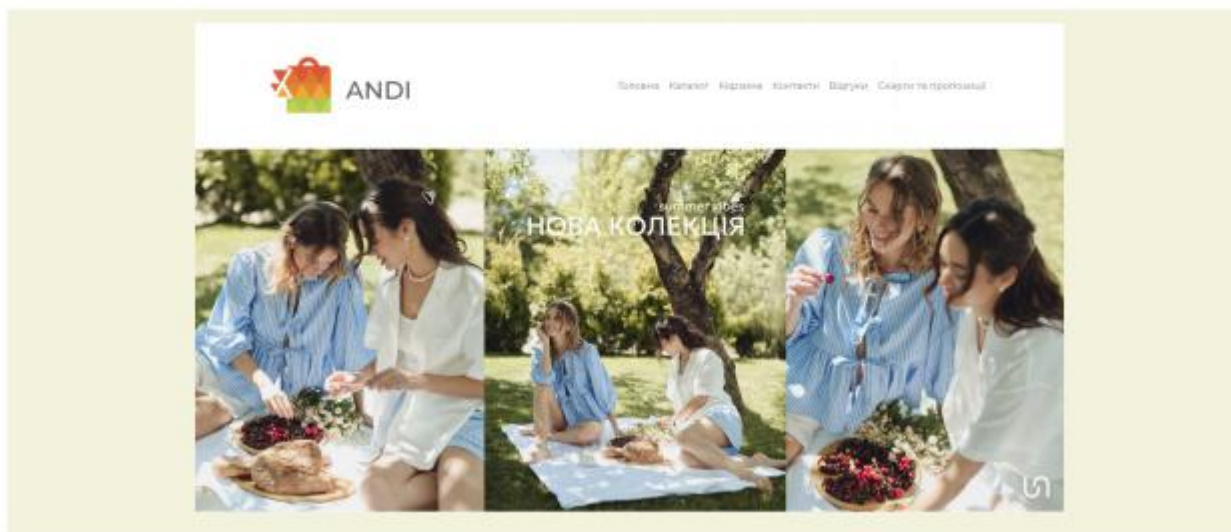


СХЕМА НАВИГАЦІЇ ВЕБ-ЗАСТОСУНКУ



ВЕБ-ДИЗАЙН ВЕБ-ЗАСТОСУНКУ

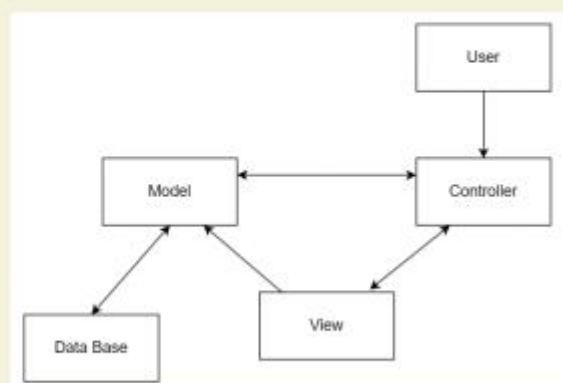


АРХІТЕКТУРА ВЕБ-ЗАСТОСУНКУ

Model відповідає за управління даними застосунку. Він обробляє доменну логіку, взаємодіє з базою даних та відповідає за збереження і обробку даних.

View займається представленням даних користувачу. Вона отримує дані з Model та відображає їх у відповідному форматі.

Controller діє як посередник між Model і View. Він отримує вхідні запити від користувача через View, обробляє їх за допомогою Model та визначає, яку View слід використовувати для відображення результату.



ПРОЦЕС РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ

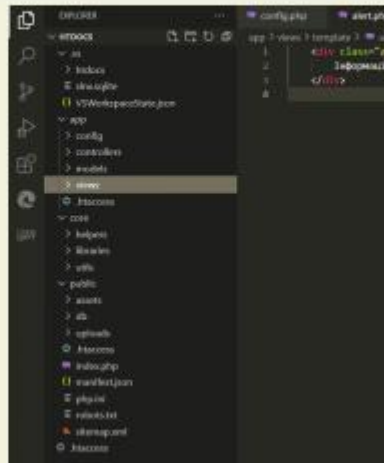


СХЕМА БАЗИ ДАНИХ ВЕБ-ЗАСТОСУНКУ



ФРАГМЕНТ КОДУ - ОТРИМАННЯ ДАНИХ З КОРЗИНИ (СЕСІЇ)

```
public function index()
{
    // Fetch information
    $information = $this->informationModel->selectById(1);

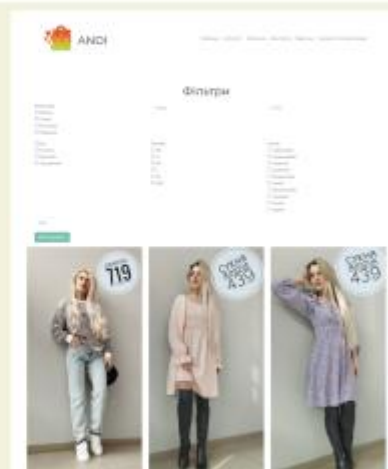
    // Get cart items from session
    $cart = isset($_SESSION['cart']) ? $_SESSION['cart'] : [];

    // Prepare data to pass to the view
    $data = [
        'information' => $information,
        'cart' => $cart, // Add cart items to data array
    ];

    // Load the view with data
    $this->view('Cart', $data);

    // Display flash message (if any)
    flash('message');
}
```

ДЕМОНСТРАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ



ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ
ФАХОВИЙ КОЛЕДЖ ОНТУ»

Дякую за увагу!

Хоменко Оксана

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Хоменко Оксани Владиславівни

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Розробка інтернет-магазину одягу для просування бренду

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 86 сторінок. У пояснювальній записці описано етапи розробки інтернет-магазину одягу для просування бренду засобами PHP та MySQL. Графічна частина складається з окремих слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувачка освіти Хоменко Оксана поступово та послідовно виконувала всі етапи, проявляла ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувачка освіти виконувала самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувачка освіти Хоменко Оксана під час роботи над дипломним проектом вивчила достатньо багато літературних та інтернет-джерел за даною тематикою.

Вважаю, що теоретична підготовка дипломниці достатня і вона готова до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувачка освіти Хоменко Оксана показала вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як MS VS Code, PHP, MySQL, MS PowerPoint, MS Visio та ін.

Оцінка розрахункової частини Відмінно
Оцінка графічної частини Добре
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Жадан Артур Сергійович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», викладач спецдисциплін циклової комісії комп'ютерної техніки та програмної інженерії

Підпис



« 10 » 06 2024 р.

РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Хоменко Оксани Владиславівни

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Жадан Артур Сергійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка інтернет-магазину одягу для просування бренду

Обсяг розрахунково-пояснювальної записки 86 сторінок

Обсяг графічної (презентаційної) частини 12 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі просування бренду магазину одягу та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту _____

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 12 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки відмінна, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту Реалізовано інтернет-магазин для просування бренду, що дозволяє публікувати товар на віртуальну вітрину з метою його просування.

Стилізація графічного інтерфейсу відповідає тематиці обраної предметної області.

Розроблена веб-система добре підходить для малого бізнесу.

д) основні недоліки дипломного проекту _____

При впровадженні веб-системи у сферу середнього бізнесу, можуть виникнути певні складнощі у зв'язку із обмеженнями веб-системи.

В основному розділі пояснювальної записки поверхнево описано етапи розробки та впровадження веб-системи. В деяких частинах пояснювальної записки присутні помилки оформлення.

Оцінка розрахункової частини _____ Добре

Оцінка графічної частини _____ Добре

Загальна оцінка _____ Добре

Прізвище, ім'я, по батькові рецензента _____ Кривченко Юрій Вікторович

Місце роботи і посада рецензента _____ ВСП «Одеський технічний фаховий коледж ОНТУ»,
голова циклової комісії комп'ютерних технологій та програмної інженерії

Підпис: _____

« 11 » _____ червень 2024 р.

Ім'я користувача:
Катерина Григоріївна Краснокутська

ID перевірки:
1016357650

Дата перевірки:
13.06.2024 17:32:07 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.06.2024 17:32:42 EEST

ID користувача:
100011688

Назва документа: 4РП-07_Хоменко_О

Кількість сторінок: 48 Кількість слів: 6678 Кількість символів: 51256 Розмір файлу: 10.93 MB ID файлу: 1016161988

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

4.24% Схожість

Найбільша схожість: 0.57% з Інтернет-джерелом (<https://gitlab.fdmci.hva.nl/oordl002/paymentplatform/commit/a1c09...>)

4.24% Джерела з Інтернету

344

Сторінка 50

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

10
сторінок

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Хоменко Оксана Владиславівна,
здобувачка освіти гр. 4РП-07, та

Жадан Артур Сергійович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка інтернет-магазину одягу для просування бренду» (авторка роботи – Хоменко О.В., керівник роботи – Жадан А.С.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Хоменко О.В. /

Керівник



/ Жадан А.С. /

«10» червня 2024 р.