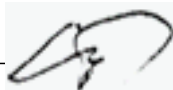


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

—  —

проф. Приходько С.Б.

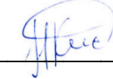
«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: **Розробка вебсервісу для магазину весільного та святкового**
вбрання "Казковий образ"

Виконала: студентка групи 4151

—  —

Кисельова Н.О.

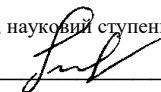
(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАС, к.т.н.,

ДОЦЕНТ

(посада, науковий ступень вчене звання)

—  —

Макарова Л.М.

(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми

 _____ доц. Макарова Л.М.
 (підпис)
 « 08 » 04 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студентці Кисельовій Наталії Олександрівні
 (Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебсервісу для магазину весільного та святкового вбрання
"Казковий образ"

Керівник роботи Макарова Лідія Миколаївна
 Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____
 Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів: Титульний слайд, Мета та завдання кваліфікаційної роботи, Аналіз вимог до вебсервісу, Архітектура вебсервісу, Діаграма класів вебсервісу, Діаграми послідовності вебсервісу, Логічна модель бази даних вебсервісу, Робочий проєкт вебсервісу, Фізична модель бази даних вебсервісу, Результати розробки вебсервісу, Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 08.04.2025 р.**КАЛЕНДАРНИЙ ПЛАН**

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	
5.	Розробка проекту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	16.06.2025	

* - за результатами переддипломної практики (ПП), яка була з 03.02.2025 до 23.03.2025 р.

Студент

(підпис)

Кисельова Н.О.

(ПІБ)

Керівник роботи

(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розв'язанню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме аналізу розробки вебсервісу для магазину весільного та святкового вбрання "Казковий образ".

Проведено аналіз практичної задачі інженерії програмного забезпечення та проаналізовано існуючі аналоги програмних рішень. Проведено постановку задачі на розробку вебсервісу для магазину весільного та святкового вбрання "Казковий образ". Розроблено ескізний, технічний та робочий проекти програмного забезпечення, представлено результати розробки та розділ з охорони праці.

Кваліфікаційна робота викладена на 90 сторінках друкованого тексту, містить 27 рисунків, 32 таблиці, список використаних джерел з 29 найменувань та 5 додатків.

ABSTRACT

The qualification work is dedicated to solving a practical problem of software engineering, characterized by complexity and uncertainty of conditions, namely the analysis of the development of a web service for the wedding and festive clothing store "Fairy Image".

An analysis of the practical problem of software engineering was carried out and existing analogues of software solutions were analyzed. The task for the development of a web service for the wedding and festive clothing store "Fairy Image" was formulated. A sketch, technical and working software project was developed, the results of the development and the section on labor protection are presented.

The qualification work is presented on 90 pages of printed text, contains 27 figures, 32 tables, a list of sources used with 29 names, and 5 appendices.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ ВЕСІЛЬНОГО ТА СВЯТКОВОГО ВБРАННЯ «КАЗКОВИЙ ОБРАЗ».....	10
1.1 Аналіз практичної задачі інженерії програмного забезпечення для розробки вебсервісу для магазину весільного та святкового вбрання «Казковий образ»	10
1.2 Аналіз існуючих програмних продуктів.....	12
1.3 Постановка задачі на розробку вебсервісу для магазину весільного та святкового вбрання «Казковий образ»	20
2 РОЗРОБКА ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ ВЕСІЛЬНОГО ТА СВЯТКОВОГО ВБРАННЯ «КАЗКОВИЙ ОБРАЗ»	25
2.1 Аналіз вимог до вебсервісу для магазину весільного та святкового вбрання «Казковий образ»	25
2.1.1 Побудова моделі варіантів використання	25
2.1.2 Специфікації варіантів використання.....	26
2.2 Архітектура вебсервісу для магазину весільного та святкового вбрання «Казковий образ»	31
2.3 Проєкт вебсервісу для магазину весільного та святкового вбрання «Казковий образ»	33
2.3.1 Ескізний проєкт вебсервісу.....	33
2.3.2 Технічний проєкт вебсервісу	38
2.3.3 Робочий проєкт програмного забезпечення.....	45
3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ ВЕСІЛЬНОГО ТА СВЯТКОВОГО ВБРАННЯ «КАЗКОВИЙ ОБРАЗ»	50
4 ОХОРОНА ПРАЦІ	52
4.1 Аналіз факторів ризику при роботі з комп'ютером	52

4.2 Основні законодавчі та нормативно-правові акти про охорону праці в Україні.....	54
4.3 Розробка заходів щодо зменшення навантаження при роботі з комп'ютером.....	56
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТОК А – Технічне завдання на розробку вебсервісу для магазину весільного та святкового вбрання "Казковий образ".....	66
ДОДАТОК Б – Код програми	70
ДОДАТОК В – Опис програми.....	80
ДОДАТОК Г – Інструкція користувача	83
ДОДАТОК Д – Програма та методика випробування програми	87

ВСТУП

В умовах сучасної цифрової епохи ефективність ведення бізнесу багато в чому визначається його представленістю в онлайн-середовищі. Магазины весільного та святкового вбрання не становлять винятку – розробка зручного, функціонального та результативного вебсервісу може суттєво зміцнити позиції магазину на ринку. Сучасні користувачі очікують, що вибір і замовлення вбрання можна буде здійснити онлайн, переглянувши каталог, підбравши розмір, стиль, колір, а також забронювавши примірku або отримавши консультацію в режимі реального часу.

Кваліфікаційна робота передбачає розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробку вебсервісу для магазину весільного та святкового вбрання "Казковий образ". Комплексність задачі полягає в тому, що є багато взаємопов'язаних елементів, різноманітних вимог до вебсервісу для магазину весільного та святкового вбрання "Казковий образ", до того ж є потреба в оптимізації ресурсів. Невизначеність умов полягає в тому, що параметри вебсервісу для магазину весільного та святкового вбрання "Казковий образ" можуть бути нечіткими або недостатньо визначеними. Вебсервіс для магазину весільного та святкового вбрання "Казковий образ" допоможе покращити обслуговування клієнтів, автоматизувати процеси замовлення та обробки товарів, забезпечити зручну навігацію та пошук вбрання за різними критеріями (тип, стиль, колір, розмір тощо), а також підтримувати зворотний зв'язок із клієнтами. Такий вебсервіс дозволить ефективно управляти асортиментом, відстежувати наявність товарів, формувати аналітичні звіти для прийняття управлінських рішень, а також реалізовувати маркетингові стратегії, такі як акції, знижки, програми лояльності. У підсумку, розроблений вебсервіс сприятиме цифровій трансформації діяльності магазину "Казковий образ",

підвищенню його конкурентоспроможності на ринку та покращенню взаємодії з цільовою аудиторією.

Мета роботи: розробка вебсервісу для магазину весільного та святкового вбрання "Казковий образ".

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз практичної задачі інженерії програмного забезпечення, в рамках якого проаналізувати, зокрема, існуючі аналогічні програмні продукти;

- на основі проведеного аналізу сформулювати вимоги до вебсервісу для магазину весільного та святкового вбрання "Казковий образ" та розробити технічне завдання;

- розробити ескізний, технічний та робочий проекти вебсервісу для магазину весільного та святкового вбрання "Казковий образ";

- провести тестування та випробування вебсервісу для магазину весільного та святкового вбрання "Казковий образ";

- розробити програмну документацію до вебсервісу для магазину весільного та святкового вбрання "Казковий образ".

В результаті буде розроблено вебсервіс для магазину весільного та святкового вбрання "Казковий образ", який забезпечить повноцінну онлайн-присутність магазину, дозволить користувачам переглядати каталог товарів, здійснювати пошук за фільтрами, оформляти замовлення, бронювати примірки, залишати відгуки та отримувати консультації в режимі реального часу. Вебсервіс матиме зручний адаптивний інтерфейс, що дозволить комфортно користуватись ним як з комп'ютерів, так і з мобільних пристроїв. Адміністративна панель дозволить персоналу магазину швидко оновлювати інформацію про товари, керувати замовленнями та взаємодіяти з клієнтами. Таким чином, вебсервіс сприятиме підвищенню ефективності бізнес-процесів, покращенню клієнтського досвіду та розширенню ринкових можливостей магазину "Казковий образ".

Об'єктом даної кваліфікаційної роботи є процес розробки вебсервісу для магазину весільного та святкового вбрання "Казковий образ".

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ ВЕСІЛЬНОГО ТА СВЯТКОВОГО ВБРАННЯ «КАЗКОВИЙ ОБРАЗ»

1.1 Аналіз практичної задачі інженерії програмного забезпечення для розробки вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Розробка та створення вебсервісу для магазину весільного та святкового вбрання – це комплексний технічний процес, що включає кілька важливих етапів: від аналізу потреб і цільової аудиторії до впровадження готового рішення. Основна мета полягає в тому, щоб створити інтуїтивно зрозумілий, візуально привабливий та функціональний вебінструмент, який полегшить користувачам процес вибору та замовлення одягу для урочистих подій, а також оптимізує внутрішні бізнес-процеси магазину.

Магазин весільного та святкового вбрання спеціалізується на продажу товарів для особливих подій: весіль, вечірок, випускних, ювілеїв та фотосесій. Асортимент включає весільні сукні, вечірні сукні, костюми, взуття, аксесуари, фати, болеро, рукавички, сумочки, прикраси тощо. Подібні магазини мають широкий асортимент і часто потребують цифрової платформи, яка забезпечить зручний доступ до продукції, актуальність наявності та можливість онлайн-бронювання послуг (примірка, пошив, консультація) [1].

Щоб ефективно організувати продаж урочистого вбрання через вебсервіс, слід врахувати наступні аспекти.

Управління асортиментом та запасами товарів

Спершу проводиться аналіз попиту на різні типи вбрання з урахуванням сезонності (весільний сезон, випускні, новорічні свята) та поточних трендів. На основі цього формуються замовлення у виробників або власного пошиву.

Необхідна система відстеження залишків на складі, щоб своєчасно оновлювати наявність на сайті, уникати розчарування покупців та оптимізувати закупівлі.

Класифікація та розміщення товарів

Вебсервіс має надавати чітку структуру каталогу з поділом на категорії: весільні сукні, вечірні сукні, для нареченої, для подружок, чоловічі костюми, аксесуари. Усередині кожної категорії – фільтри за кольором, фасоном, ціною, розміром. Важливим є візуальний контент: якісні фотографії з різних ракурсів, відеогляди, розмірна сітка, можливість "приміряти" вбрання віртуально.

Ціноутворення та акції

Формування цін базується на собівартості товару, конкурентному середовищі та платоспроможності цільової аудиторії. Для залучення покупців застосовуються цінові стратегії – наприклад, обмежені знижки, бонуси за передплату або рекомендацію друзів, сезонні розпродажі, персоналізовані пропозиції.

Презентація товару в інтерфейсі

Візуальна подача має бути естетичною, стильною та зручною. Усі моделі повинні мати розгорнутий опис, інформацію про тканини, варіанти кольорів, доступність розмірів, умови індивідуального пошиву. Бажано забезпечити функцію «запис на примірку» або «виклик стиліста».

Онлайн-продажі та обслуговування клієнтів

Вебсервіс повинен мати просту форму замовлення, підтримку кількох способів оплати (оплата карткою, переказ, післяплата), захищеність персональних даних. Додатково – наявність онлайн-чату або інтеграції з месенджерами для оперативного консультування клієнтів. Важливо впровадити систему обробки замовлень та надсилання сповіщень про їх статус.

Маркетинг та просування

Цифровий маркетинг є ключовим каналом залучення клієнтів. Вебсервіс має бути оптимізованим для пошукових систем (SEO), містити блог чи розділ з порадами, мати можливість підписки на розсилку. Варто застосовувати

таргетовану рекламу у соціальних мережах, спільні акції з весільними фотографами, організаторами, візажистами.

Фінансовий облік та аналітика

Розробка вебсервісу повинна передбачати інтеграцію з CRM- або ERP-системами для обліку замовлень, витрат, прибутків, рентабельності товарів. Потрібна також статистика щодо переглядів товарів, найпопулярніших категорій, ефективності рекламних кампаній.

Повернення та обміни

Оскільки весільне вбрання часто виготовляється або коригується індивідуально, політика повернень має бути чітко прописана. Проте в разі покупки стандартних моделей повинна бути реалізована проста процедура повернення/обміну, зручна комунікація з менеджерами та чіткі терміни.

Зв'язок з клієнтами та відгуки

Важливо організувати збір та модерацію відгуків на сайті, виводити оцінки під товарами, а також реагувати на коментарі клієнтів. Це підвищує довіру, забезпечує зворотний зв'язок і допомагає в ухваленні управлінських рішень [2].

Ці загальні положення дозволяють глибше зрозуміти специфіку цифрової трансформації магазину весільного та святкового вбрання. Вони формують методологічну основу для розроблення ефективної підсистеми вебсервісу, що задовольнить як вимоги бізнесу, так і очікування клієнтів.

1.2 Аналіз існуючих програмних продуктів

Сучасні магазини весільного та святкового вбрання працюють у конкурентному середовищі, де важливу роль відіграє не лише якість товару, а й рівень сервісу, зручність замовлення, швидкість обслуговування та персоналізований підхід до клієнта. Для досягнення цих цілей бізнес активно використовує програмні продукти, адаптовані під специфіку індустрії.

Типовий весільний салон або магазин прокату суконь поєднує кілька напрямів діяльності: продаж, оренда, індивідуальний пошив, проведення примірок і консультацій. Тому одним із ключових завдань є побудова системи, що дозволяє ефективно управляти всіма процесами одночасно.

Було проаналізовано десятки інтернет-магазинів, щоб визначити кращі з них і скласти рейтинг за наступними показниками:

- Асортимент: кількість товарів, різноманітність фасонів, наявність аксесуарів.
- Зручність навігації: UX сайту, фільтри, швидкість пошуку, адаптація до мобільних.
- Спосіб оплати: кількість варіантів (карта, готівка, накладений платіж, онлайн-оплата).
- Опис товару: деталізація, наявність розмірної сітки, фото, відео, характеристики.

Ці сайти пропонують десятки сотень товарів. Відгуки клієнтів дають чітко зрозуміти якість товару та популярність магазину. У 10-ку увійшли магазини, що зібрали на своїх сторінках розмаїтий вибір продукції з описом важливих характеристик (див. табл. 1.1).

Таблиця 1.1 – Порівняльна характеристика інтернет-магазинів весільного та святкового вбрання

№	Назва магазину	Асортимент	Зручність навігації сайту	Спосіб оплати	Опис товару
1	InLove (Київ)	5	5	4	5
2	Laura Style (Харків)	5	4	5	4
3	Ivanna Dress (Львів)	4	4	4	4
4	Primavera Sposa (Одеса)	4	5	5	5
5	FAINA Boutique (Біла Церква)	4	4	4	4
6	СВІТЛАНА (Львів)	4	3	4	3
7	Jeorjett Dress (Чернівці)	5	3	4	3
8	Wedding Market (Одеса/Київ)	5	4	5	4
9	Wedboom.ua (Чернівці)	4	4	5	5
10	White Angel (Львів)	4	3	4	3

Зовнішній вигляд головних сторінок проаналізованих інтернет-магазинів представлений на рис. 1.1 – 1.10.

InLove (Київ) позиціонує себе як бренд сучасних весільних суконь із шоурумом у центрі столиці [3]. Сайт має стильний і мінімалістичний дизайн, навігація інтуїтивна, кожна модель має якісні фото та докладний опис. Однією з головних переваг є можливість онлайн-бронювання примірки та консультації, а також орієнтація на актуальні тренди. Недоліком можна вважати обмеженість у способах оплати та порівняно невеликий вибір вечірніх суконь.

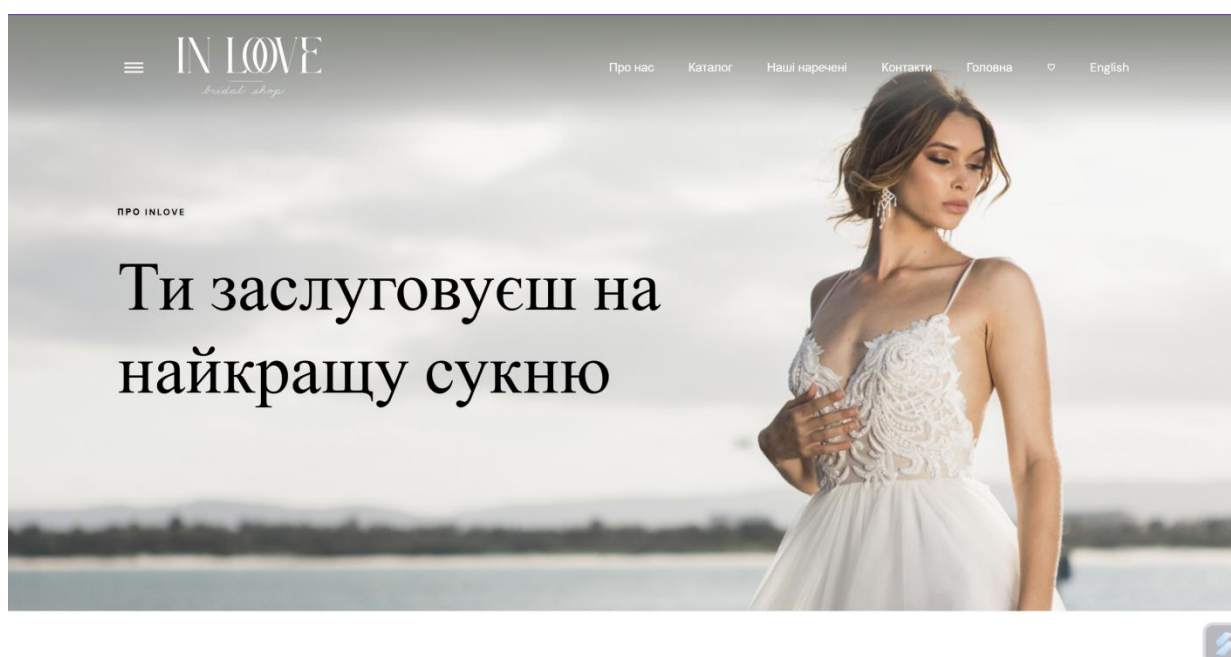


Рисунок 1.1 – Головна сторінка інтернет-магазину «InLove»

Laura Style (Харків, Київ) – це масштабна мережа з широким асортиментом весільних і вечірніх суконь, включно з чоловічими костюмами та аксесуарами [4]. Їхній сайт має дещо класичний вигляд, однак добре структурований і функціональний. Перевагою є наявність опції оренди вбрання, що знижує вартість підготовки до події. Незначним недоліком є не завжди актуальні фото та недостатньо докладні розміри для деяких товарів.



Рисунок 1.2 – Головна сторінка Інтернет-магазину «Laura Style»

Ivanna Wedding Dress (Львів) – один із найстаріших салонів Львова, що поєднує досвід із сучасним підходом до весільної моди [5]. Асортимент на сайті представлено добірно, з акцентом на класичну українську елегантність. Веб-сайт простий, зручний, хоча дещо застарілий візуально. Основною перевагою є можливість індивідуального пошиву за власними мірками. Водночас, на сайті не завжди є повна інформація про наявність моделей.

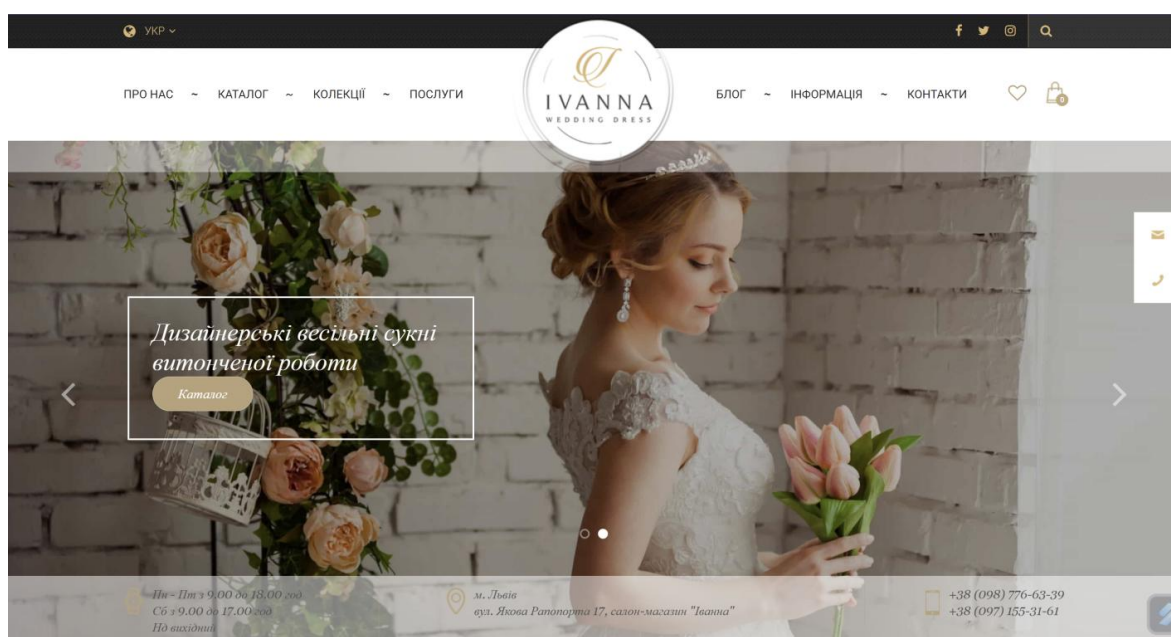


Рисунок 1.3 – Головна сторінка Інтернет-магазину «Ivanna Wedding Dress»

Primavera Sposa (Одеса) відома своїм стильним підходом і ексклюзивністю [6]. Сайт справляє враження преміального магазину: висока якість фото, креативний контент і детальні описи. Тут акцент зроблено на індивідуальності та створенні образу разом із консультантами. Ціни вищі за середні, що відповідає брендовій концепції, але може бути недоліком для економного покупця.

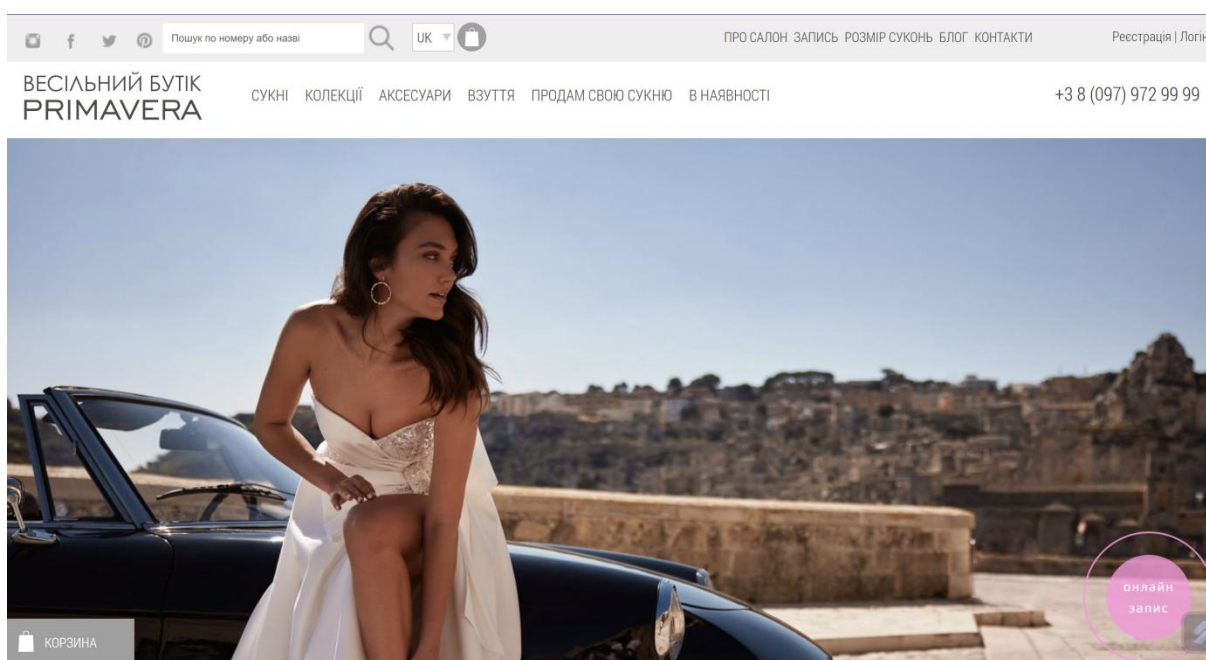


Рисунок 1.4 – Головна сторінка Інтернет-магазину «Primavera Sposa»

FAINA Boutique (Біла Церква) – це інтернет-магазин, що орієнтується на дизайнерські моделі для урочистостей [7]. Перевага – наявність авторських колекцій і помірна цінова політика. Сайт працює стабільно, хоча функціональність фільтрів обмежена. Недоліком є не дуже великий вибір весільних суконь у порівнянні з іншими магазинами.

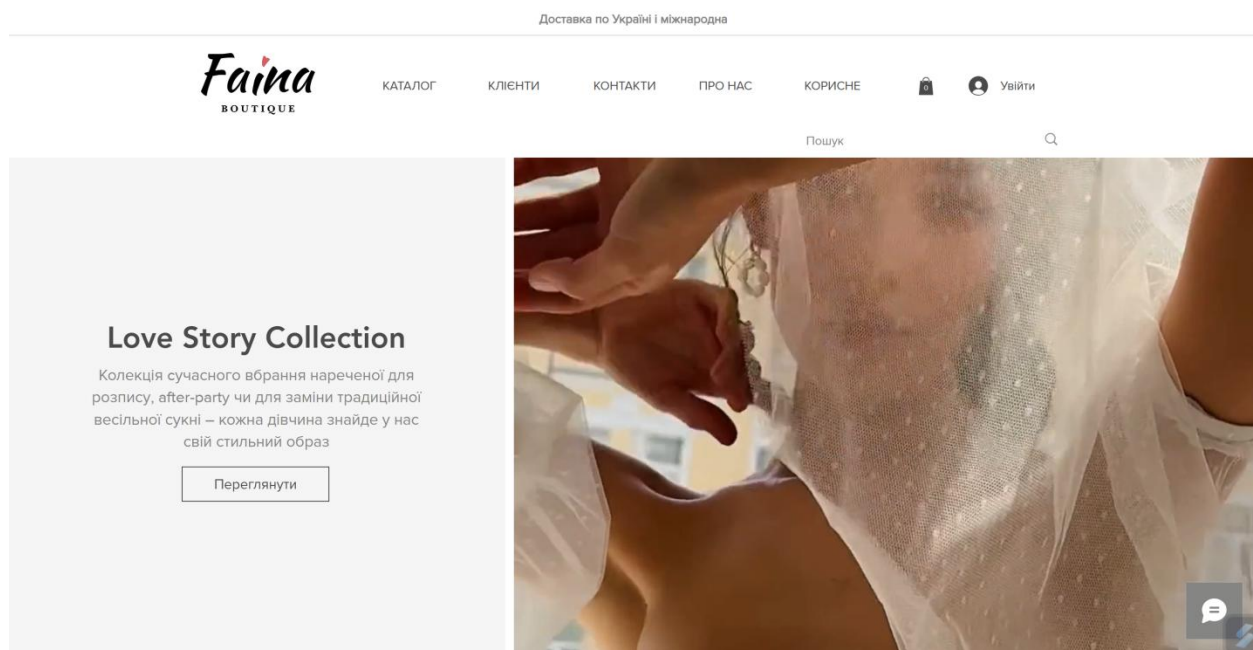


Рисунок 1.5 – Головна сторінка Інтернет-магазину «FAINA Boutique»

СВІТЛАНА (Львів) – традиційний львівський салон із багаторічною історією [8]. Вебсайт виконано в простому стилі, без надлишкової візуальної складності. Перевагою є індивідуальний пошив, можливість адаптації фасонів під клієнтку. Однак сайт потребує оновлення: відсутні інтерактивні елементи, мало фото з різних ракурсів, і загалом навігація дещо застаріла.

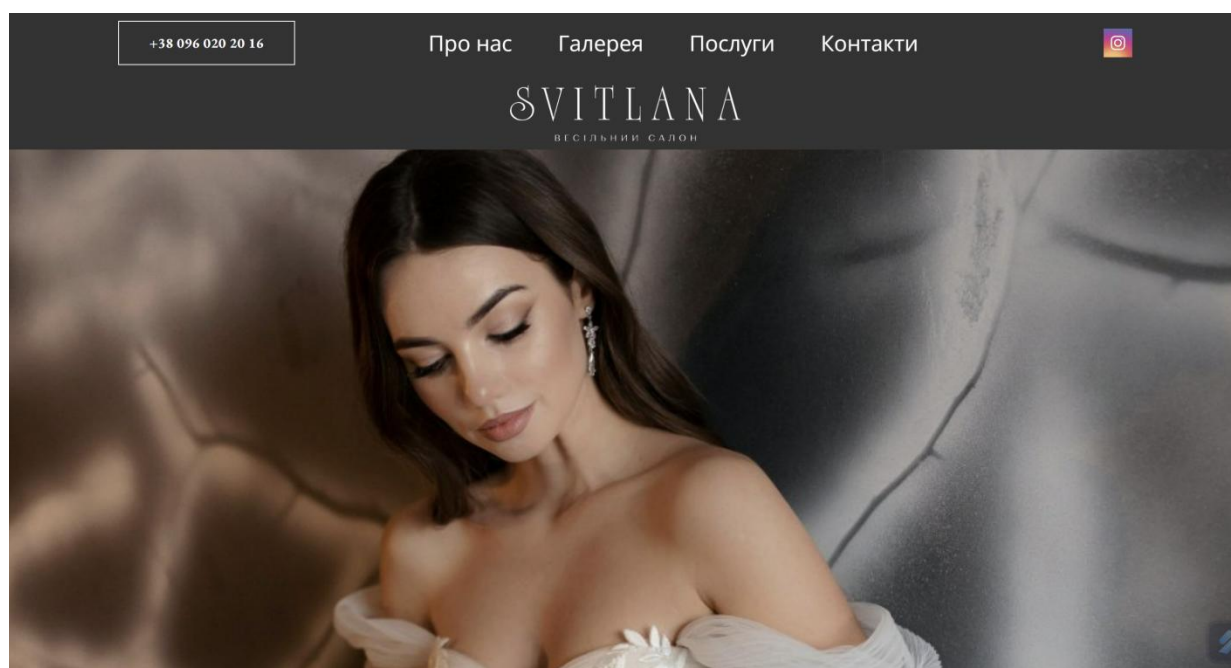


Рисунок 1.6 – Головна сторінка Інтернет-магазину «СВІТЛАНА»

Jeorjett Dress (Чернівці) працює як виробник і оптовий постачальник весільних суконь [9]. Основна перевага – широкий вибір моделей, орієнтованих на масовий продаж, включаючи експорт за кордон. Це один із небагатьох магазинів, що активно співпрацює з іншими салонами. Недолік – мінімалізм у вебдизайні: сайт більше нагадує бізнес-платформу, ніж зручний магазин для кінцевого покупця.

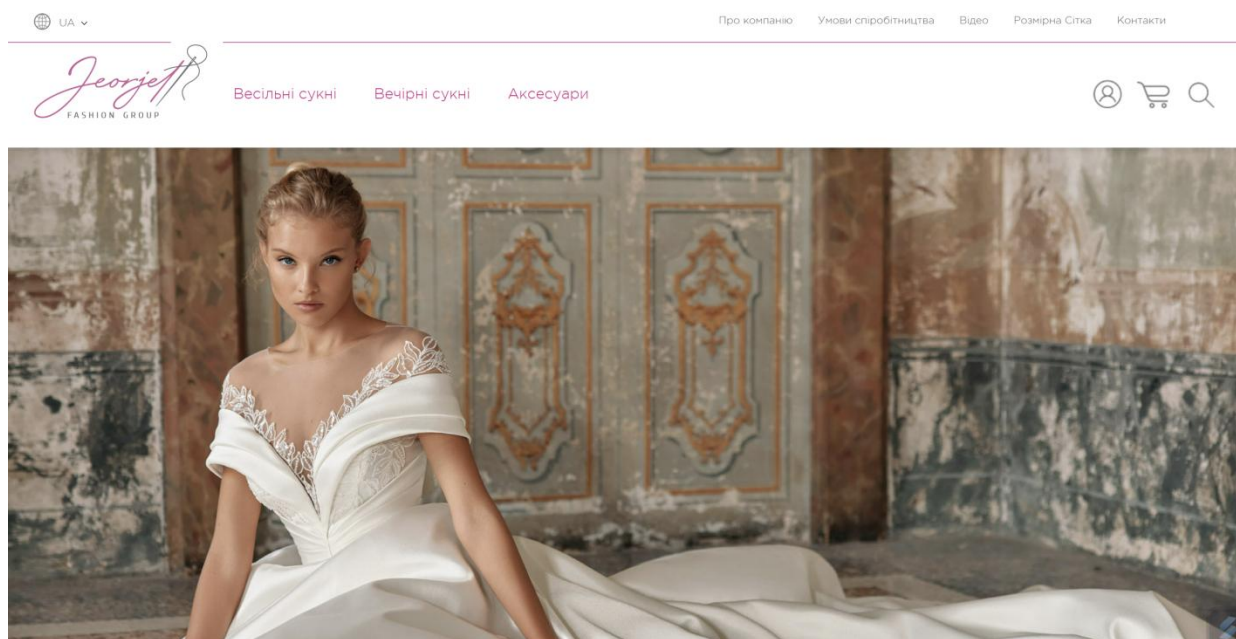


Рисунок 1.7 – Головна сторінка Інтернет-магазину «Jeorjett Dress»

Wedding–Market / Анабель (Київ, Одеса, Херсон) – доступний за ціною інтернет-магазин, який пропонує весільні та вечірні сукні з демократичним підходом до оформлення сайту [10]. Тут легко знайти потрібний товар завдяки фільтрам, описам і варіантам кольору. Перевагою є великий вибір бюджетних моделей, а недоліком – деяка одноманітність асортименту та менш індивідуальний підхід.

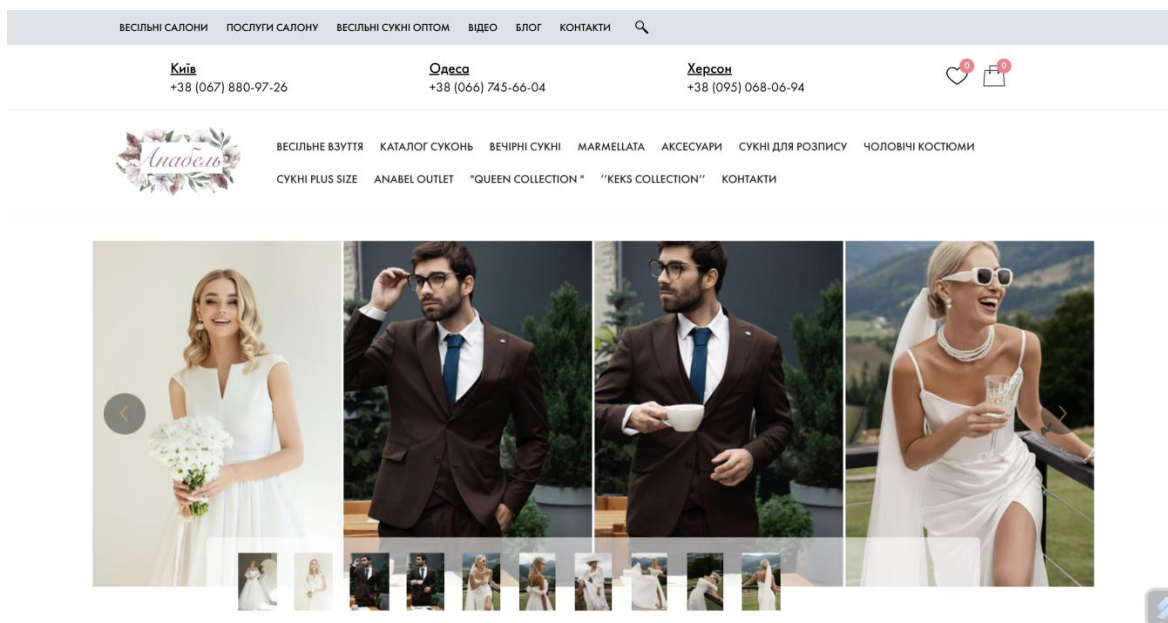


Рисунок 1.8 – Головна сторінка Інтернет-магазину «Wedding–Market/Анабель»

Wedboom.ua (Чернівці) – це платформа з великим вибором товарів: від суконь до фат, болеро й весільних підв'язок [11]. Сайт зручний, добре оптимізований, із функцією замовлення в кілька кліків. Його перевага – в доступності та мультибрендовості. Недолік – брак персоналізації, оскільки магазин працює більше як маркетплейс, ніж як бутік.

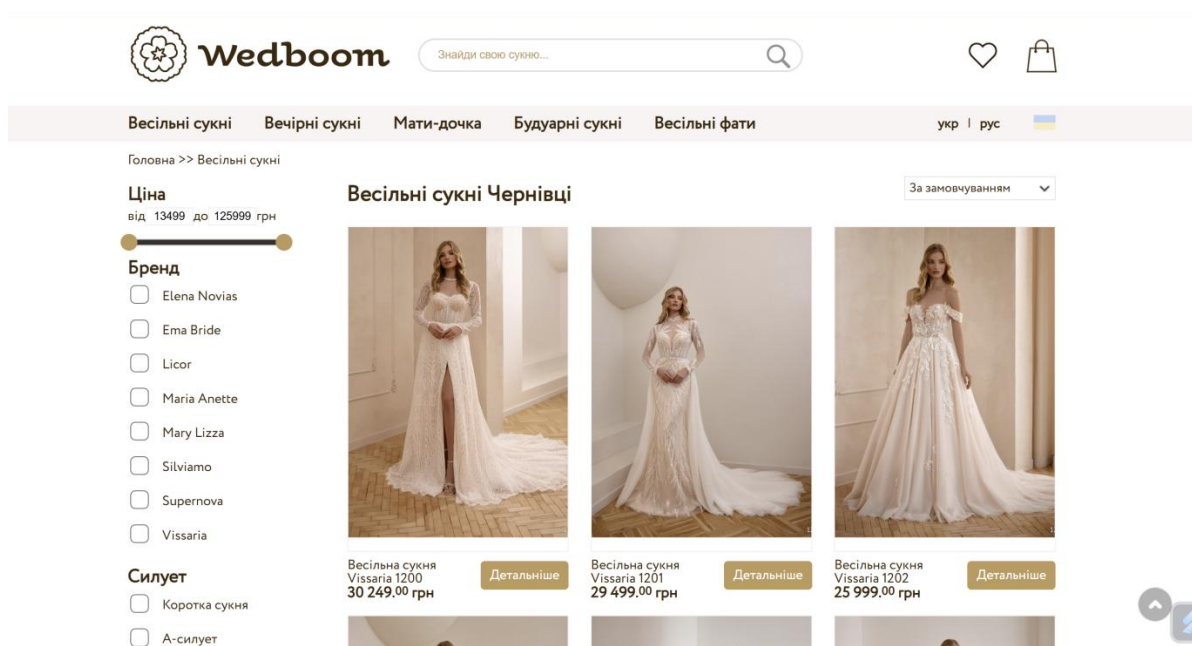


Рисунок 1.9 – Головна сторінка Інтернет-магазину «Wedboom.UA»

White Angel (Львів) – салон із широким вибором силуетів та фасонів, орієнтований на різні типи фігур [12]. Сайт простий, із базовим набором функцій, хоча іноді повільно завантажується. Перевагою є фокус на класичні стилі, а недоліком – нестача актуальних фото та обмежений функціонал для онлайн-консультацій.

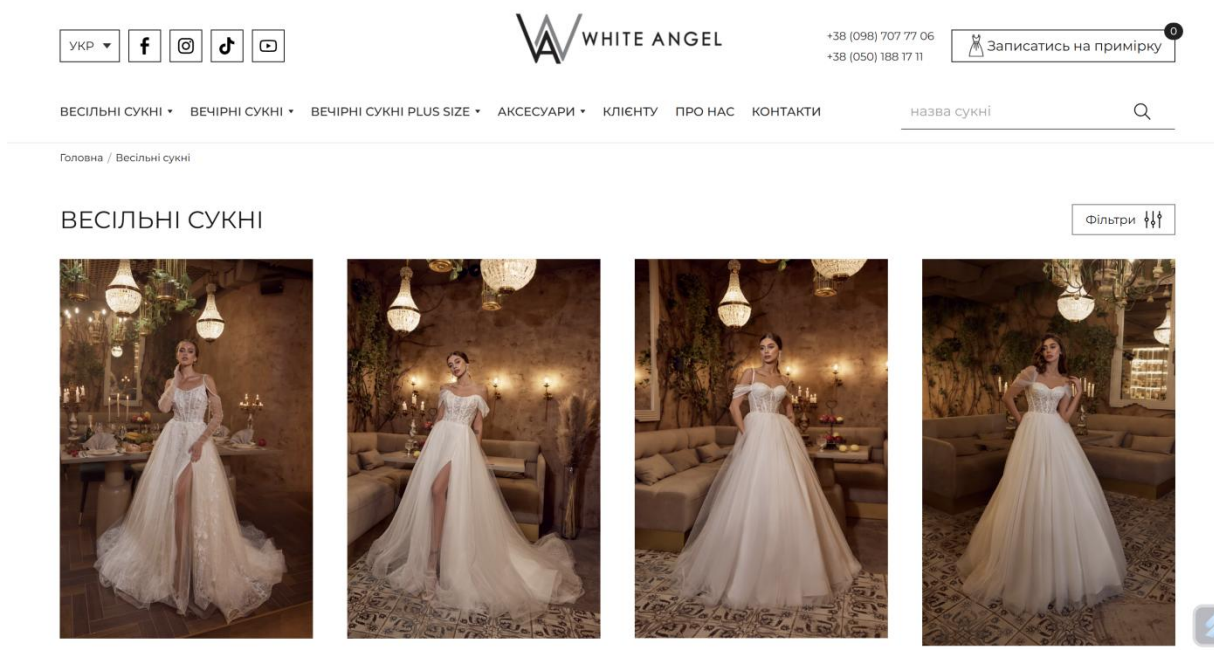


Рисунок 1.10 – Головна сторінка Інтернет-магазину «White Angel»

Після аналізу існуючих програмних продуктів переходимо до постановки задачі на розробку вебсервісу для магазину весільного та святкового вбрання «Казковий образ».

1.3 Постановка задачі на розробку вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Постановка задачі. Перед початком розробки вебсервісу для магазину весільного та святкового вбрання «Казковий образ» потрібно визначити ключові цілі та завдання для вирішуваної данної проблеми, після цього можна

буде розпочинати проектування. Перш за все, створимо стислий опис поставленого завдання.

Метою даного кваліфікаційної роботи є розробка вебсервісу для магазину весільного та святкового вбрання «Казковий образ».

Мета вебсервісу – надання всієї необхідної інформації про потрібні товари, а також можливість їх замовлення.

Розробка вебсервісу для магазину весільного та святкового вбрання «Казковий образ» базується на деяких вагомих підставах, які впливають на необхідність і доцільність такого проекту. Основні підстави для розробки вебсервісу включають:

1. Інтернет-присутність

У сучасних умовах ведення бізнесу наявність вебресурсу є необхідною складовою ефективної комунікації з цільовою аудиторією. Вебсервіс виконує роль цифрової вітрини магазину, забезпечуючи ширший доступ до потенційних покупців. За його допомогою магазин може не лише продемонструвати свій асортимент, але й надати вичерпну інформацію про продукцію, стимулюючи прийняття рішення про купівлю.

2. Інтеграція електронної комерції

Впровадження вебсервісу, створеного за допомогою фреймворку CakePHP та мови програмування PHP, відкриває можливості для повноцінної реалізації функціоналу електронної комерції. Це дозволяє не лише розширити географію клієнтської бази, а й забезпечити користувачам зручність взаємодії з магазином у цифровому форматі.

3. Підвищення конкурентоздатності

Використання сучасних вебтехнологій, зокрема React, надає можливість розробити швидкий, адаптивний і функціональний інтерфейс, що позитивно впливає на перше враження користувача. Завдяки високій продуктивності, привабливому дизайну та гнучкості налаштувань, вебсервіс може стати потужним інструментом для зміцнення конкурентних позицій на ринку.

4. Зручність користування для клієнтів

Вебсервіс забезпечує відвідувачам змогу у зручний для них час переглядати каталог товарів, ознайомлюватися з характеристиками, порівнювати моделі та здійснювати замовлення. Це сприяє підвищенню якості обслуговування, створює позитивний користувацький досвід і сприяє формуванню лояльності до бренду.

5. Маркетингові інструменти та просування

Цифрова платформа відкриває нові можливості для реалізації маркетингових стратегій: запуск email-розсилок, персоналізованих пропозицій для постійних клієнтів, акційних кампаній, а також інтеграції із соціальними мережами. Це сприяє активному залученню нових клієнтів, підвищенню впізнаваності бренду та зростанню обсягів продажу.

Таким чином, створення вебсервісу з використанням фреймворку CakePHP та мови програмування PHP є логічно обґрунтованим рішенням, яке дозволяє ефективно презентувати продукцію магазину весільного та святкового вбрання, організувати процес онлайн-продажів і забезпечити цифрову присутність бренду в умовах сучасного ринку[2].

Для ефективного створення вебсервісу необхідно чітко визначити та реалізувати ключові функціональні вимоги, що забезпечать повноцінну роботу сервісу та зручність для кінцевого користувача.

Перш за все, слід реалізувати структуру каталогу товарів, що відображатиме широкий асортимент продукції у впорядкованому вигляді. Кожна одиниця товару повинна супроводжуватися детальною інформацією: назва бренду, країна походження, категорія, текстовий опис, що сприятиме обґрунтованому вибору з боку покупця.

Для підвищення зручності пошуку потрібного товару необхідно впровадити систему фільтрації та розширеного пошуку. Це дозволить користувачам швидко орієнтуватися у великому обсязі інформації, знаходити потрібні позиції за параметрами, такими як розмір, колір, тип події, бренд тощо.

Невід’ємним елементом функціональності є інтерактивна корзина покупця, яка дозволяє зберігати обрані товари, редагувати їх кількість, видаляти позиції та переходити до оформлення замовлення.

Система реєстрації та авторизації користувачів має забезпечити можливість створення персонального кабінету з безпечним зберіганням даних. Завдяки цьому користувачі зможуть переглядати історію своїх замовлень, зберігати улюблені товари та пришвидшити процес покупки.

Оформлення замовлення має відбуватися через зручну та логічно побудовану послідовність дій із підтвердженням вибраних товарів, зазначенням контактних даних і способу доставки. Важливою складовою є збереження історії покупок, що сприяє повторним замовленням і персоналізованому обслуговуванню.

Система оплати повинна передбачати інтеграцію з кількома перевіреними платіжними сервісами. Це дасть змогу клієнтам обирати між різними методами оплати – банківськими картками, електронними гаманцями, онлайн-банкінгом тощо – з гарантією безпеки транзакцій.

Також необхідно впровадити систему адміністрування вмісту, яка дозволить співробітникам магазину керувати інформацією про товари, обробляти замовлення та оновлювати дані про клієнтів.

Для забезпечення якісного обслуговування клієнтів слід передбачити багатоканальну підтримку користувачів: онлайн-консультації, відповіді на запити електронною поштою, телефоном, а також реалізацію розділу поширених запитань (FAQ), що сприятиме самостійному вирішенню типових проблем.

Таким чином, впровадження зазначених функціональних можливостей є критично важливим для забезпечення повноцінного функціонування вебсервісу магазину весільного та святкового вбрання, що відповідатиме очікуванням сучасного користувача.

Для розробки вебсервісу враховуються наступні нефункціональні вимоги, які забезпечують оптимальну роботу і зручність використання.

1. Час відгуку системи: система повинна відповісти на запит користувача протягом 2-3 секунд, забезпечуючи швидке завантаження сторінок та відповіді на дії користувача.
2. Продуктивність під високим навантаженням: система має підтримувати стабільну роботу навіть під час інтенсивних навантажень, таких як розпродажів або промоційних акцій.
3. Надійність та відсоток часу роботи: мета – забезпечити 99,5% часу безперебійної роботи, мінімізуючи час простою системи.
4. Сумісність з браузером: система має бути сумісною з основними версіями популярних веббраузерів, таких як Chrome, Firefox, Safari та Edge.
5. Масштабованість: система має бути розроблена з можливістю легкого масштабування, аби витримувати збільшення кількості користувачів та обсягу даних.

Для успішної реалізації проєкту важливо враховувати сучасні тенденції у сфері вебдизайну, а також впроваджувати функціональні можливості, що забезпечують зручний перегляд та замовлення товарів. Завдання цього проєкту полягає у проведенні аналізу предметної галузі, обґрунтованому виборі інструментів для створення вебсервісу, розробці макету інтерфейсу, реалізації інтуїтивно зрозумілого функціоналу та відповідності очікуванням і потребам цільової аудиторії.

На основі проведеного аналізу було сформовано перелік вимог до вебсервісу магазину святкового та весільного вбрання «Казковий образ», який має сприяти ефективному просуванню і продажу продукції. Впровадження такого рішення дозволить скоротити витрати на рекламу, розширити ринок збуту, залучити нових клієнтів і, як наслідок, підвищити рівень прибутковості підприємства. Таким чином, розробка вебсервісу є доцільною як з точки зору бізнесу, так і з позиції кінцевого споживача [13].

2 РОЗРОБКА ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ ВЕСІЛЬНОГО ТА СВЯТКОВОГО ВБРАННЯ «КАЗКОВИЙ ОБРАЗ»

2.1 Аналіз вимог до вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

2.1.1 Побудова моделі варіантів використання

Діаграма варіантів використання – це візуальне представлення функціональних можливостей системи з точки зору користувачів. Вона показує, які дії або сценарії взаємодії з системою доступні для зовнішніх учасників [14].

Основною метою діаграми є зрозумілий опис функціональності системи, визначення ролей користувачів та їх взаємодію та формування основи для вимог до системи [14].

Кожна діаграма складається з акторів і варіантів використання. Відносини між ними можуть бути різних типів: association, include, extend, а також generalization.

На прикладі вебсервісу для магазину весільного та святкового вбрання акторами є Гість, Клієнт, Співробітник. Варіанти використання: створити замовлення, провести оплату, запланувати примірку, керувати товарами, тощо.

Діаграма варіантів використання вебсервісу для магазину весільного та святкового вбрання «Казковий образ» наведена на рис. 2.1.

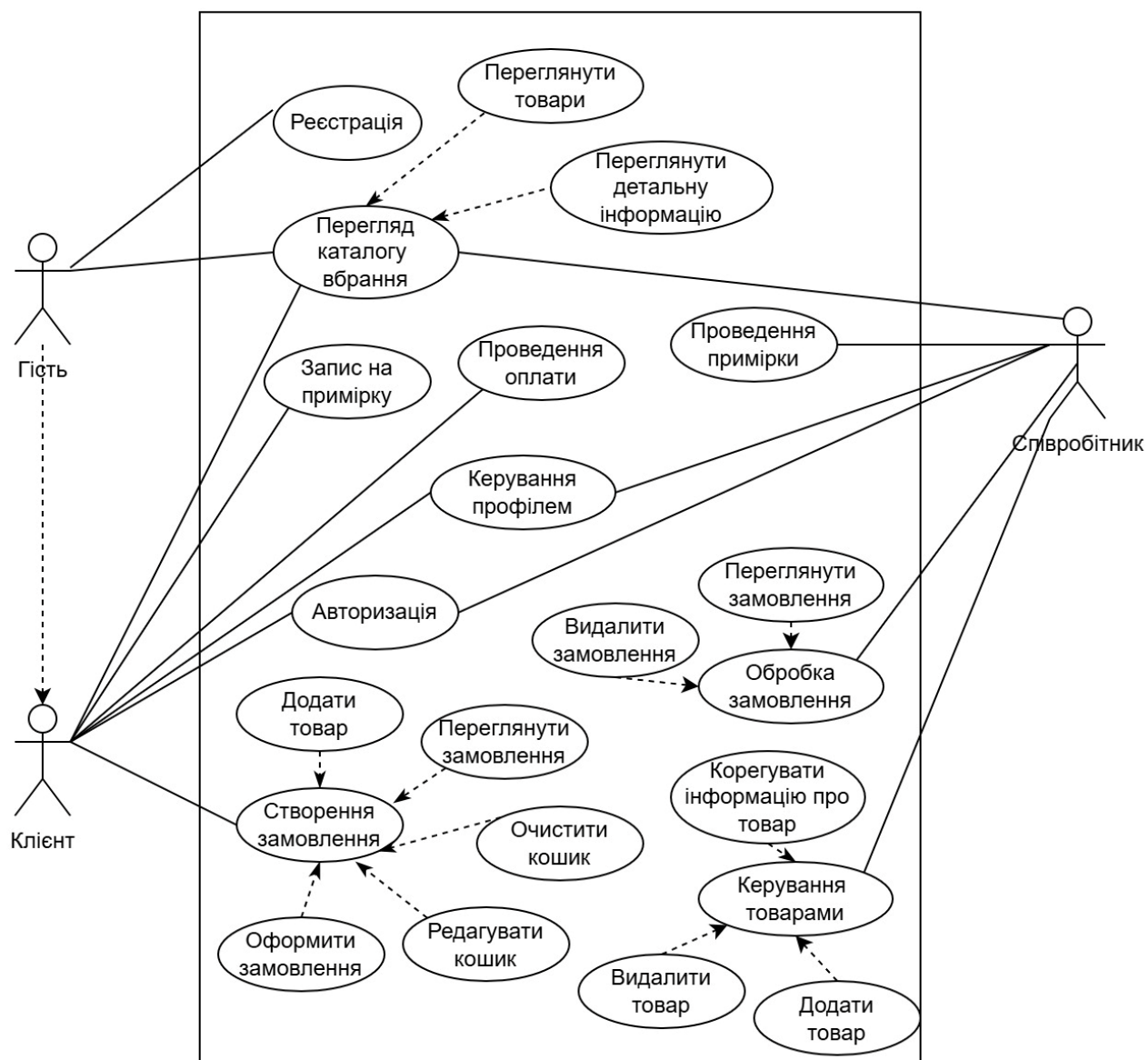


Рисунок 2.1 - Діаграма варіантів використання вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Далі необхідно розробити специфікації варіантів використання вебсервісу для магазину весільного та святкового вбрання «Казковий образ».

2.1.2 Специфікації варіантів використання

Для більш детального розгляду кожного варіанту використання використання вебсервісу для магазину весільного та святкового вбрання «Казковий образ» проаналізуємо кожний за п'ятьма складовими: короткий

опис, передумови, основний потік подій, альтернативний потік подій та постумови. Результати наведені в таблицях 2.1 – 2.10.

Таблиця 2.1 – Опис варіанту використання «Реєстрація»

Актор	Складова	Опис
Гість	Короткий опис	Створення нового облікового запису в системі.
	Передумови	Гість не має зареєстрованого акаунта.
	Основний потік подій	1. Гість відкриває форму реєстрації. 2. Вводить персональні дані (ПІБ, email, пароль). 3. Підтверджує реєстрацію. 4. Система зберігає новий акаунт.
	Альтернативний потік	Email уже існує в базі – система пропонує авторизацію. Некоректні дані – повернення до форми з повідомленням про помилку.
	Постумови	Користувача зареєстровано, доступ відкрито до особистого кабінету.

Таблиця 2.2 – Опис варіанту використання «Авторизація»

Актор	Складова	Опис
Клієнт, Співробітник	Короткий опис	Надання користувачем логіну та пароля для входу в систему.
	Передумови	Існує обліковий запис користувача.
	Основний потік подій	1. Користувач або співробітник відкриває форму входу. 2. Вводить email і пароль. 3. Система перевіряє. 4. Вхід успішний.
	Альтернативний потік	Неправильні дані – повідомлення про помилку, пропозиція відновити пароль.
	Постумови	Користувач отримує доступ до функціоналу системи.

Таблиця 2.3 – Опис варіанту використання «Керування профілем»

Актор	Складова	Опис
1	2	3
Клієнт, Співробітник	Короткий опис	Користувач редагує персональні дані, такі як ім'я, контактна інформація або пароль.
	Передумови	Користувач авторизований та має доступ до налаштувань профілю.

Продовження таблиці 2.3

1	2	3
	Основний потік подій	1. Користувач переходить до розділу “Мій профіль”. 2. Змінює необхідні поля (наприклад, email, телефон, пароль). 3. Підтверджує зміни. 4. Система зберігає оновлену інформацію.
	Альтернативний потік	- Email уже використовується іншим акаунтом → система повідомляє про конфлікт. - Зміни не відповідають формату – вивід помилки.
	Постумови	Дані успішно оновлені; профіль користувача актуалізований і синхронізований із базою.

Таблиця 2.4 – Опис варіанту використання «Перегляд каталогу вбрання»

Актор	Складова	Опис
Гість, Клієнт, Співробітник	Короткий опис	Отримання користувачем доступу до списку товарів і їх характеристик.
	Передумови	Система повинна бути доступна; товари мають бути завантажені у базу.
	Основний потік подій	1. Користувач переходить до розділу “Каталог”. 2. Використовує фільтри для пошуку. 3. Переглядає картки товарів.
	Альтернативний потік	Якщо немає товарів – система показує відповідне повідомлення.
	Постумови	Користувач ознайомлений з асортиментом товарів.

Таблиця 2.5 – Опис варіанту використання «Керування товарами»

Актор	Складова	Опис
1	2	3
Співробітник	Короткий опис	Додавання, редагування або видалення інформації про товари через внутрішній інтерфейс.
	Передумови	Співробітник авторизований, має відповідні права.
	Основний потік подій	1. Перехід до адміністративної панелі. 2. Обрання “Товари”. 3. Додавання/редагування/видалення товару.

Продовження таблиці 2.5

1	2	3
		4. Збереження змін.
	Альтернативний потік	Некоректні або неунікальні дані – система повідомляє про помилку.
	Постумови	Товар доданий, відредагований або видалений

Таблиця 2.6 – Опис варіанту використання «Створення замовлення»

Актор	Складова	Опис
Клієнт	Короткий опис	Користувач обирає товари та формує нове замовлення.
	Передумови	Користувач авторизований; товари доступні.
	Основний потік подій	1. Користувач додає товар до кошика. 2. Перевіряє вміст. 3. Обирає тип замовлення (купівля/оренда). 4. Натискає “Оформити замовлення”.
	Альтернативний потік	Товар недоступний – система виводить сповіщення. Помилка при збереженні – повторна спроба.
	Постумови	Нове замовлення створено, очікує оплати або обробку.

Таблиця 2.7 – Опис варіанту використання «Проведення оплати»

Актор	Складова	Опис
Клієнт	Короткий опис	Користувач виконує оплату за оформлене замовлення.
	Передумови	Замовлення має статус "підтверджено", сума до оплати сформована.
	Основний потік подій	1. Користувач обирає спосіб оплати. 2. Вводить реквізити або виконує дію (наприклад, QR код). 3. Система фіксує оплату.
	Альтернативний потік	Помилка транзакції – користувачу пропонується повторити. Недійсна карта – повідомлення про невдачу.
	Постумови	Платіж здійснено, замовлення оновлено до статусу "оплачене".

Таблиця 2.8 – Опис варіанту використання «Обробка замовлення»

Актор	Складова	Опис
Співробітник	Короткий опис	Співробітник переглядає замовлення, готує товари та оновлює статус.
	Передумови	Є хоча б одне нове або підтверджене замовлення у списку.
	Основний потік подій	1. Працівник авторизується. 2. Обирає замовлення зі списку. 3. Підтверджує оплату. 4. Змінює статус (наприклад, “готово”).
	Альтернативний потік	Якщо товар недоступний – працівник зв'язується з клієнтом; замовлення може бути скасовано.
	Постумови	Оновлено статус замовлення

Таблиця 2.9 – Опис варіанту використання «Запис на примірku»

Актор	Складова	Опис
Клієнт	Короткий опис	Клієнт обирає дату/час та ініціює примірku обраного вбрання.
	Передумови	Користувач авторизований; товари доступні до примірku.
	Основний потік подій	1. Клієнт переходить до розділу “Примірku”. 2. Обирає вбрання. 3. Обирає дату й час. 4. Надсилає запит.
	Альтернативний потік	Час зайнятий – система пропонує альтернативні слоти. Сеанс не створено – повідомлення про помилку.
	Постумови	Примірku створена та запланована.

Таблиця 2.10 – Опис варіанту використання «Проведення примірku»

Актор	Складова	Опис
1	2	3
Співробітник	Короткий опис	Співробітник проводить заплановану примірku з клієнтом, фіксує результати і коментарі.
	Передумови	Існує запланована примірku в системі, клієнт і співробітник ідентифіковані.
	Основний потік подій	1. Співробітник входить до системи. 2. Переглядає заплановані примірku. 3. Запрошує клієнта у кабінку.

Продовження таблиці 2.10

1	2	2
		4. Надає вбрання для примірки. 5. Вносить нотатки щодо розміру, вподобань, реакцій клієнта. 6. Завершує сесію примірки.
	Альтернативний потік	- Клієнт не з'явився → система фіксує відсутність. - Вбрання відсутнє в наявності → заміна або скасування примірки.
	Постумови	Примірка завершена, інформацію внесено в систему, дані доступні для подальших замовлень або рекомендацій.

Маючи детальні специфікації варіантів використання використання вебсервісу для магазину весільного та святкового вбрання «Казковий образ» можемо переходити для подальшого проектування вебсервісу.

2.2 Архітектура вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Архітектура вебсервісу магазину весільного та святкового вбрання «Казковий образ» побудована на основі клієнт-серверної моделі, яка передбачає чітке розмежування між користувацьким інтерфейсом і серверною логікою. Клієнтська частина виконує роль інтерфейсу, з яким безпосередньо взаємодіє користувач, а серверна – обробляє запити, виконує бізнес-логіку та забезпечує збереження й обробку даних. Вся комунікація між клієнтом і сервером здійснюється через мережеві протоколи HTTP/HTTPS, що дозволяє реалізувати зручну модель доступу через браузер незалежно від типу пристрою користувача [15].

У межах серверної частини застосовано архітектурний шаблон MVC (Model–View–Controller), що забезпечує логічний поділ функціональних обов'язків усередині застосунку. Моделі відповідають за взаємодію з базою

даних та реалізують правила доступу до сутностей системи, таких як клієнти, товари, замовлення, примірки та платежі. Контролери приймають запити від користувача, звертаються до моделей для виконання відповідних дій і передають результати у вигляді структурованих даних до представлень. Представлення формують HTML-контент, який відображається у браузері користувача і забезпечує зворотний зв'язок та візуальну взаємодію [16].

Механізми маршрутизації дозволяють обробляти запити динамічно, забезпечуючи доступ до різних функціональних модулів застосунку без дублювання логіки. Клієнтська частина забезпечує адаптивне і зрозуміле середовище для взаємодії з користувачем, включаючи перегляд товарів, оформлення замовлень, керування профілем і запис на примірку. Така організація архітектури гарантує стабільну роботу вебсервісу, зручність в експлуатації та подальшу підтримку в умовах змін бізнес-вимог [17].

Архітектура вебсервісу магазину весільного та святкового вбрання «Казковий образ» представлена на рисунку 2.2.

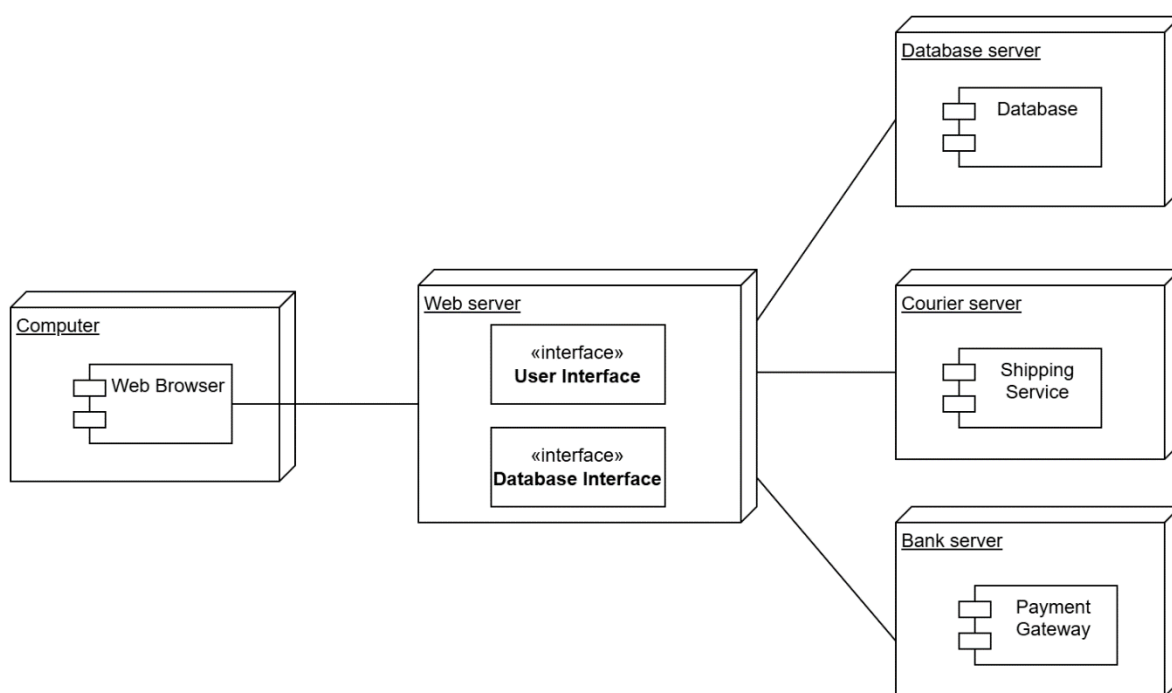


Рисунок 2.2 - Архітектура вебсервісу магазину весільного та святкового вбрання «Казковий образ»

Така архітектура системи є гнучкою, масштабованою та підтримує повторне використання коду.

2.3 Проєкт вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

2.3.1 Ескізний проєкт вебсервісу

Розробка концептуальної моделі даних

Концептуальна модель даних – це абстрактне логічне представлення інформації, яка описує об'єкти предметної області, їхні властивості та зв'язки між ними [18]. Вона не залежить від конкретної системи управління базами даних (СУБД) і призначена для того, щоб зрозуміло та систематизовано описати вимоги до структури інформації ще до технічної реалізації.

Ключовою метою концептуального моделювання є узгодження розуміння між всіма зацікавленими сторонами. У такій моделі використовуються сутності, атрибути, а також зв'язки, які описують тип взаємодії між об'єктами.

Ці моделі часто зображають у вигляді ER-діаграм, що наочно демонструють структуру даних та типи зв'язків.

Концептуальна модель даних для вебсервісу для магазину весільного та святкового вбрання «Казковий образ» представлена на рисунку 2.3.

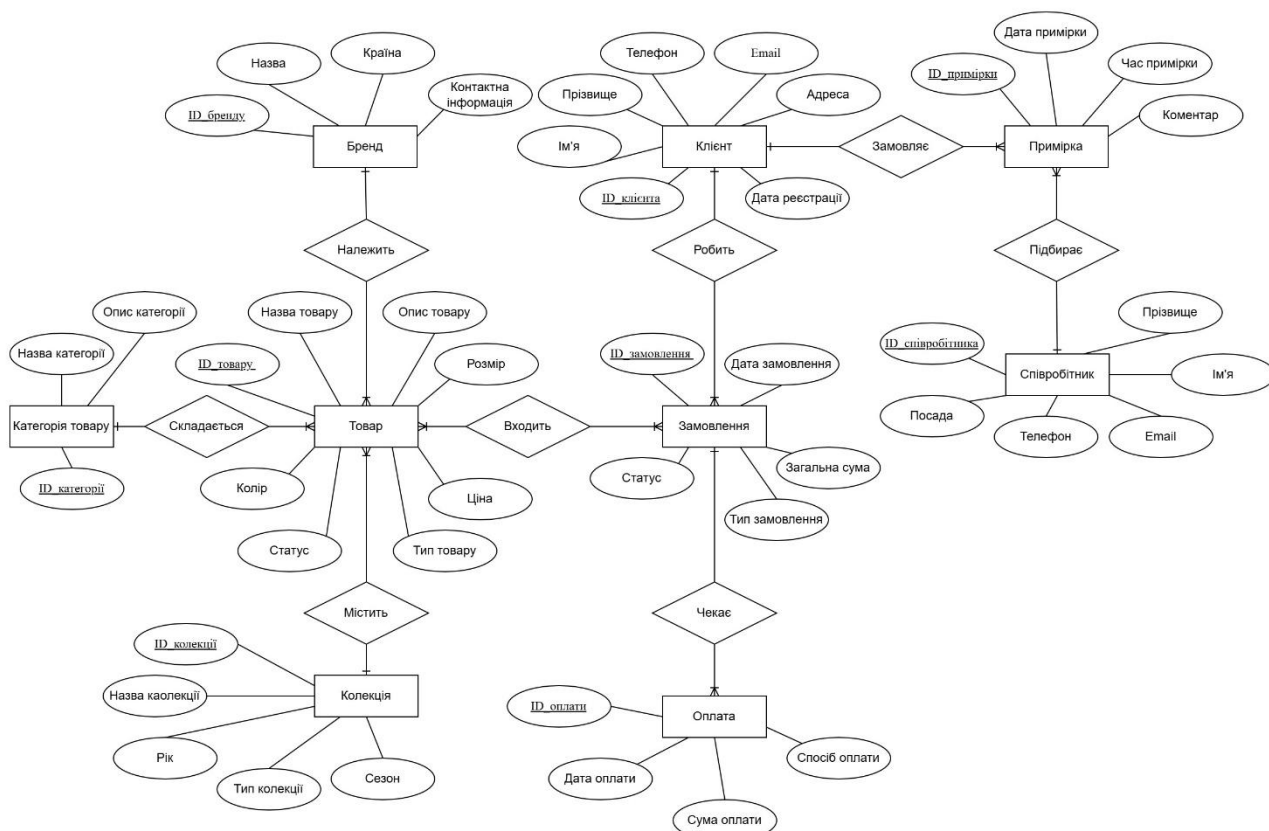


Рисунок 2.3 – Концептуальна модель даних для вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Розглянемо детальний опис основних сутностей концептуальної моделі даних для вебсервісу для магазину весільного та святкового вбрання «Казковий образ» та визначемо атрибути для кожної сутності.

Сутність «Клієнт» має наступні атрибути:

- ID_клієнта;
- Ім'я;
- Прізвище;
- Телефон;
- Email;
- Адреса;
- Дата реєстрації.

Сутність «Замовлення» має наступні атрибути:

- ID_замовлення;
- Дата замовлення;

- Статус;
- Загальна сума;
- Тип замовлення.

Сутність «Товар» має наступні атрибути:

- ID_товару;
- Назва товару;
- Опис товару;
- Розмір;
- Колір;
- Ціна;
- Тип товару;
- Статус.

Сутність «Колекція» має наступні атрибути:

- ID_колекції;
- Назва колекції;
- Рік;
- Сезон;
- Опис колекції.

Сутність «Категорія товару» має наступні атрибути:

- ID_категорії;
- Назва категорії;
- Опис категорії.

Сутність «Оплата» має наступні атрибути:

- ID_оплати;
- Дата оплати;
- Сума;
- Спосіб оплати.

Сутність «Примірка» має наступні атрибути:

- ID_примірки;
- Дата примірки;

- Час примірки;
- Коментар.

Сутність «Співробітник» має наступні атрибути:

- ID_співробітника;
- Ім'я;
- Прізвище;
- Посада;
- Телефон;
- Email.

Сутність «Бренд» має наступні атрибути:

- ID_бренду;
- Назва;
- Країна;
- Контактна інформація.

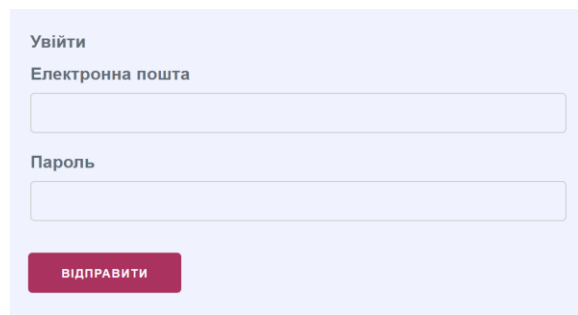
Розробка прототипу інтерфейсу користувача

Спираючись на опис діаграми варіантів використання вебсервісу для магазину весільного та святкового вбрання «Казковий образ», було розроблено інтерфейс користувача вебсервісу.

Інтерфейс користувача – це сукупність засобів і механізмів, які забезпечують взаємодію людини з програмною системою або пристроєм. Іншими словами, це те, як користувач бачить, сприймає та керує функціональністю застосунку: кнопки, форми, меню, повідомлення, а також візуальний стиль, структуру й логіку переходів [19].

Хороший інтерфейс користувача має бути інтуїтивно зрозумілим, зручним, естетично привабливим і доступним для різних типів користувачів. У випадку вебсервісу магазину весільного вбрання, це сторінки товарів, кошик, форми реєстрації, модулі примірок, повідомлення про оплату тощо. Саме інтерфейс визначає, наскільки легко й приємно користувач зможе досягти бажаних дій у системі [20].

Розроблений ескіз графічного інтерфейсу вебсервісу представлено на рисунках 2.4-2.6.



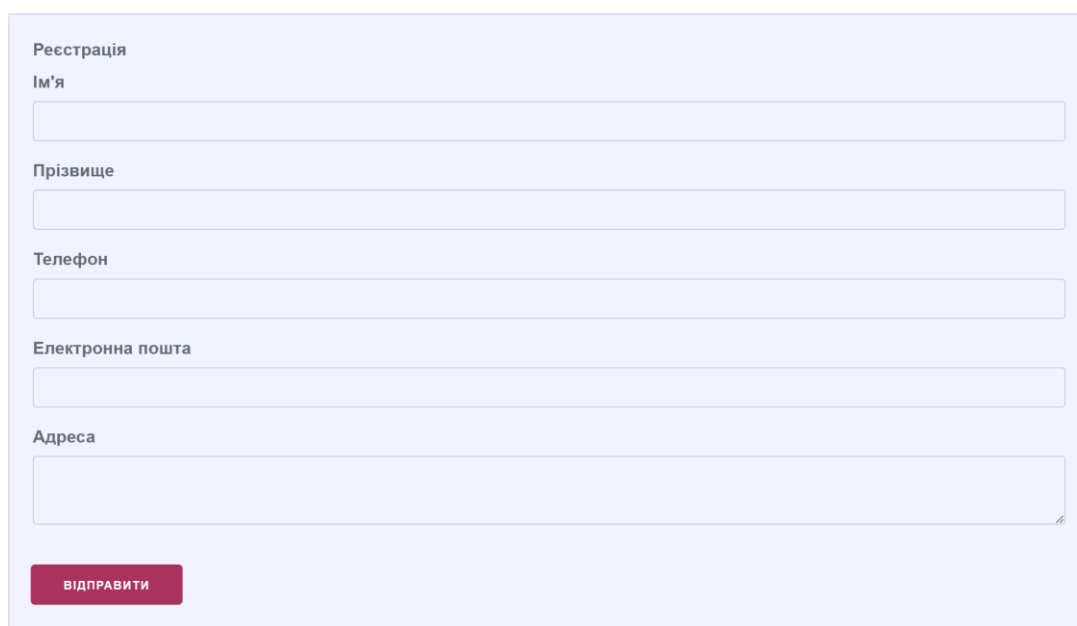
Увійти

Електронна пошта

Пароль

ВІДПРАВИТИ

Рисунок 2.4 – Форма авторизації користувача вебсервісу для магазину весільного та святкового вбрання «Казковий образ»



Реєстрація

Ім'я

Прізвище

Телефон

Електронна пошта

Адреса

ВІДПРАВИТИ

Рисунок 2.5 – Форма реєстрації користувача вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Колекція					НОВА КОЛЕКЦІЯ		
ID	Назва	Сезон	Рік	Дії			
1	Весна мрії	Весна	2 024	Переглянути Редагувати Видалити			
2	Зимова казка	Зима	2 024	Переглянути Редагувати Видалити			
3	Літній шик	Літо	2 025	Переглянути Редагувати Видалити			
4	Осінній гламур	Осінь	2 024	Переглянути Редагувати Видалити			
5	Вічна класика	Всесезонна	2 025	Переглянути Редагувати Видалити			
					< попередня наступна >		

Рисунок 2.6 – Форма співробітника для керування колекціями вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Після розробки ескізного проєкту вебсервісу для магазину весільного та святкового вбрання «Казковий образ», можна переходити до розробки технічного проєкту вебсервісу для магазину весільного та святкового вбрання «Казковий образ».

2.3.2 Технічний проєкт вебсервісу

Діаграма класів – це один із ключових типів UML-діаграм, який використовується для моделювання структури системи на об’єктно-орієнтованому рівні. Вона відображає класи, їхні атрибути, методи, а також зв’язки між класами – такі як асоціації, агрегації, композиції та наслідування [21]. Діаграма класів дозволяє наочно показати, з яких елементів складається система та як вони взаємодіють між собою.

Особливістю цієї діаграми є її орієнтація на довгострокову структуру даних – тобто вона не лише показує, що система робить, а й як вона це реалізує на рівні класів.

У контексті розробки вебсервісу для магазину весільного та святкового вбрання «Казковий образ», діаграма класів допомагає перетворити концептуальну модель даних на об’єктну, ближчу до майбутньої реалізації програмного коду (рис. 2.7).

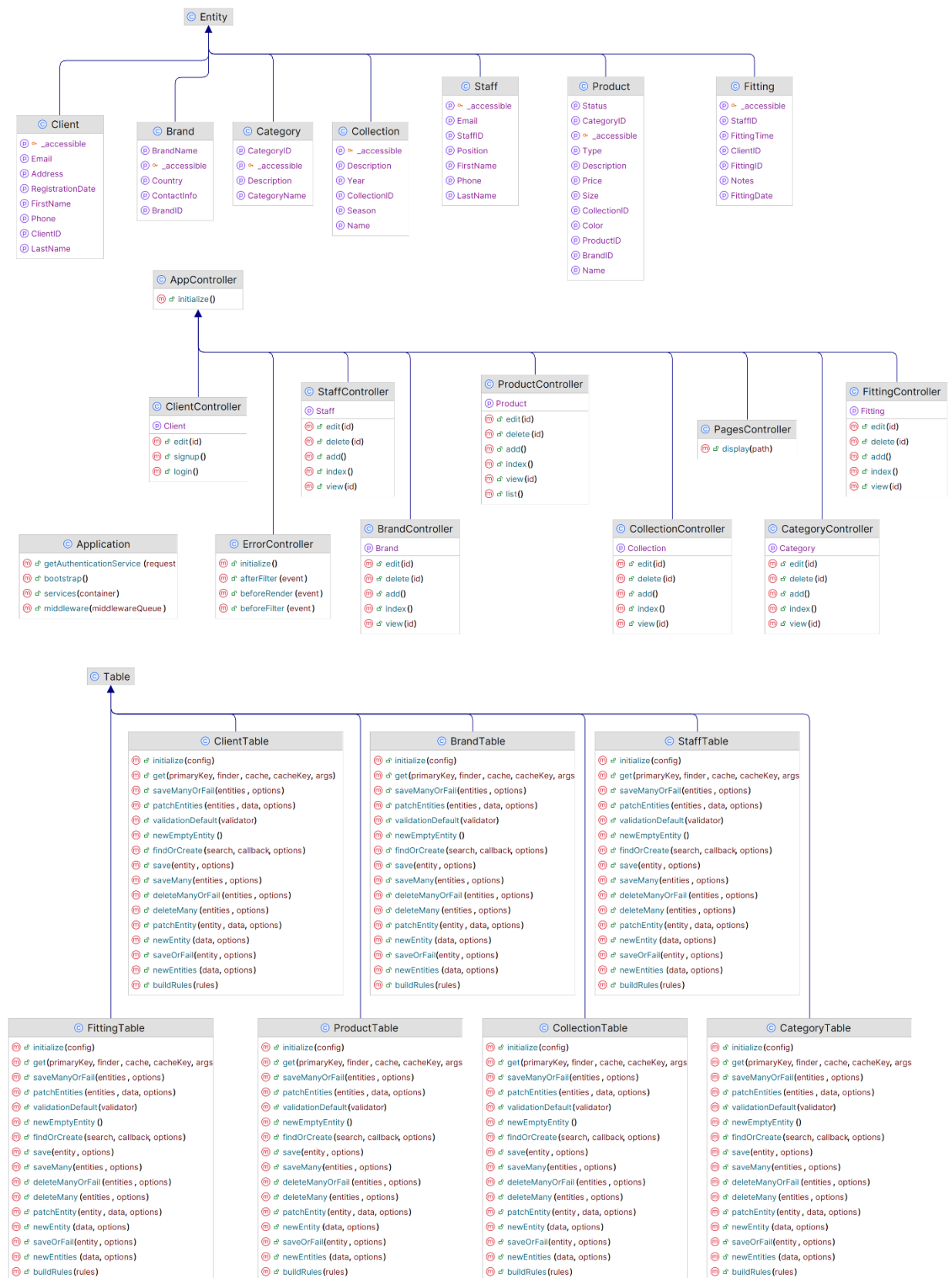


Рисунок 2.7 - Діаграма класів вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Детальна специфікація класів вебсервісу для магазину весільного та святкового вбрання «Казковий образ» наведена в таблицях 2.11-2.20.

Таблиця 2.11 – Специфікація класу «Client»

Складова	Опис
Відповідальність	Представляє зареєстрованого клієнта магазину.
Атрибути	ClientID, FirstName, LastName, Phone, Email, Address, RegistrationDate
Операції зі специфікаціями	Register() – реєстрація нового клієнта Login() – авторизація UpdateProfile() – редагування профілю ViewOrders() – перегляд власних замовлень ScheduleFitting() – запис на примірку

Таблиця 2.12 – Специфікація класу «Staff»

Складова	Опис
Відповідальність	Представляє працівника магазину, що виконує обслуговування клієнтів.
Атрибути	StaffID, FirstName, LastName, Position, Phone, Email
Операції зі специфікаціями	ViewFittings() – перегляд запланованих примірок ConductFitting() – проведення примірки ManageProducts() – керування товарами

Таблиця 2.13 – Специфікація класу «Category»

Складова	Опис
Відповідальність	Групування товарів за типом (весільне, вечірнє тощо).
Атрибути	CategoryID, CategoryName, Description
Операції зі специфікаціями	AddCategory() – створення нової категорії UpdateCategory() – редагування AssignToProduct() – зв'язування з товаром

Таблиця 2.14 – Специфікація класу «Brand»

Складова	Опис
Відповідальність	Відображає виробника або постачальника товарів.
Атрибути	BrandID, BrandName, Country, ContactInfo
Операції зі специфікаціями	CreateBrand() – додавання нового бренду EditBrand() – редагування ViewBrandProducts() – перегляд товарів бренду

Таблиця 2.15 – Специфікація класу «Collection»

Складова	Опис
Відповідальність	Відображає дизайнерську або сезонну колекцію товарів.
Атрибути	CollectionID, Name, Season, Year, Description
Операції зі специфікаціями	AddCollection() – створення колекції LinkProducts() – прив’язка товарів до колекції EditCollection() – оновлення даних

Таблиця 2.16 – Специфікація класу «Product»

Складова	Опис
Відповідальність	Представляє вбрання або аксесуар, що доступний у магазині.
Атрибути	ProductID, Name, Description, Size, Color, Price, Type, Status
Операції зі специфікаціями	ViewDetails() – перегляд інформації AddToCart() – додати до замовлення UpdateAvailability() – оновлення статусу AssignCategory() – прив’язка до категорії/колекції/бренду

Таблиця 2.17 – Специфікація класу «Order»

Складова	Опис
Відповідальність	Представляє замовлення клієнта із відповідною інформацією.
Атрибути	OrderID, OrderDate, Status, TotalAmount, OrderType
Операції зі специфікаціями	CreateOrder() – оформлення нового замовлення UpdateStatus() – зміна статусу LinkPayment() – прив’язка платежу CancelOrder() – скасування

Таблиця 2.18 – Специфікація класу «Order_Item»

Складова	Опис
Відповідальність	Представляє окрему позицію в межах замовлення.
Атрибути	OrderID, ProductID, Quantity, UnitPrice, Discount
Операції зі специфікаціями	AddItem() – додати товар до замовлення EditQuantity() – змінити кількість CalculateSubtotal() – розрахунок проміжної суми

Таблиця 2.19 – Специфікація класу «Payment»

Складова	Опис
Відповідальність	Фіксує деталі оплати за замовлення.
Атрибути	PaymentID, PaymentDate, Amount, PaymentMethod
Операції зі специфікаціями	MakePayment() – створити запис про оплату VerifyPayment() – підтвердження операції AttachToOrder() – прив’язка до замовлення

Таблиця 2.20 – Специфікація класу «Fitting»

Складова	Опис
Відповідальність	Представляє примірku одягу між клієнтом і співробітником.
Атрибути	FittingID, FittingDate, FittingTime, Notes
Операції зі специфікаціями	ScheduleFitting() – запланувати примірku RecordFeedback() – внести нотатки CompleteFitting() – завершити процес

Динамічна модель програмного забезпечення

Динамічну модель вебсервісу для магазину весільного та святкового вбрання «Казковий образ» можна представити у вигляді діаграми послідовностей.

Діаграма послідовностей – це тип UML-діаграми, що моделює взаємодію між об'єктами або компонентами системи у часовій послідовності. Вона показує, як об'єкти обмінюються повідомленнями для виконання певного сценарію, ілюструючи хто з ким взаємодіє, що передає і в якій черзі. На діаграмі зображаються учасники (актор, інтерфейс, сервіси, база даних) у вигляді вертикальних ліній життя, а повідомлення між ними – горизонтальними стрілками. Це дозволяє чітко простежити логіку процесу крок за кроком [22].

Динамічна модель вебсервісу представлена у вигляді діаграми послідовностей на рисунку 2.8.

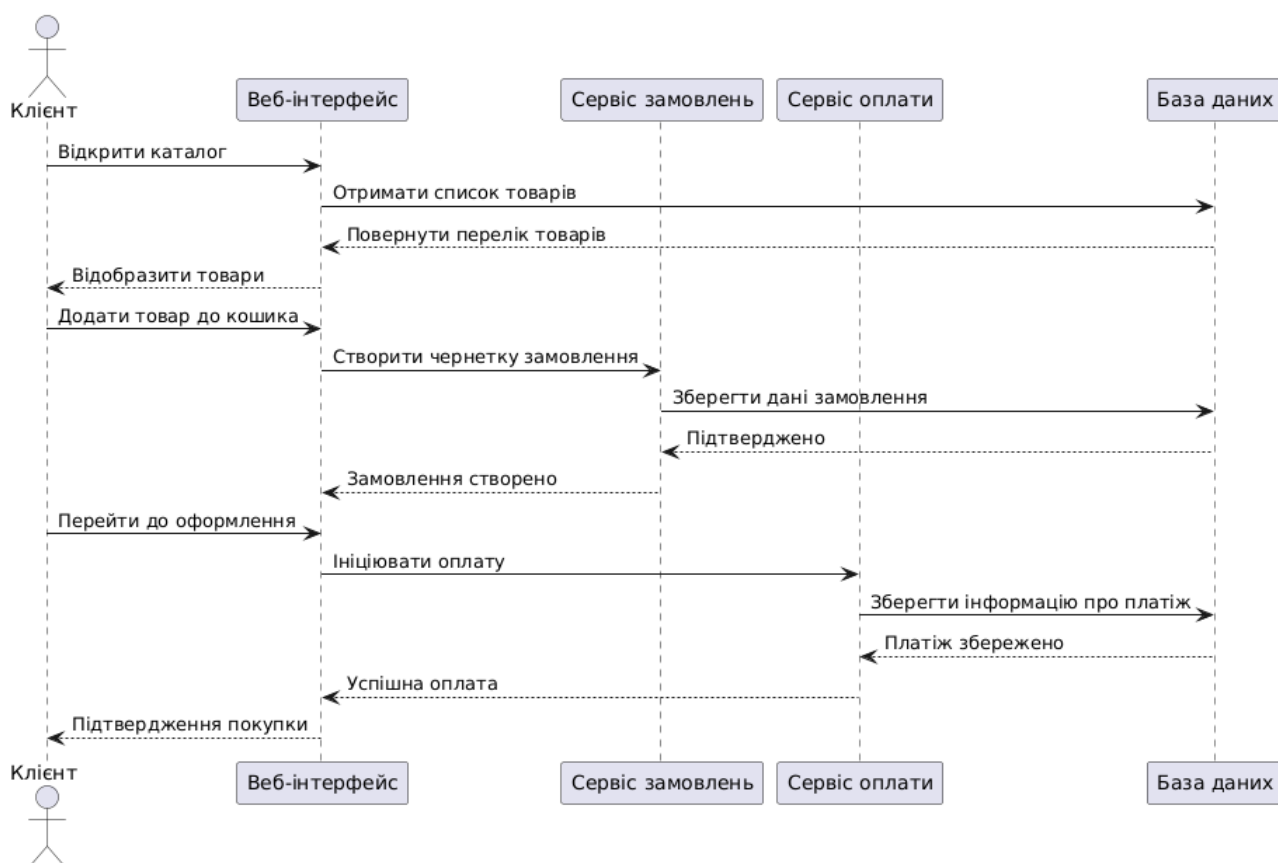


Рисунок 2.8 - Діаграма послідовності вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Діаграма послідовностей демонструє покрокову взаємодію клієнта із системою під час типового процесу замовлення вбрання через вебінтерфейс. Процес починається з того, що клієнт переглядає каталог товарів, отримуючи дані з бази даних через інтерфейс. Після перегляду клієнт додає вибраний товар до кошика, в результаті чого створюється чернетка замовлення. Цей запит обробляє сервіс замовлень, зберігаючи відповідну інформацію в базі [21].

Далі клієнт переходить до оформлення, ініціюючи платіж через сервіс оплати. Система реєструє платіж у базі даних та повідомляє клієнта про успішне завершення операції. Діаграма ілюструє не лише самі етапи процесу, а й взаємозв'язки між компонентами системи (інтерфейсом, сервісами, базою даних), що дозволяє візуалізувати архітектуру обміну повідомленнями та забезпечити цілісне розуміння логіки обробки замовлення.

Розробка логічної моделі даних

На основі концептуальної моделі даних було розроблено логічну модель даних. Логічна модель даних – це формалізоване уявлення структури даних, яке описує сутності, їхні атрибути та зв'язки між ними незалежно від конкретної СУБД. Вона деталізує концептуальну модель, надаючи чітке уявлення про логіку організації інформації в системі [22].

Логічна модель даних вебсервісу для магазину весільного та святкового вбрання «Казковий образ» наведена на рисунку 2.9.

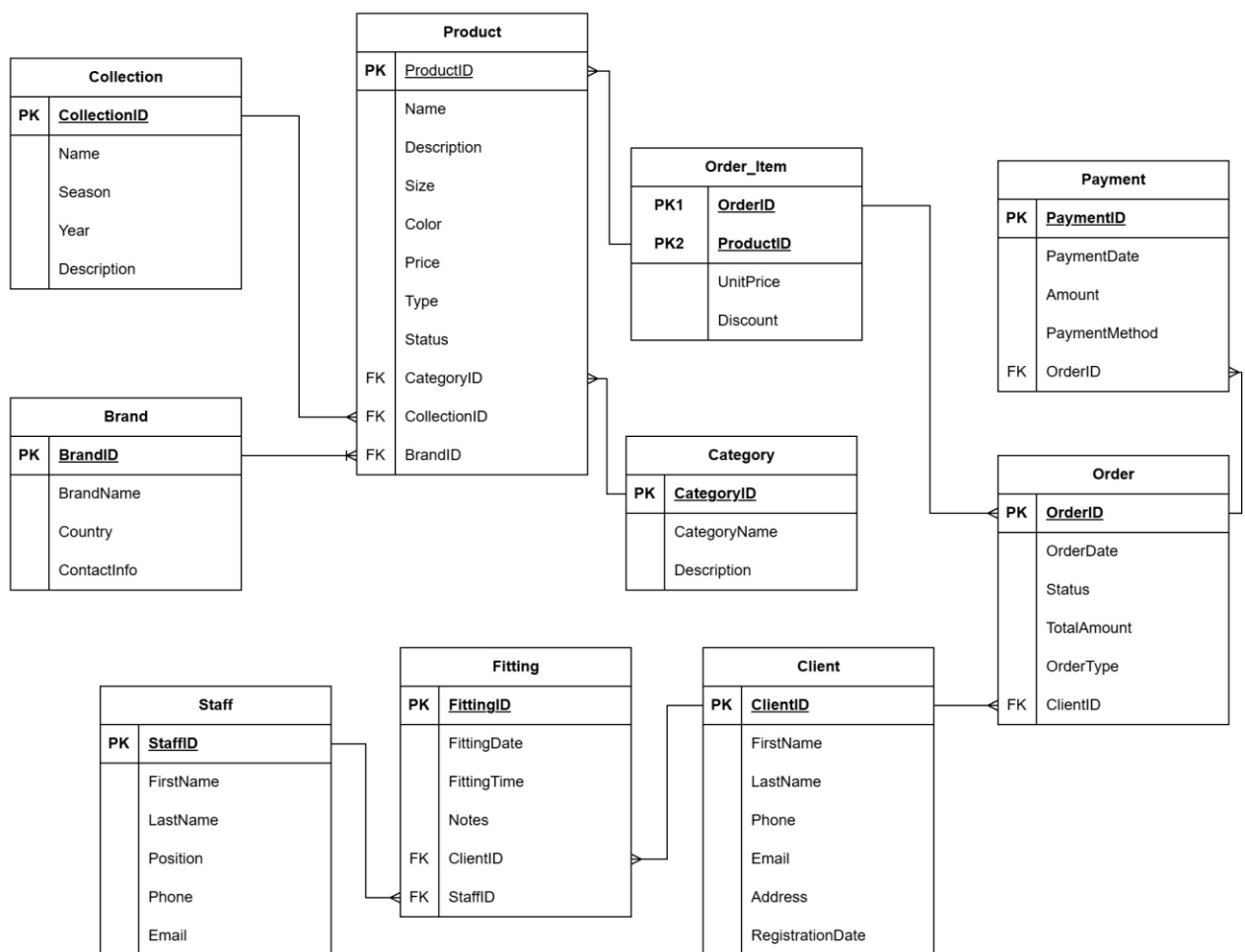


Рисунок 2.9 - Логічна модель даних вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Після розробки технічного проєкту вебсервісу для магазину весільного та святкового вбрання «Казковий образ», можна перейти до розробки робочого

проєкту вебсервісу для магазину весільного та святкового вбрання «Казковий образ».

2.3.3 Робочий проєкт програмного забезпечення

Вибір мови програмування та СКБД

У процесі розробки вебсервісу для магазину весільного та святкового вбрання було особливо важливо забезпечити поєднання зручності, надійності, швидкодії та гнучкої підтримки функціональної логіки. З огляду на це, після аналізу можливих технологічних стеків було прийнято рішення на користь використання мови програмування PHP 8.3, яка характеризується стабільністю, широкою підтримкою спільноти та високою продуктивністю при роботі з вебсервісами. Завдяки сучасній архітектурі PHP 8.3 розробка є максимально сумісною з принципами об'єктно-орієнтованого програмування, а також забезпечує покращену обробку виключень та строгішу типізацію [24].

У якості фреймворку обрано CakePHP 5, який дотримується патерну MVC (Model-View-Controller), що дозволяє чітко розмежувати бізнес-логіку, представлення та зберігання даних [25]. CakePHP 5 надає потужні засоби для маршрутизації, генерації CRUD-операцій, валідації форм, а також інтегровану ORM (Object Relational Mapping), що значно полегшує взаємодію з базою даних без необхідності прямого написання SQL-коду. Використання фреймворку дозволяє також суттєво скоротити кількість рутинного коду та забезпечити структурованість і масштабованість застосунку [26].

Для створення клієнтської частини було використано сучасні технології веб-розмітки: HTML5 – для структурування вмісту, CSS3 – для стилізації та адаптивного відображення інтерфейсу на різних пристроях, а також PHP – для інтеграції динамічного контенту [27]. Такий підхід забезпечив можливість реалізувати зручний, інтуїтивно зрозумілий і привабливий інтерфейс взаємодії з користувачем, що враховує UX/UI-принципи.

В якості системи керування базами даних було обрано MariaDB 10, яка є надійною та високопродуктивною реляційною СКБД з відкритим кодом. Вона повністю сумісна з MySQL, підтримує ACID-властивості транзакцій, зовнішні ключі, представлення, індексацію, а також інструменти для резервного копіювання та відновлення. Використання MariaDB дозволяє ефективно реалізувати складну структуру сутностей, таких як товари, замовлення, клієнти, оплати та примірки, з дотриманням вимог до цілісності й узгодженості даних.

Для розробки та налагодження проєкту було обрано інтегроване середовище розробки NetBeans IDE, яке має повну підтримку PHP, інтеграцію з фреймворками та системами контролю версій (зокрема Git), а також підтримку автозаповнення коду, відладку і профілювання. Завдяки цьому середовищу забезпечено зручну навігацію в межах проєкту, підвищено ефективність розробки та полегшено інтеграцію різних компонентів системи.

Таким чином, обраний програмно-технологічний стек забезпечує надійну основу для функціонування, подальшого розвитку та масштабування вебсервісу відповідно до вимог реального бізнесу.

Фізична модель бази даних

Фізична модель бази даних вебсервісу для магазину весільного та святкового вбрання «Казковий образ» відображає реалізацію логічної структури бази даних у СУБД, з точним визначенням типів даних, обмежень, ключів і зв'язків між таблицями. Вона дозволяє системі ефективно зберігати, обробляти та забезпечувати цілісність даних у рамках реального застосунку [28].

Фізична модель даних вебсервісу для магазину весільного та святкового вбрання «Казковий образ» представлена на рисунку 2.10.

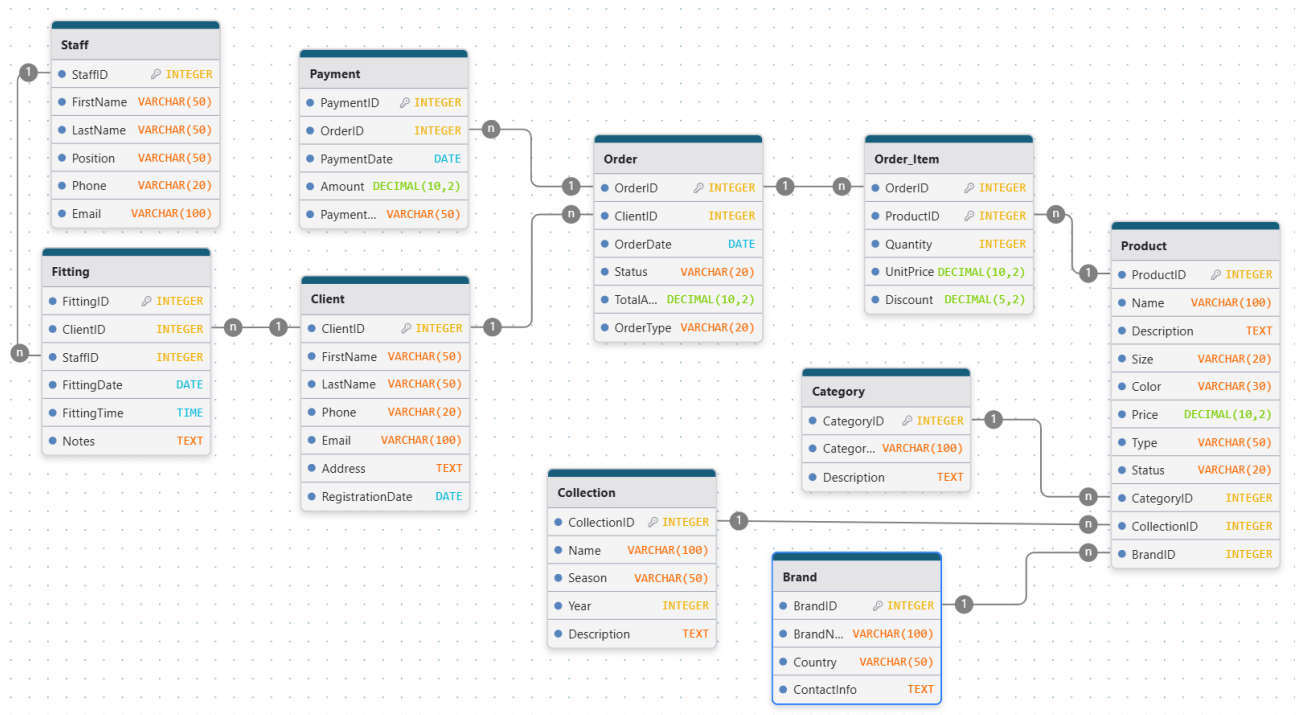


Рисунок 2.10 - Фізична модель даних вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Специфікації до фізичної моделі даних наведено в таблицях 2.21-2.30.

Таблиця 2.21 – Опис структури таблиці «Client»

Ім'я поля	Тип	Ключ	Обов'язковість
ClientID	INTEGER	PK	Так
FirstName	VARCHAR(50)		Так
LastName	VARCHAR(50)		Так
Phone	VARCHAR(20)		Так
Email	VARCHAR(100)		Так
Address	TEXT		Так
RegistrationDate	DATE		Так

Таблиця 2.22 – Опис структури таблиці «Staff»

Ім'я поля	Тип	Ключ	Обов'язковість
StaffID	INTEGER	PK	Так
FirstName	VARCHAR(50)		Так
LastName	VARCHAR(50)		Так
Position	VARCHAR(50)		Так
Phone	VARCHAR(20)		Так
Email	VARCHAR(100)		Так

Таблиця 2.23 – Опис структури таблиці «Category»

Ім'я поля	Тип	Ключ	Обов'язковість
CategoryID	INTEGER	PK	Так
CategoryName	VARCHAR(100)		Так
Description	TEXT		Так

Таблиця 2.24 – Опис структури таблиці «Brand»

Ім'я поля	Тип	Ключ	Обов'язковість
BrandID	INTEGER	PK	Так
BrandName	VARCHAR(100)		Так
Country	VARCHAR(50)		Так
ContactInfo	TEXT		Так

Таблиця 2.25 – Опис структури таблиці «Collection»

Ім'я поля	Тип	Ключ	Обов'язковість
CollectionID	INTEGER	PK	Так
Name	VARCHAR(100)		Так
Season	VARCHAR(50)		Так
Year	INTEGER		Так

Таблиця 2.26 – Опис структури таблиці «Product»

Ім'я поля	Тип	Ключ	Обов'язковість
ProductID	INTEGER	PK	Так
Name	VARCHAR(100)		Так
Description	TEXT		Так
Size	VARCHAR(20)		Так
Color	VARCHAR(30)		Так
Price	DECIMAL(10,2)		Так
Type	VARCHAR(50)		Так
Status	VARCHAR(20)		Так
CategoryID	INTEGER	FK	Так
CollectionID	INTEGER	FK	Так
BrandID	INTEGER	FK	Так

Таблиця 2.27 – Опис структури таблиці «Order»

Ім'я поля	Тип	Ключ	Обов'язковість
1	2	3	4
OrderID	INTEGER	PK	Так
ClientID	INTEGER	FK	Так

Продовження таблиці 2.27

1	2	3	4
OrderDate	DATE		Так
Status	VARCHAR(20)		Так
TotalAmount	DECIMAL(10,2)		Так
OrderType	VARCHAR(20)		Так

Таблиця 2.28 – Опис структури таблиці «Order_Item»

Ім'я поля	Тип	Ключ	Обов'язковість
OrderID	INTEGER	PK, FK	Так
ProductID	INTEGER	PK, FK	Так
Quantity	INTEGER		Так
UnitPrice	DECIMAL(10,2)		Так
Discount	DECIMAL(5,2)		Так

Таблиця 2.29 – Опис структури таблиці «Payment»

Ім'я поля	Тип	Ключ	Обов'язковість
PaymentID	INTEGER	PK	Так
OrderID	INTEGER	FK	Так
PaymentDate	DATE		Так
Amount	DECIMAL(10,2)		Так
PaymentMethod	VARCHAR(50)		Так

Таблиця 2.30 – Опис структури таблиці «Fitting»

Ім'я поля	Тип	Ключ	Обов'язковість
FittingID	INTEGER	PK	Так
ClientID	INTEGER	FK	Так
StaffID	INTEGER	FK	Так
FittingDate	DATE		Так
FittingTime	TIME		Так
Notes	TEXT		Так

Текст вебсервісу для магазину весільного та святкового вбрання «Казковий образ» наведено в додатку Б.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБСЕРВІСУ ДЛЯ МАГАЗИНУ ВЕСІЛЬНОГО ТА СВЯТКОВОГО ВБРАННЯ «КАЗКОВИЙ ОБРАЗ»

У результаті виконання кваліфікаційної роботи було вирішено практичну задачу інженерії програмного забезпечення, а саме розроблено вебсервіс для магазину весільного та святкового вбрання «Казковий образ», що характеризується складною структурою взаємодій та широким спектром функціональних вимог. Було реалізовано вебсервіс, який повністю відповідає поставленим функціональним, інформаційним та технічним критеріям, включаючи обробку замовлень, керування товарами, проведення примірок і облік оплат.

Сервіс розроблено із застосуванням сучасних технологій веб-розробки. Серверна частина реалізована на мові програмування PHP 8.3 з використанням Framework CakePHP 5, що забезпечує ефективну обробку запитів користувачів, взаємодію з базою даних та підтримку бізнес-логіки. Для зберігання даних використовується реляційна база даних MariaDB 10. Клієнтська частина створена з використанням HTML 5 / CSS 3 та PHP 8.3 і забезпечує зручний та інтуїтивно зрозумілий інтерфейс.

Головна сторінка вебсервісу для магазину весільного та святкового вбрання «Казковий образ» представлена на рисунку 3.1.

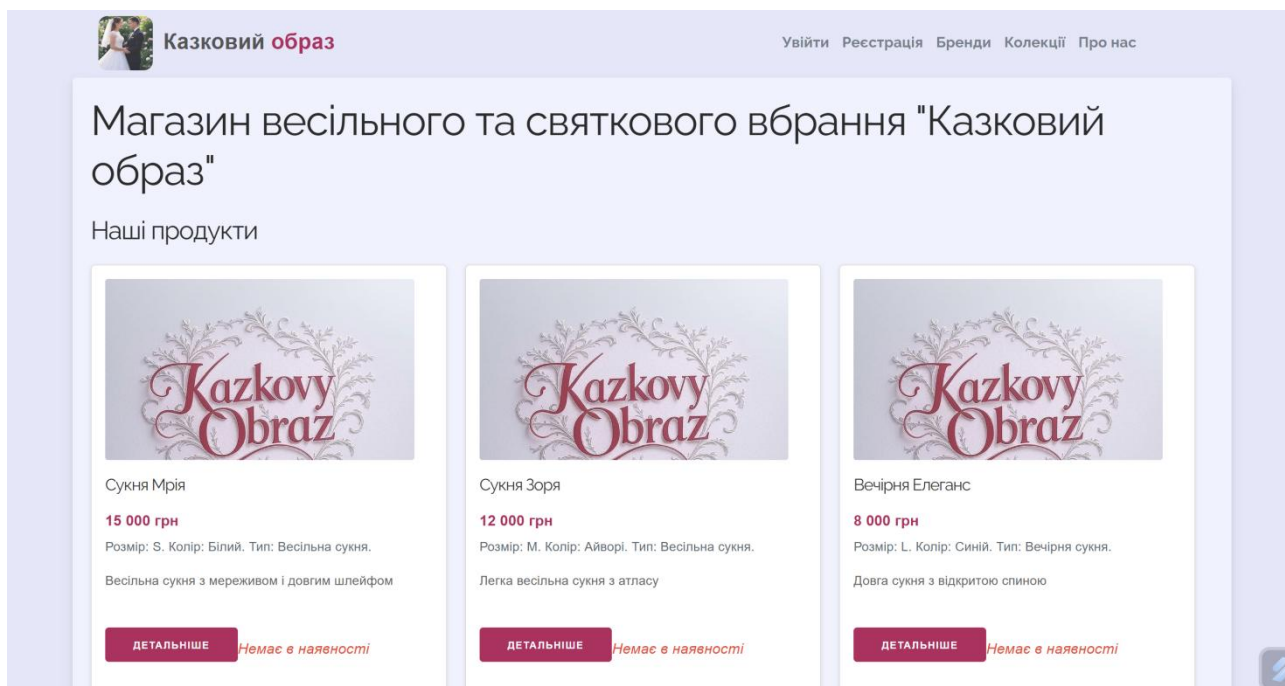


Рисунок 3.1 – Головна сторінка вебсервісу для магазину весільного та святкового вбрання «Казковий образ»

Програмне забезпечення підтримує ключові функції, необхідні для повноцінного функціонування вебсервісу: реєстрацію та авторизацію клієнтів, додавання й редагування товарів, формування замовлень, запис на примірку, оплату та перегляд історії замовлень.

Код програми наведено у додатку Б – Код програмного забезпечення для магазину весільного та святкового вбрання «Казковий образ». Опис програми представлено у додатку В – Опис програмного забезпечення для магазину весільного та святкового вбрання «Казковий образ». Для експлуатації програмного забезпечення також було розроблено інструкцію користувача, яку наведено у додатку Г – Інструкція користувача програмного забезпечення для магазину весільного та святкового вбрання «Казковий образ». Програму та методику випробувань програмного забезпечення представлено у додатку Д – Програма та методика випробування програмного для магазину весільного та святкового вбрання «Казковий образ».

4 ОХОРОНА ПРАЦІ

4.1 Аналіз факторів ризику при роботі з комп'ютером

При роботі з комп'ютером шкідливими і небезпечними чинниками є:

- електростатичні поля;
- електромагнітне випромінювання;
- наявність потужних іонізуючих випромінювань;
- локальне стомлення, загальне стомлення;
- стомлюваність очей;
- небезпека поразки електричним струмом;
- пожежонебезпека.

Всі ці чинники викликані двома основними причинами: неправильною роботою за обладнанням і неправильним вибором обладнання.

Негативний вплив комп'ютера на людину є комплексним.

По-перше, комп'ютер є джерелом електростатичного поля; електромагнітних випромінювань в низькочастотному, наднизькочастотних та високочастотному діапазонах (2 Гц – 400 кГц); випромінювання оптичного діапазону (ультрафіолетового, інфрачервоного і видимого світла), рентгенівського випромінювання.

Вчені встановили, що випромінювання низької частоти в першу чергу негативно впливає на центральну нервову систему, викликаючи головний біль / запаморочення, нудоту, депресію, безсоння, відсутність апетиту, стрес.

Низькочастотне електромагнітне поле може з'явитися причиною шкірних захворювань (вугрі, екзема, рожевий лишай та ін), хвороб серцево-судинної системи та шлунково-кишкового тракту; воно впливає на білі кров'яні тіลця, що призводить до виникнення пухлин, у тому числі і злоякісних .

Електростатичне поле великої напруженості здатне змінювати і переривати клітинне розвиток, а також в окремих випадках викликати катаракту.

Дисплеї завдяки електростатичного поля інтенсивно притягають і збирають пил, а значить, можуть бути причиною появи дерматитів на шкірі обличчя, загострення бронхіальної астми, подразнення слизових оболонок

По-друге, нерухома і напружена поза.

По-третє, інтенсивна робота з клавіатурою викликає больові відчуття в ліктьових суглобах, передпліччях, зап'ястях, в кистях та пальцях рук.

Відповідно до законодавчих актів площа робочого місця користувача ПК повинна складати не менше 4,5м². У приміщеннях повинні проводитися щоденне вологе прибирання і систематичне провітрювання після кожної години роботи. Устаткування, що виробляє багато шуму (друкуючі пристрої, сканери, сервери і тому подібні), рівні шуму якого перевищують нормативні, повинне розміщуватися поза робочими місцями співробітників.

Робочі столи слід розміщувати так, щоб монітори були орієнтовані бічною стороною до світлових отворів, щоб природне світло падало переважно зліва.

При розміщенні робочих місць відстань між робочими столами має бути не менше 2,0 м, а відстань між бічними поверхнями відеомоніторів – не менше 1,2 м. Робочі місця співробітників, що виконують творчу роботу і що вимагає значної розумової напруги або високої концентрації уваги, рекомендується ізолювати один від одного перегородками заввишки від 1,5 м.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного устаткування. Висота робочої поверхні столу повинна складати 725 мм, робоча поверхня столу повинна мати ширину 800..1400 мм і глибину 800..1000 мм. Робочий стіл повинен мати простір для ніг заввишки не менше 600 мм, шириною – не менше 500 мм, завглибшки на рівні колін – не менше 450 мм і на рівні витягнутих ніг – не менше 650 мм.

Конструкція робочого стільця або крісла повинна забезпечувати підтримку раціональної робочої пози працівника і дозволяти змінювати позу з метою зниження статичної напруги м'язів шийно-плечової області і спини. Робочий стілець або крісло мають бути підйимально-поворотними, регульованими по висоті і кутам нахилу сидіння і спинки, а також відстані спинки від переднього краю сидіння, при цьому регулювання кожного параметра має бути незалежним, легко здійснюваним і мати надійну фіксацію.

Клавіатуру слід розташовувати на поверхні столу на відстані 100..300 мм від краю, зверненого до користувача, або на спеціальній поверхні, відокремленій від основної.

Екран відеомонітора повинен знаходитися від очей користувача на відстані 600..700 мм, але не ближче 500.

4.2 Основні законодавчі та нормативно-правові акти про охорону праці в Україні

Створення системи охорони праці на підприємстві передбачене Законом України «Про охорону праці». У загальному, законодавство про охорону праці складається з цього Закону, Кодексу законів про працю України, Закону України «Про загально обов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності» та прийнятих відповідно до них нормативно-правових актів.

Відповідно до Закону України «Про охорону праці», під час укладання трудового договору роботодавець повинен проінформувати працівника під розписку про умови праці та про наявність на його робочому місці небезпечних і шкідливих виробничих факторів, які ще не усунуто, можливі наслідки їх впливу на здоров'я та про права працівника на пільги і компенсації за роботу в таких умовах відповідно до законодавства і колективного договору.

Крім того, відповідно до вказаного Закону, роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці.

Серед чималого переліку обов'язків роботодавця, передбаченого ст. 13 Закону України «Про охорону праці» є пункт про те, що роботодавець – розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства (далі – акти підприємства), та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами та актами підприємства з охорони праці.

На державному рівні створено цілу армію контролюючих органів, серед яких звісно є й такі, які можуть забороняти, зупиняти, припиняти, обмежувати експлуатацію підприємства у разі виявлення будь-яких порушень чи недоліків у системі охорони праці.

Крім того, за порушення законодавства про охорону праці, невиконання розпоряджень посадових осіб органів державного нагляду за охороною праці юридичні та фізичні особи, які відповідно до законодавства використовують найману працю, притягаються органами державного нагляду за охороною праці до сплати штрафу у порядку, встановленому законом.

Максимальний розмір штрафу не може перевищувати п'яти відсотків місячного фонду заробітної плати юридичної чи фізичної особи, яка відповідно до законодавства використовує найману працю [29].

4.3 Розробка заходів щодо зменшення навантаження при роботі з комп'ютером

Заходи щодо зменшення впливу

Заходи щодо усунення небезпеки поразки електричним струмом зводяться до правильного розміщення устаткування та електричних кабелів. Інші заходи щодо забезпечення електробезпеки, збігаються з загальними заходами пожежо- та електробезпеки.

В якості профілактичних заходів для забезпечення пожежної безпеки слід використовувати скриту електромережу, надійні розетки з пожежобезпечних матеріалів, силові мережі живлення устаткування виконувати кабелями, розрахованими на підключення в 3-5 разів більшого навантаження, включати й виключати живлення обладнання за допомогою штатних вимикачів. Треба регулярно робити очистку внутрішніх частин комп'ютерів, іншого устаткування від пилу, розташовувати комп'ютери на окремих не спалюваних столах. Для запобігання іскріння необхідно рідше встромляти і виймати штепсельні вилки з розеток.

Екран дисплея повинен бути розташованим перпендикулярно до напрямку погляду. Якщо він розташований під кутом, то стає причиною сутулості. Відстань від дисплея до очей повинна трохи перевищувати звичну відстань між книгою та очима. Перед екраном монітора, особливо старих типів, повинен бути спеціальний захисний екран. При його відсутності треба сидіти на відстані витягнутої руки від монітора. Ще одним моментом, який стосується зору, є необхідність створення неоднорідного поля зору. Для цього можна розвісити на поверхнях (стінах) плакати та картини, виконані у спокійних тонах. Наприклад, пейзажі.

Важливою є форма спинки крісла, яка повинна повторювати форму спини. Висота крісла повинна бути такою, щоб користувач не почував тиску на куприк або стегна. Крісло бажано обладнати бильцями. Його потрібно встановити так, щоб не треба було тягтися до клавіатури. Періодично

користувачу необхідно рухатися, вчасно змінювати положення тіла і робити перерви у роботі.

При напруженій роботі за комп'ютером щогодини необхідно робити перерву на 15 хвилин через кожну годину і треба займатися іншою справою. Декілька разів на годину бажано виконувати серію легких вправ для розслаблення.

Техніка безпеки при роботі на комп'ютері

Загальні положення:

1) При виконанні робіт на персональних комп'ютерах (ПК) необхідно дотримуватись вимог загальної та даної інструкцій з охорони праці.

2) До самостійної роботи на комп'ютерах допускаються особи, які пройшли медичний огляд, навчання з професії, ввідний інструктаж з охорони праці і первинний інструктаж з охорони праці на робочому місці. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя.

3) При роботі на комп'ютерах, до складу яких входять відеодисплейні термінали (ВДТ), на працівників можуть діяти наступні шкідливі та погрожуючі життю і здоров'ю виробничі фактори:

а) фізичні :

- підвищений рівень шуму (від вентилятора мікропроцесора та плат);
- високий рівень напруги в електричній мережі, торкання до якої може викликати нещасний випадок;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітного випромінювання;
- підвищений рівень напруженості електричного поля;
- пряма і відбита від екрана блескність;
- несприятливе розподілення яскравості об'єктів в полі зору.

б) психофізіологічні:

- фізичні перевантаження статичної та динамічної дії;

- нервово-психічні перевантаження (розумове перенапруження, перенапруження зорових аналізаторів, монотонність праці, емоційні перевантаження).

4) Основним обладнанням робочого місця користувача ПК є: ВДТ, системний блок, клавіатура, друкуючий пристрій (принтер), зовнішні пристрої пам'яті, сканер, "миша", блок безперервного живлення, робочий стіл, стілець та інше; допоміжним – шафи, полиці, сейф та ін.

Робочі місця з ВДТ повинні бути розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін – на відстані 1м, між собою – на відстані не менше 1,5 м. Відносно вікон робочі місця доцільно розташувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

5) Робочі місця, обладнані ВДТ, слід розташовувати таким чином, щоб запобігти попаданню в очі прямого світла. Джерела освітлення слід розташовувати з обох боків екрану, паралельно напрямку зору.

Для запобігання світлових відблисків від екрана, клавіатури в напрямку очей користувача від освітлювачів загального призначення або сонячних променів необхідно застосовувати антивідблисківі сітки, спеціальні фільтри, захисні козирки, на вікнах регульовані жалюзі.

Фільтри із металевої або нейлонової сітки використовувати не рекомендується тому, що сітка спотворює зображення внаслідок інтерференції світла. Найкраща якість зображення забезпечується скляними поляризаційними фільтрами – вони усувають практично усі відблиски і забезпечують чітке та контрастне зображення.

6) При роботі з текстовою інформацією (в режимі введення даних, редагування тексту і читання з екрану ВДТ) найбільш фізіологічним є зображення чорних знаків на світлому (білому) фоні.

7) Розташовувати ВДТ на робочому місці слід так, щоб поверхня екрану знаходилась в центрі поля зору на відстані 400-700 мм від очей користувача. Пропонується розташовувати елементи робочого місця таким чином, щоб

підтримувалась однакова відстань від очей користувача до екрана, клавіатури, пюпітра.

8) Робота комп'ютера супроводжується електромагнітним випромінюванням різних частот і рівнів, інтенсивність котрого зменшується пропорційно квадрату відстані до екрану. Оптимальною для працюючого за ПК є відстань 50 см від екрану ВДТ.

9) Раціональною робочою позою працюючого за ПК вважається таке розташування тіла, при якому ступні працівника розташовані горизонтально на підлозі або на підставці для ніг, стегна зорієнтовані в горизонтальній площині, верхні кінцівки рук – вертикально, кут ліктьового суглоба коливається в межах 70-90 град., зап'ястя зігнуті під кутом, не перевищує 20 град., нахил голови – в межах 15-20 град.

10) Для нейтралізації зарядів статичної електрики в приміщенні, де виконуються роботи на комп'ютерах, в тому числі на лазерних та світлодіодних принтерах, пропонується збільшувати вологість повітря за допомогою кімнатних зволожувачів. Не рекомендується носити одяг з синтетичних матеріалів.

Вимоги безпеки перед початком роботи:

1) Перевірити надійність встановлення апаратури на робочому столі. ВДТ не повинен стояти на краю столу. Повернути ВДТ так, щоб було зручно дивитись на екран – під прямим кутом (а не збоку) і трохи згори вниз, а екран повинен бути нахиленим під невеликим кутом (нижній край ближче до оператора).

2) Перевірити загальний стан апаратури, справність проводки електромережі, з'єднувальних шнурів, штепсельних вилок та розеток, заземлення захисного екрана.

4) Відрегулювати освітленість робочого місця.

5) Відрегулювати і зафіксувати висоту стільця (крісла), зручний для користувача нахил його спинки.

6) Під'єднати до системного блока необхідну апаратуру (принтер, сканер і т.п.). Всі кабелі, що з'єднують системний блок з іншими пристроями слід вставляти і виймати тільки при вимкненому електричному живленні.

7) Увімкнути апаратуру ПК вмикачами на корпусах в такій послідовності: стабілізатор (або блок безперервного живлення), ВДТ, системний блок, принтер (якщо передбачається друкування).

8) Відрегулювати яскравість світіння екрану ВДТ, фокусування, контрастність. Не слід робити яскравість екрану занадто великою, тому що це втомлює очі.

Рекомендується:

- яскравість світіння екрану – не менше 100 кд/ кв.м (кандел на кв.м);
- відношення яскравості екрану ВДТ до яскравості оточуючих його поверхонь в робочій зоні – не більше 3:1;
- мінімальний розмір точки світіння – не більше 0,4 мм для монохромного ВДТ і не більше 0,6 мм для кольорового;
- контрастність зображення знака – не менше 0,8.

9) При появі несправностей комп'ютерної техніки повідомити керівника.

Вимоги безпеки при закінченні роботи на ПК:

1) Закінчити і зберегти в пам'яті ПК файли, які знаходились у роботі. Виконати всі дії для коректного завершення роботи в оперативній системі.

2) Вимкнути принтер та інші периферійні пристрої, вимкнути ВДТ і системний блок. Вимкнути стабілізатор (або блок безперервного живлення при його наявності). Штепсельні вилки витягнути з розеток. Накрити клавіатуру кришкою для попередження попадання в неї пилу.

3) Навести порядок на робочому місці.

4) Вимкнути освітлення та електроприлади.

Вимоги безпеки при аварійних ситуаціях:

1) При раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності: периферійні пристрої, ВДТ, системний

блок, стабілізатор (або блок безперервного живлення). Витягнути вилки з розеток.

2) При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК. Знайти місце загоряння і виконати всі можливі заходи для його ліквідації, попередивши терміново про це керівництво.

3) У випадку виникнення пожежі негайно попередити про це пожежну частину та керівництво, виконати усі можливі заходи по евакуації людей з приміщення і розпочати гасіння пожежі первинними засобами пожежогасіння.

ВИСНОВКИ

Підвівши підсумки виконання кваліфікаційної роботи, можна відзначити, що поставлена мета досягнута - розроблено вебсервіс для магазину весільного та святкового вбрання "Казковий образ".

В результаті були вирішені наступні задачі:

- проведено аналіз практичної задачі інженерії програмного забезпечення, в рамках якого проаналізовано існуючі аналогічні програмні продукти;

- на основі проведеного аналізу сформовано вимоги до вебсервісу для магазину весільного та святкового вбрання "Казковий образ" та розроблено технічне завдання;

- розроблено ескізний, технічний та робочий проекти вебсервісу для магазину весільного та святкового вбрання "Казковий образ";

- проведено тестування та випробування вебсервісу для магазину весільного та святкового вбрання "Казковий образ";

- розроблено програмну документацію до вебсервісу для магазину весільного та святкового вбрання "Казковий образ".

В результаті було розроблено вебсервіс для магазину весільного та святкового вбрання "Казковий образ", який дозволяє:

- клієнтам: переглядати каталог товарів, здійснювати пошук за фільтрами, оформляти замовлення, бронювати примірки, залишати відгуки.

- співробітникам: оновлювати інформацію про товари, керувати замовленнями та взаємодіяти з клієнтами.

Таким чином, вебсервіс сприяє підвищенню ефективності бізнес-процесів, покращенню клієнтського досвіду та розширенню ринкових можливостей магазину "Казковий образ".

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chaffey, Dave, and Fiona Ellis-Chadwick. Digital Marketing. 8th ed. Pearson Education Limited, 2022. – 688 p.
2. Котенко Н., Жирова Т., Чибасєвський В., Десятко А. Дослідження основних тенденцій сучасної розробки веб-сайтів. Електронне фахове наукове видання "Кібербезпека: освіта, наука, техніка". Україна, 2019, 1(5), с. 6–15.
3. InLove Collection – весільні сукні в Києві [Електронний ресурс]. – Режим доступу: <https://inlove.biz/> (дата звернення: 05.03.2025).
4. Laura Style – весільні та вечірні сукні в Києві та Харкові [Електронний ресурс]. – Режим доступу: <https://laura-style.com.ua/> (дата звернення: 05.03.2025).
5. Ivanna Dress – весільний салон у Львові [Електронний ресурс]. – Режим доступу: <https://ivannalviv.com.ua/> (дата звернення: 05.03.2025).
6. Primavera Sposa – весільний салон в Одесі [Електронний ресурс]. – Режим доступу: <https://primaverasposa.com.ua/uk/> (дата звернення: 05.03.2025).
7. FAINA Boutique – інтернет-магазин суконь та аксесуарів [Електронний ресурс]. – Режим доступу: <https://www.faina.boutique/> (дата звернення: 05.03.2025).
8. СВІТЛАНА – весільний салон у Львові [Електронний ресурс]. – Режим доступу: <https://svitlana.in.ua/> (дата звернення: 05.03.2025).
9. Jeorjett Dress – виробник весільних суконь (Чернівці) [Електронний ресурс]. – Режим доступу: <https://jeorjettddress.com/ua> (дата звернення: 05.03.2025).
10. Wedding Market – весільний маркет (Київ/Одеса) [Електронний ресурс]. – Режим доступу: <https://wedding-market.ua/> (дата звернення: 05.03.2025).

11. Wedboom – інтернет-магазин весільних та вечірніх суконь [Електронний ресурс]. – Режим доступу: <https://wedboom.net/> (дата звернення: 05.03.2025).
12. White Angel – весільний бутік у Львові [Електронний ресурс]. – Режим доступу: <https://whiteangel.com.ua/> (дата звернення: 05.03.2025).
13. Методичні рекомендації щодо підготовки звіту з переддипломної практики та виконання кваліфікаційної роботи бакалавра зі спеціальності 121 «Інженерія програмного забезпечення» / С. Б. Приходько, Л. М. Макарова, Л. О. Латанська, С. В. Суслов, Т. А. Фаріонова. – Миколаїв : НУК, 2024. – 56 с.
14. Діаграма варіантів використання [Електронний ресурс] // Вікіпедія : вільна енциклопедія. – URL: https://uk.wikipedia.org/wiki/Діаграма_варіантів_використання (дата звернення: 01.04.2025)
15. Клієнт-серверна архітектура [Електронний ресурс] // Вікіпедія : вільна енциклопедія. – URL: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура (дата звернення: 17.04.2025)
16. Модель–вид–контролер (MVC) [Електронний ресурс] // Вікіпедія : вільна енциклопедія. – URL: <https://uk.wikipedia.org/wiki/Модель-вид-контролер> (дата звернення: 17.04.2025)
17. MVC Framework Tutorial for Beginners [Електронний ресурс] // Guru99. – URL: <https://www.guru99.com/uk/mvc-tutorial.html> (дата звернення: 17.04.2025)
18. Концептуальна модель – Вікіпедія [Електронний ресурс]. – URL: https://uk.wikipedia.org/wiki/Концептуальна_модель (дата звернення: 23.04.2025)
19. Ткаченко Т.О. Проектування інтерфейсів користувача: навч. посібник. Харків: ХНЕУ ім. С. Кузнеця, 2021. 194 с.
20. React – A JavaScript library for building user interfaces. [Електронний ресурс]. – Режим доступу: <https://react.dev/> (дата звернення: 18.02.2025).

21. Діаграма класів [Електронний ресурс] // Вікіпедія : вільна енциклопедія. – URL: https://uk.wikipedia.org/wiki/Діаграма_класів (дата звернення: 02.05.2025).
22. Діаграма послідовності [Електронний ресурс] // Вікіпедія : вільна енциклопедія. – URL: https://uk.wikipedia.org/wiki/Діаграма_послідовності (дата звернення: 12.05.2025).
23. Data Modeling: Conceptual, Logical, & Physical [Електронний ресурс] // Guru99. – URL: <https://www.guru99.com/uk/data-modelling-conceptual-logical.html> (дата звернення: 17.05.2025)
24. PHP 8.3.0 Release Announcement [Електронний ресурс] // PHP.net. – URL: <https://www.php.net/releases/8.3/en.php> (дата звернення: 20.05.2025).
25. Database Access & ORM - 5.x [Електронний ресурс] // CakePHP Documentation. – URL: <https://book.cakephp.org/5/en/orm.html> (дата звернення: 20.05.2025).
26. Routing - 5.x [Електронний ресурс] // CakePHP Documentation. – URL: <https://book.cakephp.org/5/en/development/routing.html> (дата звернення: 20.05.2025).
27. Новий етап у розвитку веб-технологій: HTML5+CSS3 [Електронний ресурс] // UniversalWeb. – URL: <https://universalweb.com.ua/blog/tehnology/html5-css3.html> (дата звернення: 20.05.2025).
28. Фізична модель даних [Електронний ресурс] // Data-Life-UA. – URL: <https://data-life-ua.com/db/fizychna-model-danykh/> (дата звернення: 20.05.2025)
29. Про охорону праці : Закон України від 14.10.1992 № 2694-XII. Дата оновлення: 27.02.2021. [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення: 01.03.2025).

ДОДАТОК А – Технічне завдання на розробку вебсервісу для магазину весільного та святкового вбрання "Казковий образ"

Вступ

Назва розроблюваного проекту: «Вебсервіс для магазину весільного та святкового вбрання "Казковий образ". Галузь застосування – магазин весільного та святкового вбрання "Казковий образ".

1 Підстава для розробки

Розробка програмного забезпечення виконується на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова.

2 Призначення розробки

2.1 Експлуатаційне призначення

Вебсервіс призначений для використання клієнтами магазину "Казковий образ" для перегляду, вибору та замовлення весільного або святкового вбрання онлайн. Також він має забезпечувати зручне керування замовленнями для адміністраторів та менеджерів магазину.

2.2 Функціональне призначення

Вебсервіс має забезпечити наступні функції:

- Реєстрація та авторизація користувачів.
- Перегляд каталогу товарів.
- Пошук товарів за категоріями та ключовими словами.
- Додавання товарів до кошика.
- Оформлення замовлення та оплата онлайн.
- Відстеження статусу замовлення.
- Оцінка та відгуки про товари.

- Управління користувачькими акаунтами (зміна інформації, історія замовлень тощо).
- Інтерфейс для адміністраторів для управління товарами, замовленнями та користувачами.

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програмне забезпечення повинно виконувати наступні функції:

- 1) Реєстрація/авторизація користувачів (включаючи соціальні мережі та email);
- 2) Пошук та фільтрація товарів;
- 3) Перегляд детальної інформації про товар (опис, ціна, наявність, фото);
- 4) Додавання товарів до кошика та оформлення замовлення;
- 5) Обробка оплат (інтеграція з платіжними системами);
- 6) Система знижок та акцій;
- 7) Відгуки та оцінки товарів;
- 8) Адміністративна панель для управління асортиментом та замовленнями.

3.1.2 Вимоги до організації вхідних та вихідних даних.

Дані зберігаються в файлі БД з довільним доступом типу .myd.

Введення даних здійснюється за допомогою пристроїв введення даних (клавіатури та миші). Дані або вводяться вручну, або обираються із представлених списків.

Виведення даних здійснюється за допомогою пристроїв виведення даних (монітор та принтер).

3.2 Вимоги до надійності

Програмний продукт повинен функціонувати при безперебійній роботі комп'ютера. При виникненні збоїв в роботі комп'ютера, відновлення нормальної роботи повинне проводитися після перезавантаження програмного продукту.

Для забезпечення надійності інформації повинна використовуватися СКБД, що забезпечує цілісність транзакцій і несе відповідальність за цілісність інформації.

У разі введення користувачем некоректної інформації програмний продукт повинен повідомити про помилку і надати можливість виправити її.

3.3 Вимоги до умов експлуатації

Необхідний рівень підготовки користувачів: мінімальні навички в користуванні комп'ютером. Комп'ютер призначений для роботи в закритому опалювальному приміщенні.

3.4 Вимоги до складу та параметрів технічних засобів

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплекції:

- CPU: Intel(R) i5 2,16 МГц або аналогічний за параметрами AMD;
- RAM: 4 GB;
- HDD: 10 GB вільного місця.

3.5 Вимоги до інформаційної та програмної сумісності

Для повноцінного функціонування системи необхідно використовувати ОС Windows версії не нижче 7.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;

- текст програми;
- опис програми;
- інструкцію користувача;
- програму і методику випробувань ПЗ.

5 Стадії та етапи розробки

Етапи розробки представлені в таблиці А.1.

Таблиця А.1 – Етапи розробки програмного забезпечення

Номер п/п	Назва етапів роботи	Термін виконання
1.	Аналіз практичної задачі інженерії ПЗ	17.02.2025
2.	Аналіз вимог до ПЗ	28.03.2025
3.	Розробка архітектури ПЗ	08.04.2025
4.	Розробка проєкту ПЗ	15.04.2025
5.	Реалізація та тестування ПЗ	28.04.2025

6 Порядок контролю та прийому

Контроль за аналізом та проектуванням кожної окремої частини програмного забезпечення відбувається на кожному етапі з урахуванням вимог, визначених у технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником.

Прийом проводять відповідно до програми і методики випробувань і документують за допомогою протоколу проведення випробувань. У разі знаходження помилок під час прийому програмного виробу складається акт про знайдені помилки, який підписуються представниками замовника і розробника і затверджуються керівником організації – замовника та організації – розробника. Розробник повинен на протязі не більше ніж 2 тижнів виправити зазначені помилки і оповістити замовника про повторне проведення перевірки.

ДОДАТОК Б – Код програми

Текст файлу ProductController.php

```
<?php
declare(strict_types=1);

namespace App\Controller;

/**
 * Product Controller
 *
 * @property \App\Model\Table\ProductTable $Product
 */
class ProductController extends AppController
{
    /**
     * Index method
     *
     * @return \Cake\Http\Response|null|void Renders view
     */
    public function list()
    {
        $query = $this->Product->find();
        $product = $this->paginate($query);

        $this->set(compact('product'));
    }

    /**
     * Index method
     *
     * @return \Cake\Http\Response|null|void Renders view
     */
    public function index()
    {
        $query = $this->Product->find();
        $product = $this->paginate($query);

        $this->set(compact('product'));
    }

    /**
     * View method
     *
     * @param string|null $id Product id.
     * @return \Cake\Http\Response|null|void Renders view
     * @throws \Cake\Datasource\Exception\RecordNotFoundException
     When record not found.
     */
}
```

```

public function view($id = null)
{
    $product = $this->Product->get($id, contain: []);
    $this->set(compact('product'));
    $this->set('categories', $this->fetchTable('Category')-
>find('list')->all()->toArray());
    $this->set('collections', $this->fetchTable('Collection')-
>find('list')->all()->toArray());
    $this->set('brands', $this->fetchTable('Brand')-
>find('list')->all()->toArray());
}

/**
 * Add method
 *
 * @return \Cake\Http\Response|null|void Redirects on
successful add, renders view otherwise.
 */
public function add()
{
    $product = $this->Product->newEmptyEntity();
    if ($this->request->is('post')) {
        $product = $this->Product->patchEntity($product,
$this->request->getData());
        if ($this->Product->save($product)) {
            $this->Flash->success(__('The product has been
saved.'));

            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('The product could not be
saved. Please, try again.'));
    }
    $this->set(compact('product'));
    $this->set('categories', $this->fetchTable('Category')-
>find('list')->all());
    $this->set('collections', $this->fetchTable('Collection')-
>find('list')->all());
    $this->set('brands', $this->fetchTable('Brand')-
>find('list')->all());
}

/**
 * Edit method
 *
 * @param string|null $id Product id.
 * @return \Cake\Http\Response|null|void Redirects on
successful edit, renders view otherwise.
 * @throws \Cake\Datasource\Exception\RecordNotFoundException
When record not found.
 */
public function edit($id = null)
{

```



```

        $product = $this->Product->get($id, contain: []);
        if ($this->request->is(['patch', 'post', 'put'])) {
            $product = $this->Product->patchEntity($product,
$this->request->getData());
            if ($this->Product->save($product)) {
                $this->Flash->success(__('The product has been
saved.'));

                return $this->redirect(['action' => 'index']);
            }
            $this->Flash->error(__('The product could not be
saved. Please, try again.'));
        }
        $this->set(compact('product'));
        $this->set('categories', $this->fetchTable('Category')->
find('list')->all());
        $this->set('collections', $this->fetchTable('Collection')->
find('list')->all());
        $this->set('brands', $this->fetchTable('Brand')->
find('list')->all());
    }

    /**
     * Delete method
     *
     * @param string|null $id Product id.
     * @return \Cake\Http\Response|null Redirects to index.
     * @throws \Cake\Datasource\Exception\RecordNotFoundException
     When record not found.
     */
    public function delete($id = null)
    {
        $this->request->allowMethod(['post', 'delete']);
        $product = $this->Product->get($id);
        if ($this->Product->delete($product)) {
            $this->Flash->success(__('The product has been
deleted.'));
        } else {
            $this->Flash->error(__('The product could not be
deleted. Please, try again.'));
        }

        return $this->redirect(['action' => 'index']);
    }
}

```

Текст файла ProductController.php

```

<?php
declare(strict_types=1);

namespace App\Controller;

/**
 * Collection Controller
 *
 * @property \App\Model\Table\CollectionTable $Collection
 */
class CollectionController extends ApplicationController
{
    /**
     * Index method
     *
     * @return \Cake\Http\Response|null|void Renders view
     */
    public function index()
    {
        $query = $this->Collection->find();
        $collection = $this->paginate($query);

        $this->set(compact('collection'));
    }

    /**
     * View method
     *
     * @param string|null $id Collection id.
     * @return \Cake\Http\Response|null|void Renders view
     * @throws \Cake\Datasource\Exception\RecordNotFoundException
     When record not found.
     */
    public function view($id = null)
    {
        $collection = $this->Collection->get($id, contain: []);
        $this->set(compact('collection'));
    }

    /**
     * Add method
     *
     * @return \Cake\Http\Response|null|void Redirects on
     successful add, renders view otherwise.
     */
    public function add()
    {
        $collection = $this->Collection->newEmptyEntity();
        if ($this->request->is('post')) {

```

```

        $collection = $this->Collection->
>patchEntity($collection, $this->request->getData());
        if ($this->Collection->save($collection)) {
            $this->Flash->success(__('The collection has been
saved.'));

            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('The collection could not be
saved. Please, try again.'));
    }
    $this->set(compact('collection'));
}

/**
 * Edit method
 *
 * @param string|null $id Collection id.
 * @return \Cake\Http\Response|null|void Redirects on
successful edit, renders view otherwise.
 * @throws \Cake\Datasource\Exception\RecordNotFoundException
When record not found.
 */
public function edit($id = null)
{
    $collection = $this->Collection->get($id, contain: []);
    if ($this->request->is(['patch', 'post', 'put'])) {
        $collection = $this->Collection->
>patchEntity($collection, $this->request->getData());
        if ($this->Collection->save($collection)) {
            $this->Flash->success(__('The collection has been
saved.'));

            return $this->redirect(['action' => 'index']);
        }
        $this->Flash->error(__('The collection could not be
saved. Please, try again.'));
    }
    $this->set(compact('collection'));
}

/**
 * Delete method
 *
 * @param string|null $id Collection id.
 * @return \Cake\Http\Response|null Redirects to index.
 * @throws \Cake\Datasource\Exception\RecordNotFoundException
When record not found.
 */
public function delete($id = null)
{
    $this->request->allowMethod(['post', 'delete']);
    $collection = $this->Collection->get($id);

```

```

        if ($this->Collection->delete($collection)) {
            $this->Flash->success(__('The collection has been
deleted.'));
        } else {
            $this->Flash->error(__('The collection could not be
deleted. Please, try again.'));
        }

        return $this->redirect(['action' => 'index']);
    }
}

```

Текст файла index.php

```

<?php
/**
 * @uses \App\Controller\ProductController::index()
 * @var \App\View\AppView $this
 * @var iterable<\App\Model\Entity\Product> $product
 */
?>
<div class="product index content">
    <?= $this->Html->link(__('New Product'), ['action' => 'add'],
['class' => 'button float-right']) ?>
    <h3><?= __('Product') ?></h3>
    <div class="table-responsive">
        <table>
            <thead>
                <tr>
                    <th><?= $this->Paginator->sort('ProductID',
__('ID')) ?></th>
                    <th><?= $this->Paginator->sort('Name',
__('Name')) ?></th>
                    <th><?= $this->Paginator->sort('Size',
__('Size')) ?></th>
                    <th><?= $this->Paginator->sort('Color',
__('Color')) ?></th>
                    <th><?= $this->Paginator->sort('Price',
__('Price')) ?></th>
                    <th><?= $this->Paginator->sort('Type',
__('Type')) ?></th>
                    <th><?= $this->Paginator->sort('Status',
__('Status')) ?></th>
                    <th class="actions"><?= __('Actions') ?></th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($product as $product): ?>
                <tr>
                    <td><?= $this->Number->format($product-
>ProductID) ?></td>
                    <td><?= h($product->Name) ?></td>

```

```

        <td><?= h($product->Size) ?></td>
        <td><?= h($product->Color) ?></td>
        <td><?= $this->Number->format($product->Price)
?></td>

        <td><?= h($product->Type) ?></td>
        <td><?= h($product->Status) ?></td>
        <td class="actions">
            <?= $this->Html->link(__('View'),
['action' => 'view', $product->ProductID]) ?>
            <?= $this->Html->link(__('Edit'),
['action' => 'edit', $product->ProductID]) ?>
            <?= $this->Form->postLink(
                __('Delete'),
                ['action' => 'delete', $product-
>ProductID],
                [
                    'method' => 'delete',
                    'confirm' => __('Are you sure you
want to delete # {0}?', $product->ProductID),
                ]
            ) ?>
        </td>
    </tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
<div class="paginator">
    <ul class="pagination">
        <?= $this->Paginator->first('< ' . __('first')) ?>
        <?= $this->Paginator->prev('< ' . __('previous')) ?>
        <?= $this->Paginator->numbers() ?>
        <?= $this->Paginator->next(__('next') . ' >') ?>
        <?= $this->Paginator->last(__('last') . ' >>') ?>
    </ul>
</div>
</div>

```

Текст файла main.sql

```

CREATE TABLE `Client` (
  `ClientID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
  `FirstName` VARCHAR(50) NOT NULL,
  `LastName` VARCHAR(50) NOT NULL,
  `Phone` VARCHAR(20) NOT NULL,
  `Email` VARCHAR(100) NOT NULL,
  `Address` TEXT NOT NULL,
  `RegistrationDate` DATE NOT NULL,
  PRIMARY KEY(`ClientID`)
);

```

```

CREATE TABLE `Staff` (

```

```

        `StaffID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
        `FirstName` VARCHAR(50) NOT NULL,
        `LastName` VARCHAR(50) NOT NULL,
        `Position` VARCHAR(50) NOT NULL,
        `Phone` VARCHAR(20) NOT NULL,
        `Email` VARCHAR(100) NOT NULL,
        PRIMARY KEY(`StaffID`)
    );

CREATE TABLE `Category` (
    `CategoryID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
    `CategoryName` VARCHAR(100) NOT NULL,
    `Description` TEXT NOT NULL,
    PRIMARY KEY(`CategoryID`)
);

CREATE TABLE `Brand` (
    `BrandID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
    `BrandName` VARCHAR(100) NOT NULL,
    `Country` VARCHAR(50) NOT NULL,
    `ContactInfo` TEXT NOT NULL,
    PRIMARY KEY(`BrandID`)
);

CREATE TABLE `Collection` (
    `CollectionID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
    `Name` VARCHAR(100) NOT NULL,
    `Season` VARCHAR(50) NOT NULL,
    `Year` INTEGER NOT NULL,
    `Description` TEXT NOT NULL,
    PRIMARY KEY(`CollectionID`)
);

CREATE TABLE `Product` (
    `ProductID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
    `Name` VARCHAR(100) NOT NULL,
    `Description` TEXT NOT NULL,
    `Size` VARCHAR(20) NOT NULL,
    `Color` VARCHAR(30) NOT NULL,
    `Price` DECIMAL(10,2) NOT NULL,
    `Type` VARCHAR(50) NOT NULL,
    `Status` VARCHAR(20) NOT NULL,
    `CategoryID` INTEGER NOT NULL,
    `CollectionID` INTEGER NOT NULL,
    `BrandID` INTEGER NOT NULL,
    PRIMARY KEY(`ProductID`)
);

```

```
CREATE TABLE `Order` (
  `OrderID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
  `ClientID` INTEGER NOT NULL,
  `OrderDate` DATE NOT NULL,
  `Status` VARCHAR(20) NOT NULL,
  `TotalAmount` DECIMAL(10,2) NOT NULL,
  `OrderType` VARCHAR(20) NOT NULL,
  PRIMARY KEY(`OrderID`)
);
```

```
CREATE TABLE `Order_Item` (
  `OrderID` INTEGER NOT NULL,
  `ProductID` INTEGER NOT NULL,
  `Quantity` INTEGER NOT NULL,
  `UnitPrice` DECIMAL(10,2) NOT NULL,
  `Discount` DECIMAL(5,2) NOT NULL,
  PRIMARY KEY(`OrderID`, `ProductID`)
);
```

```
CREATE TABLE `Payment` (
  `PaymentID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
  `OrderID` INTEGER NOT NULL,
  `PaymentDate` DATE NOT NULL,
  `Amount` DECIMAL(10,2) NOT NULL,
  `PaymentMethod` VARCHAR(50) NOT NULL,
  PRIMARY KEY(`PaymentID`)
);
```

```
CREATE TABLE `Fitting` (
  `FittingID` INTEGER NOT NULL AUTO_INCREMENT UNIQUE,
  `ClientID` INTEGER NOT NULL,
  `StaffID` INTEGER NOT NULL,
  `FittingDate` DATE NOT NULL,
  `FittingTime` TIME NOT NULL,
  `Notes` TEXT NOT NULL,
  PRIMARY KEY(`FittingID`)
);
```

```
ALTER TABLE `Product`
ADD FOREIGN KEY(`CategoryID`) REFERENCES `Category`(`CategoryID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Product`
ADD FOREIGN KEY(`CollectionID`) REFERENCES
`Collection`(`CollectionID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Product`
ADD FOREIGN KEY(`BrandID`) REFERENCES `Brand`(`BrandID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Order`
```

```
ADD FOREIGN KEY(`ClientID`) REFERENCES `Client`(`ClientID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Order_Item`
ADD FOREIGN KEY(`OrderID`) REFERENCES `Order`(`OrderID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Order_Item`
ADD FOREIGN KEY(`ProductID`) REFERENCES `Product`(`ProductID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Payment`
ADD FOREIGN KEY(`OrderID`) REFERENCES `Order`(`OrderID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Fitting`
ADD FOREIGN KEY(`ClientID`) REFERENCES `Client`(`ClientID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE `Fitting`
ADD FOREIGN KEY(`StaffID`) REFERENCES `Staff`(`StaffID`)
ON UPDATE NO ACTION ON DELETE NO ACTION;
```


ДОДАТОК В – Опис програми

1 Загальні відомості

Розроблений вебсервіс для магазину весільного та святкового вбрання "Казковий образ" є програмою для клієнтів та співробітників магазину весільного та святкового вбрання "Казковий образ", який займається вводом та формуванням інформації про заявки на поставку товару та продаж товару.

Програма призначена для підвищення ефективності бізнес-процесів, покращенню клієнтського досвіду та розширенню ринкових можливостей магазину "Казковий образ".

Для розробки вебсервісу були обрані наступні засоби:

- в якості мови програмування серверної частини була обрана PHP 8.3 з використанням Framework CakePHP 5, клієнтської – HTML 5 / CSS 3 та PHP 8.3;
- в якості СКБД була обрана MariaDB 10;
- в якості середовища розробки було обрано середовище NetBeans IDE.

2 Функціональне призначення

Програма дозволяє виконувати наступні основні функції:

- клієнтам: переглядати каталог товарів, здійснювати пошук за фільтрами, оформляти замовлення, бронювати примірки, залишати відгуки;
- співробітникам: оновлювати інформацію про товари, керувати замовленнями та взаємодіяти з клієнтами.

3 Опис логічної структури

Вебсервіс розроблений на модульній основі, тобто кожний компонент (модуль) програми відповідає за конкретну дію, наприклад: формування заявки на примірку вбрання (відповідає за формування заявки на примірку вбрання). Дані, необхідні для роботи вебсервісу, зберігаються у нормалізованих таблицях бази даних.

4 Технічні та програмні засоби

Система повинна задовольняти вказаним вимогам на комп'ютері

наступної мінімальної комплекції:

- CPU: Intel(R) i5 2,16 МГц або аналогічний за параметрами AMD;
- RAM: 4 GB;
- HDD: 10 GB вільного місця.

Для повноцінного функціонування системи необхідно використовувати ОС Windows версії не нижче 7.

5 Виклик та завантаження

Розроблений вебсервіс для магазину весільного та святкового вбрання є клієнт-серверним застосунком, у якому клієнтська частина взаємодіє із серверною за допомогою HTTP-запитів, а обробка запитів реалізована засобами фреймворку CakePHP 5. При переході користувача на вебсторінку здійснюється ініціалізація HTML/CSS-розмітки та PHP-компонентів клієнтської частини, що відповідають за формування інтерфейсу та логіки відображення.

Коли користувач викликає певну функцію (наприклад, авторизується, додає товар до кошика або оформлює замовлення), відбувається звернення до відповідного маршруту (route) у системі маршрутизації CakePHP, після чого контролер обробляє запит, звертається до моделей, взаємодіє з базою даних MariaDB 10 і формує відповідь. Ця відповідь повертається у форматі HTML або JSON, залежно від конкретного типу запиту (синхронного чи асинхронного), що дозволяє забезпечити як повноцінну перезавантажувану взаємодію, так і динамічне оновлення окремих частин сторінки без її повного оновлення.

Виклик кожної сторінки або дії відбувається за структурованими URL-шляхами відповідно до шаблонів фреймворку, наприклад: /products/view/15 – перегляд картки товару з ідентифікатором 15. Серверні сценарії обробки запитів реалізовані у вигляді контролерів та екшенів, які автоматично завантажуються згідно з конфігурацією CakePHP. Завдяки використанню MVC-підходу забезпечено чітке розмежування відповідальностей, що спрощує обробку запитів та повторне використання компонентів. Усі компоненти застосунку розгортаються локально або на віддаленому сервері, викликаються через браузер користувача та завантажуються з урахуванням кешування стилів,

зображень для підвищення швидкодії.

6 Вхідні та вихідні дані

Дані зберігаються в файлі БД з довільним доступом типу .myd.

Вхідні дані:

Введення даних здійснюється за допомогою пристроїв введення даних (клавіатура та миша). Дані або вводяться вручну, або обираються із представлених списків.

Вихідні дані:

Виведення даних здійснюється за допомогою пристроїв виведення даних (монітор та принтер).

ДОДАТОК Г – Інструкція користувача

Розроблений вебсервіс для магазину весільного та святкового вбрання є клієнт-серверним застосунком, у якому клієнтська частина взаємодіє із серверною за допомогою HTTP-запитів, а обробка запитів реалізована засобами фреймворку CakePHP 5. При переході користувача на вебсторінку здійснюється ініціалізація HTML/CSS-розмітки та PHP-компонентів клієнтської частини, що відповідають за формування інтерфейсу та логіки відображення.

Усі компоненти застосунку розгортаються локально або на віддаленому сервері, викликаються через браузер користувача та завантажуються з урахуванням кешування стилів, зображень для підвищення швидкодії.

Після відкриття сайту користувач потрапляє на головну сторінку вебзастосунку, де відображено загальну інформацію та каталог вбрання. Головна сторінка наведена на рисунку Г.1.

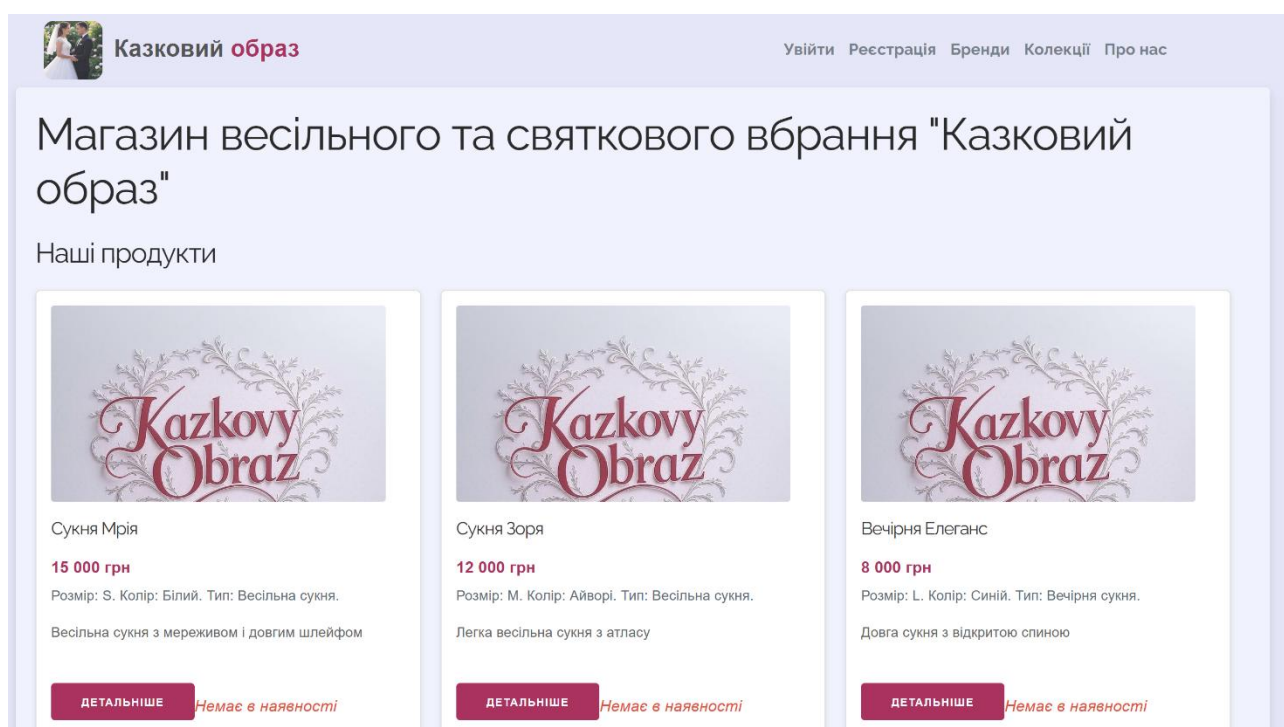


Рисунок Г.1 - Головна сторінка вебзастосунку

У верхній частині сторінки знаходиться навігаційна панель, що забезпечує перехід до таких основних розділів: форма авторизації, особистий кабінет, каталог брендів, каталог колекцій.

Якщо користувач не має акаунта, то він може зареєструватися, натиснувши на кнопку «Реєстрація». Сторінка реєстрації представлена на рисунку Г.2.

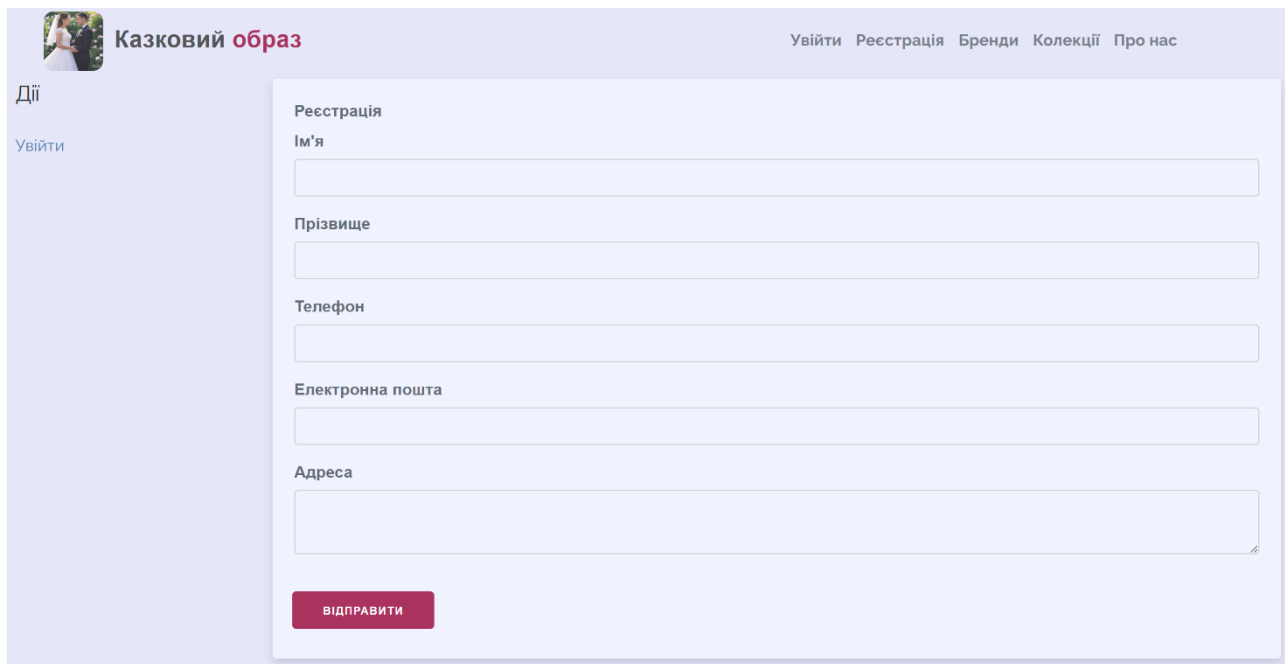


Рисунок Г.2 - Сторінка реєстрації користувача

Увійти до системи можна за допомогою введення логіну й пароля, після чого користувач отримує доступ до персоналізованих функцій: створення замовлень, перегляду історії та редагування профілю. Сторінка авторизації представлена на рисунку Г.3.

Рисунок Г.3 - Сторінка авторизації користувача

У розділі «Каталог» користувач може переглядати перелік доступного вбрання. Кожен товар представлено у вигляді блоку з коротким описом, фотографією, ціною. За потреби користувач може натиснути на кнопку «Детальніше», щоб переглянути його повний опис та характеристики (рисунок Г.4).

Інформація про товар	
Категорія	Весільні сукні
Колекція	Весна мрії
Бренд	Elegance

Рисунок Г.4 – Сторінка детального перегляду обраного товару

Для замовлення примірки вбрання клієнт обирає варіант та натискає на кнопку «Замовити примірку», яка відкриває сторінку з формою для бронювання дати та часу. Сторінка запису на примірку зображена на рисунку Г.5.

The screenshot shows a web interface for 'Казковий образ' (Fairytale Image). The header includes a logo, the brand name, and navigation links: 'Увійти', 'Реєстрація', 'Бренди', 'Колекції', and 'Про нас'. On the left, there is a sidebar with 'Дії' (Actions) and 'Список примірок' (Appointment list). The main content area features a form titled 'Додати примірку' (Add appointment). The form contains the following fields: 'Дата примірки' (Appointment date) with a date picker showing 'ДД.ММ.ГГГГ', 'Час примірки' (Appointment time) with a time picker showing '--:--' and a clock icon, and 'Примітки' (Notes) with a text area. A red button labeled 'ВІДПРАВИТИ' (SEND) is at the bottom of the form.

Рисунок Г.5 – Сторінка запису на примірку

Після вибору бажаного товару для придбання клієнт додає його до кошика – система створює чернетку замовлення, де відображаються всі обрані позиції. Перехід до кошика здійснюється через відповідне посилання в шапці сайту. У вікні кошика можна змінити кількість товарів або видалити непотрібні. Після підтвердження вмісту натискається кнопка «Оформити замовлення», після чого відкривається форма введення контактних даних.

Форма оформлення замовлення передбачає введення ПІБ, електронної пошти, номера телефону, а також додаткової інформації. Після натискання кнопки підтвердження система записує замовлення в базу даних і відображає його статус на сторінці підтвердження.

Співробітник магазину може керувати інформацією про бренди, колекції, товари, а саме: додавання нових, корегування існуючих та видалення неактуальних. Для переходу на сторінки керування інформацією необхідно обрати відповідний пункт в головному меню.

Сторінка для керування інформацією про бренди представлена на рисунку Г.6.

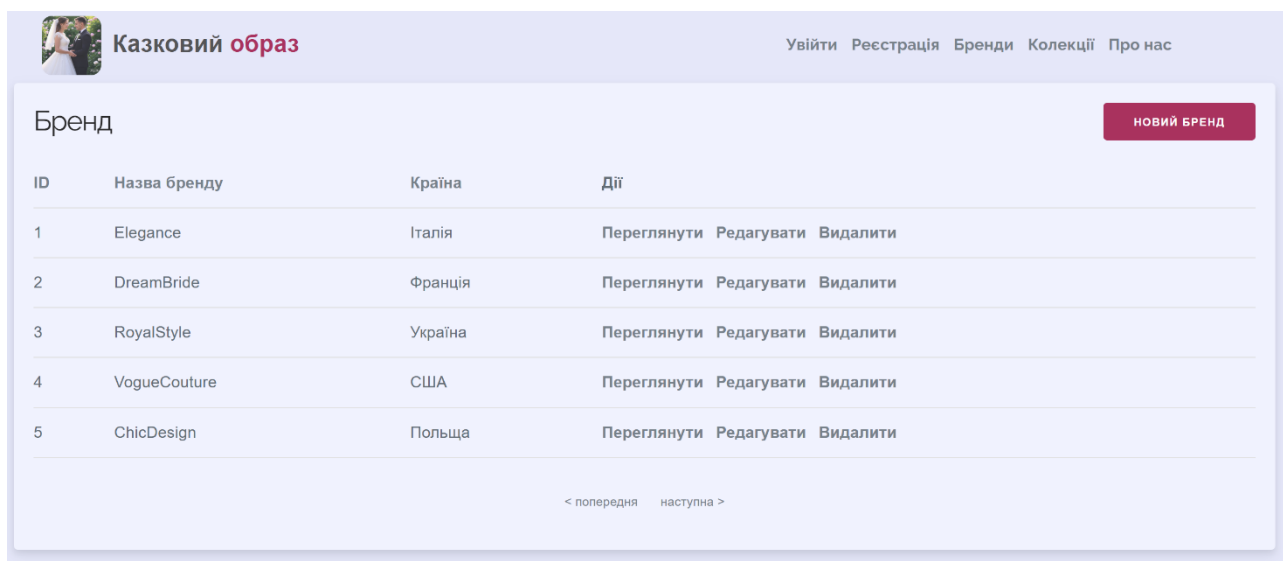


Рисунок Г.6 – Сторінка для керування інформацією про бренди

Сторінка для керування інформацією про колекції представлена на рисунку Г.7.

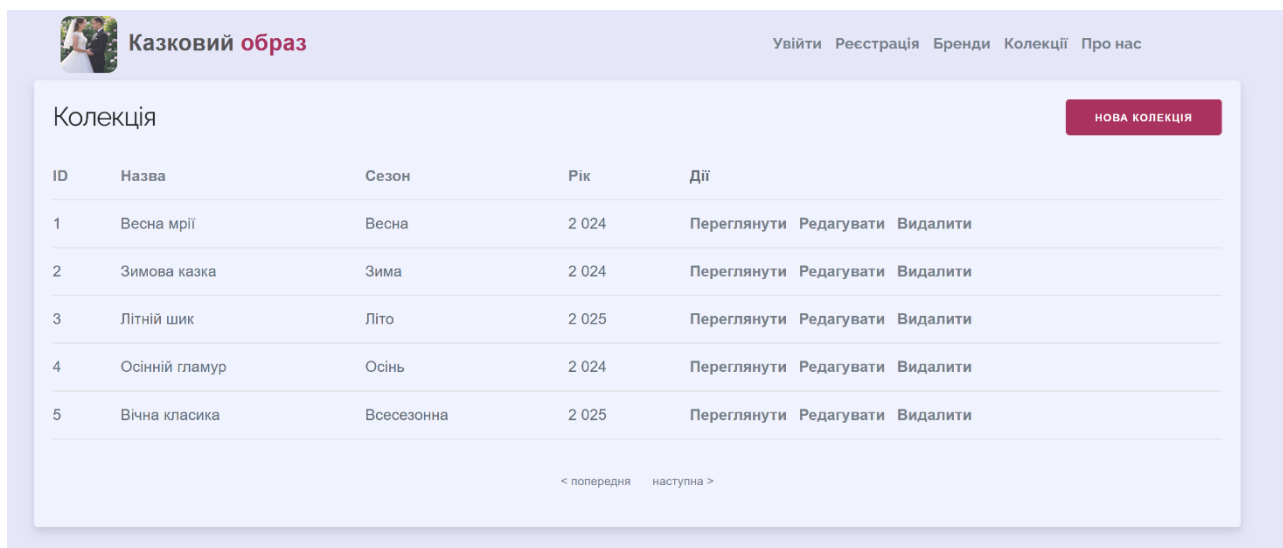


Рисунок Г.7 – Сторінка для керування інформацією про колекції

ДОДАТОК Д – Програма та методика випробування програми

1. Об'єкт випробувань

1.1 Назва об'єкту

Повна назва об'єкта випробувань: вебсервіс для магазину весільного та святкового вбрання "Казковий образ".

Коротка назва: вебсервіс.

1.2 Область застосування

Вебсервіс створений для підвищення ефективності бізнес-процесів, покращення клієнтського досвіду та розширення ринкових можливостей магазину "Казковий образ". Розроблений вебсервіс дозволяє:

- клієнтам: переглядати каталог товарів, здійснювати пошук за фільтрами, оформляти замовлення, бронювати примірки, залишати відгуки.
- співробітникам: оновлювати інформацію про товари, керувати замовленнями та взаємодіяти з клієнтами.

2 Мета випробувань

Мета проведення випробувань – оцінка експлуатаційних характеристик вебсервіса, перевірка і підтвердження працездатності програми в умовах, максимально наближених до умов реальної експлуатації.

3 Вимоги до програми

Вебсервіс має реалізовувати функції, описані в технічному завданні, яке наведено в додатку А.

4 Вимоги до програмної документації

Для проведення тестування повинна бути надана наступна документація:

- технічне завдання на розробку програмного забезпечення;
- текст програми;
- опис програми;
- інструкція користувача;
- програма і методика випробувань програмного забезпечення.

5 Склад, порядок та методи випробувань

5.1 Перевірка програми на відповідність технічному завданню:

- Реєстрація та авторизація користувачів.
- Перегляд каталогу товарів.
- Пошук товарів за категоріями та ключовими словами.
- Додавання товарів до кошика.
- Оформлення замовлення та оплата онлайн.
- Відстеження статусу замовлення.
- Оцінка та відгуки про товари.

5.2 Перевірка програми на її програмно-апаратну сумісність: тестування на апаратних-програмних платформах різної конфігурації, але не нижче за мінімальні вимоги, наведені в технічному завданні; перевірка на наявність і усунення всіх помилок, зазначених у п. 5.1.

5.3 Візуальний перегляд програми: остаточне налагодження на предмет виявлення та усунення дрібних помилок.

Методи випробувань

1) Методика проведення перевірки вимог до програмної документації

Перевірка дотримання вимог програмної документації на програмний продукт виконується візуально представником служби, відповідальної за експлуатацію. У ході перевірки зіставляється склад і комплектність програмної документації, представленої розробником, з переліком програмної документації, наведеним у пункті «Склад програмної документації, пропонованої на випробування» цього документа.

Перевірка вважається завершеною у випадку відповідності складу та комплектності програмної документації, представленої розробником, переліком програмної документації, наведеному у зазначеному вище пункті.

2) Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» технічного завдання.

Перевірка вважається завершеною у разі відповідності складу і послідовності дій, при виконанні даної перевірки, вказаною вище пункту технічного завдання.

В процесі тестування потрібно перевірити функціональність вебсервіс для магазину весільного та святкового вбрання "Казковий образ". У таблиці Д.1, що наведена нижче, вказано перелік випробувань основних функціональних можливостей.

Таблиця Д.1 – Методика випробування

№	Дія	Очікуваний результат	Результат перевірки	Зауваження
1	2	3	4	5
1	Перевірка правильності авторизації у програмі	Авторизацію пройдено при вводі правильного логіну та паролю		
2	Перевірка правильності переходів на вкладку колекцій	Відкривається вкладка з колекціями		
3	Перевірка правильності додання запису про товар	Програма додає запис про товар		
4	Перевірка правильності видалення даних товар	Програма видаляє потрібний запис про товар		
5	Перевірка правильності формування замовлення	Програма формує замовлення з товарами, які обрав клієнт		
6	Перевірка правильності функції пошуку товарів	Програма виводить список товарів, які відповідають запиту		