

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему: «РОЗРОБКА ВЕБ-САЙТУ ДЛЯ УПРАВЛІННЯ ПОСЛУГАМИ В СФЕРІ
КРАСИ ТА ЗДОРОВ'Я»

здобувача Маліневської Єви Едуардівни

_____ (Підпис)

Науковий керівник:

к.т.н., доцент

_____ Васильєва Л.В.

**Робота допущена до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)
освітньо-професійної програми

_____ Помазун О.М.

“ _____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Тішков Б.О.

“ _____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти *Маліневської Єви Едуардівни*

заочної форми навчання

на підготовку кваліфікаційної бакалаврської роботи

на тему: «Розробка веб-сайту для управління послугами в сфері краси та здоров'я»

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р.
№ 467- ст.

Кваліфікаційна бакалаврська робота виконується на матеріалах

наукових публікацій, інтернет-ресурсів

План кваліфікаційної бакалаврської роботи

Розділ I Характеристика та аналіз предметної галузі.

Розділ II Розробка вимог і моделювання інформаційної системи.

Розділ III Проектування та реалізація компонентів системи.

Об'єкт дослідження процес проектування та розробки сучасного веб-сайту для управління послугами у сфері краси та здоров'я, включаючи створення функціональної веб-системи з інтерактивними можливостями для адміністраторів, майстрів і клієнтів.

Предмет дослідження веб-сайт управління послугами в сфері краси та здоров'я його структура, функціональні можливості, дизайн та технічна реалізація.

Мета кваліфікаційної бакалаврської роботи полягає у проектуванні та розробці зручного, адаптивного веб-сайту, який дозволяє автоматизувати ключові бізнес-процеси (онлайн-запис, управління записами, адміністрування послуг) у закладах індустрії краси та здоров'я.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I Провести аналіз предметної галузі. Дослідити особливості роботи закладів краси та здоров'я, визначити основні потреби потенційних користувачів (адміністраторів, майстрів та клієнтів), проаналізувати існуючі аналоги веб-рішень, а також обґрунтувати доцільність розробки нового ресурсу.

У розділі II Представити реалізацію проєктної частини та визначити бізнес-вимоги до веб-системи. Сформулювати функціональні та нефункціональні вимоги до проєктованої веб-системи, розробити структуру бази даних, описати архітектуру системи, а також постановку та алгоритм розв'язання задачі. Здійснити вибір інструментів реалізації проєкту.

У розділі III Реалізувати розроблений програмний продукт. Описати процес розробки веб-сайту, його ключові функції (реєстрація, авторизація, онлайн-запис, каталог послуг та вибір оплати), провести тестування функціональних модулів і наведено приклади роботи системи в реальному середовищі.

**Завдання підготував
науковий керівник**

_____ **Васильєва Людмила Володимирівна**
« 10 » березня 2025 р.

**Завдання одержав
здобувач**

_____ **Маліневська Єва Едуардівна**
« 10 » березня 2025 р..

Відгук
про кваліфікаційну бакалаврську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Маліневської Єви Едуардівни

на тему

«Розробка веб-сайту для управління послугами в сфері краси та здоров'я»

1. Актуальність теми: обрана тема є актуальною, оскільки цифровізація сфери послуг краси та здоров'я є необхідною умовою підвищення її ефективності. Веб-сайт для управління такими послугами значно полегшує організацію роботи, покращує обслуговування клієнтів та автоматизує процеси.
2. Позитивні риси кваліфікаційної бакалаврської роботи: структурованість і логічність подачі матеріалу, якісне оформлення та обґрунтування технічних рішень, високий рівень володіння сучасними технологіями.
3. Наявність самостійних розробок автора: автор самостійно реалізував інтерфейс користувача, структуру бази даних, розробив та протестував основний функціонал системи. Проект демонструє вміння автора працювати з веб-технологіями та орієнтуватися у предметній області.
4. Цінність теоретичних висновків та практичних рекомендацій: отримані теоретичні узагальнення та практичні рекомендації можуть бути застосовані при створенні аналогічних інформаційних систем у сфері послуг. Робота має прикладне значення і демонструє здатність автора до аналізу та оптимізації бізнес-процесів.
5. Наявність недоліків: у роботі місцями спостерігається надмірна деталізація другорядних аспектів, також варто було б приділити більше уваги питанням безпеки даних та масштабування системи
6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК: кваліфікаційний бакалаврський проект є добре структурованим і виконаним дослідженням із дотриманням усіх вимог. Рекомендую допустити проект до захисту перед екзаменаційною комісією.

Науковий керівник



доцент, к.т.н. Васильєва Л.В.

“ 8 ” _____ 06 _____ 2025 р.

Рецензія
на кваліфікаційну бакалаврську роботу
здобувача вищої освіти

Маліневської Єви Едуардівни

(прізвище, ім'я, по батькові)

Тема «Розробка веб-сайту для управління послугами в сфері краси та здоров'я»

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення: тема є актуальною, оскільки автоматизація запису клієнтів та управління послугами в сфері краси відповідає сучасним запитам бізнесу та користувачів. Доцільність розробки обґрунтована зростаючою потребою в онлайн-сервісах у цій галузі.

Якість проведеного дослідження: у роботі представлено чіткий аналіз предметної області, використано сучасні інструменти — Django, SQLite, Bootstrap. Надано UML-діаграми, моделі бази даних, описано функціонал системи. Проєкт логічно структурований, технічно обґрунтований.

Позитивні риси кваліфікаційної бакалаврської роботи: представлено реалістичність та практичність проєкту, використані сучасні технології. Добре оформлена технічна документація, також наявність діаграм і структурованої бази даних.

Зауваження: було б доцільно додати оплату онлайн для зручності користувачів, також більш детально описати процедуру тестування системи.

Практична значимість висновків і рекомендацій: розроблена система може бути впроваджена в реальні заклади сфери краси. Вона спрощує запис клієнтів, управління послугами та зберігання інформації, що позитивно впливає на ефективність роботи сервісу.

Зав. відділення «Програмування»
Відокремленого структурного підрозділу
«Фаховий коледж інформаційних
систем і технологій»



Світлана Миколаївна ЯНЧУК

“10” червня 2025 р.

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «Розробка веб-сайту для управління послугами в сфері краси
та здоров'я»

Київ: кафедра інформаційних систем в економіці, 2025 р.

Кваліфікаційна бакалаврська робота присвячена розробці веб-сайту для управління послугами у сфері краси та здоров'я. Метою роботи є створення зручної, функціональної та адаптивної інформаційної системи, що дозволяє автоматизувати процеси запису клієнтів, адміністрування послуг, ведення бази даних клієнтів, також їх різноманітність у виборі послуг салону.

У першому розділі роботи був проведений аналіз предметної галузі. Дослідили особливості роботи закладів краси та здоров'я, визначили основні потреби потенційних користувачів (адміністраторів, майстрів та клієнтів), проаналізувати існуючі аналоги веб-рішень, а також обґрунтувати доцільність розробки нового ресурсу.

Другий розділ присвячено проєктній частині та визначення бізнес-вимог до веб-системи. В ньому сформульовано функціональні та нефункціональні вимоги до проєктованої веб-системи, розроблено структуру бази даних, описано архітектуру системи, а також постановку та алгоритм розв'язання задачі. Здійснено вибір інструментів реалізації, таких як мова програмування Python, фреймворк Django, база даних SQLite, а також клієнтські технології HTML5, CSS3, Bootstrap і JavaScript (jQuery).

У третьому розділі представлено реалізацію розробленого програмного продукту. Описано процес розробки веб-сайту, його ключові функції (реєстрація, авторизація, онлайн-запис, каталог послуг та вибір оплати), проведено тестування функціональних модулів і наведено приклади роботи системи в реальному середовищі. Результати демонструють ефективність і зручність запропонованого рішення для організації щоденної діяльності салонів краси та центрів здоров'я.

Для клієнтської частини застосовано HTML5, CSS3 і JavaScript з використанням бібліотеки jQuery та фреймворку Bootstrap, що забезпечує адаптивний дизайн інтерфейсу на різних пристроях (ПК, планшети, смартфони). Завдяки використанню AJAX-технологій реалізовано динамічну взаємодію з сервером без перезавантаження сторінок. Передача даних між клієнтом і сервером здійснюється у форматі JSON, що покращує швидкодію та зручність роботи системи.

Описано структуру бази даних SQLite, форми вхідних і вихідних даних, а також проведено тестування працездатності веб-сайту.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 70 сторінок, 2 таблиці, 27 рисунків, список використаних джерел з 11 найменувань та 3 додатки.

«Розробка веб-сайту для управління послугами в сфері краси та здоров'я»

Об'єкт дослідження. Процес проєктування та розробки сучасного веб-сайту для управління послугами у сфері краси та здоров'я, включаючи створення функціональної веб-системи з інтерактивними можливостями для адміністраторів, майстрів і клієнтів.

Предмет дослідження. Веб-сайт управління послугами в сфері краси та здоров'я його структура, функціональні можливості, дизайн та технічна реалізація.

Мета та завдання дослідження. Проєктування та розробка зручного та адаптивного веб-сайту для автоматизації ключових бізнес-процесів (онлайн-реєстрація, управління обліком, управління послугами) закладів індустрії краси та здоров'я.

Виходячи з поставлених цілей, було визначено такі завдання:

Розроблено сучасний веб-сайт для послуг краси та здоров'я, впроваджено ключові функціональні модулі, а також проведено тестування системи на працездатність, що доводить її застосовність у реальному середовищі. Веб-сайт повинен забезпечити зручний доступ до послуг, оптимізує внутрішні процеси салонів краси та покращує якість обслуговування клієнтів.

Теоретична значущість полягає в обґрунтуванні підходів до автоматизації управління послугами у сфері краси та здоров'я, дослідженні особливостей функціонування веб-застосунків, а також у застосуванні методологій об'єктно-орієнтованого аналізу та проєктування з використанням мови моделювання UML. Це створює базу для подальших наукових досліджень і впровадження подібних систем у суміжних сферах.

Методична значущість полягає в розробці послідовної структури реалізації інформаційної системи: від аналізу вимог до впровадження та тестування. Запропонована методика може бути використана як зразок для створення подібних

веб-застосунків у сфері обслуговування, а також у навчальному процесі для студентів ІТ-спеціальностей.

Практична значущість розробленого веб-сайту полягає в його безпосередньому застосуванні у сфері краси та здоров'я. Система дозволяє автоматизувати процес запису клієнтів на послуги, зберігати історію взаємодії з клієнтами, покращувати сервіс та зменшувати навантаження на адміністративний персонал. Веб-застосунок може бути впроваджений у реальні бізнес-процеси салонів краси, SPA-центрів, косметологічних клінік тощо, сприяючи підвищенню ефективності їхньої роботи.

Отримані результати можуть бути використані для подальшого розвитку та комерційного впровадження.

Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: проєкт, веб-сайт, послуги, клієнти, онлайн-запис.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1	7
ХАРАКТЕРИСТИКА ТА АНАЛІЗ РОЗРОБКИ ВЕБ-САЙТУ КРАСИ ТА ЗДОРОВ'Я.....	7
1.1 Характеристика предметної галузі та об'єкта дослідження	7
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі	8
РОЗДІЛ 2	15
РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ НАДАННЯ ПОСЛУГ В СФЕРІ КРАСИ ТА ЗДОРОВ'Я.....	15
2.1 Аналіз і специфікація вимог до інформаційної системи.....	15
2.2 Постановка та алгоритм розв'язання задачі	24
2.3 Моделювання інформаційної системи	29
РОЗДІЛ 3	32
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ.....	32
3.1 Інформаційне забезпечення	32
3.2 Технічне забезпечення	33
3.3 Програмне забезпечення.....	34
3.4 Результати реалізації інформаційної системи	37
ВИСНОВКИ	46
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ДОДАТКИ	49

ВСТУП

У сучасному середовищі цифрової трансформації все більше галузей потребують інтеграції інформаційних технологій для підвищення операційної ефективності. Індустрія краси та здоров'я не є винятком. Зіткнувшись зі зростаючою конкуренцією, зростанням очікувань клієнтів та потребою в гнучкому управлінні робочим часом, підтримці онлайн-бронювання та зворотного зв'язку, сучасні, зручні та потужні веб-рішення є вкрай необхідними. Створення веб-сайту для управління послугами в цій галузі може автоматизувати ключові бізнес-процеси: управління записами, управління базою даних клієнтів, організацію співробітників, доступ до списків послуг та ефективну комунікацію з користувачами. Такі рішення допомагають оптимізувати корпоративні ресурси, покращити якість послуг та підвищити загальну конкурентоспроможність організації. Тому розробка веб-сайту для управління послугами в індустрії краси та здоров'я є важливим та значущим завданням, яке відповідає сучасним технологічним тенденціям та потребам ринку.

В контексті сучасної цифрової трансформації бізнесу веб-технології займають ключове місце в автоматизації сфери послуг, зокрема індустрії краси та здоров'я. В останні роки спостерігається зростання інтересу до рішень, які дозволяють оптимізувати процеси бронювання, взаємодії з клієнтами, обліку послуг та персоналізованого маркетингу. Це відображено в багатьох наукових дослідженнях, публікаціях та практичних розробках.

У спеціалізованій літературі питанням моделювання інформаційних систем, створення зручного користувацького інтерфейсу та побудови архітектур веб-додатків з використанням сучасних фреймворків, таких як Django, Laravel, Node.js тощо, приділяється велика увага. Наприклад, у роботах Іванченка В.М. Веб-технології: навчальний посібник [1] та Мельниченка С.О. Основи розробки веб-систем: монографія [2] досліджуються методи побудови інтерактивних веб-систем, що враховують особливості поведінки користувачів та вимоги UI/UX.

Ще одним самостійним напрямком досліджень є розробка адаптивного дизайну веб-інтерфейсів, яка була досліджена в роботі Петренко Н.В. Адаптивний веб-дизайн у сфері сервісних послуг // Інформаційні технології і системи. [3], яка проаналізувала ефективність використання CSS-фреймворків (таких як Bootstrap) для створення зручних мобільних інтерфейсів для платформ онлайн-сервісів.

У роботах Козлової О.П. Впровадження CRM у закладах індустрії краси: досвід та перспективи [4] та Дмитренка Ю.В. Інформаційна безпека при роботі з персональними даними клієнтів у веб-середовищі // Управління розвитком складних систем [5] досліджуються практичні питання автоматизації бізнес-процесів у сфері послуг. Перша досліджує інтеграцію CRM-модулів у систему запису на прийом салонів краси, а друга аналізує безпеку веб-систем обробки персональних даних клієнтів.

Крім того, важливий внесок у розробку типових рішень також зробили огляди популярних SaaS-платформ для салонів краси (Fresha, Booksy, Bookon), опубліковані на професійних ІТ-блогах та бізнес-виданнях. Ці платформи продемонстрували ефективність впровадження автоматизованих веб-систем у реальних середовищах.

На основі аналізу наукових та практичних джерел можна зробити висновок, що питання цифровізації процесів управління послугами є актуальними, добре дослідженими та активно розробляються, що створює передумови для впровадження власного програмного рішення з урахуванням існуючого досвіду та сучасних технологій.

Мета і завдання дослідження. Полягає у проєктуванні та розробці зручного, адаптивного веб-сайту, який дозволяє автоматизувати ключові бізнес-процеси (онлайн-запис, управління записами, адміністрування послуг) у закладах індустрії краси та здоров'я.

Об'єкт дослідження. Процес проєктування та розробки сучасного веб-сайту для управління послугами у сфері краси та здоров'я, включаючи створення функціональної веб-системи з інтерактивними можливостями для адміністраторів, майстрів і клієнтів.

Предмет дослідження. Веб-сайт управління послугами в сфері краси та здоров'я його структура, функціональні можливості, дизайн та технічна реалізація.

Методи дослідження. У процесі виконання дипломного проєкту були застосовані такі наукові методи дослідження:

- Аналіз та синтез – для вивчення предметної галузі, виявлення вимог до системи та визначення функціональних і нефункціональних характеристик майбутнього веб-застосунку;
- Порівняльний аналіз – для аналізу наявних аналогічних інформаційних систем (Fresha, Booksy, Bookon) та виділення їхніх переваг і недоліків;
- Моделювання – для побудови UML-діаграм, що описують структуру та функціонування системи;
- Проєктування – для розробки структури бази даних, архітектури веб-сайту та користувацького інтерфейсу;
- Метод спостереження та експерименту – для перевірки працездатності програмного продукту, оцінювання зручності користувача та виявлення можливих покращень.

Теоретична значущість полягає в обґрунтуванні підходів до автоматизації управління послугами у сфері краси та здоров'я, дослідженні особливостей функціонування веб-застосунків, а також у застосуванні методологій об'єктно-орієнтованого аналізу та проєктування з використанням мови моделювання UML. Це створює базу для подальших наукових досліджень і впровадження подібних систем у суміжних сферах.

Методична значущість полягає в розробці послідовної структури реалізації інформаційної системи: від аналізу вимог до впровадження та тестування. Запропонована методика може бути використана як зразок для створення подібних веб-застосунків у сфері обслуговування, а також у навчальному процесі для студентів ІТ-спеціальностей.

Практична значущість розробленого веб-сайту полягає в його безпосередньому застосуванні у сфері краси та здоров'я. Система дозволяє

автоматизувати процес запису клієнтів на послуги, зберігати історію взаємодії з клієнтами, покращувати сервіс та зменшувати навантаження на адміністративний персонал. Веб-застосунок може бути впроваджений у реальні бізнес-процеси салонів краси, SPA-центрів, косметологічних клінік тощо, сприяючи підвищенню ефективності їхньої роботи.

Було розроблено сучасний веб-сайт для управління послугами у сфері краси та здоров'я, реалізовано ключові функціональні модулі, протестовано працездатність системи, що підтвердило її придатність для використання у реальному середовищі. Веб-сайт забезпечує зручний доступ до послуг, оптимізує внутрішні процеси салону краси та підвищує якість обслуговування клієнтів.

Використані інструментальні засоби: середовище розробки PyCharm, операційна система Windows 11, текстовий редактор Microsoft Word 2016 для підготовки та оформлення пояснювальної записки.

Структура роботи. Загальний обсяг роботи становить 70 сторінок друкованого тексту, 27 рисунків на 12 сторінках, 2 таблиці на 2 сторінках, 3 додатки на 20 сторінках. Список використаних джерел налічує 11 найменування.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА АНАЛІЗ РОЗРОБКИ ВЕБ-САЙТУ КРАСИ ТА ЗДОРОВ'Я

1.1 Характеристика предметної галузі та об'єкта дослідження

Індустрія краси та здоров'я є однією з перспективніших та динамічніших галузей у сучасній сфері послуг. Вона охоплює широкий спектр послуг, включаючи: перукарські послуги, косметичні процедури, манікюр та педикюр, масаж, SPA, солярій, дерматологію тощо. Галузь є висококонкурентною, тенденції швидко змінюються, а вимоги до якості послуг та обслуговування клієнтів стають все вищими.

Тому компаніям у галузі краси та здоров'я необхідно застосовувати сучасні цифрові технології, особливо веб-рішення, які можуть автоматизувати типові бізнес-процеси та підвищити рівень обслуговування.

Веб-сайти можуть вирішити такі основні проблеми в галузі:

- Ведення ручних або неструктурованих процесів реєстрації клієнтів, що призводить до плутанини та потенційної втрати прибутку;
- Відсутність єдиної інформаційної платформи, що охоплює адміністраторів, керівників та клієнтів;
- Недостатній контроль над робочим навантаженням співробітників та неефективне планування роботи;
- Обмежене просування послуг та комунікація з клієнтами;
- Відсутність використання функцій CRM (збереження історії відвідувань, уподобань, знижок тощо).

Все це робить необхідним створення веб-сайту, який не лише має функцію відображення, а й служить платформою для взаємодії між клієнтами, керівництвом та співробітниками агентства.

Об'єкт дослідження. Процес проєктування та розробки сучасного веб-сайту для управління послугами у сфері краси та здоров'я, включаючи створення функціональної веб-системи з інтерактивними можливостями для адміністраторів, майстрів і клієнтів.

Дослідження охоплює такі ключові елементи:

- Управління обліком клієнтів (записи, нагадування, скасування);
- Управління каталогом послуг та головним каталогом;
- Контроль робочого навантаження співробітників.

Важливо також, що цифровізація процесів обслуговування позитивно впливає на репутацію бренду, допомагає створити постійну клієнтську базу та підвищити лояльність споживачів до закладу.

Предметна галузь, яку охоплює проєкт, – це цифрові інструменти для сфери послуг, зокрема програмне забезпечення для веб-автоматизації процесів ведення обліку, бухгалтерського обліку, комунікації та управління в закладах краси та велнесу. Цей проєкт вивчає та застосовує сучасні методи веб-програмування, бази даних, дизайн інтерфейсу користувача та адаптивні методи організації онлайн-послуг у галузі.

Таким чином, розробка веб-сайту для управління послугами є відповіддю на сучасні ринкові виклики та прикладом застосування сучасних інформаційних технологій для покращення реальних бізнес-процесів.

1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Аналіз літератури показує, що застосування цифровізації в індустрії краси та здоров'я стимулює науковий та практичний інтерес. У низці робіт Ковальчук Л.М. Ковальчук Л.М. «Інформаційні системи в сервісній діяльності», 2021 [6] та Петренко О.С. «Моделі та методи автоматизації процесів у сфері послуг», 2020 [7] також зазначають, що цифровізація процесів обслуговування дозволяє ефективно

керувати ресурсами, покращувати якість обслуговування клієнтів та точніше аналізувати дані про корпоративну діяльність.

Основні функціональні вимоги до таких систем включають:

- Автоматична реєстрація на послуги;
- Управління графіками та робочим навантаженням керівників;
- Надання інформації про послуги та експертів;
- Збереження історії відвідувачів та уподобань клієнтів;
- Забезпечення онлайн-спілкування.

У літературі також наголошується, що такі інформаційні системи (ІС) підтримують адаптивність (відображення на різних пристроях), безпеку обробки персональних даних, резервне копіювання інформації та масштабованість.

У літературі, присвяченій веб-розробці, детально розглядаються технічні аспекти впровадження таких систем:

- HTML5/CSS3 та JavaScript є базовими технологіями для клієнтської сторони;
- PHP (Laravel), Python (Django) або Node.js є поширеними рішеннями для серверної логіки;
- MySQL, PostgreSQL, MongoDB;
- REST API – системні компоненти.

У деяких джерелах згадується реалізація принципів UX/UI[8]. Забезпечення багатомовних інтерфейсів та локалізації для користувачів у різних регіонах.

Сьогодні індустрія краси та здоров'я активно впроваджує інформаційні системи (ІС) для автоматизації бізнес-процесів, покращення комунікації з клієнтами та підвищення ефективності управління. Це дозволяє компаніям цієї галузі вести клієнтську базу, організовувати електронний облік, керувати плануванням персоналу, вести фінансовий облік та реалізовувати маркетингові стратегії.

Розглянемо три популярні програмні рішення, які широко використовуються як у міжнародній, так і в українській практиці: Fresha, Booksy та Bookon (Binotel).

Fresha [9] [<https://www.fresha.com/>] – одна з найпопулярніших міжнародних платформ, що спеціалізується на автоматизації бізнесу в секторах краси, здоров'я та велнесу та співпрацює з тисячами салонів краси, масажних салонів, перукарень, клінік естетичної медицини тощо.

Основні переваги Fresha:

- Управління бронюванням: Клієнти можуть записуватися на прийом онлайн у режимі реального часу.
- Гнучкий графік: Платформа може легко керувати змінами в робочому графіку агента.
- Маркетинг: Вбудовані інструменти для постійних клієнтів (автоматичні акції, знижки), SMS та email-маркетингові кампанії для постійних клієнтів.
- Фінансовий аналіз: Система може зберігати звіти про дохід, відвідування та статистику співробітників.
- Інвентаризація: Облік товарів, продажів та наявності продукції в салоні.
- Інтеграції: Форми онлайн-реєстрації можна створювати на веб-сайті, в Instagram, Facebook та Google.

Система має адаптивну версію для мобільних пристроїв та два мобільні додатки – один для адміністраторів/спеціалістів та один для клієнтів, знайдених через її платформу маркетплейсу.

Недоліки:

- Відсутність україномовного інтерфейсу;
- Функціональність недостатньо локалізована для задоволення потреб українських користувачів;
- Опора на комісійну модель.

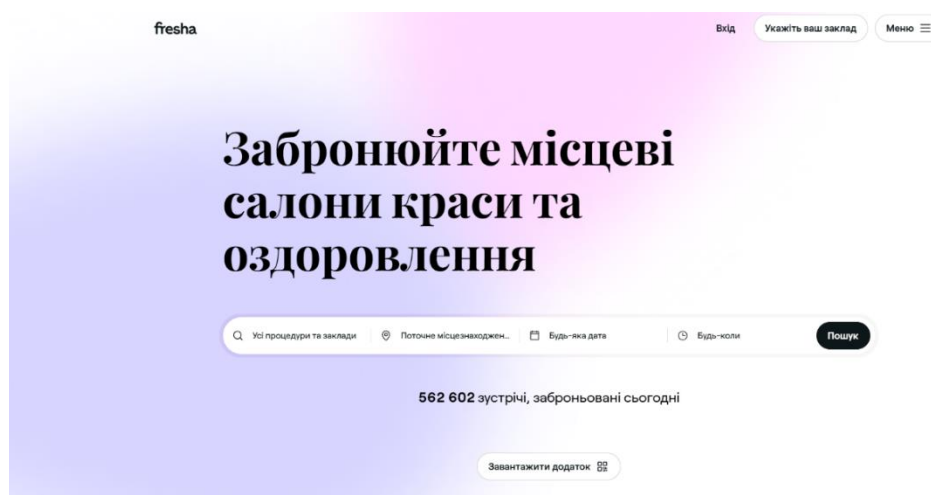


Рисунок 1.1 – Головна сторінка «Fresha»

Джерело: [9]

Booksy [10] [<https://booksy.com/>] – це міжнародна система управління салонами краси, що походить зі Сполучених Штатів. Система широко використовується в понад 20 країнах, включаючи Польщу, Велику Британію, Іспанію, Бразилію та Південну Африку. Система орієнтована на окремих косметологів та великі мережі салонів краси.

Основні характеристики:

- Цілодобове онлайн-бронювання: запис на прийом через веб-сайт, додаток або сторінку косметолога.
- CRM-система: база даних клієнтів, історія відвідувань, нотатки косметолога, минулі замовлення, служба рекомендацій клієнтів.
- Мобільний додаток: для клієнтів та косметологів (iOS та Android).
- Нагадування: автоматично повідомляє клієнтів про майбутні візити (SMS, push).
- Інтеграція: Google Reserve, Instagram, Facebook та власний веб-сайт.
- Фінансовий облік: контроль доходів та витрат, розрахунок комісій, формування звітів.
- Онлайн-оплата: прийом платежів банківськими картками через платформу.

Booksy підтримує кілька тарифних планів, включаючи фіксовані щомісячні платежі або оплату за бронювання.

Недоліки:

- Дорогі для малого бізнесу;
- Відсутність української локалізації;
- Деякі інструменти доступні лише в певних країнах.

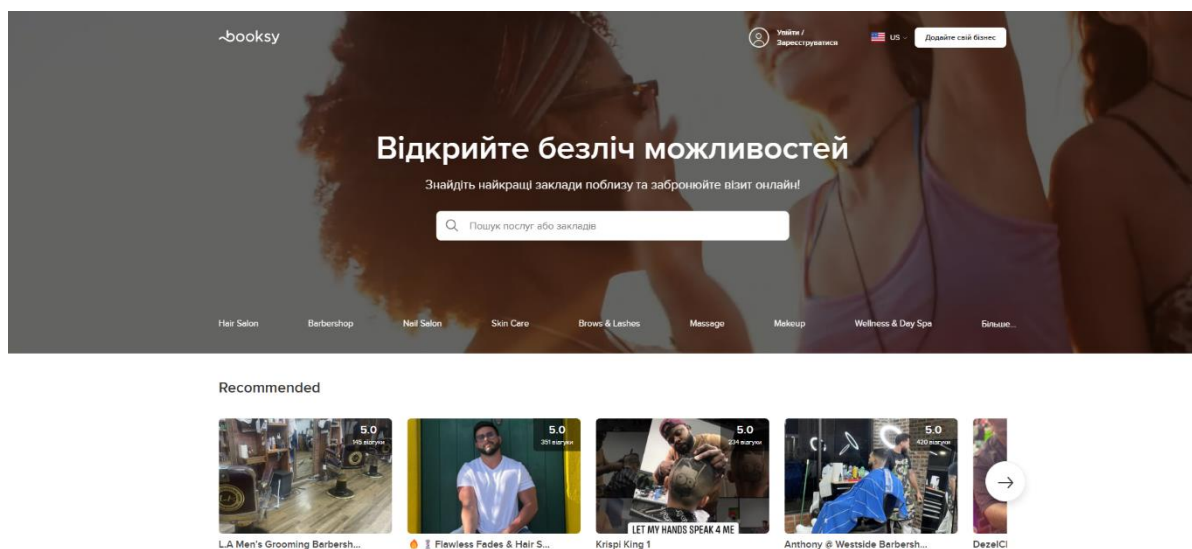


Рисунок 1.2 – Головна сторінка «Booksy»

Джерело: [10]

Bookon (Binotel) [11] [<https://bookon.binotel.ua/>] — українська онлайн-система для автоматизації запису клієнтів і керування послугами в сфері краси, розроблена компанією Binotel. Сервіс орієнтований на ринок України, а також частково країн СНД.

Ключові особливості:

- Зручний веб-інтерфейс українською мовою;
- Гнучке налаштування послуг та спеціалістів;
- Автоматизований онлайн-запис: через віджет на сайт, соціальні мережі або унікальне посилання;
- SMS- та e-mail-нагадування клієнтам про запис;
- Повна інтеграція з IP-телефонією Binotel: запис дзвінків, ведення історії комунікацій, розпізнавання клієнта під час вхідного дзвінка;
- CRM-модуль: облік візитів, статистика, повторні записи, історія клієнта;

- Панель аналітики: відстеження завантаженості майстрів, популярності послуг, доходу.

Платформа працює за передплатною моделлю, де вартість залежить від обраного пакету функцій. Підтримка доступна українською мовою, що робить Bookon зручним і доступним рішенням для малого та середнього бізнесу в Україні.

Переваги:

- Повна локалізація;
- Власна технічна підтримка;
- Сумісність із популярними українськими CRM та телефонією.

Недоліки:

- Відсутність мобільного застосунку для клієнтів;
- Менший обсяг аналітики у порівнянні з міжнародними системами.

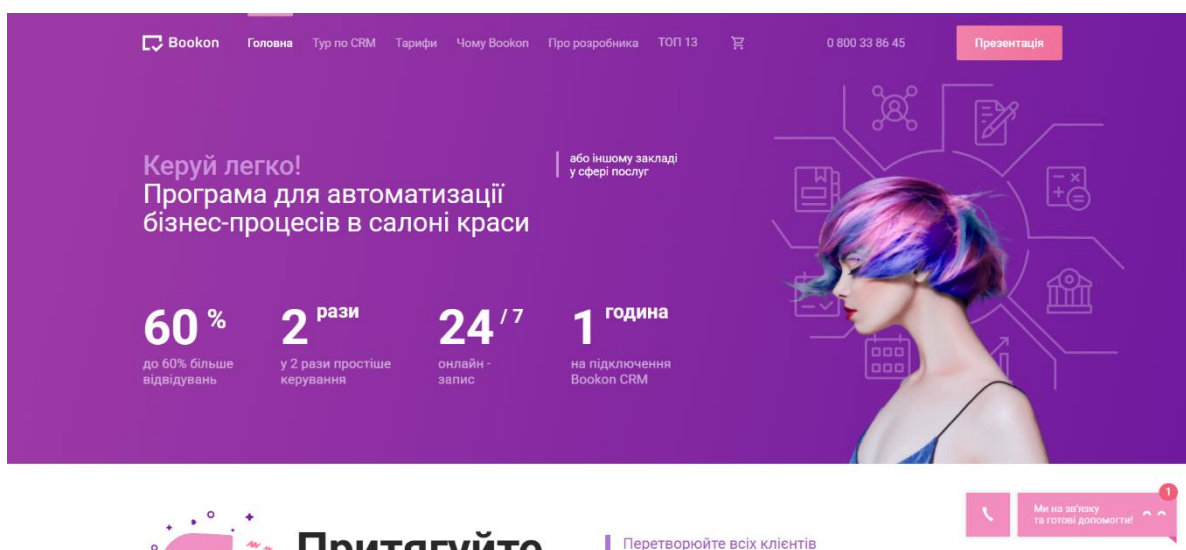


Рисунок 1.3 – Головна сторінка «Bookon (Binotel)»

Джерело: [11]

Таблиця 1.1 – Порівняльна характеристика платформ.

Платформа	Країна	Онлайн-запис	CRM-система	SMS / Email-нагадування	Мобільний застосунок	Інтеграція з соцмережами	Мова інтерфейсу	Комерційні умови
<i>Fresha</i>	Велика Британія	Так	Так	Так	Так	Instagram, Facebook	Англійська	Безкоштовно, комісія за нових клієнтів
<i>Booksy</i>	США / ЄС	Так	Так	Так	Так	Google, Instagram	Англійська	Платна підписка або комісія
<i>Bookon (Binotel)</i>	Україна	Так	Так	Так	Ні	Немає	Українська	Платна модель, гнучке ціноутворення

Джерело: сформовано автором на основі виконаного дослідження

Усі платформи пропонують ефективні інструменти для автоматизації бізнесу в секторі краси та здоров'я. Міжнародні рішення (Fresha, Booksy) мають широкий функціонал, але не підходять для місцевого українського ринку через мовні особливості та особливості інтеграції.

У цьому контексті український сервіс Bookon є вигідною альтернативою, яка забезпечує зручну локалізацію, підтримку та функції, адаптовані до потреб малого та середнього бізнесу. Це підтверджує доцільність розробки професійного веб-сайту, який враховує місцеві потреби, функціональні вимоги та очікування користувачів.

РОЗДІЛ 2

РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ НАДАННЯ ПОСЛУГ В СФЕРІ КРАСИ ТА ЗДОРОВ'Я

2.1 Аналіз і специфікація вимог до інформаційної системи

Розробка інформаційної системи, яка забезпечує автоматизацію ключових бізнес-процесів у сфері краси та здоров'я, передбачає детальне формулювання бізнес-вимог. Ці вимоги відображають очікування від проєкту як з боку замовника, так і з боку кінцевих користувачів, а також визначають базові умови для реалізації проєкту командою розробки.

Бізнес-вимоги можна умовно поділити на три групи: загальні (що визначають функціональну мету проєкту), вимоги користувача (з урахуванням ролей у системі) та вимоги розробника (що задають рамки технічної реалізації). Нижче наведено розгорнуту характеристику кожної групи.

Загальні бізнес-вимоги визначаються з урахуванням стратегічних цілей бізнесу, для якого створюється веб-сайт. Вони визначають, якими ключовими функціями має володіти система, щоб бути ефективною, конкурентоспроможною та зручною у використанні.

- Система має забезпечувати централізоване керування послугами, клієнтськими записами, розкладом майстрів, фінансовими даними та аналітикою.
- Користування платформою повинно бути можливим у будь-який час доби через сучасні браузерери та пристрої, включаючи смартфони та планшети.
- Веб-сайт має забезпечувати користувачам інтуїтивно зрозумілий інтерфейс для взаємодії з основним функціоналом – зокрема, перегляду послуг, цін, вільних слотів для запису та здійснення бронювання.

- Усі дані про клієнтів, візити, платежі та інші транзакції мають зберігатися у захищеному середовищі з дотриманням принципів конфіденційності та актуального законодавства.
- Система повинна підтримувати розширення функціоналу без потреби кардинальної переробки коду.
- Необхідна можливість підключення зовнішніх сервісів, зокрема платіжних систем, засобів комунікації, CRM, а також підтримка маркетингових інструментів.

Бізнес-вимоги користувача формуються у процесі взаємодії з системою користувачі виконують різні функції в залежності від своєї ролі – клієнт, майстер, адміністратор. Для кожної ролі визначаються окремі вимоги, які відображають їх потреби та очікування.

Клієнт:

- Повинен мати змогу зручно переглядати каталог послуг, дізнаватися ціни, читати описи та переглядати профілі спеціалістів.
- Має мати доступ до швидкого й простого механізму онлайн-запису, із вибором дати, часу та майстра.
- Має отримувати автоматичні повідомлення (SMS або email) з підтвердженням запису та нагадуваннями перед візитом.
- Повинен мати особистий кабінет, у якому зберігається історія записів, улюблені послуги та можливість оцінити роботу салону.
- Повинна існувати можливість скасування або зміни запису без необхідності телефонного дзвінка.

Майстер:

- Повинен мати інструменти для перегляду власного графіка роботи, списку клієнтів та призначених процедур.
- Має мати змогу залишати службові нотатки щодо клієнта (наприклад, особливості процедур або побажання).
- Повинен мати змогу редагувати свій розклад або позначати години недоступності.

Адміністратор/власник:

- Повинен мати повний доступ до панелі керування, де можна редагувати послуги, налаштовувати ціни, акції, знижки.
- Має мати змогу призначати ролі іншим користувачам (майстер, клієнт, менеджер).
- Повинен отримувати статистичні звіти (щоденні, тижневі, місячні) щодо кількості візитів, завантаженості працівників, рівня доходів тощо.
- Має мати можливість експортувати дані для зовнішнього аналізу або обліку.

Бізнес-вимоги розробника визначаються для команди розробки з метою забезпечення ефективної, масштабованої та безпечної реалізації функціоналу веб-сайту.

- Архітектура повинна бути модульною, що забезпечує легке додавання нових функцій.
- Бекенд реалізовується на сучасному технологічному стеку (наприклад, Node.js, Django або Laravel), а фронтенд – з використанням React, Vue або аналогічного фреймворку.
- База даних повинна бути реляційною (MySQL, PostgreSQL) із чітко структурованими зв'язками між таблицями.
- Веб-сайт має бути повністю адаптивним – коректно відображатися як на ПК, так і на мобільних пристроях.
- Розробка має включати налаштування системи автентифікації з підтримкою JWT або OAuth.
- Повинна бути реалізована система логування подій та обробки помилок для моніторингу стану сайту.
- Для роботи з конфіденційними даними слід застосовувати шифрування та SSL-сертифікати.
- Необхідно реалізувати механізм регулярного резервного копіювання бази даних.

- Усі зміни повинні відслідковуватись у системі контролю версій (Git), з можливістю відкотитися до стабільної версії.

Таблиця 2.1 – Пріоритет цілей.

Назва вимоги	Опис	Пріоритет цілі
Централізоване управління	Система повинна забезпечувати централізоване керування послугами, розкладом, клієнтами та фінансами.	1 – Критично важливий
Доступність 24/7	Платформа має бути доступна у будь-який час доби на будь-якому пристрої.	2 – Важливий
Інтуїтивний інтерфейс	Користувач повинен легко взаємодіяти з сайтом без потреби в інструкції.	1 – Критично важливий
Захист даних	Особисті дані мають зберігатися у зашифрованому вигляді згідно з чинним законодавством.	1 – Критично важливий
Можливість масштабування	Архітектура повинна підтримувати подальший розвиток функціоналу.	3 – Помірна важливість
Онлайн-запис	Клієнт повинен мати змогу обрати послугу, дату, майстра та записатись.	1 – Критично важливий
Нагадування	Автоматичні нагадування (SMS, email) мають надсилатися заздалегідь.	2 – Важливий
Календар записів	Доступ до власного розкладу візитів.	1 – Критично важливий
Панель керування	Повний доступ до створення, редагування та видалення даних.	1 – Критично важливий

Джерело: сформовано автором на основі виконаного дослідження

На рисунку 2.1 вказані бізнес вимоги для цього проєкту.

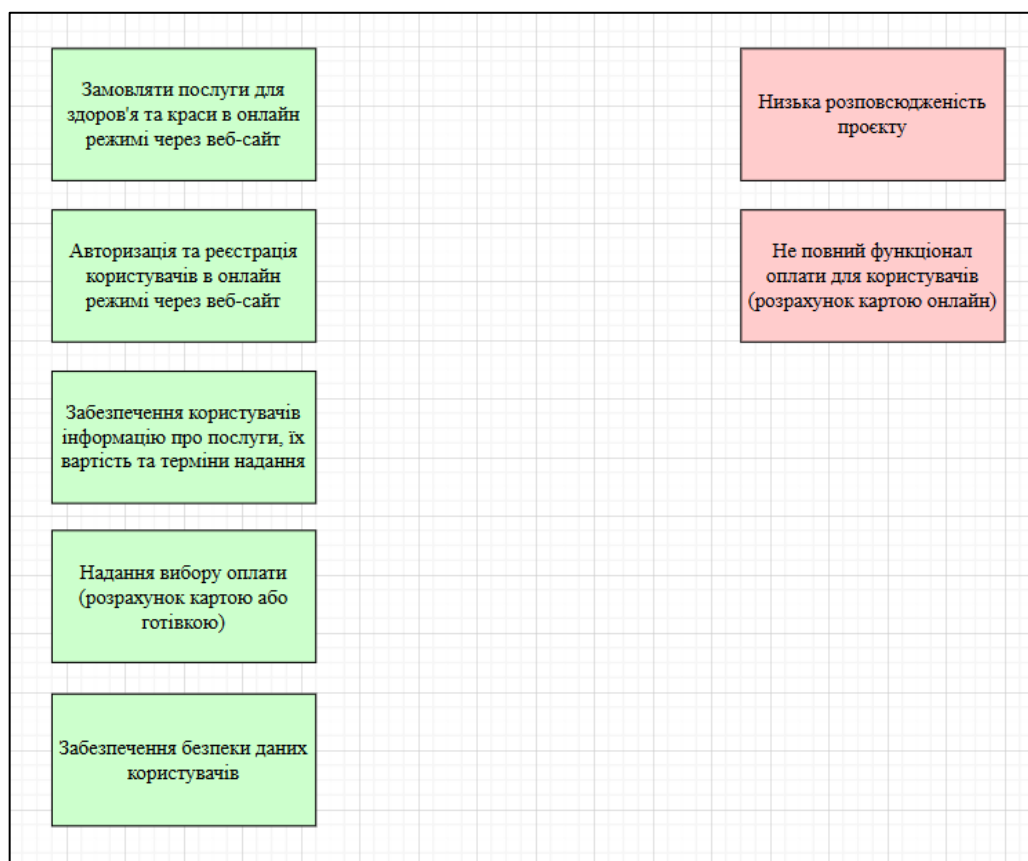


Рисунок 2.1 – Бізнес вимоги до проєкту

Джерело: розроблено автором

Функціональні вимоги визначають, які функції, можливості та завдання повинна виконувати веб-система, щоб задовольнити потреби користувачів і досягти бізнес-цілей. Вони конкретизують очікувану поведінку системи та обмеження, які мають бути враховані під час розробки.

Функціональні вимоги описують різні функції, які система має виконувати, такі як обробка даних, взаємодія з користувачами, збереження та вилучення інформації, автоматизація процесів та інші дії, які повинні бути підтримані системою. Вони визначають необхідну функціональність, яка гарантує ефективне функціонування системи.

Функціональні вимоги також можуть включати умови та обмеження, наприклад, часові рамки, обсяги даних, продуктивність, стандарти безпеки та інші

вимоги, які впливають на роботу системи. Вони допомагають забезпечити відповідність системи встановленим стандартам та вимогам користувачів.

Нижче наведено можливі функціональні вимоги до веб-сайту з управління послугами краси та здоров'я:

- Реєстрація користувачів:
 - Можливість зареєструвати нових користувачів, обов'язково заповнюючи певні поля.
 - Можливість увійти в систему для зареєстрованих користувачів.
- Перегляд інформації про послуги:
 - Представлення переліку доступних послуг краси та здоров'я.
 - Відображення вартості, термінів виконання та умов оплати кожної конкретної послуги.
- Організація вибору оплати:
 - Забезпечення способів оплат, таких як оплата картою чи готівкою.
- Історія замовлень:
 - Можливість перегляду історії замовлень користувача та їх поточних статусів.
- Забезпечення безпеки даних:
 - Захист особистих даних користувачів, історії замовлень та їх статус.

Для забезпечення зручного та ефективного процесу управління послугами веб сайту краси та здоров'я, ці функціональні вимоги мають бути втілені в дизайні та розробці веб-сайт.

На рисунку 2.2 вказані функціональні вимоги для цього проєкту.

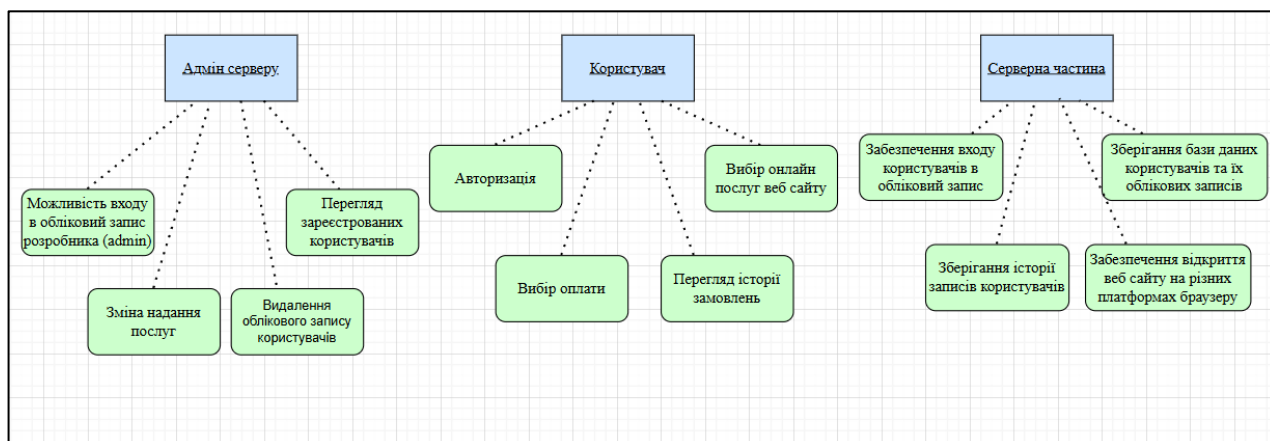


Рисунок 2.2 – Функціональні вимоги до проекту

Джерело: розроблено автором

Нефункціональні вимоги при розробці веб-системи визначають характеристики та обмеження системи, які не впливають на її функціональність, але мають значення для її якості, ефективності та поведінки. На перший погляд, вони можуть здатися менш важливими, але насправді грають важливу роль у забезпеченні успішного функціонування системи.

Основні категорії нефункціональних вимог включають такі аспекти:

- **продуктивність:** ці вимоги визначають швидкодію та ефективність системи. Вони охоплюють такі показники, як час відгуку системи, час виконання операцій, пропускну здатність та швидкодію обробки даних;
- **надійність:** вимоги до надійності стосуються стабільності та безвідмовної роботи системи. Вони визначають тривалість безвідмовної роботи, можливість відновлення після помилок, обробку помилок та відновлення системи після відмови;
- **безпека:** ці вимоги встановлюють заходи для захисту системи від несанкціонованого доступу, злому, витоку даних та інших загроз безпеці. Вони можуть включати аутентифікацію користувачів, авторизацію, шифрування даних, контроль доступу та інші механізми безпеки;
- **масштабованість:** вимоги до масштабованості визначають здатність системи працювати ефективно без втрати продуктивності при збільшенні обсягу

даних, кількості користувачів або навантаженні системи. Вони також стосуються легкості розширення та інтеграції з іншими системами;

- удосконаленість: ці вимоги відображають можливість системи бути модифікованою, розширеною та покращеною для відповіді на змінні потреби користувачів. Вони визначають гнучкість системи та легкість внесення змін без порушення роботи системи;
- сумісність: вимоги до сумісності стосуються здатності системи взаємодіяти та інтегруватися з іншими системами, стандартами або протоколами. Вони визначають підтримку певних форматів даних, протоколів комунікації або стандартів безпеки;
- використання: ці вимоги стосуються зручності, доступності та інтуїтивного розуміння системи для користувачів. Вони охоплюють ергономіку, зручність використання, документацію та навчання користувачів.

Це лише декілька основних категорій нефункціональних вимог, а конкретні вимоги будуть залежати від потреб та вимог конкретної інформаційної системи.

При проєктуванні веб-сайту для управління послугів у сфері краси та здоров'я можна реалізувати такі нефункціональні вимоги:

- продуктивність:
 - швидкодія веб-сайту, щоб користувачі могли ефективно взаємодіяти з веб-системою без затримок;
 - ефективна обробка та відображення інформації про послуги, їх вартість, терміни виконання та умови оплати;
- надійність:
 - стабільна робота веб-сайту без відмов та падінь;
 - забезпечення надійного збереження даних користувачів та історії замовлень;
- безпека:
 - захист особистих даних користувачів, забезпечення конфіденційності та цілісності інформації;

- механізми аутентифікації та авторизації користувачів для обмеження доступу до системи та її функціональності;
- масштабованість:
 - здатність системи працювати ефективно та зберігати продуктивність при збільшенні обсягу замовлень та користувачів;
- удосконаленість:
 - легкість модифікації та розширення функціональності веб-сайту для врахування змінних потреб користувачів та розвитку бізнесу;
 - можливість внесення змін без перебоїв у роботі системи та без необхідності повного перепроєктування;
- сумісність:
 - взаємодія з іншими системами, стандартами або протоколами, що може включати інтеграцію з платіжними шлюзами, електронною поштою або іншими сервісами;
 - підтримка стандартних форматів даних та протоколів комунікації для забезпечення сумісності з іншими системами;
- використання:
 - зручність і інтуїтивне розуміння інтерфейсу веб-сайту для користувачів;
 - доступність документації, навчальних матеріалів та підтримки користувачів для полегшення використання системи.

Нефункціональні вимоги до системи управління послугами краси та здоров'я наведено нижче на рисунку 2.3.

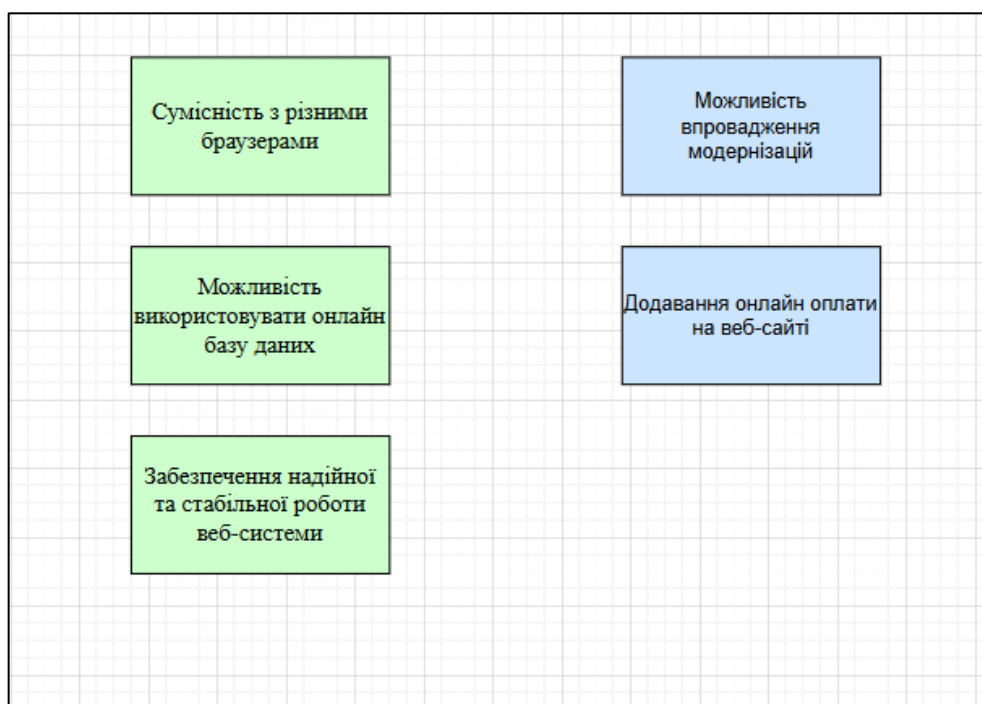


Рисунок 2.3 – Діаграма нефункціональних вимог до проєкту

Джерело: розроблено автором

2.2 Постановка та алгоритм розв'язання задачі

Постановка задачі: Розробка веб-сайту для управління послугами в сфері краси та здоров'я

Характеристика задачі

Задача полягає в розробці веб-сайту для управління послугами в сфері краси та здоров'я. Ця система призначена для вибору, бронювання послуг, авторизації користувачів, перегляду історії замовлень.

Призначення і техніко-економічна сутність задачі:

Основне призначення цієї веб-системи полягає в спрощенні та автоматизації процесу замовлення послуг здоров'я та краси. Вона дозволяє клієнтам

авторизуватися, вибрати необхідні послуги, оформити їх та вибрати спосіб оплати (готівка чи карта).

Задача розробки веб-сайту для управління послугами в сфері краси та здоров'я спрямована на створення ефективного інструменту автоматизації бізнес-процесів у закладах, які надають естетичні та оздоровчі послуги (салони краси, барбершопи, масажні кабінети, SPA-центри тощо). Система має забезпечити:

- Зручний механізм онлайн-запису клієнтів;
- Ефективне управління клієнтською базою;
- Автоматизоване інформування про візити;

Система орієнтована як на внутрішніх користувачів (адміністратори), так і на зовнішніх (клієнти), забезпечуючи швидку взаємодію між ними через єдиний веб-інтерфейс.

З технічної точки зору, реалізація задачі передбачає використання сучасних веб-технологій (HTML/CSS, JavaScript, Python, PHP або інші серверні мови, фреймворки), систем керування базами даних (MySQL/PostgreSQL) та хмарних або локальних серверів для розміщення веб-сайту. Система має бути побудована з урахуванням принципів безпеки, масштабованості та зручності обслуговування.

З економічної точки зору, впровадження такого веб-сайту дозволяє:

- Знизити витрати на адміністрування завдяки автоматизації записів, нагадувань та обліку;
- Підвищити прибутковість за рахунок зменшення кількості пропущених візитів і зростання числа повторних клієнтів;
- Скоротити час обслуговування клієнтів за рахунок оптимізації процесів взаємодії;
- Підвищити конкурентоспроможність бізнесу шляхом впровадження сучасних цифрових сервісів.

Таким чином, задача має високу практичну цінність, сприяє розвитку малого та середнього бізнесу в сфері послуг, відповідає сучасним вимогам ринку та очікуванням користувачів.

Об'єкти, за управлінням яких розв'язується задача:

- Послуги
 - Типи послуг (перукарські, косметологічні, масаж, манікюр тощо).
 - Опис, ціна, тривалість, категорія.
- Клієнти
 - Реєстрація та авторизація користувачів.
 - Зберігання контактної інформації.
 - Історія записів.
- Записи/бронювання
 - Онлайн-запис на послугу.
 - Вибір майстра, дати, часу.
- Платежі
 - Інформація про оплату (готівка, картка)

Призначення й використання вихідної інформації

Вхідною інформацією для оброблення в комп'ютерній системі буде:

- логін користувача;
- пароль користувача;
- назва послуги;
- опис послуги;
- ціна послуги;
- каталог послуг.

На рисунку 2.4 наведено вхідну інформацію, яку отримує сервер.

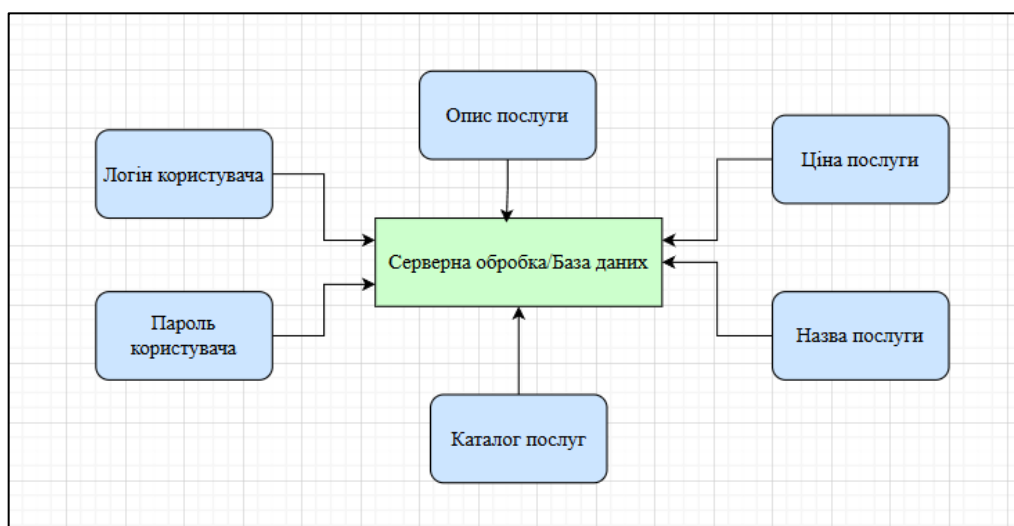


Рисунок 2.4 – Діаграма вхідної інформації

Джерело: розроблено автором

Призначення й використання вхідної інформації

Вихідна інформація, яка генерується в результаті роботи розробленої веб-системи, є ключовим елементом для підтримки управлінських, аналітичних та операційних рішень у сфері діяльності закладів краси та здоров'я. Вона створюється на основі введених користувачами даних (клієнтів, адміністраторів) та формалізованих бізнес-процесів системи.

Вихідною інформацією для оброблення в комп'ютерній системі буде:

- назва послуги;
- опис послуги;
- ціна послуги;
- каталог послуг;
- кількість послуг в кошику;
- загальна ціна послуг в кошику;
- статус замовлення.

На рисунку 2.5 наведено діаграму вихідної інформації.

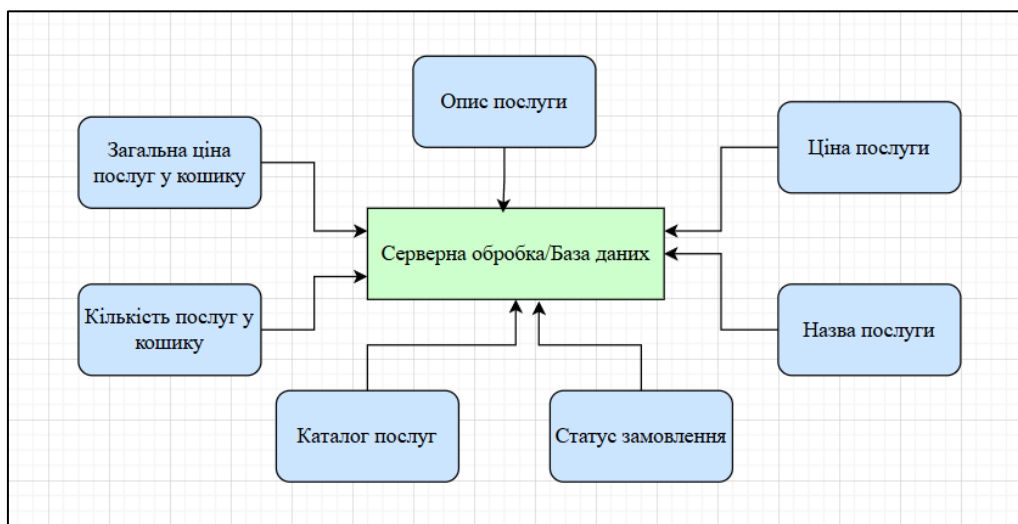


Рисунок 2.5 – Діаграма вихідної інформація

Джерело: розроблено автором

У рамках розробки веб-сайту для управління послугами в сфері краси та здоров'я було обрано SQLite як засіб зберігання структурованих даних. SQLite – це легка вбудована реляційна база даних, яка не потребує окремого серверного середовища та ідеально підходить для веб-додатків середнього навантаження.

Модель бази даних показана на рисунку 2.6

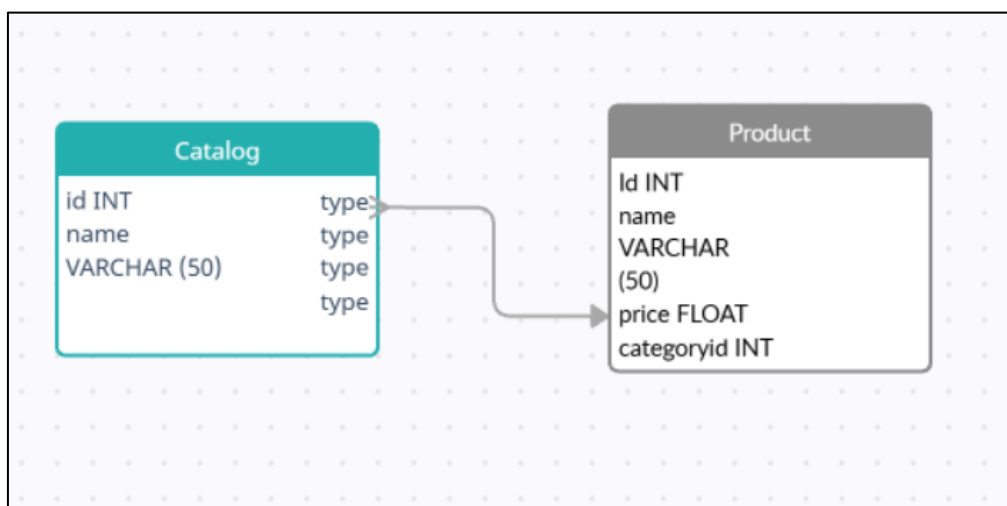


Рисунок 2.6 – Модель бази даних

Джерело: розроблено автором

2.3 Моделювання інформаційної системи

Інформаційна підсистема є ключовим елементом автоматизованої системи управління послугами в сфері краси та здоров'я. Її основною функцією є забезпечення збору, зберігання, обробки та відображення даних, необхідних для функціонування веб-додатку.

Моделювання інформаційної підсистеми дозволяє на етапі проєктування систематизувати логіку обробки даних, визначити основні об'єкти, зв'язки між ними, а також структуру взаємодії користувачів із системою.

Архітектуру комп'ютерної системи та взаємодію модулів представлено на рисунку 2.7.

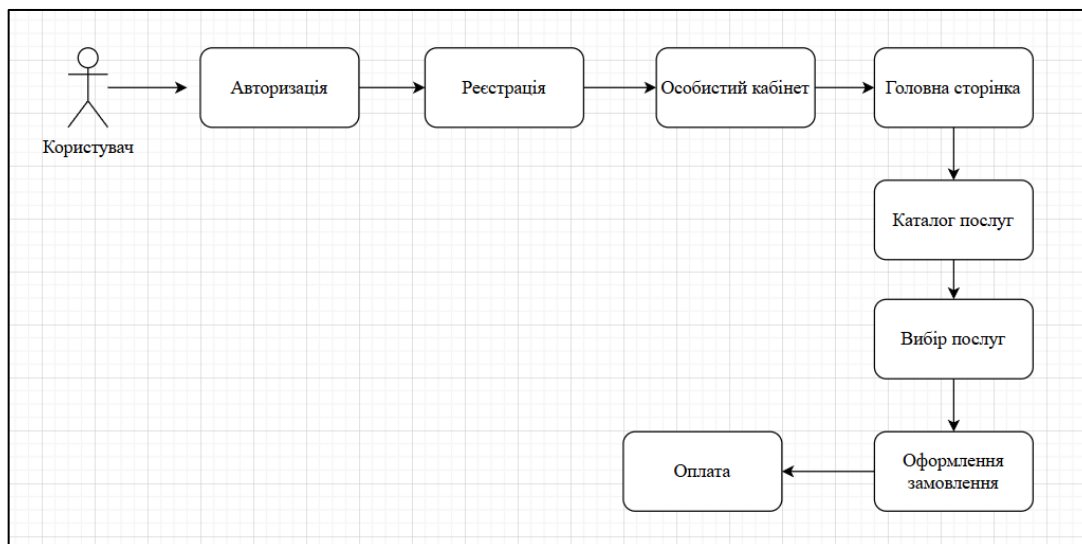


Рисунок 2.7 – Діаграма компонентів системи

Джерело: розроблено автором

Діаграма прецедентів ілюструє взаємодію основних користувачів веб-сайту – користувача та адміністратора (адміна) – з функціональністю системи. Вона відображає перелік основних дій (прецедентів), які можуть виконувати ці актори, та зв'язки між ними.

Опис діаграми прецедентів:

- Користувач – звичайний відвідувач або клієнт платформи, який може переглядати та замовляти послуги.

- Адмін – адміністратор системи, відповідальний за управління каталогом послуг, категоріями та користувачами.

Прецеденти для користувача:

- Вхід – автентифікація користувача в системі.
- Вибір оплати – вибір методу оплати після замовлення послуги.
- Перегляд послуг – відображення переліку доступних послуг.
- Перегляд каталогу – доступ до категоризованого списку послуг.
- Замовити послугу – створення замовлення на обрану послугу.

Прецеденти для адміністратора:

- Додати послугу – створення нової послуги в системі.
- Додати категорію – створення нової категорії послуг для структуризації.
- Перегляд замовлення – можливість бачити та контролювати створені користувачами записи.
- Перегляд користувачів – доступ до списку зареєстрованих користувачів, з можливістю їх модерації.

Зв'язки:

- Обидва актори мають свої окремі прецеденти, які не перетинаються, за винятком функцій перегляду послуг і каталогу – ці прецеденти використовуються як частини інших (наприклад, "Додати послугу" пов'язане з "Переглядом послуг", як логічне продовження).
- Лінії зв'язку між прецедентами вказують на їхню залежність, зокрема за допомогою умовного позначення label (яке в UML слід уточнити як «include» або «extend» для коректності).

Ця діаграма прецедентів є основою для проєктування користувацького інтерфейсу та реалізації логіки роботи веб-додатку. Вона демонструє базові функції, які мають бути реалізовані для забезпечення ефективної взаємодії клієнта з послугами та адміністратора з системними даними.

На рисунку 2.8 наведена діаграма прецедентів.

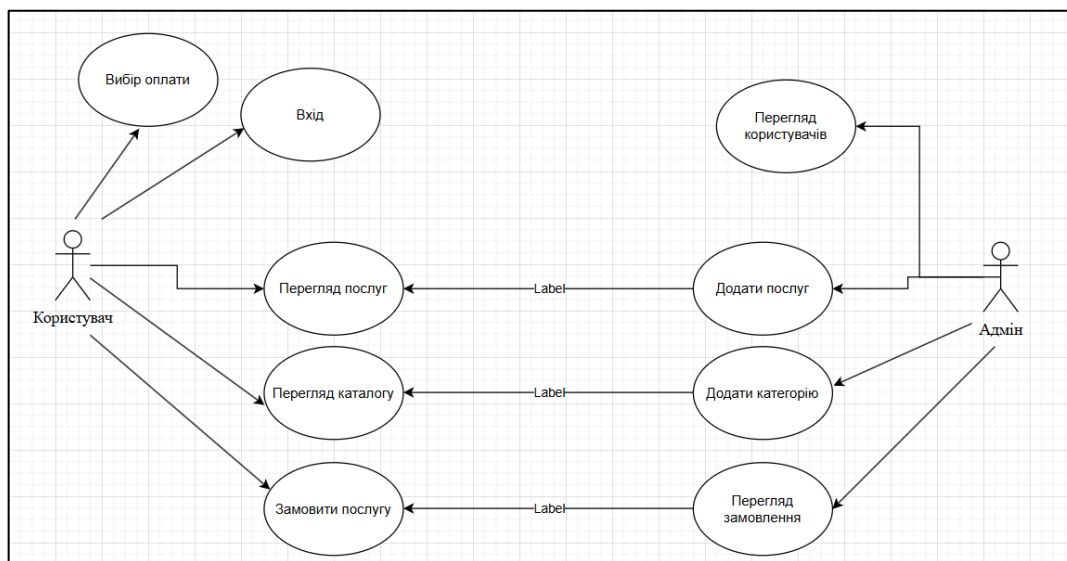


Рисунок 2.8 – Діаграма прецедентів

Джерело: розроблено автором

Алгоритм роботи веб-сайту для послуг краси та здоров'я подано у вигляді UML – діаграми представленої на рисунку 2.9.

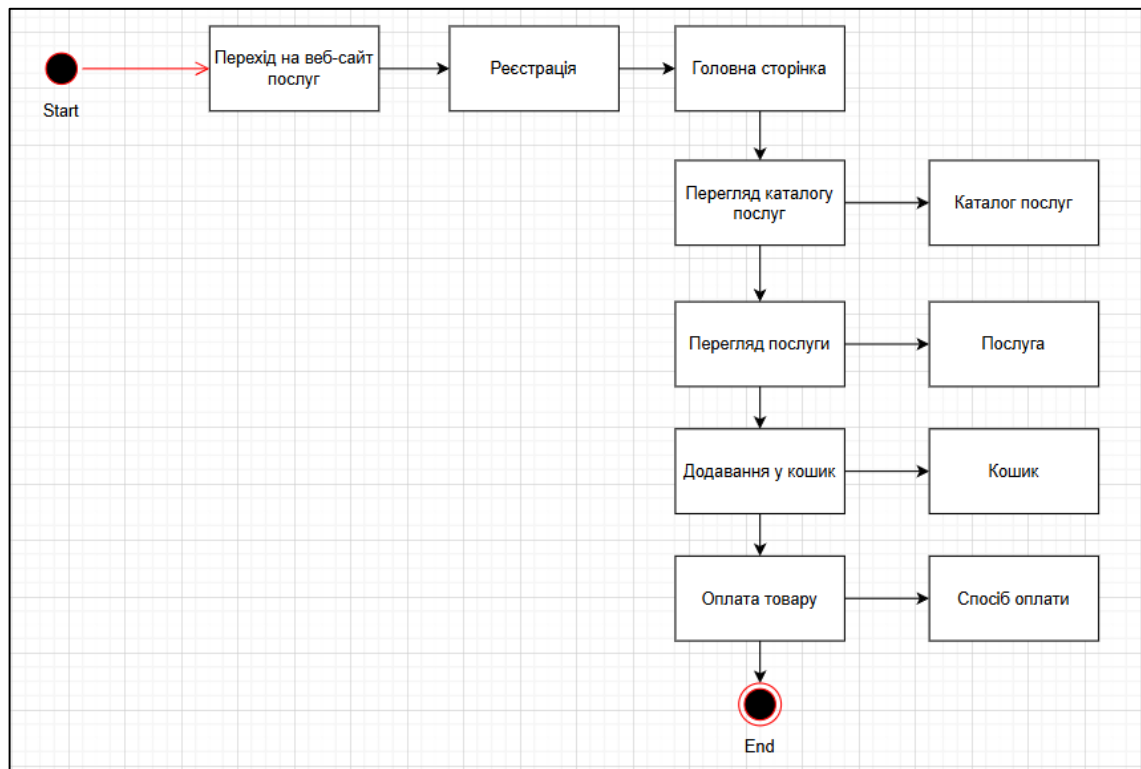


Рисунок 2.9 – Діаграма алгоритму роботи системи веб-сайту

Джерело: розроблено автором

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

Інформаційне забезпечення веб-сайту для управління послугами у сфері краси та здоров'я охоплює комплекс процесів, пов'язаних зі збором, зберіганням, обробкою та передачею інформації, яка забезпечує ефективне функціонування сервісу. Основними компонентами інформаційного забезпечення є база даних, система управління даними, а також механізми захисту, валідації та цілісності інформації.

У проєкті використовується база даних SQLite, що є легким і вбудованим реляційним сховищем даних. Вона зберігає ключову інформацію про користувачів, послуги, категорії, графік роботи, замовлення та інші об'єкти, що використовуються в системі.

Серверна логіка реалізована на основі фреймворку Django, який забезпечує обробку даних, управління маршрутами, взаємодію з базою даних через ORM, а також автентифікацію і авторизацію користувачів. Django також відповідає за генерацію HTML-сторінок через шаблонізатор DTL (Django Template Language).

Клієнтська частина побудована з використанням HTML, CSS, JavaScript, а також фреймворку Bootstrap, що забезпечує адаптивність і зручність користувацького інтерфейсу. Для інтерактивної взаємодії та динамічного оновлення даних застосовується jQuery, що спрощує обробку подій та реалізацію AJAX-запитів.

Збір первинної інформації в системі відбувається за допомогою інтерактивних веб-форм. Користувачі можуть реєструватися, авторизуватись, замовляти послуги, залишати відгуки або повідомлення. Адміністратори, у свою

чергу, мають можливість додавати нові послуги, редагувати інформацію, керувати замовленнями та графіком.

Структура бази даних організована у вигляді реляційної моделі, що дозволяє ефективно зберігати та обробляти інформацію, забезпечуючи швидкий доступ до потрібних даних і зручне масштабування системи.

Таким чином, інформаційне забезпечення забезпечує надійну основу для функціонування веб-сайту, підтримує всі ключові процеси взаємодії між користувачами та адміністрацією платформи, а також гарантує збереження, актуальність і безпеку даних.

3.2 Технічне забезпечення

Інформаційна підсистема для веб-сайту управління послугами у сфері краси та здоров'я передбачає використання хостингового середовища або віддаленого сервера (наприклад, PythonAnywhere, Heroku або хмарні сервіси типу AWS), де буде розгорнуто ядро системи, базу даних (SQLite або іншу за потреби), а також усі необхідні файли веб-додатку. Доступ користувачів до функціоналу системи здійснюється через сучасні веб-браузери на ПК, смартфонах чи планшетах при наявності інтернет-з'єднання.

Адміністративна частина системи (АРМ адміністратора) працює через окремий розділ інтерфейсу, доступний після автентифікації, з можливістю віддаленого доступу. У разі потреби адміністратори можуть працювати із системою через захищене з'єднання з додатковим рівнем авторизації.

Архітектура інформаційної підсистеми веб-платформи включає наступні ключові компоненти:

- Інтерфейс користувача – засіб інтерактивної взаємодії з системою. Містить сторінки: головна, про салон, каталог, послуги, графік роботи, контактна інформація, форма замовлення, особистий кабінет, реєстрація/вхід, політика конфіденційності.

- База даних – зберігає інформацію про користувачів, адміністраторів, категорії послуг, самі послуги, замовлення, повідомлення зворотного зв'язку, графіки та інше.
- Модуль «Головна» – головна сторінка сайту, яка відображає загальну інформацію про платформу та швидкий доступ до ключових функцій (пошук послуг, замовлення, реєстрація).
- Модуль «Реєстрація та авторизація» – забезпечує створення облікового запису, вхід у систему та керування особистим кабінетом.
- Модуль «Послуга» – надає детальну інформацію про кожну послугу, включаючи опис, вартість, тривалість, фото, доступність за часом.
- Модуль «Каталог послуг» – дозволяє зручно переглядати послуги у каталозі (догляд за волоссям, обличчям, тілом, нігтями тощо).
- Модуль «Замовлення» – дає змогу користувачам здійснювати онлайн-бронювання послуг із вибором дати, часу.
- Модуль «Адміністрування» – дає адміністраторам змогу додавати, редагувати або видаляти послуги, переглядати запити клієнтів та іншими об'єктами системи.

Таким чином, інформаційна підсистема забезпечує стабільну та логічно організовану взаємодію всіх користувачів з функціоналом веб-додатку, дозволяючи ефективно керувати послугами у сфері краси та здоров'я.

3.3 Програмне забезпечення

Програмне забезпечення веб-сайту для управління послугами у сфері краси та здоров'я складається з комплексу технологічних компонентів, які забезпечують повноцінне функціонування, зручність розробки, підтримку і масштабування інформаційної системи.

Основою серверної частини є Django – популярний фреймворк для розробки веб-додатків на мові програмування Python. Django надає розробникам

набір інструментів для реалізації бізнес-логіки, управління даними, безпеки і формування інтерфейсів.

- **Бізнес-логіка:** Вся логіка обробки замовлень, управління користувачами, графіками майстрів, послугами і оплатою реалізована у вигляді модулів Django.
- **ORM (Object-Relational Mapping):** Django ORM забезпечує роботу з базою даних, дозволяючи оперувати об'єктами Python замість написання SQL-запитів. Це підвищує продуктивність розробки і робить код більш підтримуваним.
- **Маршрутизація (Routing):** Django обробляє HTTP-запити, спрямовуючи їх на відповідні функції чи класи, які генерують відповіді.
- **Обробка форм:** Використання вбудованих механізмів для валідації і обробки даних, введених користувачем.
- **Автентифікація та авторизація:** Стандартні системи Django для реєстрації, входу, відновлення пароля та управління правами доступу.
- **Шаблони:** Система шаблонів Django дозволяє формувати HTML-сторінки динамічно, підставляючи дані з сервера.

База даних використовується для збереження інформації про користувачів, послуги, замовлення, майстрів та інші об'єкти використовується реляційна база даних SQLite.

- **Локальний файл бази:** Усі дані зберігаються у єдиному файлі, що спрощує управління і розгортання.
- **Інтеграція з Django:** SQLite підтримується без додаткових налаштувань, що дозволяє швидко почати розробку і тестування.
- **Підтримка транзакцій і зовнішніх ключів:** Забезпечує цілісність даних і надійність операцій.
- **Простота та продуктивність:** Ідеально підходить для невеликих та середніх проєктів, де немає потреби у складній інфраструктурі.

Клієнтська частина

Інтерфейс користувача реалізовано за допомогою стандартних веб-технологій:

- HTML5: Відповідає за структуру веб-сторінок.
- CSS3: Забезпечує стилізацію та оформлення сторінок.
- Bootstrap: CSS-фреймворк, який дозволяє створювати адаптивний дизайн – інтерфейс автоматично підлаштовується під різні пристрої: ПК, планшети, смартфони. Bootstrap також надає готові UI-компоненти, такі як кнопки, форми, навігаційні панелі та модальні вікна, що прискорює розробку і покращує юзабіліті.
- JavaScript: Додає динамічність і взаємодію з користувачем.
- jQuery: Бібліотека, яка спрощує роботу з JavaScript, дозволяє легко писати анімації, обробляти події і виконувати AJAX-запити – асинхронні звернення до сервера без перезавантаження сторінки.

Обмін даними

- Формат JSON (JavaScript Object Notation) використовується для обміну інформацією між клієнтом і сервером, особливо при роботі з API. JSON є легким, зрозумілим і широко підтримуваним форматом, що ідеально підходить для передачі структурованих даних.

Організація проєкту

- Використовується стандартна структура проєкту Django, що дозволяє поділити код на окремі додатки (apps), шаблони, статичні файли (стили, скрипти, зображення) і конфігураційні файли. Це підвищує масштабованість, зручність підтримки та розширення системи.
- Можливість розгортання як локально на сервері розробника, так і у хмарних середовищах, таких як PythonAnywhere, Heroku або AWS, що дає гнучкість у виборі платформи залежно від потреб.

Компоненти програмного забезпечення веб-сайту створюють єдину екосистему, що забезпечує:

- Надійність і стабільність роботи серверної частини (Django + SQLite),
- Зручність і адаптивність інтерфейсу (HTML5 + CSS3 + Bootstrap),
- Динамічну взаємодію з користувачем (JavaScript + jQuery + AJAX),
- Гнучкість у розгортанні та масштабуванні проєкту.

Цей комплекс технологій забезпечує ефективне управління послугами у сфері краси та здоров'я, задовольняючи потреби як кінцевих користувачів, так і адміністраторів системи.

3.4 Результати реалізації інформаційної системи

Реалізовано веб-сайт для управління послугами в сфері краси та здоров'я.

На сайті є 10 сторінок. Сайт містить горизонтальне меню, яке має посилання на каталог послуг, їх бронювання, авторизацію, реєстрацію та 3 посилання на інформацію про компанію.

Всі сторінки сайту є адаптивними. Послуги, які відображені на екрані після завантаження сторінки каталогу можна додати в корзину. Якщо користувач введе неіснуючий URL, йому відобразиться помилка 404.

Результат роботи «Розробка веб-сайту для управління послугами в сфері краси та здоров'я» показано на рисунках 3.1–3.14.

Програмний код наведено у додатках.

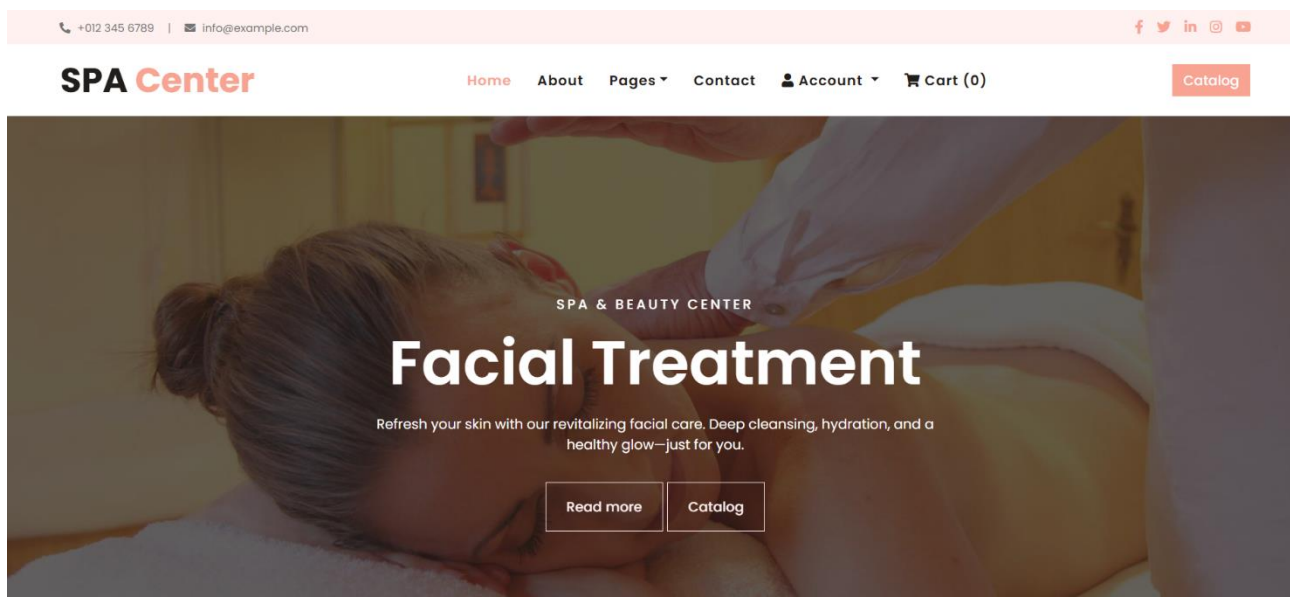


Рисунок 3.1 – Головна сторінка веб-сайту

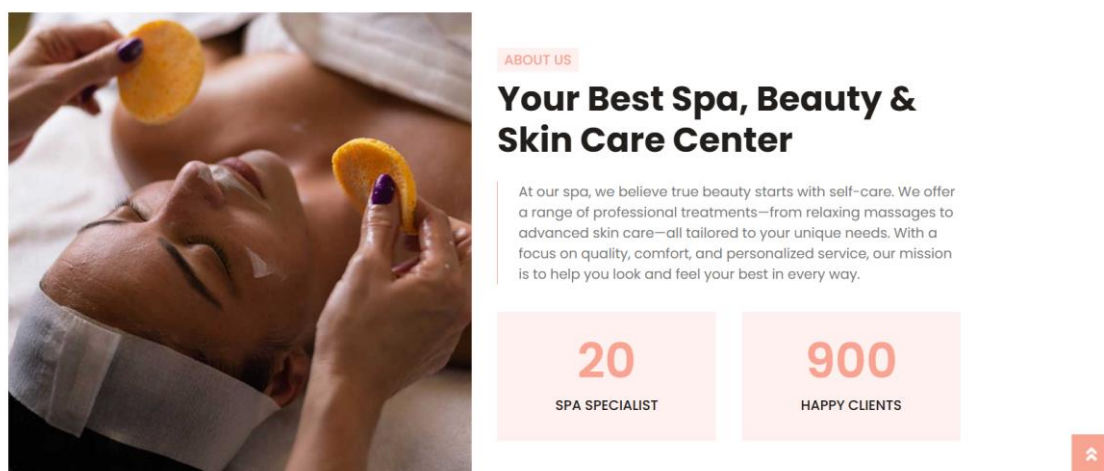


Рисунок 3.2 – Сторінка веб-сайту «Про Нас»

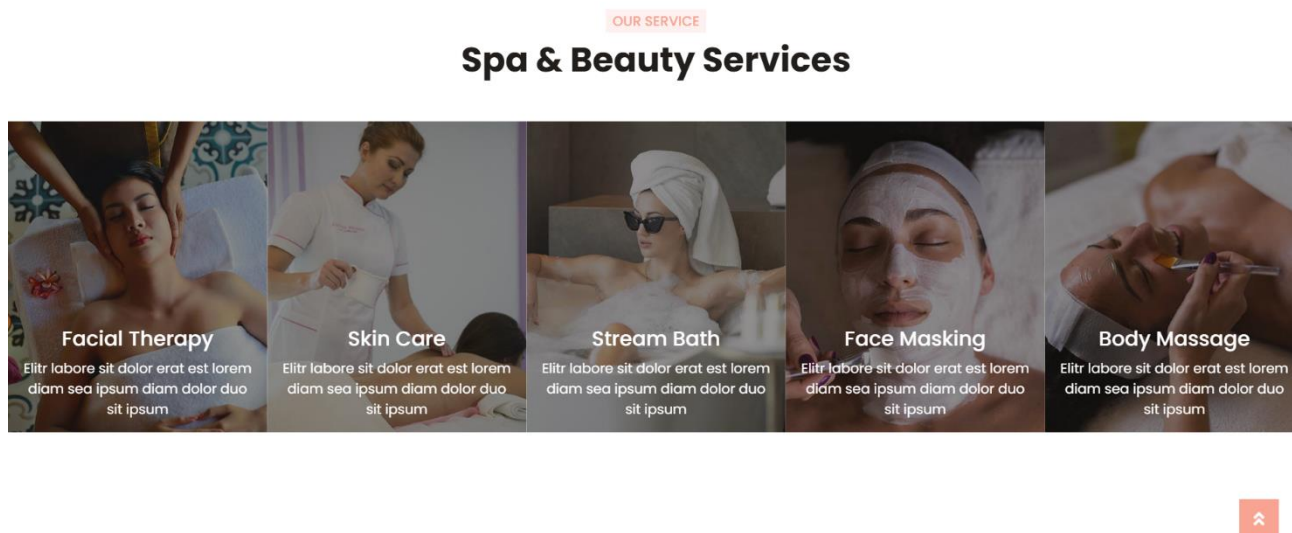


Рисунок 3.3 – Сторінка веб-сайту «Про наш сервіс»

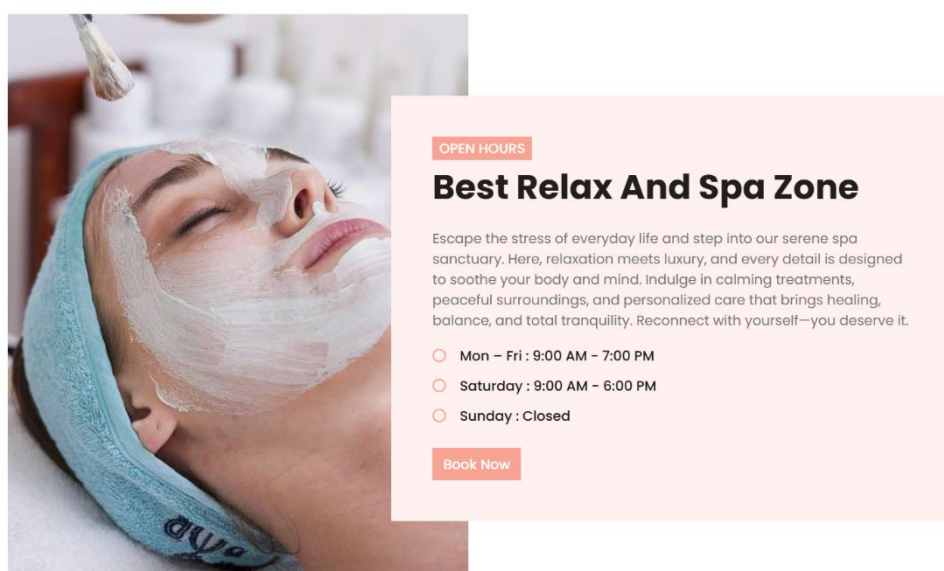


Рисунок 3.4 – Сторінка веб-сайту «Графік роботи»

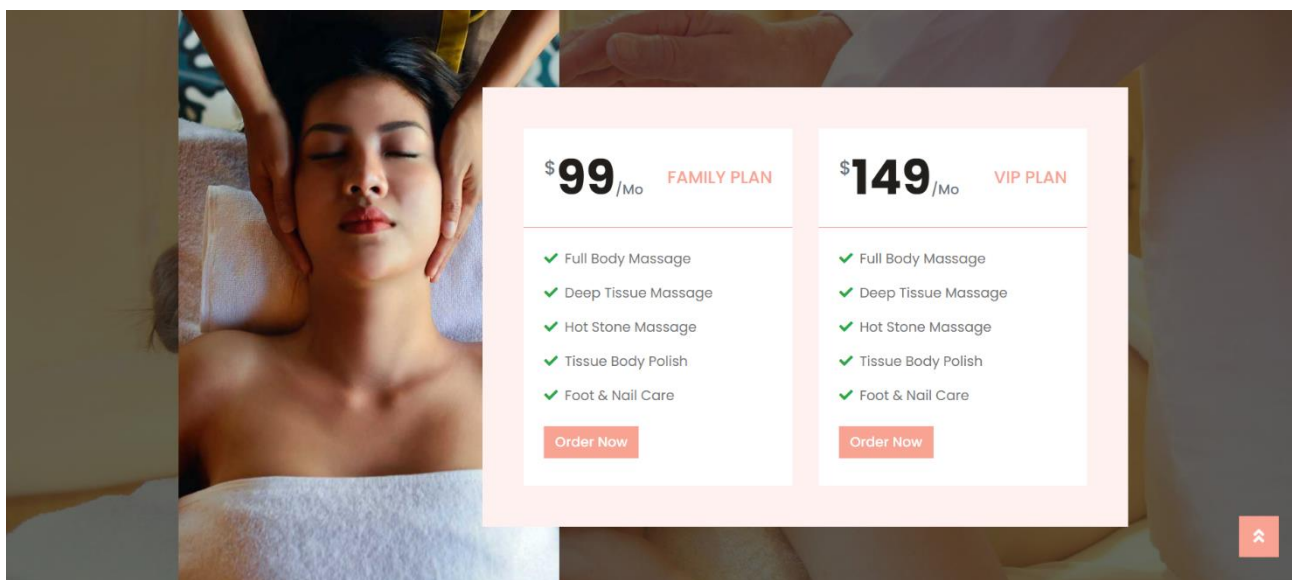


Рисунок 3.5 – Сторінка веб-сайту «Місячна підписка на надання послуг»

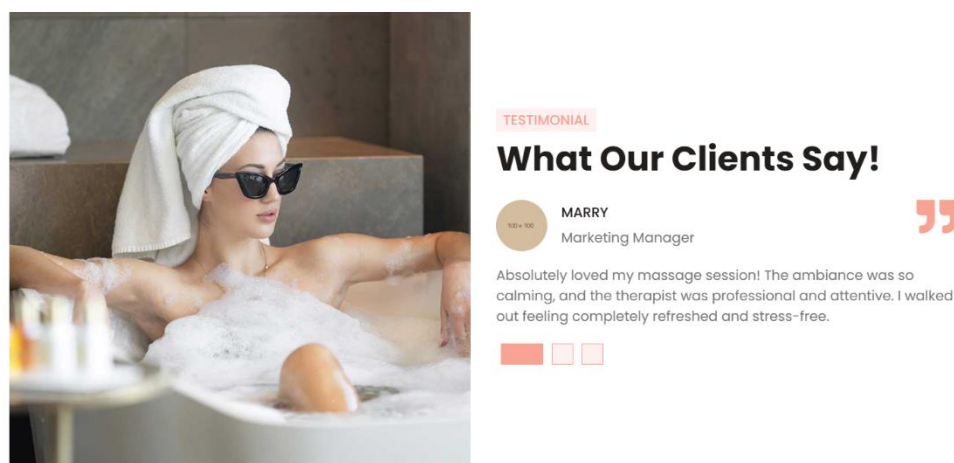


Рисунок 3.6 – Сторінка веб-сайту «Відгуки»

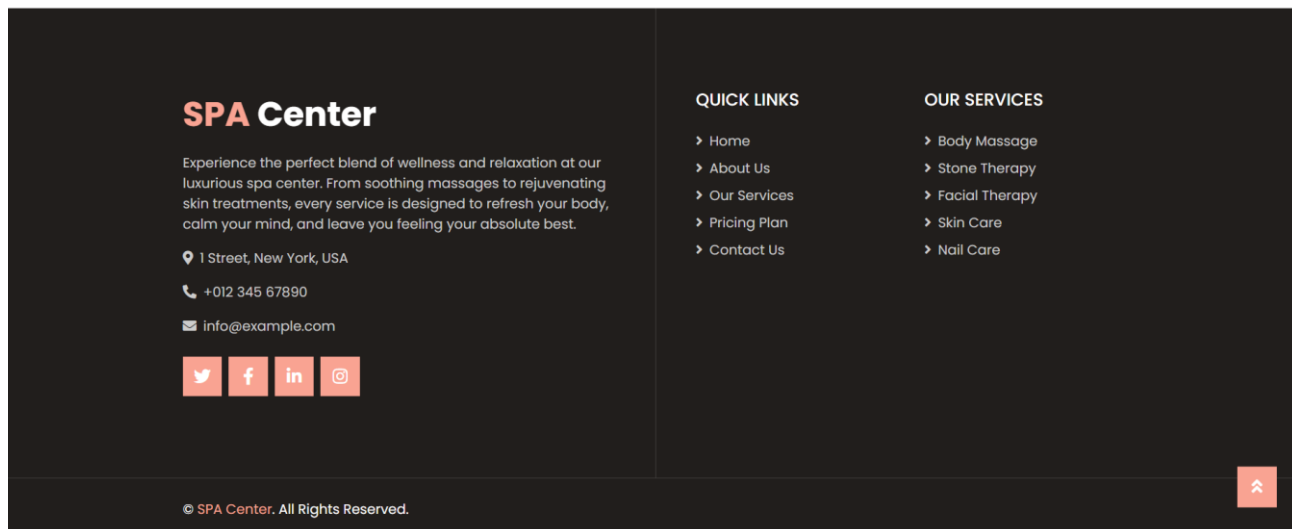


Рисунок 3.7 – Сторінка веб-сайту «Контакти»

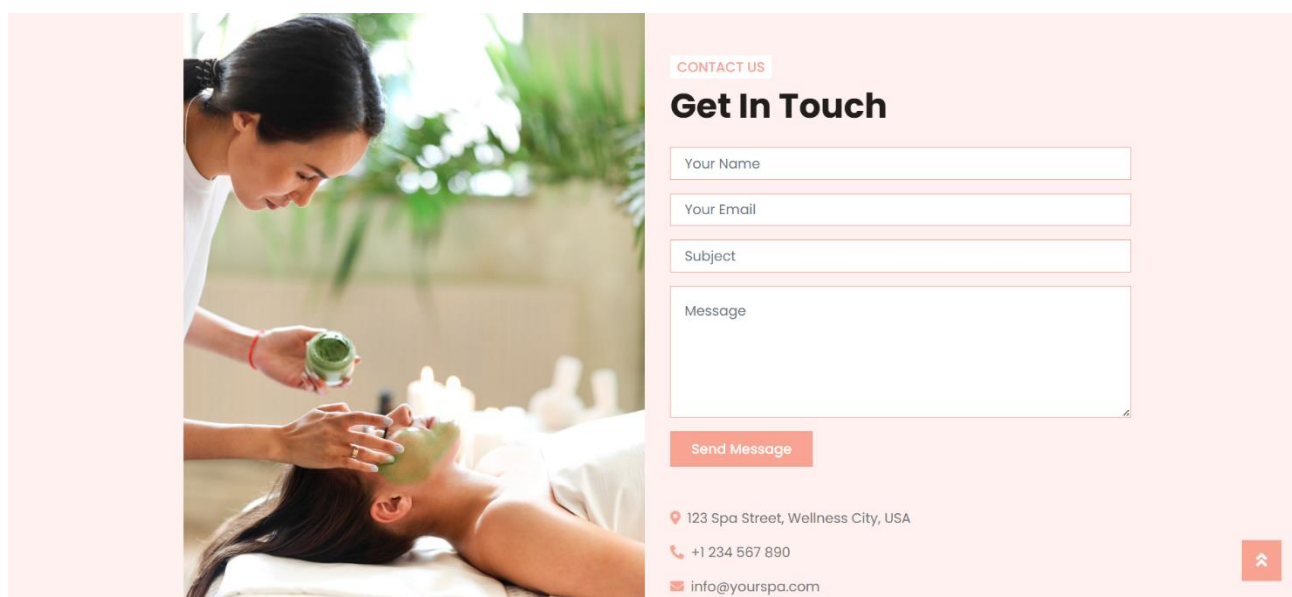


Рисунок 3.8 – Сторінка веб-сайту «Зв'яжіться з нами»

The screenshot shows the login page of the SPA Center website. The header includes the SPA Center logo, navigation links (Home, About, Pages, Contact, Account, Cart (0)), and a Catalog button. The main content area features a central login form with a 'LOGIN' tab, a 'Welcome Back' heading, and input fields for Username and Password. A 'Forgot Password?' link is located below the password field. A 'Login' button is at the bottom of the form, and a link to 'Register Now' is provided for users without an account.

+012 345 6789 | info@example.com

SPA Center

Home About Pages Contact Account Cart (0) Catalog

LOGIN

Welcome Back

Username

Enter username

Password

Enter password

[Forgot Password?](#)

Login

Don't have an account? [Register Now](#)

Рисунок 3.9 – Сторінка веб-сайту «Вхід»

The screenshot shows the registration page of the SPA Center website. The header is identical to the login page. The main content area features a central registration form with a 'REGISTER' tab, a 'Create Your Account' heading, and input fields for Username, First name, Last name, Email, Password, and Confirm Password. A 'Register' button is at the bottom of the form, and a link to 'Login Here' is provided for users who already have an account.

+012 345 6789 | info@example.com

SPA Center

Home About Pages Contact Account Cart (0) Catalog

REGISTER

Create Your Account

Username

Username

First name

First Name

Last name

Last Name

Email

Email Address

Password

Password

Confirm Password

Confirm Your Password

Register

Already have an account? [Login Here](#)

Рисунок 3.10 – Сторінка веб-сайту «Реєстрація»

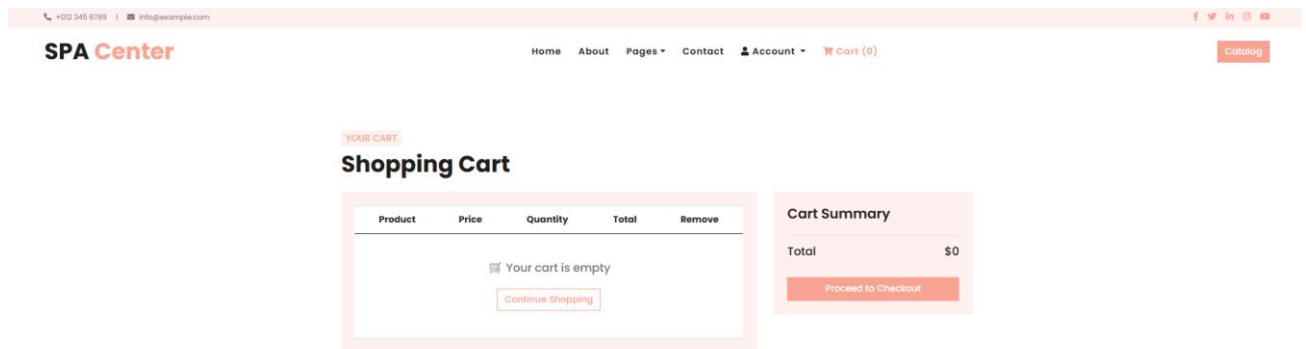


Рисунок 3.11 – Сторінка веб-сайту «Корзина»

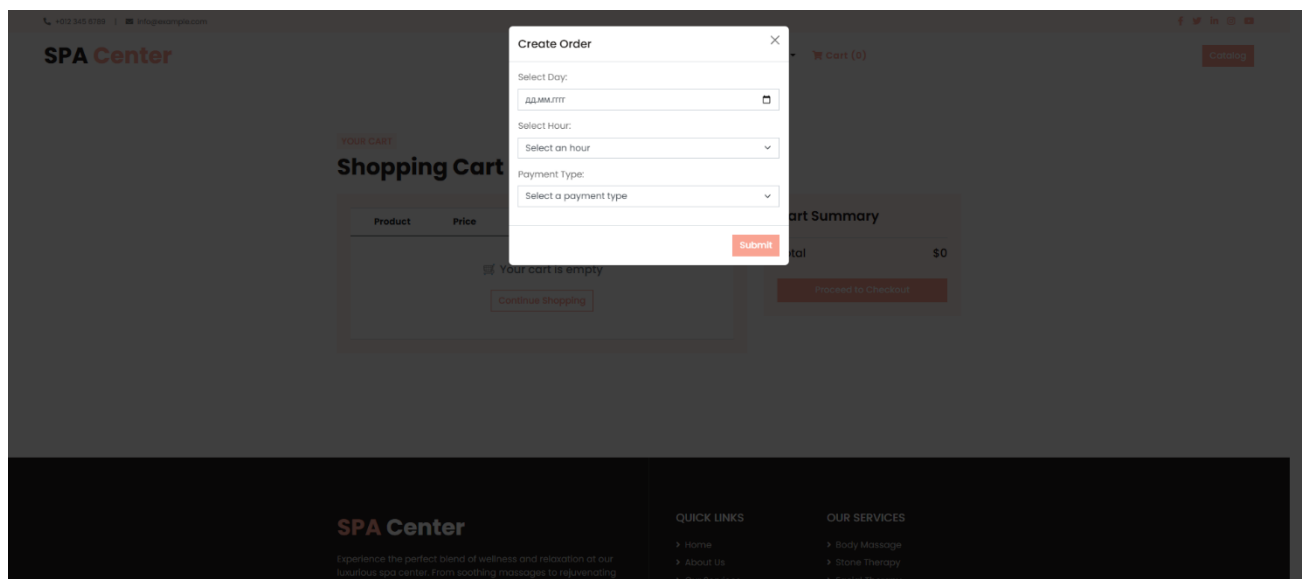


Рисунок 3.12 – Вспливаюче вікно веб-сайту «Оформлення послуг»

YOUR ACCOUNT

Profile Details

Basic Information of "admin" account

Full Name:
first admin

Email Address:
admin@email.com

Phone:

Member Since:
April 13, 2025, 2:11 p.m.

Change Password

Edit Profile

Username

admin

Required: 100 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

first

Last name

admin

Email

admin@email.com

Update Info

Your Orders

Рисунок 3.13 – Сторінка веб-сайту «Особистий кабінет»

your orders

Your Orders

Order #19 - 05 May, 2025	View
Order #18 - 18 Apr, 2025	View
Order #17 - 18 Apr, 2025	View
Order #16 - 18 Apr, 2025	View
Order #15 - 18 Apr, 2025	View
Order #14 - 18 Apr, 2025	View
Order #13 - 18 Apr, 2025	View
Order #12 - 18 Apr, 2025	View
Order #11 - 18 Apr, 2025	View
Order #10 - 18 Apr, 2025	View
Order #9 - 18 Apr, 2025	View
Order #8 - 18 Apr, 2025	View
Order #7 - 18 Apr, 2025	View
Order #6 - 18 Apr, 2025	View
Order #5 - 18 Apr, 2025	View
Order #4 - 18 Apr, 2025	View

Рисунок 3.13 – Сторінка веб-сайту «Ваші замовлення»

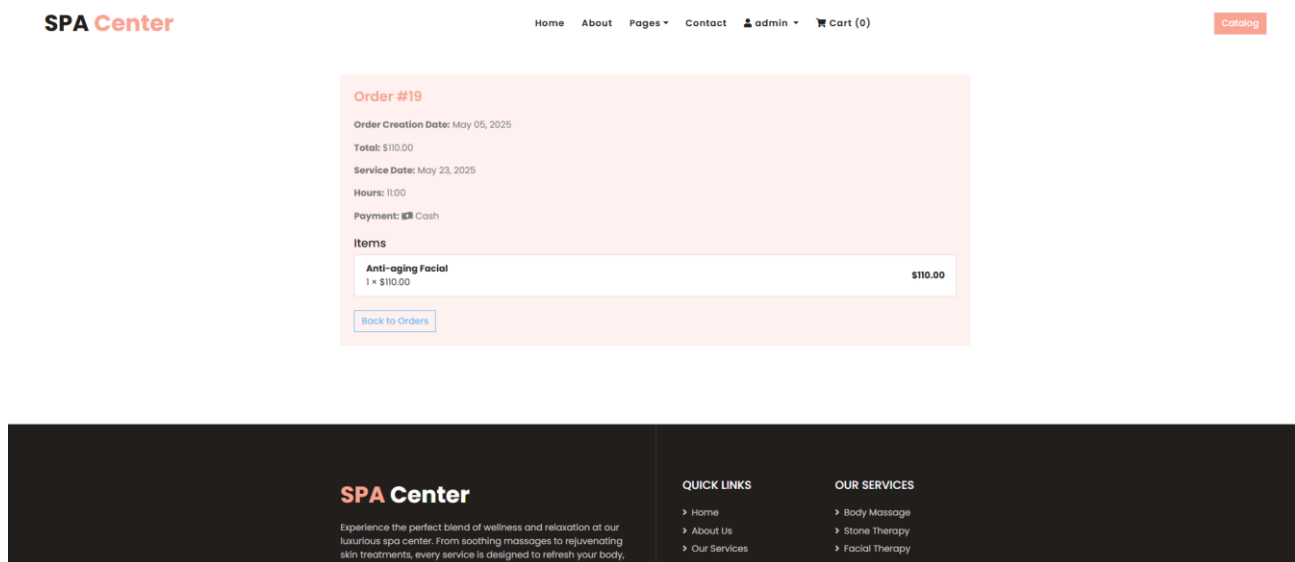


Рисунок 3.14 – Сторінка веб-сайту «Деталі замовлення»

ВИСНОВКИ

У результаті виконання кваліфікаційної бакалаврської роботи на тему «Розробка веб-сайту для управління послугами в сфері краси та здоров'я» було досягнуто поставленої мети – створено сучасний і функціональний веб-сайт, який забезпечує ефективне управління замовленнями, послугами та клієнтами, сприяючи підвищенню якості обслуговування у сфері краси та здоров'я.

Основні результати роботи полягають у наступному:

1. Проведено детальний аналіз предметної галузі сфери краси та здоров'я, що дозволило визначити ключові функціональні та нефункціональні вимоги до веб-сайту, врахувати сучасні тенденції у сфері онлайн-сервісів та виявити недоліки існуючих рішень, зокрема складність управління замовленнями і відсутність адаптивного інтерфейсу.
2. Розроблено архітектуру інформаційної системи, що включає опис функціональних модулів, UML-діаграми (прецедентів, діяльності, компонентів), а також реляційну базу даних SQLite, яка забезпечує надійне зберігання і обробку даних про користувачів, послуги, майстрів, записи на прийом та замовлення.
3. Реалізовано адаптивний веб-сайт із використанням технологічного стеку: Python, фреймворк Django для серверної частини, HTML5, CSS3, Bootstrap та JavaScript (jQuery) для клієнтської частини. Веб-сайт включає модулі реєстрації та авторизації користувачів, каталог послуг, онлайн-запис до майстрів, особистий кабінет клієнта, а також адміністративну панель для управління даними.
4. Веб-сайт забезпечує зручний і надійний спосіб отримання якісних послуг у сфері краси та здоров'я. Система створена для ефективного замовлення послуг і забезпечує комфортну взаємодію між клієнтами та спеціалістами. Користувачам пропонується широкий спектр послуг, а також можливість швидкого зв'язку з майстрами для уточнення деталей та погодження часу.

5. Проведено тестування веб-сайту, яке підтвердило його стабільність, коректність взаємодії з базою даних, а також адаптивність інтерфейсу на пристроях (ПК), що забезпечує зручність і доступність для користувачів.
6. Для подальшого розвитку проєкту рекомендовано розширення функціональності веб-сайту, покращення користувацького досвіду, впровадження додаткових інструментів маркетингу та аналітики для залучення більшої кількості клієнтів та підвищення конкурентоспроможності платформи, також додання оплати онлайн, що забезпечить більш зручний спосіб оформлення послуг.

В цілому, розробка веб-сайту для управління послугами у сфері краси та здоров'я є важливим кроком у створенні ефективної інформаційної системи, що дозволяє компаніям надавати якісні сервіси, підвищувати рівень клієнтської задоволеності та успішно конкурувати на ринку послуг.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іванченко В.М. Веб-технології: навчальний посібник. – К.: КНУ, 2020. – 214 с. (дата звернення: 12.04.2025)
2. Мельниченко С.О. Основи розробки веб-систем: монографія. – Харків: ХНУРЕ, 2021. – 180 с. (дата звернення: 14.04.2025)
3. Петренко Н.В. Адаптивний веб-дизайн у сфері сервісних послуг // Інформаційні технології і системи. – 2022. – №2(42). – С. 55–60. (дата звернення: 14.04.2025)
4. Козлова О.П. Впровадження CRM у закладах індустрії краси: досвід та перспективи // Бізнес-інформ. – 2023. – №3. – С. 73–78. (дата звернення: 15.04.2025)
5. Дмитренко Ю.В. Інформаційна безпека при роботі з персональними даними клієнтів у веб-середовищі // Управління розвитком складних систем. – 2022. – №51. – С. 47–52. (дата звернення: 15.04.2025)
6. Ковальчук Л.М. «Інформаційні системи в сервісній діяльності», 2021. (дата звернення: 20.04.2025)
7. Петренко О.С. «Моделі та методи автоматизації процесів у сфері послуг», 2020. (дата звернення: 20.04.2025)
8. Дьяков О. «Основи UX-дизайну», 2021 (дата звернення: 21.04.2025)
9. Fresha – безкоштовна платформа для салонів краси та SPA [Електронний ресурс]. – Режим доступу: <https://www.fresha.com/> (дата звернення: 23.03.2025)
10. Booksy – онлайн-сервіс бронювання для клієнтів та адміністраторів [Електронний ресурс]. – Режим доступу: <https://booksy.com/> (дата звернення: 23.03.2025)
11. Bookon – український веб-сервіс для автоматизації запису клієнтів [Електронний ресурс]. – Режим доступу: <https://bookon.binotel.ua/> (дата звернення: 23.03.2025)

ДОДАТКИ

Додаток А

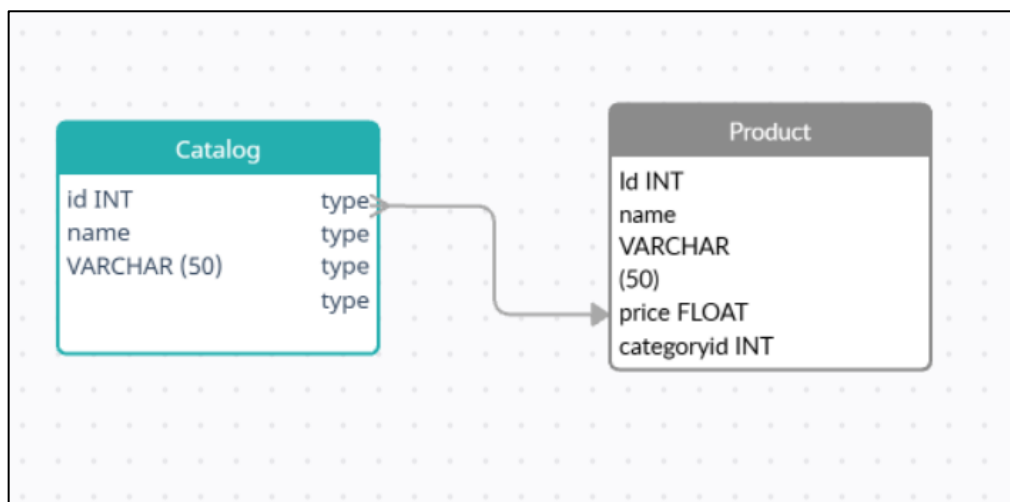


Рисунок А.1 – Схема бази даних

```
# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-hi@)j43#2vk8i(walaec-ekurariyh%kbw!)fjo6#2%s3!4*x!'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'store',
    'cart',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'ecom_spasalon.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
    },
]
```

```

'APP_DIRS': True,
'OPTIONS': {
    'context_processors': [
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
        'cart.context_processors.cart',
    ],
},
},
]

WSGI_APPLICATION = 'ecom_spa_salon.wsgi.application'

# Database
# https://docs.djangoproject.com/en/5.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/5.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/5.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

```

```

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.2/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIRS = [
    BASE_DIR / 'static',
]

MEDIA_URL = 'media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

# Default primary key field type
# https://docs.djangoproject.com/en/5.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

```

from django.contrib import admin
from django.urls import path, include

from .settings import MEDIA_URL, MEDIA_ROOT
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('store.urls')),
    path('cart/', include('cart.urls')),
] + static(MEDIA_URL, document_root=MEDIA_ROOT)

```

```

from django.contrib import admin

from .models import Category, Customer, Order, Product

admin.site.register(Category)
admin.site.register(Customer)
admin.site.register(Order)
admin.site.register(Product)

```

```

from django.apps import AppConfig

```

```
class StoreConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'store'
```

```
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm, UserChangeForm, SetPasswordForm
from django import forms
```

```
class ChangePasswordForm(SetPasswordForm):
    class Meta:
        model = User
        fields = ('new_password1', 'new_password2')

    def __init__(self, *args, **kwargs):
        super(ChangePasswordForm, self).__init__(*args, **kwargs)
        self.label_suffix = ""

        self.fields['new_password1'].widget.attrs['class'] = 'form-control'
        self.fields['new_password1'].widget.attrs['placeholder'] = 'New Password'
        self.fields['new_password1'].label = 'New Password'
        self.fields['new_password1'].help_text = '<ul class="form-text text-muted small"><li>Your
password can\'t be too similar to your other personal information.</li><li>Your password must contain
at least 8 characters.</li><li>Your password can\'t be a commonly used password.</li><li>Your
password can\'t be entirely numeric.</li></ul>'

        self.fields['new_password2'].widget.attrs['class'] = 'form-control'
        self.fields['new_password2'].widget.attrs['placeholder'] = 'Confirm Your Password'
        self.fields['new_password2'].label = 'Confirm New Password'
        self.fields['new_password2'].help_text = '<span class="form-text text-muted"><small>Enter the
same password as before, for verification.</small></span>'
```

```
class UpdateUserForm(UserChangeForm):
    email = forms.EmailField(label="Email", required=False,
widget=forms.TextInput(attrs={'class':'form-control', 'placeholder':'Email Address'}))
    first_name = forms.CharField(label="First name", required=False, max_length=100,
widget=forms.TextInput(attrs={'class':'form-control', 'placeholder':'First Name'}))
    last_name = forms.CharField(label="Last name", required=False, max_length=100,
widget=forms.TextInput(attrs={'class':'form-control', 'placeholder':'Last Name'}))
    password = None

    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'email')

    def __init__(self, *args, **kwargs):
        super(UpdateUserForm, self).__init__(*args, **kwargs)
        self.label_suffix = ""

        self.fields['username'].widget.attrs['class'] = 'form-control'
        self.fields['username'].widget.attrs['placeholder'] = 'Username'
```

```

        self.fields['username'].label = 'Username'
        self.fields['username'].help_text = '<span class="form-text text-muted"><small>Required. 150
characters or fewer. Letters, digits and @/./+/-/_ only.</small></span>'

class SignUpForm(UserCreationForm):
    email = forms.EmailField(label="Email", widget=forms.TextInput(attrs={'class':'form-control',
'placeholder':'Email Address'}))
    first_name = forms.CharField(label="First name", max_length=100,
widget=forms.TextInput(attrs={'class':'form-control', 'placeholder':'First Name'}))
    last_name = forms.CharField(label="Last name", max_length=100,
widget=forms.TextInput(attrs={'class':'form-control', 'placeholder':'Last Name'}))

    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'email', 'password1', 'password2')

    def __init__(self, *args, **kwargs):
        super(SignUpForm, self).__init__(*args, **kwargs)
        self.label_suffix = "

        self.fields['username'].widget.attrs['class'] = 'form-control'
        self.fields['username'].widget.attrs['placeholder'] = 'Username'
        self.fields['username'].label = 'Username'
        self.fields['username'].help_text = '<span class="form-text text-muted"><small>Required. 150
characters or fewer. Letters, digits and @/./+/-/_ only.</small></span>'

        self.fields['password1'].widget.attrs['class'] = 'form-control'
        self.fields['password1'].widget.attrs['placeholder'] = 'Password'
        self.fields['password1'].label = 'Password'
        self.fields['password1'].help_text = '<ul class="form-text text-muted small"><li>Your password
can\'t be too similar to your other personal information.</li><li>Your password must contain at least 8
characters.</li><li>Your password can\'t be a commonly used password.</li><li>Your password can\'t
be entirely numeric.</li></ul>'

        self.fields['password2'].widget.attrs['class'] = 'form-control'
        self.fields['password2'].widget.attrs['placeholder'] = 'Confirm Your Password'
        self.fields['password2'].label = 'password2'
        self.fields['password2'].label = 'Confirm Password'
        self.fields['password2'].help_text = '<span class="form-text text-muted"><small>Enter the same
password as before, for verification.</small></span>'

```

```

import datetime

from django.db import models
from django.contrib.auth.models import User
from django.db.models.signals import post_save

class CustomerProfile(models.Model):
    pass

```

```
# Products categories
class Category(models.Model):
    name = models.CharField(max_length=60)

    def __str__(self):
        return self.name

    class Meta:
        verbose_name_plural = 'Categories'

class Customer(models.Model):
    first_name = models.CharField(max_length=50)
    last_name = models.CharField(max_length=50)
    email = models.EmailField()
    phone = models.CharField(max_length=20)
    password = models.CharField(max_length=50)

    def __str__(self):
        return f'{self.first_name} - {self.email}'

class Product(models.Model):
    name = models.CharField(max_length=100)
    price = models.DecimalField(default=0, decimal_places=2, max_digits=6)
    category = models.ForeignKey(Category, on_delete=models.CASCADE, default=1)
    description = models.CharField(max_length=250, default="", blank=True, null=True)
    image = models.ImageField(upload_to='uploads/product/')

    def __str__(self):
        return self.name

class Order(models.Model):
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=1)
    phone = models.CharField(max_length=20, default="")
    date = models.DateField(default=datetime.datetime.today)
    status = models.BooleanField(default=False)

    def __str__(self):
        return f'{self.product} ({self.quantity})'
```

```
from django.test import TestCase
```

```
# Create your tests here.
```

```
from django.urls import path
```

```
from . import views
```

```
urlpatterns = [
    path("", views.index, name='index'),
    path('about/', views.about, name='about'),
    path('open_hours/', views.open_hours, name='open_hours'),
    path('contact/', views.contact, name='contact'),

    path('catalog/', views.catalog, name='catalog'),
    path('user/', views.user_info, name='user_info'),
    path('user/update_password/', views.update_password, name='update_password'),
    path('orders/<int:order_id>', views.order_detail, name='order_detail'),

    path('login/', views.login_page, name='login'),
    path('logout/', views.logout_page, name='logout'),
    path('register/', views.register_page, name='register'),
    path('register/password_reset/', views.password_reset, name='password_reset'),
]
```

```
from django.contrib.auth.decorators import login_required
from django.contrib.auth.models import User
from django.shortcuts import get_object_or_404, render, redirect
from django.contrib.auth import authenticate, login, logout
from django.contrib import messages

from cart.models import Order

from .models import Product, Category
from django.db.models import Q
from .forms import SignUpForm, UpdateUserForm, ChangePasswordForm

def index(request):
    return render(request, 'index.html', {})

@login_required(login_url='login')
def user_info(request):
    user = User.objects.get(id=request.user.id)
    orders = Order.objects.filter(user=request.user).order_by('-created_at').order_by('-id')
    form = UpdateUserForm(request.POST or None, instance=user)

    if request.method == 'POST':
        if form.is_valid():
            form.save()

            login(request, user)
            messages.success(request, 'User data has been updated.')
        else:
            messages.warning(request, 'Something went wrong.')
```

```

        return redirect('user_info')

    context = {
        'form': form,
        'orders': orders,
    }
    return render(request, 'user_account.html', context)

@login_required(login_url='login')
def update_password(request):
    user = User.objects.get(id=request.user.id)
    form = ChangePasswordForm(user, request.POST or None)

    if request.method == 'POST':
        if form.is_valid():
            form.save()
            messages.success(request, 'Password was changed.')

            login(request, user)
            return redirect('user_info')
        else:
            messages.warning(request, 'Something went wrong.')
            return render(request, 'update_password.html', {'form': form})

    else:
        return render(request, 'update_password.html', {'form': form})

def about(request):
    return render(request, 'about.html', {})

def open_hours(request):
    return render(request, 'open_hours.html', {})

def contact(request):
    return render(request, 'contact.html', {})

def login_page(request):
    if request.method == "POST":
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user:
            login(request, user)
            messages.success(request, "You have been logged in.")
            return redirect('index')
        else:
            messages.warning(request, "Something went wrong.")

```

```

        return redirect('login')
    else:
        return render(request, 'login.html', {})

def logout_page(request):
    logout(request)
    messages.success(request, 'You have been logged out.')
    return redirect('index')

def password_reset(request):
    return render(request, 'password_reset.html', {})

def register_page(request):
    form = SignUpForm()
    if request.method == 'POST':
        form = SignUpForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data['username']
            password = form.cleaned_data['password1']
            user = authenticate(request, username=username, password=password)
            login(request, user)
            messages.success(request, 'User have been created.')
            return redirect('index')
        else:
            messages.warning(request, "Something went wrong.")
            return render(request, 'register.html', {'form': form})
    else:
        return render(request, 'register.html', {'form': form})

def catalog(request):
    categories = Category.objects.all()
    products = Product.objects.all()

    category = request.GET.get('category')
    price_filter = request.GET.get('price')
    query = request.GET.get('q')

    if category:
        products = products.filter(category__name__icontains=category)

    if price_filter:
        if price_filter == 'under-50':
            products = products.filter(price__lt=50)
        elif price_filter == '50-100':
            products = products.filter(price__gte=50, price__lte=100)
        elif price_filter == '100-150':
            products = products.filter(price__gte=100, price__lte=150)

```

```
elif price_filter == 'over-150':
    products = products.filter(price__gt=150)

if query:
    products = products.filter(
        Q(name__icontains=query) | Q(description__icontains=query)
    )

context = {
    'products': products,
    'categories': categories
}
return render(request, 'catalog.html', context=context)

@login_required
def order_detail(request, order_id):
    order = get_object_or_404(Order, id=order_id, user=request.user)
    return render(request, 'order_detail.html', {'order': order})
```

StrikePlagiarism.com

СЕРТИФІКАТ

Дата звіту

6/8/2025

Дата редагування

Звіт не був оцінений

Звіт подібності

метадані

Назва організації

Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок

Розробка веб-сайту для управління послугами в сфері краси і здоров'я

Автор

Науковий керівник / Експерт

Маліневська Єва ЕдуардівнаВасильська Л.В.

підрозділ

кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

4.84%

4.84%

КП 1

25

Довжина фрази для коефіцієнта подібності 2

1.19%

1.19%

КП 2

6307

Кількість слів

1.71%

1.71%

КЦ

49611

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		2
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		20

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копії тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копії тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ПЕРАМЕНТІВ)
1	https://ir.kneu.edu.ua/server/api/core/bitstreams/1b35cb3a-b293-4677-9674-83983838c914/content	43 0.68 %
2	https://ir.kneu.edu.ua/server/api/core/bitstreams/add8ce43-9c83-4412-bccd-41d87f874e83/content	32 0.51 %
3	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3922cf3566a2/download	21 0.33 %
4	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3922cf3566a2/download	20 0.32 %
5	https://revolution.alibest.ru/programming/01177452_0.html	18 0.29 %

6	http://ir.nusta.edu.ua/bitstream/123456789/7341/1/122_%D0%97%D0%80%D0%B3%D1%96%D1%80%D0%BD%D0%B8%D0%B9_2020.pdf	14 0.22 %
7	https://ir.kneu.edu.ua/bitstreams/57721939-fba5-4699-afde-625c26f42ccb/download	14 0.22 %
8	Програме забезпечення для систем управління барами 3/15/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	14 0.22 %
9	https://ela.kpi.ua/server/api/core/bitstreams/f53c4163-d46c-499a-84b0-5a1bab8a7b7c/content	12 0.19 %
10	https://ir.kneu.edu.ua/server/api/core/bitstreams/add9ce43-9c83-4412-bccd-41d87f874e83/content	11 0.17 %
з бази даних RefBooks (0.00 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИФІКАЦІЙНИХ СЛІВ (ФРАГМЕНТІВ)
з домашньої бази даних (0.19 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИФІКАЦІЙНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розробка інтегрованої інформаційної системи нарахування заробітної плати на картковій рахунок 6/19/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	7 (1) 0.11 %
2	КОМП'ЮТЕРИЗАЦІЯ ПРОЦЕСІВ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ З УПРАВЛІННЯ ЯКІСТЮ ПРОГРАМНОГО ПРОДУКТУ 6/18/2019 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	5 (1) 0.08 %
з програми обміну базами даних (0.54 %)		
ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИФІКАЦІЙНИХ СЛІВ (ФРАГМЕНТІВ)
1	Система управління медичною логістикою 3/16/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	20 (2) 0.32 %
2	Програме забезпечення для систем управління барами 3/15/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	14 (1) 0.22 %
з Інтернету (4.11 %)		
ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИФІКАЦІЙНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ir.kneu.edu.ua/server/api/core/bitstreams/add9ce43-9c83-4412-bccd-41d87f874e83/content	58 (4) 0.92 %
2	https://ir.kneu.edu.ua/server/api/core/bitstreams/fb35cb3a-b293-4677-9674-83983838c914/content	49 (2) 0.78 %
3	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3923cf3565a2/download	46 (3) 0.73 %
4	https://ir.kneu.edu.ua/bitstreams/57721939-fba5-4699-afde-625c26f42ccb/download	21 (2) 0.33 %
5	https://revolution.albest.ru/engamming/01177452_0.html	18 (1) 0.29 %
6	https://ir.kneu.edu.ua/bitstreams/22acc485-0461-4d18-a9a4-f2bd8c10ff/download	17 (2) 0.27 %

7	http://ir.nusta.edu.ua/bitstream/123456789/7341/1/122_%D0%97%D0%B0%D0%B3%D1%96%D1%80%D0%BD%D0%BB%D0%B9_2020.pdf	14 (1) 0.22 %
8	https://ir.kneu.edu.ua/bitstreams/850a2498-073e-45fe-874b-28be792ab58d/download	13 (2) 0.21 %
9	https://ela.kpi.ua/server/api/core/bitstreams/f53c4163-d46c-498a-84b0-6a1bab8a7b7c/content	12 (1) 0.19 %
10	https://eir.zp.edu.ua/bitstreams/bce97f30-93dc-43de-ba53-a62787d383df/download	11 (1) 0.17 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗВІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------