

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»
на тему: «Web-орієнтована система підтримки діяльності салону краси»

Здобувачки групи ІТ.мз-21с Дегтярєва Анна Сергіївна
(шифр групи) (прізвище, ім'я, по батькові)

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

(підпис)

Анна ДЕГТЯРЬОВА
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник старший викладач кафедри ІТ, к.т.н., Ольга БОЙКО _____
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентів

Дегтярьова Анна Сергіївна

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи «Web-орієнтована система підтримки діяльності салона краси»

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом кваліфікаційної роботи «11» грудня 2023 р.

3 Вхідні дані до кваліфікаційної роботи дані про салон краси, дані про послуги салона краси, дані про користувачів

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їй належить розробити) аналіз предметної області, моделювання та проектування, розробка Web-орієнтованої системи підтримки діяльності салона краси

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації) тема, актуальність, постановка задачі, огляд програмних продуктів-аналогів, таблиця порівняння аналогів, функціональні вимоги до Web-орієнтованої системи, інструменти реалізації, структурно-функціональне моделювання, моделювання бази даних, моделювання варіантів використання, практична реалізація, демонстрація роботи, висновок

6. Консультанти випускної роботи із зазначенням розділів, що їм стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та постановка задачі дослідження		
2	Планування робіт		
3	Вибір методів дослідження		
4	Моделювання/проектування		
5	Практична реалізація результатів дослідження		
6	Написання розділів пояснювальної записки та надання керівнику на перевірку		
7	Підготовка заключної версії кваліфікаційної роботи, перевірка на плагіат		
8	Подача до ЕК комплекту документів до захисту		

Магістрант _____

Анна ДЕГТЯРЬОВА

Керівник роботи _____

к.т.н., Ольга БОЙКО

АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Web-орієнтована система підтримки діяльності салона краси».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 30 найменувань, додатків. Загальний обсяг роботи – 78 сторінок, у тому числі 62 сторінок основного тексту, 3 сторінки списку використаних джерел, 13 сторінок додатків.

Актуальність роботи полягає в розробці та впровадженню Web-орієнтованої системи для оптимізації управління салоном краси.

Мета роботи: створення Web-орієнтованої системи, базованого на фреймворку Laravel, що автоматизує процеси запису клієнтів, а також забезпечує зручний інтерфейс для аналізу та маркетингу.

Обсяг анотації: 1 сторінка.

Ключові слова: Web-орієнтована система, HTML, CSS, PHP, JavaScript, MySQL, PHPMyAdmin, Laravel.

ЗМІСТ

Вступ	6
1 Аналіз предметної області	8
1.1 Аналіз програмних продуктів-аналогів	8
1.2 Салон краси Backstage	10
1.3 Салон краси Wizard	11
1.4 Салон краси "Mira"	13
2 Постановка задачі та методи дослідження	16
2.1 Мета та задачі	16
2.2 Вибір засобів реалізації	17
3 Моделювання та проектування	22
3.1 Розробка структурно-функціональної моделі системи	22
3.2 Проектування моделі бази даних	24
3.3 Моделювання варіантів використання	30
4 Розробка web-орієнтованої системи підтримки	32
4.1 Розробка інтерфейсу (HTML, CSS)	32
4.2 Розробка бізнес-логіки (PHP, Laravel)	40
4.3 Інтеграція з базою даних (MySQL)	48
4.4 Результат розробки	52
4.5 Тестування web-орієнтованої системи	61
Висновки	62
Список використаних джерел	63
Додаток А	66
Додаток Б	73

ВСТУП

У сучасному світі, де високотехнологічні рішення швидко стають необхідністю в різних сферах життя, салони краси не залишаються осторонь впровадження новітніх інформаційних технологій для покращення своєї діяльності. У цьому контексті, актуальність створення та впровадження Web-орієнтованої системи підтримки для салонів краси виходить за рамки простої модернізації та стає важливим чинником розвитку цієї галузі.

На сьогоднішній день, салони краси є важливими елементами галузі краси та здоров'я, де люди звертаються для отримання послуг з догляду за собою та покращення свого зовнішнього вигляду. Проте, велика конкуренція та зростання вимог клієнтів вимагають від салонів ефективного управління, щоб забезпечити високу якість обслуговування та відповідати сучасним стандартам.

Сутність проблеми полягає в відсутності ефективних інструментів управління, які враховують усі аспекти роботи салону краси та сприяють його успішній діяльності. Задача полягає у розробці та впровадженні Web-орієнтованої системи підтримки, яка відповідає специфіці цієї галузі та вирішує конкретні проблеми, з якими стикаються власники та керівники салонів краси.

Мета даної магістерської дипломної роботи – розробити та впровадити ефективну Web-орієнтовану систему підтримки для салонів краси на основі фреймворку Laravel, яка забезпечить автоматизацію рутинних процесів та оптимізацію управління різними аспектами діяльності салону.

Об'єкт дослідження – підтримка діяльності салону краси.

Предмет дослідження – web-орієнтована система підтримки діяльності салону краси.

Основні задачі:

- проведення аналізу предметної області у сфері краси;

- дослідження методів, технологій реалізації задач для сфери послуг на прикладах салонів краси;
- проектування, моделювання web-орієнтованої системи;
- розробка та тестування web-орієнтованої системи підтримки діяльності салону краси;
- Оформлення документації.

Використання сучасних технологій та інноваційного підходу до створення системи підтримки салонів краси може позитивно вплинути на їхню конкурентоспроможність та забезпечити високий рівень задоволення клієнтів. Результати дослідження та розробки Web-орієнтованої системи вирішать практичні проблеми салонів краси та будуть важливим внеском у розвиток галузі.

Практичне застосування одержаних результатів полягає в можливості впровадження розробленої системи в реальних умовах роботи салонів краси. Це сприятиме підвищенню ефективності управління, збільшенню кількості задоволених клієнтів та покращенню фінансових показників салонів.

Таким чином, дана магістерська дипломна робота визначає важливість впровадження Web-орієнтованої системи підтримки для салонів краси та визначає напрямки для подальших досліджень у цій області.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз програмних продуктів-аналогів

Сучасні салони краси відіграють важливу роль у суспільстві, не лише надаючи послуги краси, але й створюючи простір для відпочинку, релаксації та самовираження. Попит на такі послуги зростає, але разом із цим збільшується і конкуренція, що ставить перед салонами виклики в управлінні та вдосконаленні сервісу.

Однією з ключових проблем для салонів краси є необхідність вдосконалення системи управління персоналом. Багато з них залишаються вірними традиційним методам, таким як ручне бронювання та облік, що може призводити до неупорядкованості та незручностей для клієнтів. Важливість впровадження сучасних технологій у сфері управління стає першочерговою, оскільки вони допомагають оптимізувати робочі процеси, покращуючи якість обслуговування та забезпечуючи більшу ефективність.

Ще однією важливою аспектом є розвиток інформаційних систем для салонів краси. Створення спеціалізованих Web-орієнтованих систем дозволяє салонам вдосконалити процес бронювання, обліку послуг та взаємодії з клієнтами. Такі системи можуть включати в себе онлайн-календарі для бронювання, зручний інтерфейс для ведення бази даних, а також аналітичні засоби для вивчення попиту та управління маркетинговими кампаніями.

Інтернет-сайти та профілі в соціальних мережах стають важливим інструментом для просування послуг, взаємодії з клієнтами та отримання зворотного зв'язку. Важливою стає інтеграція цих платформ із системами управління для максимальної ефективності.

Однак разом із ростом можливостей та технологій появляються й нові виклики. Зокрема, важливо забезпечити безпеку та конфіденційність клієнтських даних в онлайн-системах. Захист від несанкціонованого доступу

та ефективного керування даними стають необхідною частиною інфраструктури сучасного салону краси.

Головна ціль проекту – це розробити Web-орієнтовану систему, для перегляду послуг салону та зручного запису, побудувати довіру з клієнтами за допомогою зворотного зв'язку, збільшити попит на послуги.

Для більш коректного висновку розглянемо та порівняємо декілька вже існуючих веб сторінок для салонів краси. Для цього сформуємо перелік критеріїв для порівняння.

1. Привабливість дизайну:

- Використовуються естетичні та гармонійні елементи дизайну.
- Дизайн відповідає цільовій аудиторії сайту.

2. Зручність навігації:

- Навігація по сайту зрозуміла та проста.
- Користувач може швидко знайти потрібну інформацію.

3. Інтуїтивність використання:

- Користувач може зрозуміти, як використовувати сайт, без попереднього навчання.
- Кнопки, іконки, меню легко помітні та підсвідомо зрозумілі.

4. Адаптивність для різних пристроїв:

- Сайт добре виглядає та працює на різних пристроях, включаючи мобільні телефони та планшети.

5. Інформаційний вміст:

- Користувач бачить на веб-сайті актуальну інформацію.
- Відображається ціна на кожну існуючу послугу.
- Відвідувач може переглядати всі наявні акції та знижки.
- Сайт містить інформацію про команду та фахівців.

6. Запис на послуги:

- Користувач може здійснити онлайн-бронювання на сайті.
- Здійснення онлайн-замовлення зручне та інтуїтивно-зрозуміле.

7. Відгуки та рейтинги:

- На сайті міститься розділ з відгуками клієнтів для чіткого визначення рейтингу фахівців та послуг.

8. Інтеграція з Соціальними Мережами:

- Є можливість поділитися результатами послуг у соціальних мережах.
- Сайт містить посилання на свої сторінки в соціальних мережах.

9. Технічна Продуктивність – швидкість завантаження сторінок сайту, робота онлайн-сервісів без збоїв.

Під час порівняння важливо враховувати потреби цільової аудиторії та стратегічні цілі салону краси.

Тож розглянемо декілька салонів краси.

1.2. Салон краси Backstage

Backstage, сучасний салон краси, визначається не лише розмаїттям наданих послуг, але й старанним підходом до створення враження для клієнтів через свій веб-сайт, представлено на рисунку 1.1.

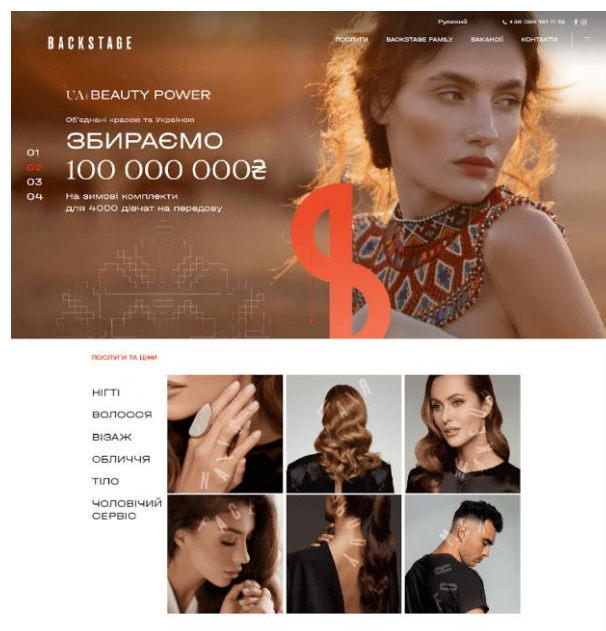


Рисунок 1.1 – Головна сторінка сайту Backstage

Джерело: побудовано автором (знімок з екрану)

Інтерфейс Backstage привертає увагу своєю сучасністю та стильністю. Навігація є легкою та логічною, що сприяє комфортному користуванню.

Сайт адаптований для різних пристроїв, надаючи клієнтам однаково зручний доступ з комп'ютерів чи мобільних телефонів.

Важливою рисою є повнота інформації. Клієнти отримують не лише перелік послуг, а й їх докладний опис, ціни та акції. Розділ, присвячений команді салону, дозволяє клієнтам знайомитися з фахівцями ближче.

Backstage пропонує зручний онлайн-календар для бронювання послуг, а процес онлайн-замовлення є легким і прозорим, що сприяє швидкому вибору бажаних послуг.

Сайт має окремий розділ для відгуків клієнтів, що дозволяє ділитися враженнями та дозволяє іншим отримати об'єктивну оцінку фахівців та послуг.

Сайт активно взаємодіє з соціальними мережами. Можливість поділитися результатами та наявність посилань на сторінки салону у соціальних мережах робить взаємодію з аудиторією більш ефективною.

Backstage вражає своєю швидкістю завантаження та надійністю роботи. Сучасні технічні можливості забезпечують користувачам швидкий та зручний доступ до інформації.

1.3. Салон краси Wizard

Салон краси "Wizard" намагається відрізнятись не лише своїми високотехнологічними послугами, але й привабливим та інтуїтивно зрозумілим веб-сайтом, представлено на рисунку 1.2.



Рисунок 1.2 – Головна сторінка сайту Wizard

Джерело: побудовано автором (знімок з екрану)

Дизайн сайту "Wizard" привабливий та сучасний, з акцентом на важливості естетики.

Навігація робиться зручною за допомогою логічно розташованих елементів, а адаптивність забезпечує зручний доступ до інформації на різних пристроях.

Сайт містить комплексну та актуальну інформацію про послуги, включаючи ціни та акції. Деталізована інформація про команду та фахівців дозволяє клієнтам краще ознайомитися з експертами салону.

Салон пропонує зручний онлайн-календар для бронювання, а процес онлайн-замовлення послуг є швидким та безпечним, що створює позитивний досвід для клієнтів.

Розділ із відгуками дозволяє клієнтам ділитися своїми враженнями, а рейтинг фахівців та послуг надає об'єктивну оцінку для майбутніх клієнтів.

Сайт активно взаємодіє з соціальними мережами, дозволяючи клієнтам ділитися результатами та швидко отримувати оновлення через сторінки салону в соціальних мережах.

Сайт швидко завантажується, а робота онлайн-сервісів проходить без збоїв, що гарантує ефективний взаємодію з користувачами.

Інформація про акції та знижки легко доступна, а можливість підписки на розсилку надає клієнтам можливість отримувати ексклюзивні пропозиції.

1.4. Салон краси "Mira"

Салон краси "Mira" вражає своїм веб-сайтом, який не лише інформує клієнтів про послуги, але й надає їм приємний інтерфейс для взаємодії. Дизайн "Mira" є привабливим і сучасним.

Навігація є зручною, а інтуїтивний інтерфейс полегшує використання сайту навіть новачкам.

Сайт також адаптивний до різних пристроїв, що забезпечує комфортне користування навіть на мобільних телефонах та планшетах, представлено на рисунку 1.3.

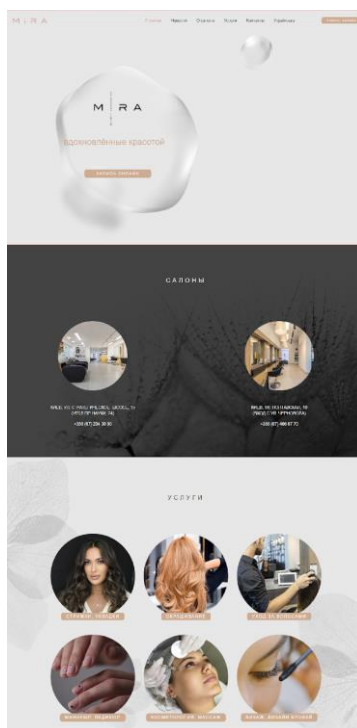


Рисунок 1.3 – Головна сторінка сайту Mira

Джерело: побудовано автором (знімок з екрану)

Сайт містить докладну та актуальну інформацію про послуги, включаючи ціни та акції. Інформація про команду та фахівців дозволяє клієнтам знайомитися з експертами салону перед візитом.

Можливість використовувати онлайн-календар для бронювання та легкість процесу онлайн-замовлення послуг робить візит в салон швидким та зручним для клієнтів.

Розділ із відгуками клієнтів надає можливість поділитися думками, а рейтинг фахівців та послуг допомагає клієнтам знайти найкращий вибір.

"Mira" активно використовує соціальні мережі, дозволяючи клієнтам легко ділитися своїми результатами та отримувати оновлення через соціальні сторінки салону.

Сайт завантажується швидко, і робота онлайн-сервісів не викликає збоїв, забезпечуючи надійну та продуктивну роботу.

Інформація про акції та знижки доступна на сайті, а можливість підписки на розсилку дозволяє клієнтам отримувати персоналізовані пропозиції.

Узагальнюючи, сучасні салони краси повинні активно впроваджувати інноваційні технології та системи управління для покращення якості обслуговування, оптимізації робочих процесів та забезпечення конкурентоспроможності в галузі. Розвиток інформаційних та комунікаційних технологій стає стратегічним кроком у напрямку сучасного та успішного салону краси.

Таблиця 1.1 – Порівняльна таблиця характеристик аналогів Web-додатку

Джерело: побудовано автором

Критерій	"Backstage"	"Wizard"	"Mira"	"Beauty Studio"
Привабливість дизайну	+	+	+	+
Зручність навігації	+	+	+	+
Інтуїтивність використання	+	+	+	+
Адаптивність для різних пристроїв	+	+	+	+

Продовження таблиці 1.1

Інформаційний вміст	-	+	+	+
Запис на послуги	+	+	+	+
Відгуки та рейтинги	+	+	+	+
Інтеграція з соціальними мережами	+	+	+	+
Технічна продуктивність	+	+	+	+

Сайти мають сучасний, привабливий та мінімалістичний стиль, що відповідає цільовій аудиторії. Також в них зрозуміла і проста навігація, за допомогою якої користувач може швидко знаходити потрібну інформацію. Відвідувач може користуватися сайтами без попереднього навчання, адже меню, кнопки й іконки легко помітні та інтуїтивно зрозумілі. Сайти адаптовані для різних пристроїв, включаючи телефони та планшети.

Відвідувач бачить на сайтах актуальну інформацію, ціни на послуги, інформацію про фахівців. Всі аналоги, крім “Backstage”, містять розділ з акціями та знижками. Відвідувачі можуть з легкістю здійснити онлайн-запис та написати відгук про певного спеціаліста. На сайтах є посилання на соціальні мережі салонів краси. Сторінки швидко завантажуються, працюють без збоїв.

Порівнявши характеристики аналогів Web-додатку, прийнято рішення, зробити сайт зручним, зрозумілим та естетичного дизайну. Сайт матиме інформацію про наявні послуги салону та дані про кожного працівника. Відвідувач буде мати можливість при здійсненні онлайн-запису обрати стиліста та зручний час, а також при потребі відмінити свій запис. Через посилання можна потрапити на соціальну сторінку салону. Для орієнтування обрано сайт «Mira», через його зрозуміле та комфортне користування.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1. Мета та задачі

Головна мета кваліфікованої роботи – розробити та впровадити ефективну Web-орієнтовану систему підтримки для салонів краси на основі фреймворку Laravel, яка забезпечить автоматизацію рутинних процесів та оптимізацію управління різними аспектами діяльності салону.

Об’єкт дослідження – підтримка діяльності салону краси.

Предмет дослідження – web-орієнтована система підтримки діяльності салону краси.

Цільова аудиторія сайту – користувачі, які планують відвідати салон, попередньо ознайомившись з послугами, ціною та здійснити онлайн-запис.

Основні вимоги до функціонала додатка: перегляд інформації про наявні послуги салону та дані про кожного працівника, здійснення онлайн-запису з можливістю обрати стиліста та зручний час, відміна онлайн-запису, перехід через посилання на соціальні сторінки салону, можливість написати відвідувачу відгук про певного спеціаліста.

Основні задачі:

- проведення аналізу предметної області у сфері краси;
- дослідження методів, технологій реалізації задач для сфери послуг на прикладах салонів краси;
- проектування, моделювання web-орієнтованої системи;
- розробка та тестування web-орієнтованої системи підтримки діяльності салону краси;
- Оформлення документації.

2.2. Вибір засобів реалізації

Сучасні салони краси визначають новий стандарт обслуговування, виходячи за рамки звичайних косметичних послуг. Вони перетворюються на місця, де клієнти можуть не лише покращити свій зовнішній вигляд, а й насолодитися процесом, знайти відпочинок та виразити свою індивідуальність. Зростаючий інтерес до краси та естетики породжує підвищену конкуренцію серед салонів. У цьому контексті виникає необхідність у вдосконаленні систем управління, адже багато салонів із застарілими методами бронювання та обліку послуг потрапляють в ситуації плутанини та незручностей для клієнтів.

Відповідь на цей виклик полягає у створенні Web-орієнтованої системи, яка об'єднає в собі різноманітні функції для зручної взаємодії з клієнтами та ефективного управління. Ключові завдання включають розробку бази даних для детального обліку послуг, створення інтерактивного календаря для онлайн-бронювання та розробку інтуїтивного інтерфейсу для користувачів. Використання Laravel та інших технологій, таких як PHP, PHPMyAdmin, MySQL гарантує надійність та ефективність функціоналу системи.

Паралельно з цим, необхідно докладно проаналізувати існуючі Web-системи салонів краси, визначити їх переваги та недоліки. Однак важливо не лише адаптувати кращі практики, а й створити унікальну систему, яка враховує особливості сучасних потреб клієнтів.

Реалізація проекту включає в себе етапи проектування та моделювання, створення бази даних для послуг, а також систему бронювання. Задачею є також створення інтуїтивно зрозумілого інтерфейсу, що дозволить клієнтам легко користуватися системою. Застосування Laravel та інших інструментів для безпеки забезпечить надійність та захищеність даних клієнтів.

HTML (HyperText Markup Language) є стандартною мовою для створення веб-сторінок. В основі його функціонування лежить використання

тегів, які визначають структуру документа та його вміст. HTML дозволяє включати текст, зображення, посилання, таблиці та інші елементи, які роблять вміст доступним для веб-браузерів.

HTML дозволяє також створювати форми для збору інформації від користувачів, таблиці для представлення даних у табличній формі, а також вбудовувати мультимедійні елементи.

Його простота та доступність допомагають веб-розробникам швидко створювати сторінки, які можуть бути легко розгорнуті в Інтернеті. HTML виступає основою для інших технологій веб-розробки, таких як CSS для стилізації та JavaScript для взаємодії з користувачем. Вивчення HTML є ключовим етапом для розпочатку кар'єри в веб-розробці.

HTML було обрано за доступний та легкий синтаксис, зрозумілу розмітку. Також великою перевагою є те, що мова безплатна та розташована у відкритому доступі.

CSS (Cascading Style Sheets) є мовою стилізації, яка використовується для задання вигляду та форматування веб-документів, написаних мовою HTML. Основні цілі CSS полягають в розділенні вмісту та його представлення на сторінці, щоб забезпечити більш гнучкий та ефективний дизайн веб-сайтів.

CSS дозволяє визначати стилі для різних медіа (екрани комп'ютерів, пристроїв з екранами високої роздільної здатності, мобільних пристроїв і т. д.), що робить його адаптивним для різних пристроїв і розмірів екранів.

Мова CSS дозволяє визначати стилі для різних елементів HTML, включаючи кольори, шрифти, розташування, розміри, відступи, рамки та інші характеристики. Створюючи зовнішній вигляд веб-сайту, CSS допомагає розділити структуру та представлення, сприяючи більшій гнучкості в управлінні дизайном та поліпшенню користувацького досвіду.

З використанням CSS можна створювати анімації, переходи та інші візуальні ефекти, що робить його потужним інструментом для веб-

розробників. Опанування CSS є важливим етапом для досягнення високого рівня професійної компетентності в галузі веб-розробки.

CSS було обрано за можливість легко функціонувати з рядками та полями та зручно змінювати їх параметри. Також мова дозволяє швидко змінювати шрифт та колір, працювати з фотографіями, взаємодіяти з елементами.

JavaScript – це високорівнева, інтерпретована мова програмування, яка використовується для створення інтерактивних веб-сайтів. Розроблена компанією Netscape, JavaScript стала однією з найпопулярніших та найбільш використовуваних мов у сфері веб-розробки.

Основна роль JavaScript полягає в наданні можливостей взаємодії користувача з веб-сторінками. Зазвичай використовується в парі з HTML та CSS, JavaScript дозволяє динамічно змінювати вміст сторінки, обробляти події, валідувати дані в формах та взаємодіяти з сервером без перезавантаження сторінки.

Однією з ключових особливостей JavaScript є його асинхронний та подієвий характер. Код може виконуватися асинхронно, що означає, що виконання програми може продовжуватися без очікування завершення певних завдань, таких як завантаження ресурсів.

Мова постійно розвивається, і її нові версії регулярно випускаються з новими можливостями та поліпшеннями. Наявність різноманітних інструментів та розширень робить JavaScript важливим елементом в арсеналі веб-розробників.

Мова JavaScript обрано для реалізації можливостей реакції веб сторінки або її елементів на дії користувача.

PHPMyAdmin – це вільний та відкритий інструмент для адміністрування системи управління базами даних MySQL за допомогою веб-інтерфейсу. Важливим елементом в розробці Web-орієнтованої системи для салону краси є використання PHPMyAdmin для ефективного управління та обслуговування бази даних.

PHPMyAdmin дозволяє зручно взаємодіяти з базою даних через інтуїтивний веб-інтерфейс, що полегшує виконання таких завдань, як створення таблиць, вставка, оновлення та видалення даних, а також управління користувачами та їхніми привілеями. Завдяки PHPMyAdmin розробникам та адміністраторам стає доступним проведення рутинних операцій без прямого використання команд SQL.

Один із ключових аспектів використання PHPMyAdmin у контексті салону краси – це забезпечення безпеки та надійності бази даних. За допомогою PHPMyAdmin можна ефективно керувати доступом до бази даних, встановлювати паролі, налаштовувати привілеї користувачів, що є критичним для забезпечення конфіденційності та цілісності даних клієнтів салону краси.

Також важливою функціональністю є можливість експорту та імпорту даних, що полегшує резервне копіювання та відновлення інформації у випадку непередбачуваних ситуацій.

Узагальнюючи, використання PHPMyAdmin стає ключовим елементом розробки Web-орієнтованої системи для салону краси на базі MySQL, забезпечуючи зручність, безпеку та надійність в управлінні базою даних.

MySQL є однією з найбільш впізнаваних технологій у сучасній екосистемі великих даних. Її часто називають найпопулярнішою базою даних, яка наразі користується широким та ефективним використанням незалежно від галузі. Зрозуміло, що будь-хто, хто займається корпоративними даними чи загальними ІТ-технологіями, повинен прагнути принаймні до базових знань MySQL.

Порівняно з багатьма рішеннями для зберігання та обробки даних, представленими сьогодні на ринку, MySQL є старішою технологією, але вона не демонструє жодних ознак втрати популярності чи корисності. Насправді MySQL нещодавно відродилася в порівнянні з більш спеціалізованими сучасними системами зберігання даних завдяки своїй швидкості, надійності, простоті використання та широкій сумісності.

MySQL – популярне, перевірене часом, але також сучасне та повнофункціональне програмне забезпечення для керування реляційними базами даних. Підприємства всюди використовують його для зберігання та обробки критично важливих корпоративних даних, як серверну частину для основних програм, орієнтованих на клієнтів, і як частину потужних стеків веб-програм.

Laravel – це високорівневий веб-фреймворк для розробки програмного забезпечення з використанням мови програмування PHP. Розроблений Тейлором Отвеллом, Laravel вирізняється елегантністю, чистотою коду та потужними можливостями для швидкого створення високоякісних веб-додатків.

Laravel став популярним веб-фреймворком завдяки своїй зручності, ефективності та активній спільноті розробників. Він відмінно підходить для розробки різноманітних веб-додатків, включаючи корпоративні сайти, електронні магазини та веб-сервіси.

Для створення інтерфейсу буде використовуватися HTML та CSS. PHP, Laravel обрано для розробки бізнес-логіки. Для інтеграції бази даних найкраще підійде MySQL.

3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

3.1. Розробка структурно-функціональної моделі системи

В сучасному суспільстві салони краси виконують більше, ніж просто функцію надання косметичних послуг. Вони стали центрами для відпочинку, релаксації та самовираження. З плином часу зростає попит на різноманітні косметичні та естетичні процедури, але разом із цим збільшується конкуренція серед салонів. Управління салоном та покращення рівня обслуговування стають складнішими завданнями.

Один із ключових викликів для сучасних салонів краси – це необхідність модернізації систем управління. Багато з них залишаються при старих методах, таких як ручне бронювання та облік, що може призводити до плутанини та незручностей для клієнтів. Впровадження сучасних технологій управління стає критично важливим для оптимізації робочих процесів, підвищення якості послуг і досягнення більшої ефективності.

Розробка інформаційних систем стає ще одним важливим аспектом. Створення спеціалізованих веб-систем, що включають в себе онлайн-календар бронювання, зручний інтерфейс для управління базами даних клієнтів та аналітичні інструменти, дозволяє салонам оптимізувати процеси бронювання та взаємодії з клієнтами.

Зростання популярності онлайн-платформ і соціальних мереж робить актуальним питання про активну онлайн-присутність салонів краси. Веб-сайти та профілі в соціальних мережах стають ключовими інструментами для просування послуг, взаємодії з клієнтами та отримання зворотного зв'язку. Інтеграція цих платформ з системами управління стає важливим елементом інфраструктури.

З неабиякою кількістю можливостей та технологій з'являються нові виклики. Забезпечення безпеки та конфіденційності даних клієнтів в онлайн-системах стає ключовим завданням. Захист від несанкціонованого доступу та

ефективне управління даними стають важливими елементами інфраструктури сучасного салону краси.

Отже, сучасні салони краси повинні активно впроваджувати інноваційні технології та системи управління для підвищення якості послуг, оптимізації робочих процесів та забезпечення конкурентоспроможності у галузі. Розвиток інформаційно-комунікаційних технологій стає стратегічним кроком на шляху до сучасних та успішних салонів краси.

Для відображення процесів в системі створено IDF0 модель системи, дана модель представлена на рисунку 3.1.

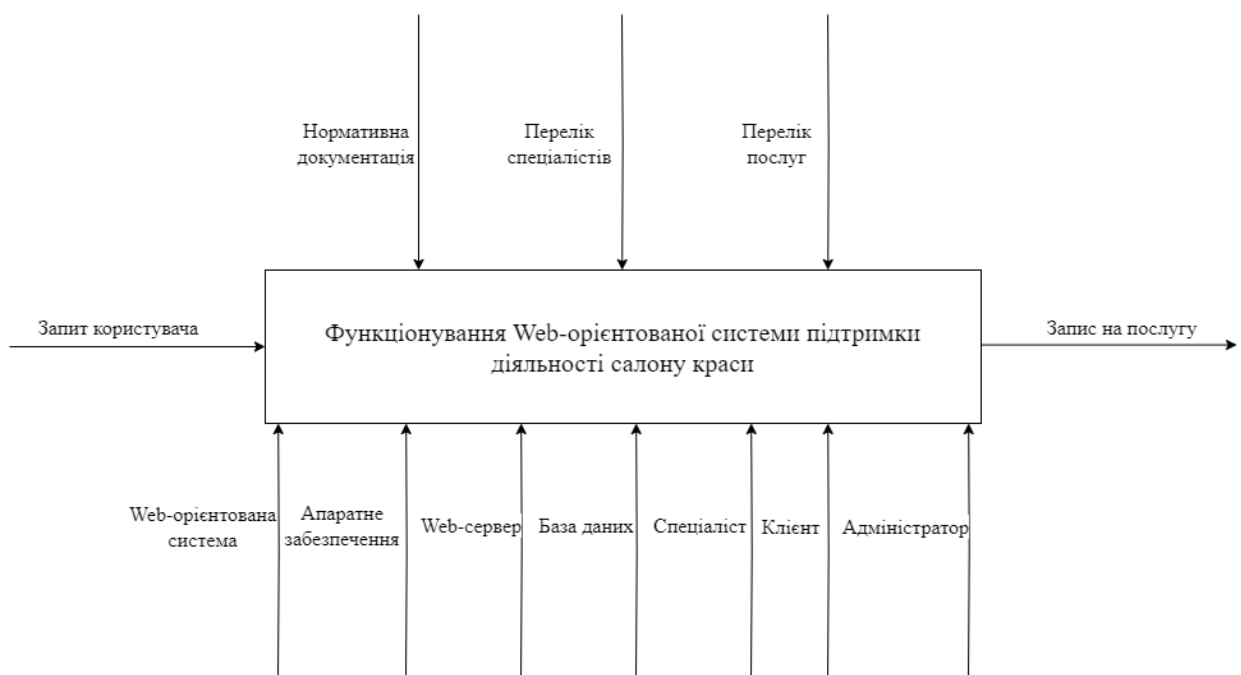


Рисунок 3.1 – IDF0 модель системи

Джерело: побудовано автором

На рисунку 3.2 представлено декомпозицію IDF0 моделі систему салону краси.

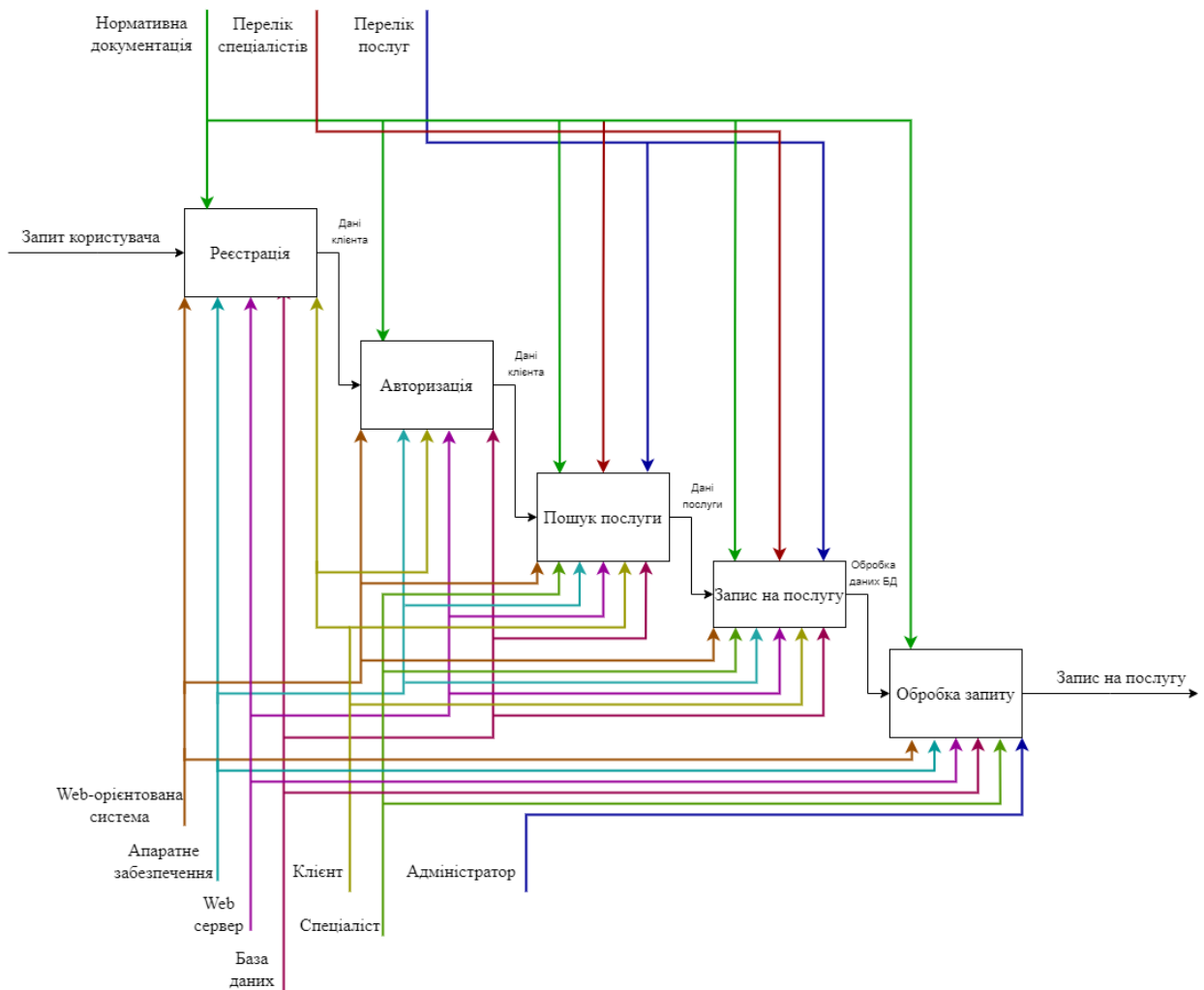


Рисунок 3.2 – Декомпозиція IDF0 моделі системи

Джерело: побудовано автором

3.2. Проектування моделі бази даних

Проектування моделі бази даних для системи управління салоном краси – це важливий етап, спрямований на ефективне зберігання та обробку інформації.

В основі моделі бази даних лежить сутність-зв'язок, що означає ідентифікацію ключових сутностей та їх взаємозв'язки.

Для бази даних салону краси розроблено фізичну модель даних, яка представлена на рисунку 3.3.

Сервер: 127.0.0.1:3306 / База даних: beauty / Таблиця: appointments

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Тригери

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	user_id	bigint(20)		UNSIGNED	Ні	Немає			Більше
3	service_id	bigint(20)		UNSIGNED	Ні	Немає			Більше
4	stylist_id	bigint(20)		UNSIGNED	Ні	Немає			Більше
5	status_id	bigint(20)		UNSIGNED	Ні	1			Більше
6	appointment_date	date			Ні	Немає			Більше
7	created_at	timestamp			Так	NULL			Більше
8	updated_at	timestamp			Так	NULL			Більше

Позначити все Вибрані:

Рисунок 3.4 – Структура таблиці appointments

Джерело: побудовано автором (знімок з екрану)

Таблиця categories:

- id: унікальний ідентифікатор категорії.
- title: назва категорії.
- photo_url: URL-адреса фото для категорії.

Детально структуру таблиці представлено на рисунку 3.5.

Сервер: 127.0.0.1:3306 / База даних: beauty / Таблиця: categories

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Тригери

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	photo_url	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше

Позначити все Вибрані:

Рисунок 3.5 – Структура таблиці categories

Джерело: побудовано автором (знімок з екрану)

Таблиця reviews:

- id: унікальний ідентифікатор відгуку.
- stylist_id: ідентифікатор фахівця, якому залишено відгук.

- user_id: ідентифікатор користувача, який залишив відгук.
- rating: оцінка відгуку.
- comment: коментар користувача.
- created_at, updated_at: мітки часу.

Детально структуру таблиці представлено на рисунку 3.6.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	✎ ⌵ Більше
2	stylist_id	bigint(20)		UNSIGNED	Ні	Немає			✎ ⌵ Більше
3	user_id	bigint(20)		UNSIGNED	Ні	Немає			✎ ⌵ Більше
4	rating	int(11)			Ні	Немає			✎ ⌵ Більше
5	comment	text	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵ Більше
6	created_at	timestamp			Так	NULL			✎ ⌵ Більше
7	updated_at	timestamp			Так	NULL			✎ ⌵ Більше

Рисунок 3.6 – Структура таблиці reviews

Джерело: побудовано автором (знімок з екрану)

Таблиця services:

- id: унікальний ідентифікатор послуги.
- service_name: назва послуги.
- description: опис послуги.
- price: ціна послуги.
- photo_url: URL-адреса фото для послуги.
- category_id: ідентифікатор категорії, до якої належить послуга.

Детально структура таблиці представлено на рисунку 3.7.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	service_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	description	text	utf8mb4_unicode_ci		Ні	Немає			Більше
4	price	int(11)			Ні	Немає			Більше
5	photo_url	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
6	category_id	bigint(20)		UNSIGNED	Ні	Немає			Більше

Рисунок 3.7 – Структура таблиці services

Джерело: побудовано автором (знімок з екрану)

Таблиця statuses:

- id: унікальний ідентифікатор статусу.
- title: назва статусу.

Детально структуру таблиці представлено на рисунку 3.8.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше

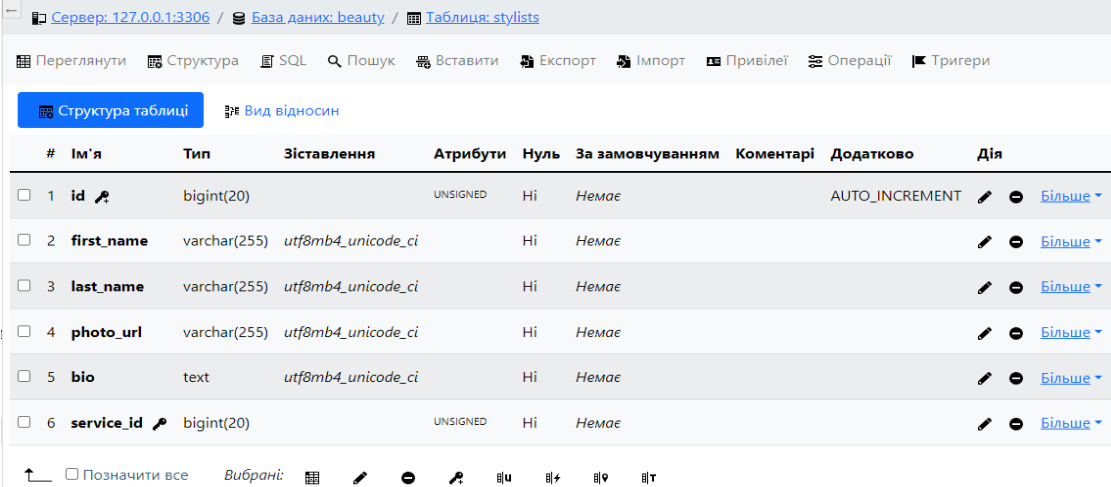
Рисунок 3.8 – Структура таблиці statuses

Джерело: побудовано автором (знімок з екрану)

Таблиця stylists:

- id: унікальний ідентифікатор фахівця.
- first_name, last_name: ім'я та прізвище фахівця.
- photo_url: URL-адреса фото фахівця.
- bio: біографія фахівця.
- service_id: ідентифікатор послуги, яку надає фахівець.

Детально структуру таблиці представлено на рисунку 3.9.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	first_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	last_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
4	photo_url	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
5	bio	text	utf8mb4_unicode_ci		Ні	Немає			Більше
6	service_id	bigint(20)		UNSIGNED	Ні	Немає			Більше

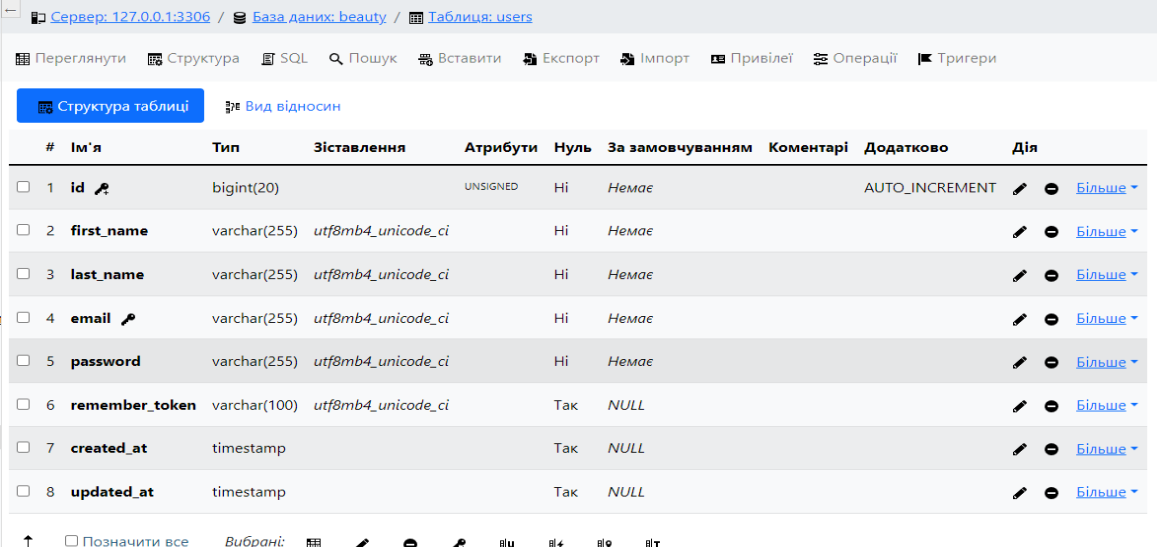
Рисунок 3.9 – Структура таблиці stylists

Джерело: побудовано автором (знімок з екрану)

Таблиця users:

- id: унікальний ідентифікатор користувача.
- first_name, last_name: ім'я та прізвище користувача.
- email: електронна адреса користувача.
- password: хеш пароля користувача.
- remember_token: токен для запам'ятовування користувача.
- created_at, updated_at: мітки часу.

Детально структуру таблиці представлено на рисунку 3.10.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	first_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	last_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
4	email	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
5	password	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
6	remember_token	varchar(100)	utf8mb4_unicode_ci		Так	NULL			Більше
7	created_at	timestamp			Так	NULL			Більше
8	updated_at	timestamp			Так	NULL			Більше

Рисунок 3.10 – Структура таблиці users

Джерело: побудовано автором (знімок з екрану)

3.3. Моделювання варіантів використання системи

Моделювання варіантів використання системи – це методологія для опису функціональності системи з точки зору користувача. Варіанти використання описують взаємодію між системою та її користувачами, визначаючи, як система реагує на конкретні запити.

Давайте розглянемо приклад моделювання варіантів використання для системи підтримки діяльності салону краси.

На рисунках 3.11-3.14 представлено діаграму класів додатку.

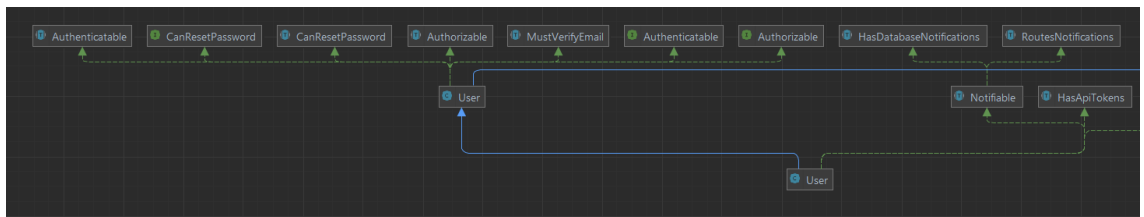


Рисунок 3.11 – Діаграма класів додатку салону краси

Джерело: побудовано автором (знімок з екрану)

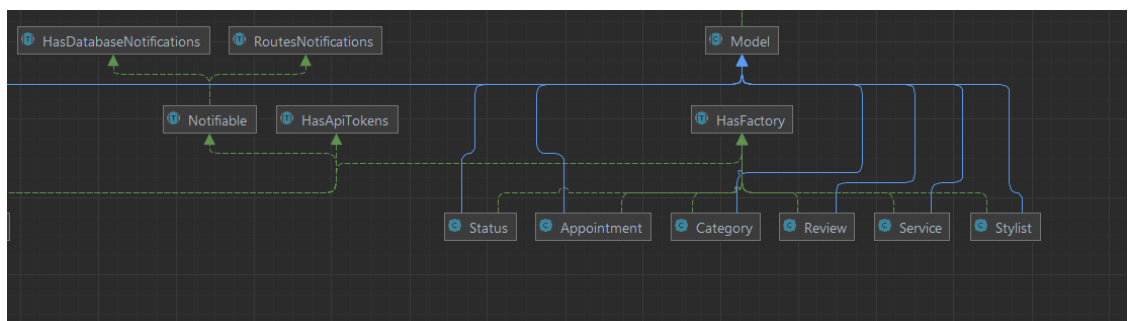


Рисунок 3.12 – Діаграма класів додатку салону краси

Джерело: побудовано автором (знімок з екрану)

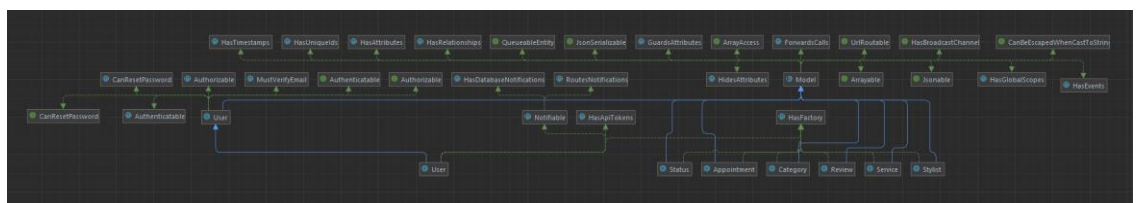


Рисунок 3.13 – Діаграма класів додатку салону краси

Джерело: побудовано автором (знімок з екрану)

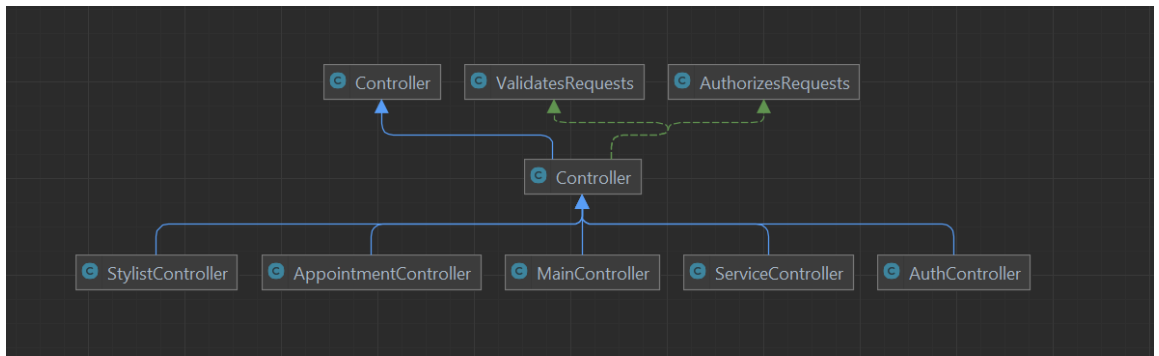


Рисунок 3.14 – Діаграма класів додатку салону краси

Джерело: побудовано автором (знімок з екрану)

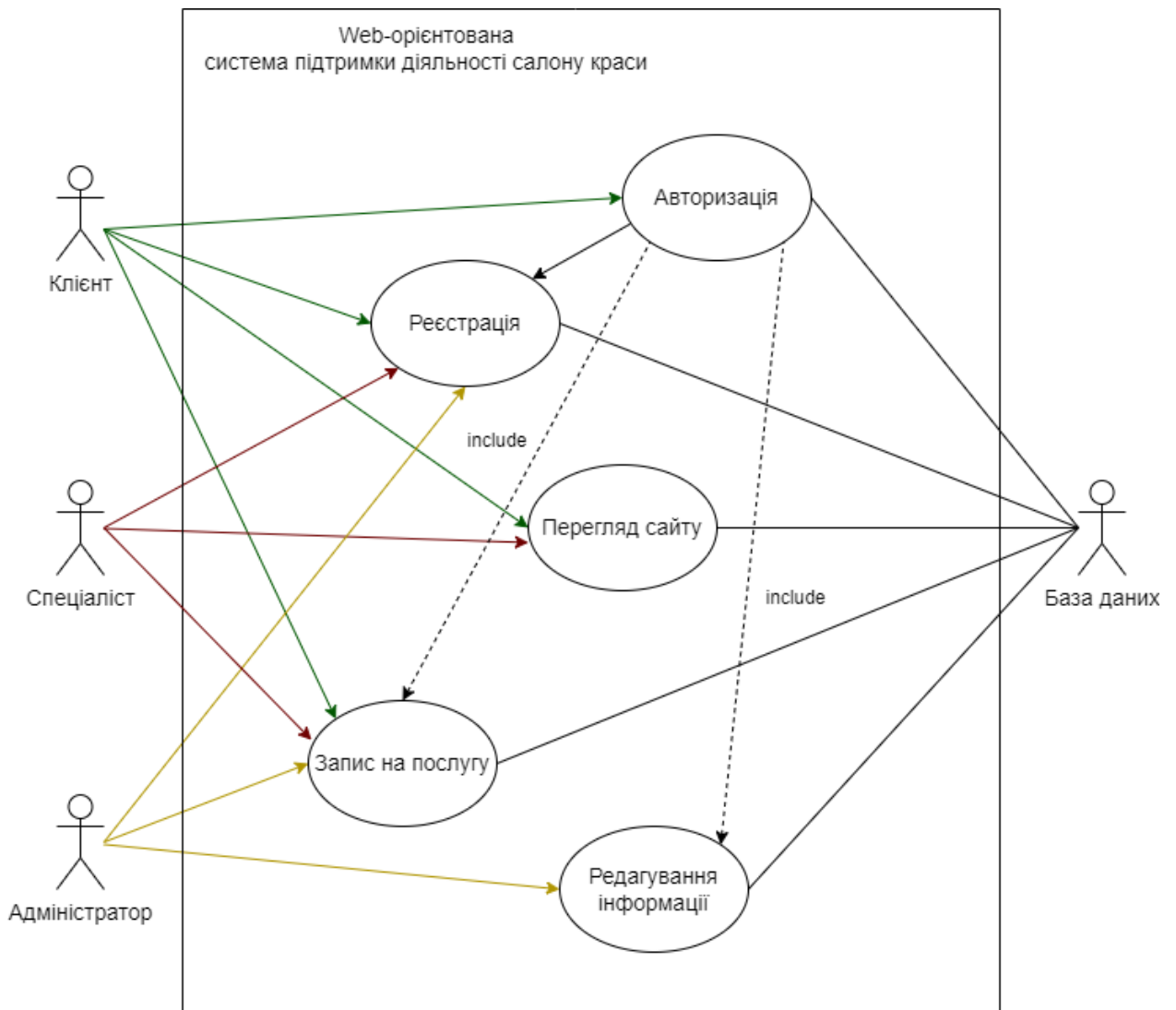


Рисунок 3.15 – Діаграма варіантів використання

Джерело: побудовано автором

4 РОЗРОБКА WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ

4.1. Розробка інтерфейсу (HTML, CSS)

Розробка інтерфейсу для сайту салону краси включає в себе створення зручного та естетичного дизайну, який забезпечить користувачам приємний та ефективний досвід перебування на веб-сторінці.

Для більш детального розуміння того як буде відображено сторінки, було створено макет у редакторі Figma.

Було розроблено макет наступних сторінок:

- головна;
- сторінка з послугами;
- сторінка з інформацією про працівників;
- сторінку з інформацією про певного працівника;
- сторінку оформлення замовлення;
- сторінку профілю;
- сторінку авторизації користувача;
- сторінку авторизації адміністратора;
- сторінку реєстрації користувача;
- адміністративну панель.

На рисунку 4.1 представлено макет головної сторінки сайту.

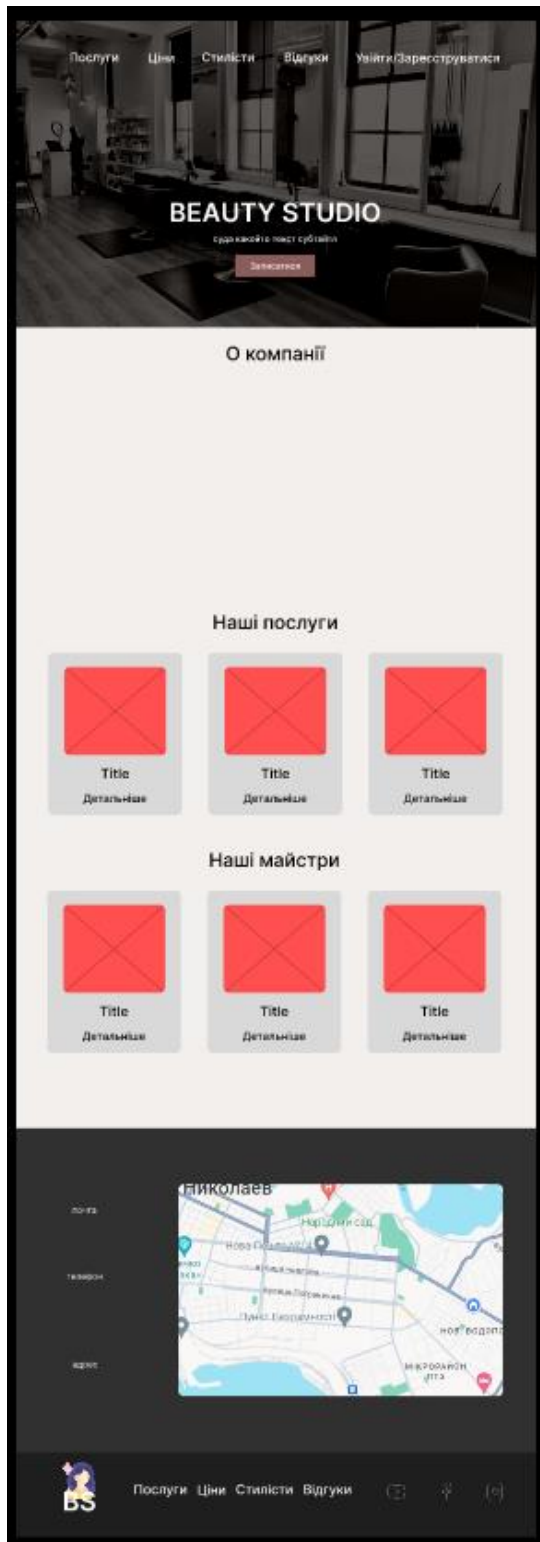


Рисунок 4.1 – Макет головної сторінки сайту

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.2 представлено макет сторінки з послугами та цінами.

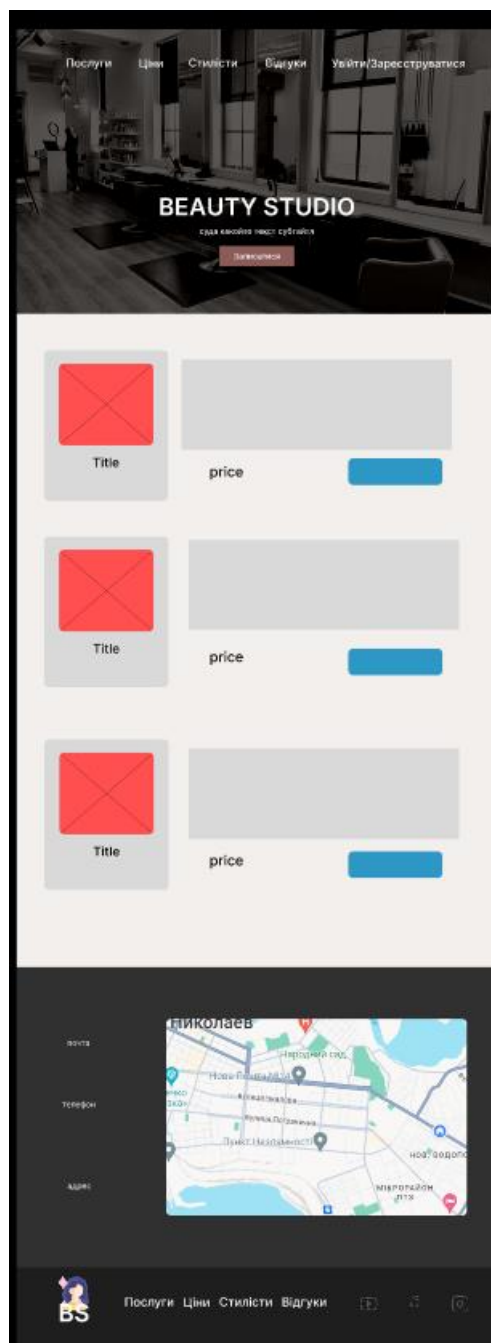


Рисунок 4.2 – Макет сторінки з послугами

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.3 представлено макет сторінки з інформацією про працівників.

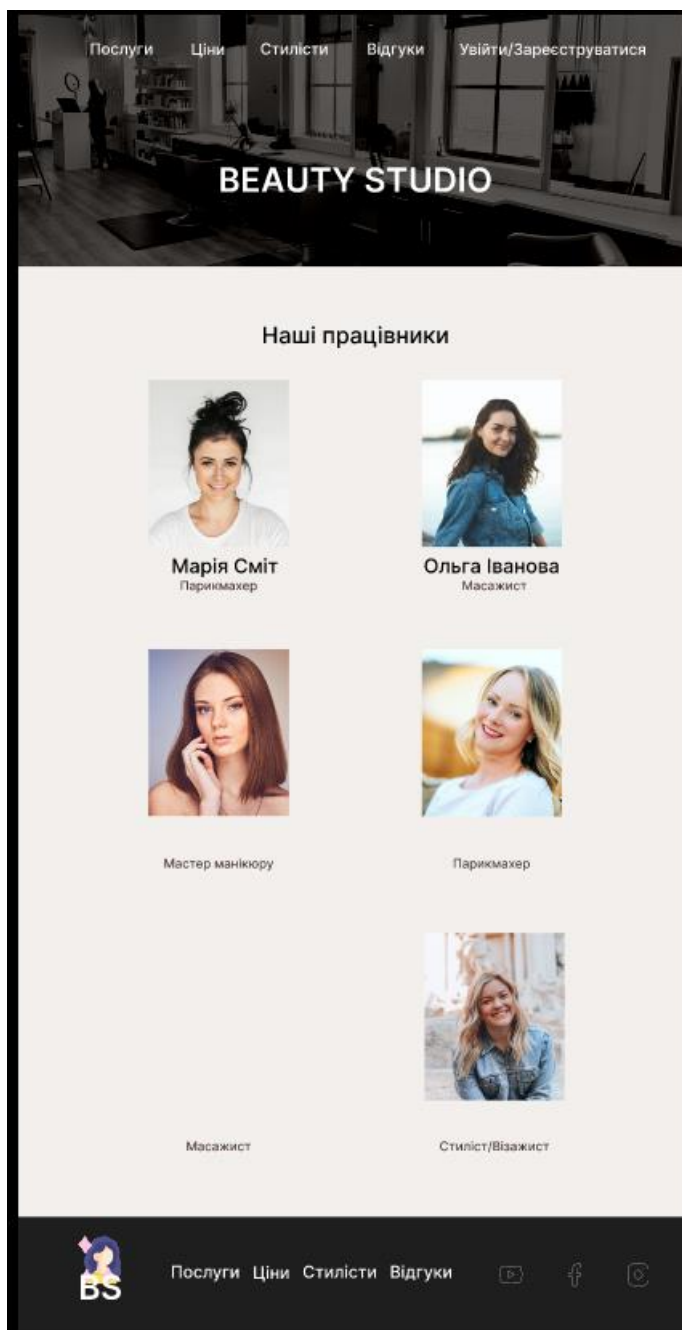


Рисунок 4.3 – Макет сторінки з працівниками

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.4 представлено макет сторінки для певного працівника

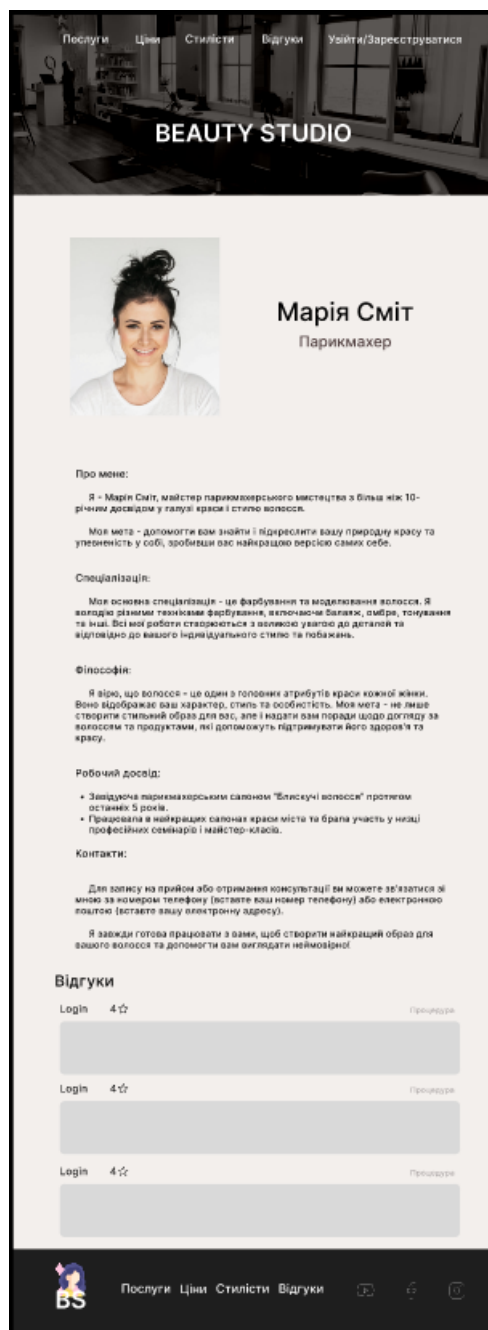


Рисунок 4.4 – Макет сторінки для певного працівника

Джерело: побудовано автором (знімок з екрану)

На рисунку 4.5 представлено макет сторінки профілю.

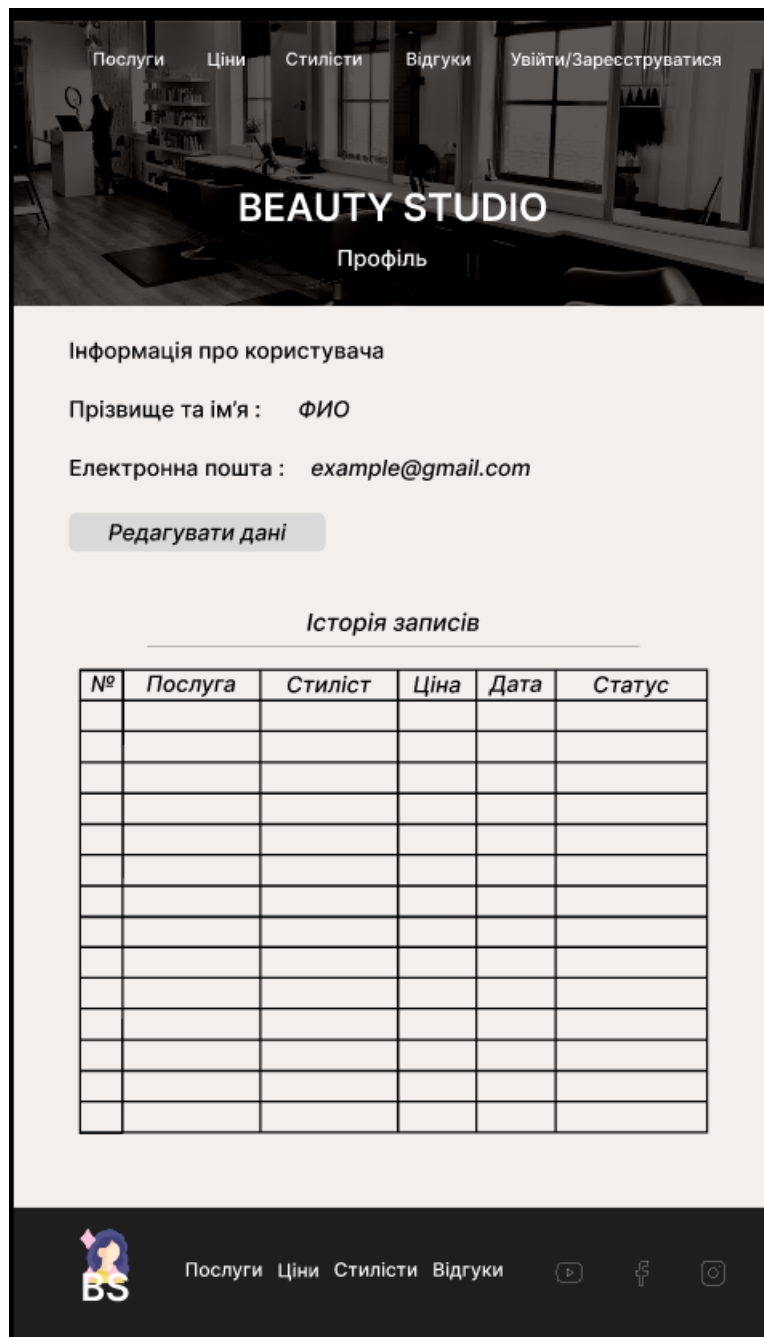


Рисунок 4.5 – Макет сторінки профілю

Джерело: побудовано автором (знімок з екрану)

На рисунках 4.6-4.8 представлено форми для реєстрації та авторизації нового користувача, а також форму авторизації адміністратора в адмін панелі.

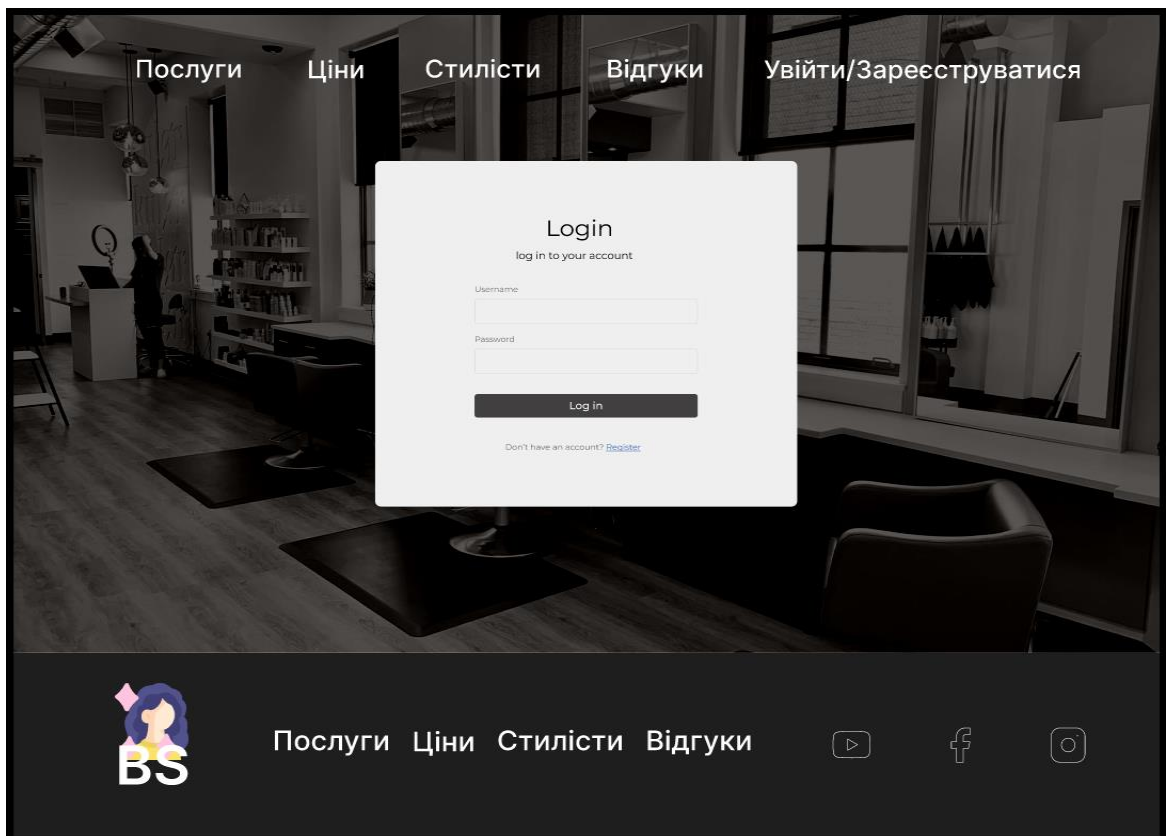


Рисунок 4.6 – Макет форми авторизації користувача

Джерело: побудовано автором (знімок з екрану)

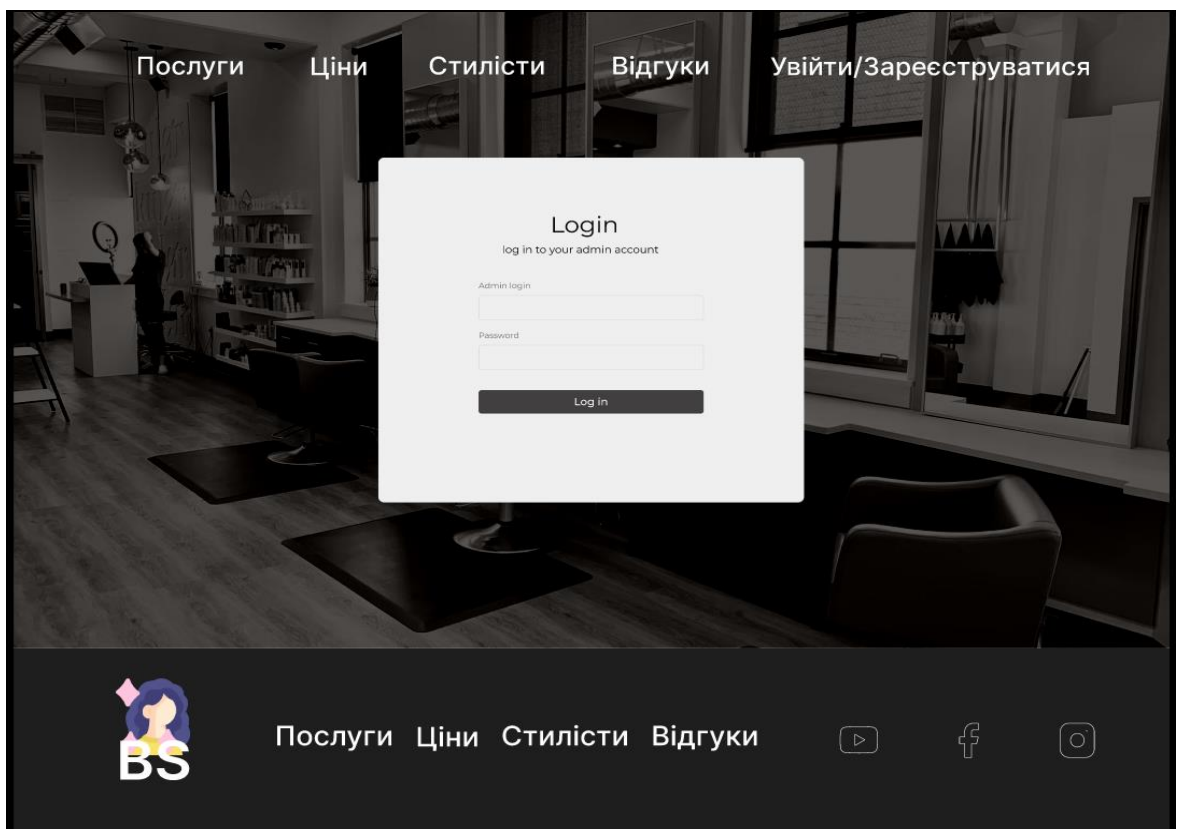


Рисунок 4.7 – Форма авторизації адміністратора в адмін панелі

Джерело: побудовано автором (знімок з екрану)

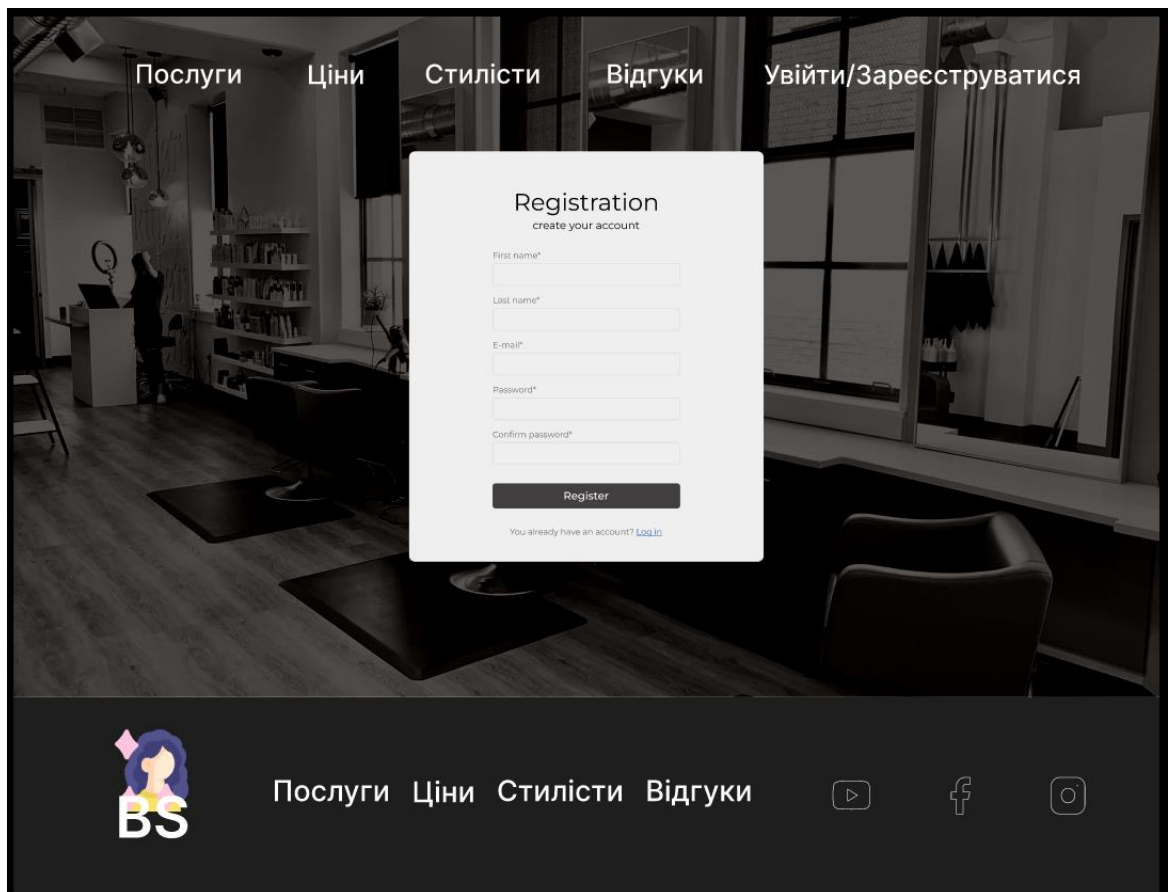


Рисунок 4.8 – Макет форми реєстрації нового користувача

Джерело: побудовано автором (знімок з екрану)

Адміністративна панель – це веб-інтерфейс, призначений для управління та адмініструванням веб-сайту чи додатку. Вона надає доступ до різноманітних функцій та інструментів, які дозволяють адміністраторам контролювати та налаштовувати різні аспекти системи безпосередньо через веб-браузер.

Основні можливості адміністративної панелі включають управління контентом, користувачами, правами доступу, налаштуваннями системи, аналітикою, безпекою та іншими параметрами, які визначені функціональністю конкретного веб-застосунку чи сайту.

В сучасних веб-додатках, таких як системи управління контентом (CMS), електронна комерція, платформи для реєстрації та бронювання, адміністративна панель є ключовим інструментом для забезпечення ефективного та зручного управління.

Макет сторінки адмін панелі представлено на рисунку 4.9.

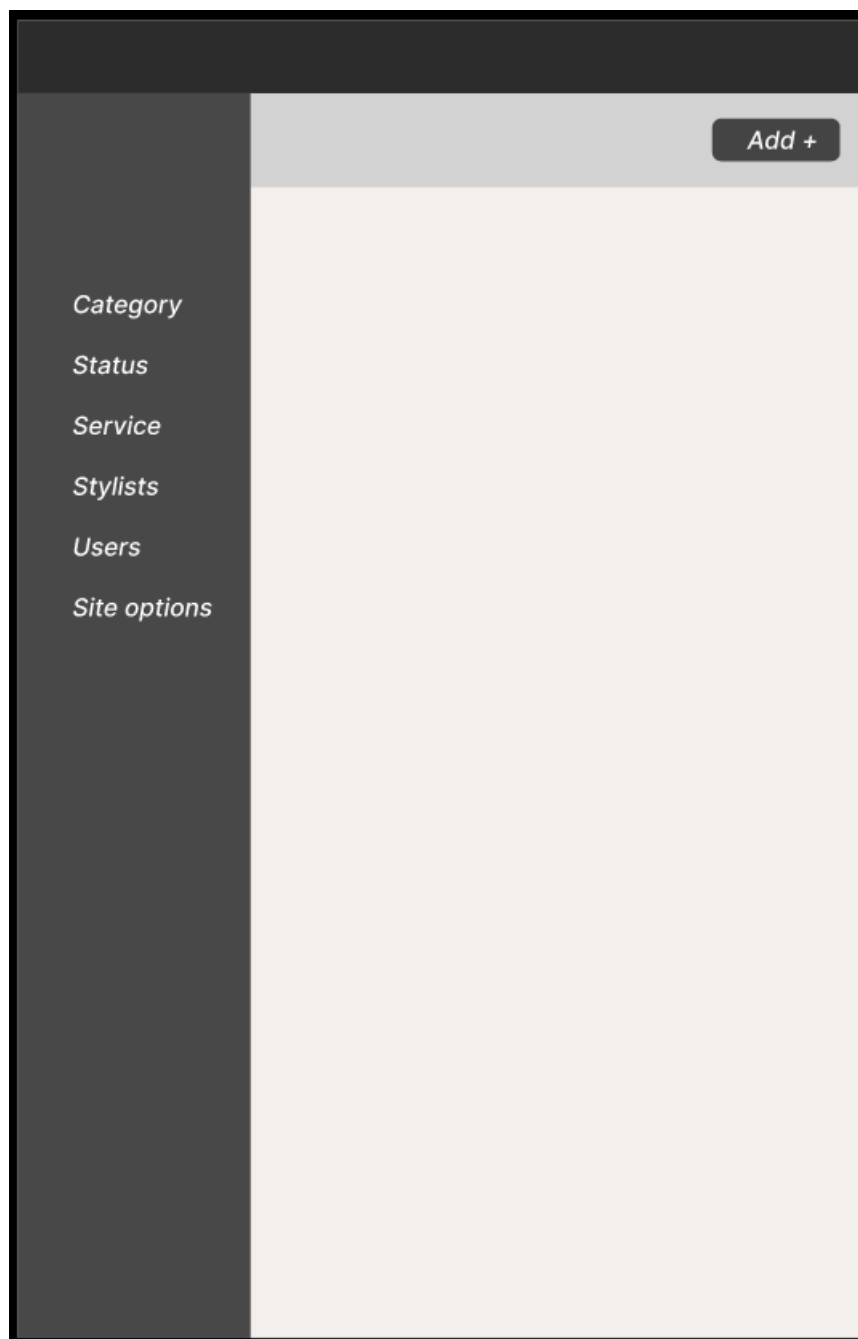


Рисунок 4.9 – Макет адмін панелі сайту
Джерело: побудовано автором (знімок з екрану)

4.2. Розробка бізнес-логіки (PHP, Laravel)

Розробка бізнес-логіки для системи підтримки салону краси, використовуючи PHP та Laravel, включатиме реалізацію різноманітних функцій та процесів для оптимізації управління та покращення взаємодії з клієнтами.

Реєстрація та аутентифікація користувачів:

- Використання Laravel для створення системи реєстрації та аутентифікації.

- Зберігання та керування обліковими записами користувачів.

Керування послугами:

- Реалізація CRUD-операцій для послуг (додавання, читання, оновлення, видалення).

- Використання Laravel Eloquent для роботи з базою даних та моделями.

Бронювання та управління записами:

- Реалізація системи онлайн-бронювання для клієнтів.

- Створення календаря для перегляду доступності та управління часом фахівців.

Зберігання та відображення відгуків:

- Створення механізму для додавання та відображення відгуків про послуги та фахівців.

Адміністративна панель:

- Розробка інтерфейсу для адміністратора з можливістю керування фахівцями, послугами та замовленнями.

На рисунках 4.10 – 4.11 представлено моделі та контролери які відповідають за отримання, обробку та відправку даних на відповідні сторінки.

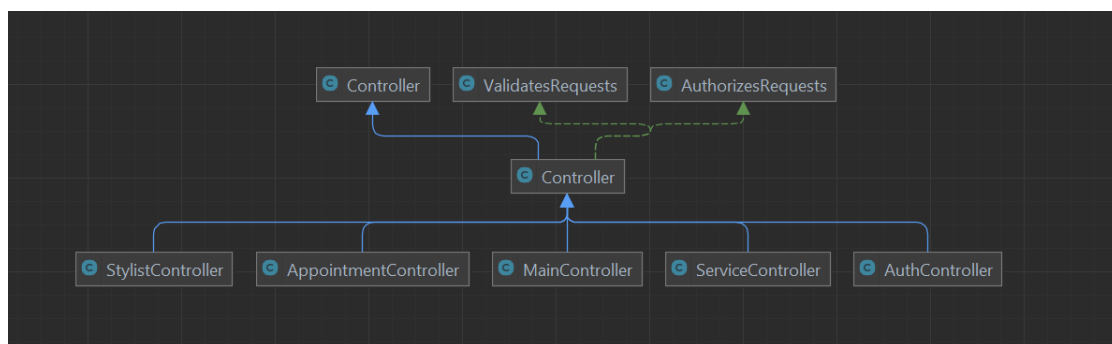


Рисунок 4.10 – Схема контролерів веб-сайту

Джерело: побудовано автором (знімок з екрану)

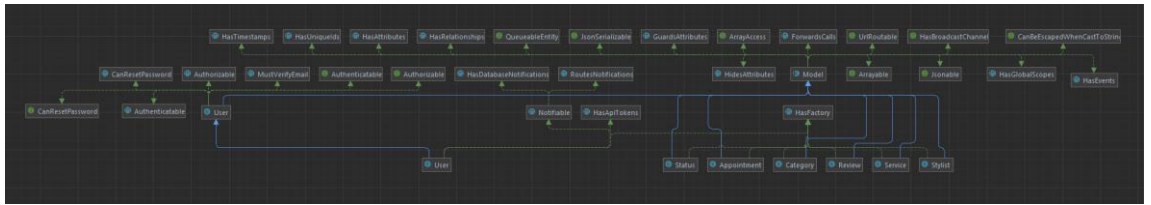


Рисунок 4.11 – Схема моделей веб-сайту

Джерело: побудовано автором (знімок з екрану)

Для керування даними з таблиць бази даних створено відповідні моделі:

Модель для таблиці замовлень представлено на рисунку 4.12. Дана модель відображує створення кожного поля таблиці та створення зв'язків між іншими таблицями.

```

14 usages
class Appointment extends Model
{
    use HasFactory;
    no usages
    protected $fillable = [
        "user_id",
        "service_id",
        "stylist_id",
        "status_id",
        "appointment_date",
    ];
    public function user(){
        return $this->belongsTo( related: User::class);
    }
    public function service(){
        return $this->belongsTo( related: Service::class);
    }
    public function stylist(){
        return $this->belongsTo( related: Stylist::class);
    }
    no usages
    public function status(){
        return $this->belongsTo( related: Status::class);
    }
}

```

Рисунок 4.12 – Модель для таблиці замовлень

Джерело: побудовано автором (знімок з екрану)

Модель для таблиці категорій представлено на рисунку 4.13.

```
8 usages
class Category extends Model
{
    use HasFactory;

    no usages
    public $timestamps = false;

    no usages
    protected $fillable = [
        'title',
        'photo_url',
    ];

    public function service(){
        return $this->hasMany(related: Service::class);
    }
}
```

Рисунок 4.13 – Модель для таблиці категорій

Джерело: побудовано автором (знімок з екрану)

Модель для таблиці відгуків представлено на рисунку 4.14.

```
4 usages
class Review extends Model
{
    use HasFactory;

    no usages
    protected $fillable = [
        "stylist_id",
        "user_id",
        "rating",
        "comment",
    ];

    public function user(){
        return $this->belongsTo(related: User::class);
    }

    public function stylist(){
        return $this->belongsTo(related: Stylist::class);
    }
}
```

Рисунок 4.14 – Модель для таблиці відгуків

Джерело: побудовано автором (знімок з екрану)

Модель для таблиці послуг представлено на рисунку 4.15.

```
13 usages
class Service extends Model
{
    use HasFactory;

    no usages
    protected $fillable = [
        "service_name",
        "description",
        "price",
        "photo_url",
        "category_id",
    ];

    public function category(){
        return $this->belongsTo( related: Category::class);
    }
    public function appointment(){
        return $this->hasMany( related: Appointment::class);
    }
    public function stylist(){
        return $this->hasMany( related: Stylist::class);
    }
}
```

Рисунок 4.15 – Модель для таблиці послуг

Джерело: побудовано автором (знімок з екрану)

Модель для таблиці статусів представлено на рисунку 4.16.

```
1 usage
class Status extends Model
{
    use HasFactory;

    no usages
    protected $fillable = [
        "title",
    ];

    public function appointment(){
        return $this->hasMany( related: Appointment::class);
    }
}
```

Рисунок 4.16 – Модель для таблиці статусів

Джерело: побудовано автором (знімок з екрану)

Модель для таблиці робітників представлено на рисунку 4.17.

```
class Stylist extends Model
{
    use HasFactory;

    no usages
    public $timestamps = false;

    no usages
    protected $fillable = [
        'first_name',
        'last_name',
        'photo_url',
        'bio',
        'service_id',
    ];

    public function service(){
        return $this->belongsTo(related: Service::class);
    }
    public function appointment(){
        return $this->hasMany(related: Appointment::class);
    }
    no usages
    public function review(){
        return $this->hasMany(related: Review::class);
    }
}
```

Рисунок 4.17 – Модель для таблиці робітників

Джерело: побудовано автором (знімок з екрану)

Модель для таблиці користувачів представлено на рисунку 4.18.

```
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /** The attributes that are mass assignable. ...*/
    no usages
    protected $fillable = [
        'first_name',
        'last_name',
        'email',
        'password',
    ];

    /** The attributes that should be hidden for serialization. ...*/
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /** The attributes that should be cast. ...*/
    no usages
    protected $casts = [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];

    public function appointment(){
        return $this->hasMany(related: Appointment::class)->orderBy( column: 'created_at');
    }
}
```

Рисунок 4.18 – Модель для таблиці користувачів

Джерело: побудовано автором (знімок з екрану)

Для правильного відображення та обробки запитів потрібно створити правильну маршрутизацію яка буде отримувати та перенаправляти дані у контролери. Маршрутизація веб-додатку представлена на рисунках 4.19-4.23.

```
// Main page route
Route::get( uri: '/', action: 'App\Http\Controllers\MainController@index' )->name( name: 'home');

// admin route
Route::get( uri: '/auth', function (){
    return view( view: 'auth.auth');
} );
Route::get( uri: '/admin', action: 'App\Http\Controllers\AuthController@admin' )->name( name: 'admin');
```

Рисунок 4.19 – Маршрутизація для головної сторінки та форми авторизації

Джерело: побудовано автором (знімок з екрану)

```
// auth routes
Route::get( uri: '/login', action: 'App\Http\Controllers\AuthController@showLoginForm' )->name( name: 'login' );
Route::post( uri: '/login_process', action: 'App\Http\Controllers\AuthController@login' )->name( name: 'login_process' );

Route::get( uri: '/logout', action: 'App\Http\Controllers\AuthController@logout' )->name( name: 'logout' );
Route::get( uri: '/profile', action: 'App\Http\Controllers\AuthController@profile' )->name( name: 'profile' );

Route::get( uri: '/register', action: 'App\Http\Controllers\AuthController@showRegisterForm' )->name( name: 'register' );
Route::post( uri: '/register_process', action: 'App\Http\Controllers\AuthController@register' )->name( name: 'register_process' );
```

Рисунок 4.20 – Маршрутизація авторизації, реєстрації та профілю

Джерело: побудовано автором (знімок з екрану)

```
// routes for service page
Route::get( uri: '/services', action: 'App\Http\Controllers\ServiceController@index' )->name( name: 'services.index' );
Route::get( uri: '/services/create', action: 'App\Http\Controllers\ServiceController@create' )->name( name: 'service.create' );
Route::post( uri: '/services', action: 'App\Http\Controllers\ServiceController@store' )->name( name: 'services.store' );
Route::get( uri: '/services/{service}', action: 'App\Http\Controllers\ServiceController@show' )->name( name: 'service.show' );
Route::get( uri: '/services/{service}/edit', action: 'App\Http\Controllers\ServiceController@edit' )->name( name: 'service.edit' );
Route::patch( uri: '/services/{service}', action: 'App\Http\Controllers\ServiceController@update' )->name( name: 'service.update' );
Route::delete( uri: '/services/{service}', action: 'App\Http\Controllers\ServiceController@destroy' )->name( name: 'service.delete' );
```

Рисунок 4.21 – Маршрутизація для таблиці послуг

Джерело: побудовано автором (знімок з екрану)

```
// routes for stylist page
Route::get( uri: '/stylists', action: 'App\Http\Controllers\StylistController@index' )->name( name: 'stylists.index' );
Route::get( uri: '/stylists/create', action: 'App\Http\Controllers\StylistController@create' )->name( name: 'stylist.create' );

Route::post( uri: '/stylists', action: 'App\Http\Controllers\StylistController@store' )->name( name: 'stylists.store' );
Route::get( uri: '/stylists/{stylist}', action: 'App\Http\Controllers\StylistController@show' )->name( name: 'stylist.show' );
Route::get( uri: '/stylists/{stylist}/edit', action: 'App\Http\Controllers\StylistController@edit' )->name( name: 'stylist.edit' );
Route::patch( uri: '/stylists/{stylist}', action: 'App\Http\Controllers\StylistController@update' )->name( name: 'stylist.update' );
Route::delete( uri: '/stylists/{stylist}', action: 'App\Http\Controllers\StylistController@destroy' )->name( name: 'stylist.delete' );
```

Рисунок 4.22 – Маршрутизація для таблиці робітників

Джерело: побудовано автором (знімок з екрану)

```
// routes for appointments page
Route::get( uri: '/appointments', action: 'App\Http\Controllers\AppointmentController@index' )->name( name: 'appointments.index' );
Route::get( uri: '/appointments/create', action: 'App\Http\Controllers\AppointmentController@create' )->name( name: 'appointment.create' );

Route::post( uri: '/appointments', action: 'App\Http\Controllers\AppointmentController@store' )->name( name: 'appointments.store' );
Route::get( uri: '/appointments/{appointment}', action: 'App\Http\Controllers\AppointmentController@show' )->name( name: 'appointment.show' );
Route::get( uri: '/appointments/{appointment}/edit', action: 'App\Http\Controllers\AppointmentController@edit' )->name( name: 'appointment.edit' );
Route::patch( uri: '/appointments/{appointment}', action: 'App\Http\Controllers\AppointmentController@update' )->name( name: 'appointment.update' );
Route::delete( uri: '/appointments/{appointment}', action: 'App\Http\Controllers\AppointmentController@destroy' )->name( name: 'appointment.delete' );
```

Рисунок 4.23 – Маршрутизація для таблиці замовлень

Джерело: побудовано автором (знімок з екрану)

4.3. Інтеграція з базою даних (MySQL)

Для інтеграції бази даних у Laravel додаток створюються міграції для кожної таблиці.

Міграції для таблиць представлено на рисунках 4.24 – 4.30.

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create( table: 'users', function (Blueprint $table) {
            $table->id();
            $table->string( column: 'first_name');
            $table->string( column: 'last_name');
            $table->string( column: 'email')->unique();
            $table->string( column: 'password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists( table: 'users');
    }
};
```

Рисунок 4.24 – Код міграції для таблиці користувачів

Джерело: побудовано автором (знімок з екрану)


```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create( table: 'categories', function (Blueprint $table) {
            $table->id();
            $table->string( column: 'title');
            $table->string( column: 'photo_url');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists( table: 'categories');
    }
};

```

Рисунок 4.25 – Код міграції для таблиці категорій
Джерело: побудовано автором (знімок з екрану)

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create( table: 'statuses', function (Blueprint $table) {
            $table->id();
            $table->string( column: 'title');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists( table: 'statuses');
    }
};

```

Рисунок 4.26 – Код міграцій для таблиці статусів
Джерело: побудовано автором (знімок з екрану)

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create( table: 'services', function (Blueprint $table) {
            $table->id();
            $table->string( column: 'service_name');
            $table->text( column: 'description');
            $table->integer( column: 'price');
            $table->string( column: 'photo_url');
            $table->foreignId( column: 'category_id')->references( column: 'id')->on( table: 'categories')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists( table: 'services');
    }
};
```

Рисунок 4.27 – Код міграції для таблиці послуг
Джерело: побудовано автором (знімок з екрану)

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create( table: 'stylists', function (Blueprint $table) {
            $table->id();
            $table->string( column: 'first_name');
            $table->string( column: 'last_name');
            $table->string( column: 'photo_url');
            $table->text( column: 'bio');
            $table->foreignId( column: 'service_id')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists( table: 'stylists');
    }
};
```

Рисунок 4.28 – Код міграцій для таблиці робітників
Джерело: побудовано автором (знімок з екрану)

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create( table: 'appointments', function (Blueprint $table) {
            $table->id();
            $table->foreignId( column: 'user_id')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
            $table->foreignId( column: 'service_id')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
            $table->foreignId( column: 'stylist_id')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
            $table->foreignId( column: 'status_id')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists( table: 'appointments');
    }
};

```

Рисунок 4.29 – Код міграцій для таблиці замовлень

Джерело: побудовано автором (знімок з екрану)

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create( table: 'reviews', function (Blueprint $table) {
            $table->id();
            $table->foreignId( column: 'stylist_id')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
            $table->foreignId( column: 'user_id')->constrained()->cascadeOnDelete()->cascadeOnUpdate();
            $table->integer( column: 'rating');
            $table->text( column: 'comment');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists( table: 'reviews');
    }
};

```

Рисунок 4.30 – Код міграцій для таблиці відгуків

Джерело: побудовано автором (знімок з екрану)

4.4 Результат розробки

В результаті розробки програми, було створено наступні сторінки:

- головна;
- сторінка з послугами;
- сторінка з інформацією про працівників;
- сторінку з інформацією про певного працівника;
- сторінку оформлення замовлення;
- сторінку профілю;
- сторінку авторизації користувача;
- сторінку авторизації адміністратора;
- сторінку реєстрації користувача;
- адміністративну панель.

Кожен веб-сайт розроблюється по різному, але у більшості випадків різні елементи сайту створюються як окремі компоненти, наприклад хедер та футер сайту. Компоненти сайту салону краси представлено на рисунках 4.31 – 4.32.

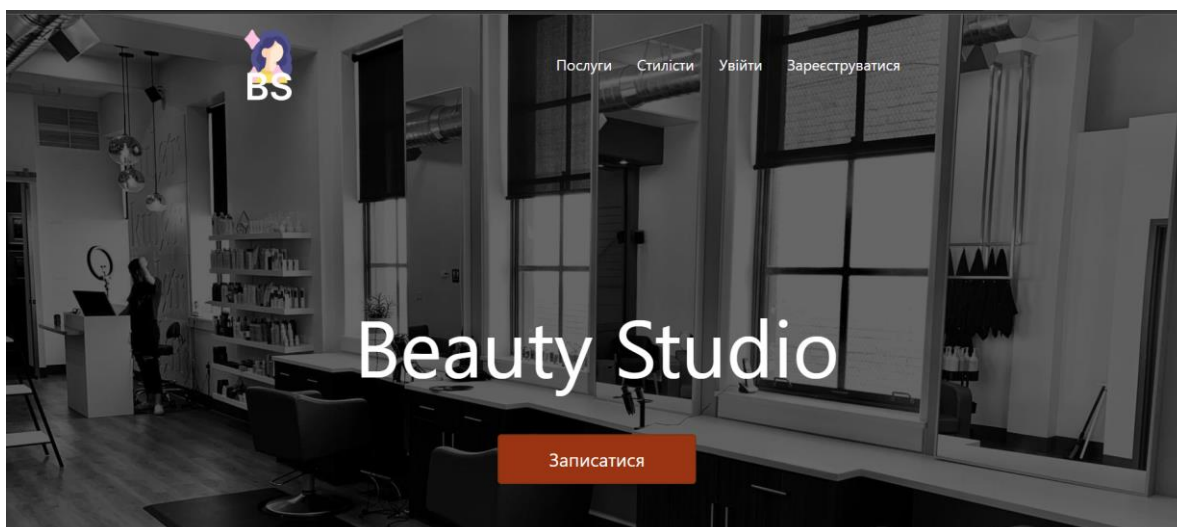


Рисунок 4.31 – Хедер сайту салону краси

Джерело: побудовано автором (знімок з екрану)

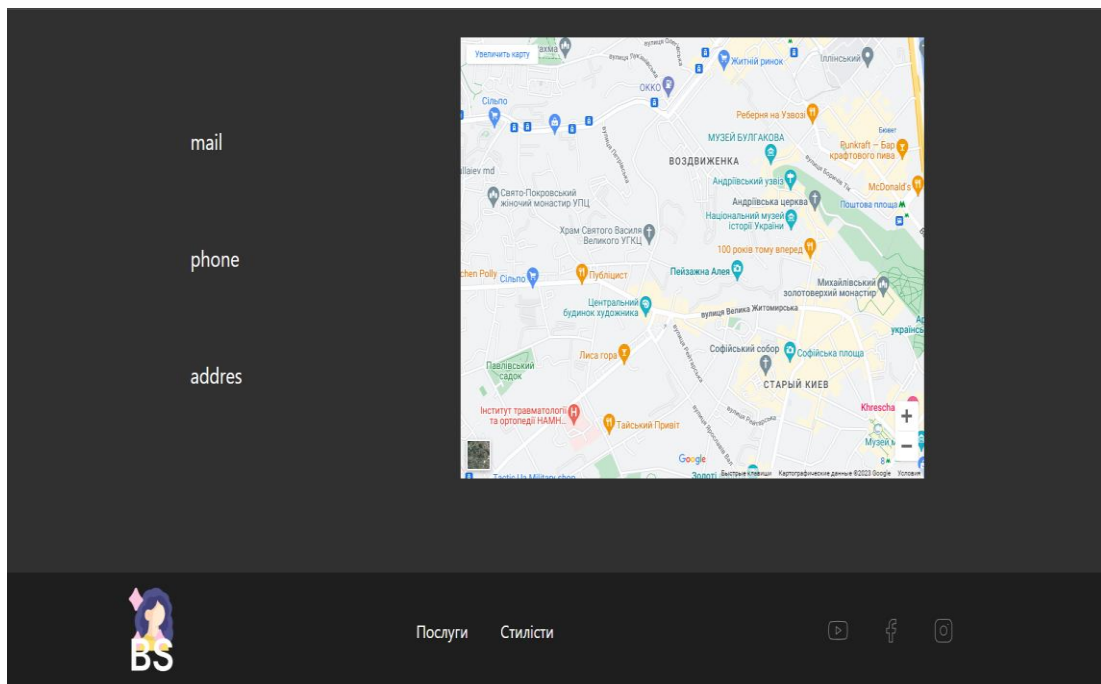
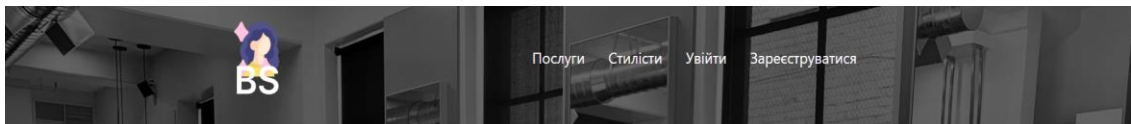


Рисунок 4.32 – Футер сайту салону краси

Джерело: побудовано автором (знімок з екрану)

На рисунках 4.33 – 4.40 представлено відображення розроблених сторінок.



Послуги які надає наш заклад

- All categories
- Парикмахерські послуги
- Нігтьовий сервіс
- SPA-процедури
- Віаж
- Масаж
- Чоловічі послуги
- Депіляція
- Фізіотерапія
- Вечірні зачіски
- Сонячні процедури

Стрижка
500

Професійна стрижка волосся за допомогою сучасних технік та стилів, яка підкреслить вашу індивідуальність. Для чоловіків і жінок.

[Записатися →](#)

Манікюр
400

Догляд за нігтями, включаючи обрізання, формування та лакування, щоб ваші руки виглядали доглянутими та стильними.

[Записатися →](#)

Педикюр
450

Догляд за нігтями та ступнями, включаючи обрізання, формування, пілінг і лакування, для красивих і доглянутих ніг.

[Записатися →](#)

Фарбування
600

Змініть свій вигляд за допомогою фарбування волосся, обираючи відтінок, який вам подобається.

[Записатися →](#)

Масаж обличчя
350

Розслаблюючий масаж обличчя, який допоможе вам відчути себе свіжими і відпочиним.

[Записатися →](#)

Макіяж
450

Професійний макіяж для важливих заходів або спеціальних подій, щоб ви виглядали вражаюче.

[Записатися →](#)

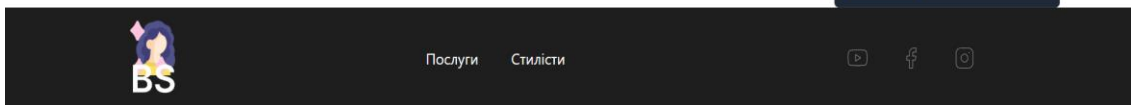
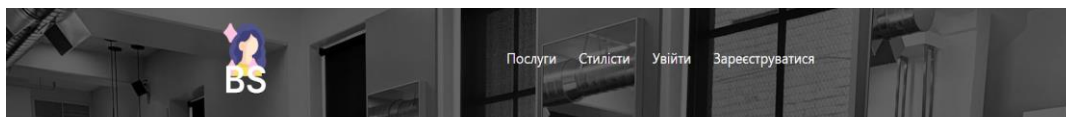




Рисунок 4.33 – Створення сторінки послуг
Джерело: побудовано автором (знімок з екрану)



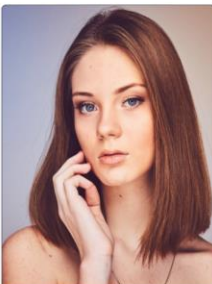
Наші майстри



Анна Петрова
Спеціалізація: Стрижка
[Детальніше →](#)



Марія Сидорова
Спеціалізація: Манікюр
[Детальніше →](#)



Ольга Іванова
Спеціалізація: Стрижка
[Детальніше →](#)



Наталія Ковальчук
Спеціалізація: Педикюр
[Детальніше →](#)



Катерина Григоренко
Спеціалізація: Фарбування
[Детальніше →](#)

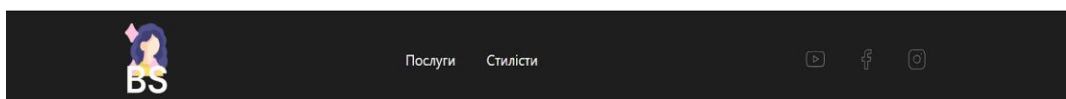


Рисунок 4.34 – Створена сторінка працівників
Джерело: побудовано автором (знімок з екрану)

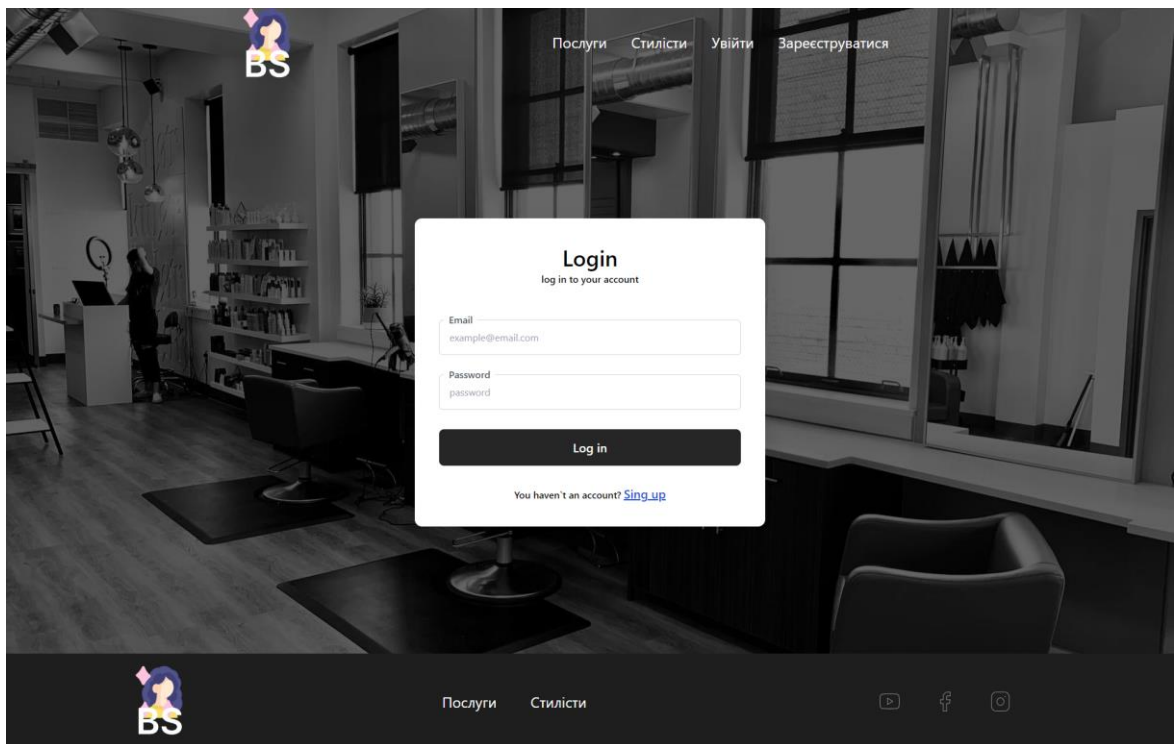


Рисунок 4.35 – Розроблена форма авторизації користувачів

Джерело: побудовано автором (знімок з екрану)

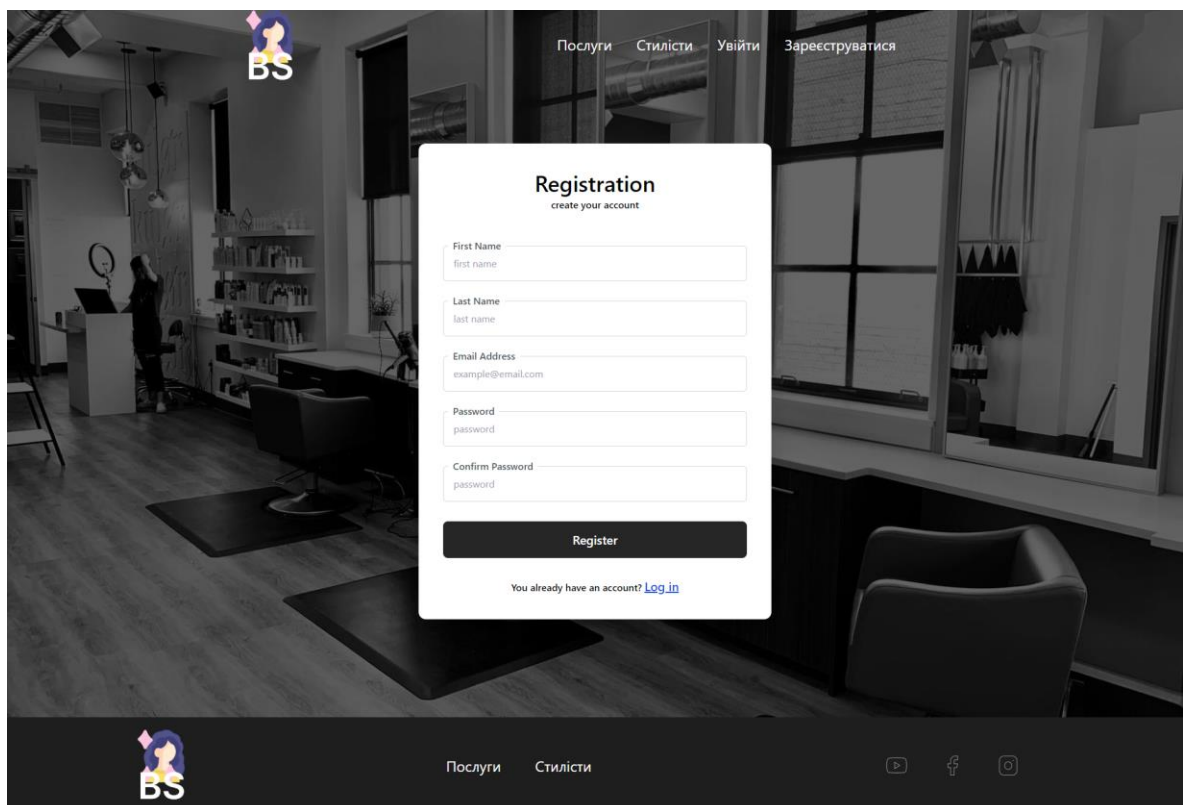


Рисунок 4.36 – Розроблена форма реєстрації користувачів

Джерело: побудовано автором (знімок з екрану)

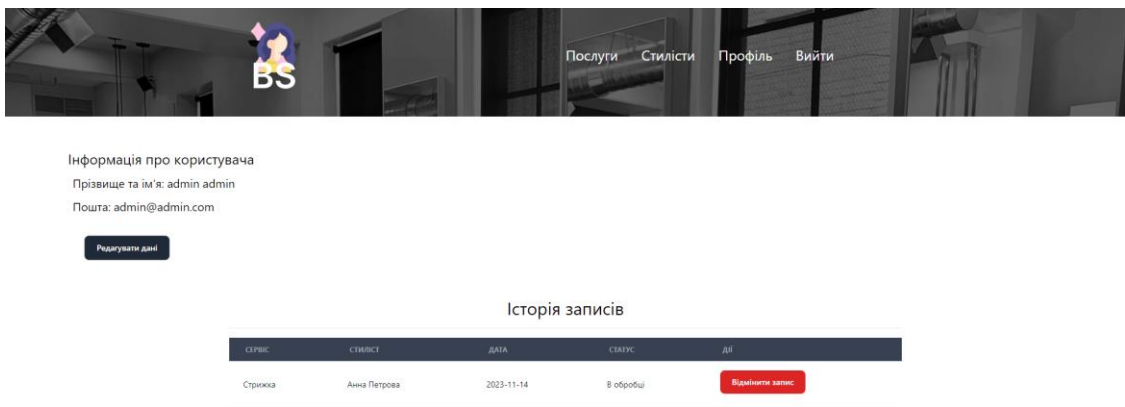


Рисунок 4.37 – Розроблена сторінка профілю користувача з таблицею історії заказів

Джерело: побудовано автором (знімок з екрану)

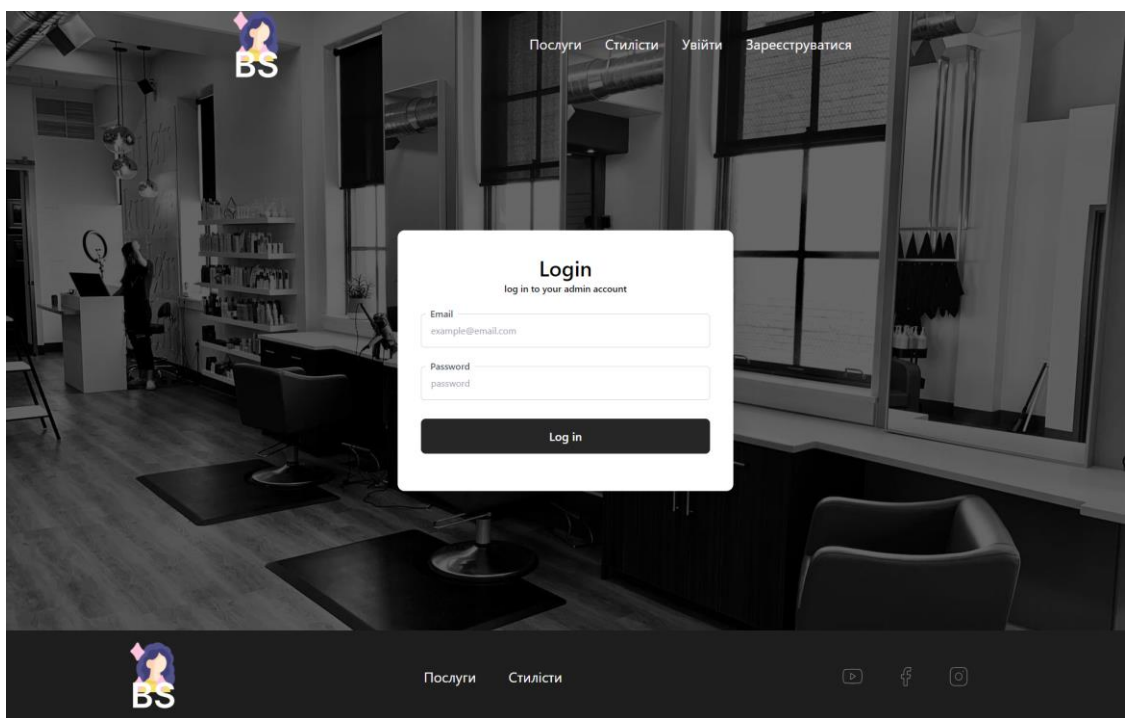
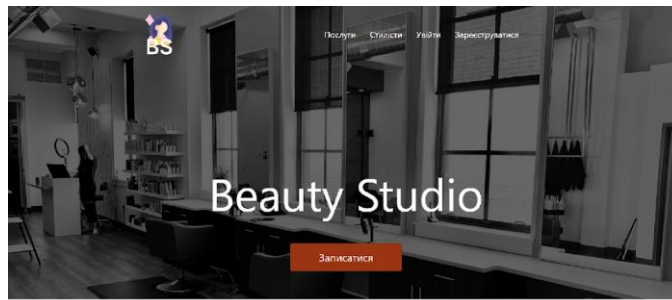


Рисунок 4.38 – Форма авторизації адміністраторів

Джерело: побудовано автором (знімок з екрану)



О компанії

Про салон краси

Усемак просило в наш салон краси, де ваша краса і добробут є нашими головними пріоритетами. Наш салон пропонує розкішні послуги і професійний догляд, які допоможуть вам відчувати себе особливою і розквітаючою.

Наша місія

Ми віримо, що кожна людина заслуговує на найвищий та догляд, і ми прагнемо зробити ваше перебування в нашому салоні найприємливішим і найефективнішим досвідом. Наша місія - підкреслити вашу природну красу та надати вам відчуття впевненості.

Наші послуги

У нашому салоні ви знайдете широчайший вибір послуг: від стрижки і чистки до масажу і масажу обличчя. Наші досвідчені стилісти та майстри краси завжди готові надати вам найкращий результат. Ми працюємо з високоякісними продуктами і інноваційними методами, щоб забезпечити вам найкращий догляд.

Наша команда

Наша команда складається з професіоналів, які мають багаторічний досвід у сфері краси та стилю. Ми завжди відокремлюємо свої навички і стежимо за останніми тенденціями в галузі краси, щоб надавати нашим клієнтам те, що вони заслуговують - найкращий сервіс і результат.

Наше обладнання та середовище

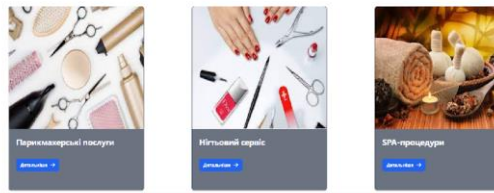
Наш салон обладнаний сучасними засобами та устаткуванням, яке допомагає нам досягти відмінних результатів. Ми також прагнемо велику увагу зберігати природу і екологічності асвтан нашій роботі. Ми використовуємо екологічно чисті продукти та практикуємо відходи.

Запрошуємо до нашого салону

Ми завжди раді бачити нових клієнтів і вельми вас вітаємо в наш салон краси. Незалежно від того, чи ви шукаєте підтримку, масаж обличчя, маніж або інші послуги, ми готові надати вам вищий стандарт обслуговування. Ви заслуговуєте на розкіш і увагу, і ми гарантуємо, що ви залишитеся задоволені результатом.

Запишіться з нами сьогодні, щоб зробити запис або дізнатися більше про наші послуги. Ми завжди готові допомогти вам виглядати найкраще!

Наші послуги



Наші працівники

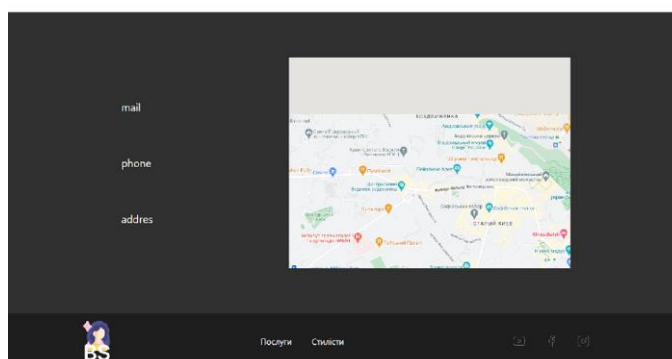
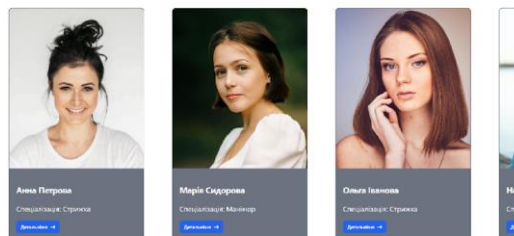
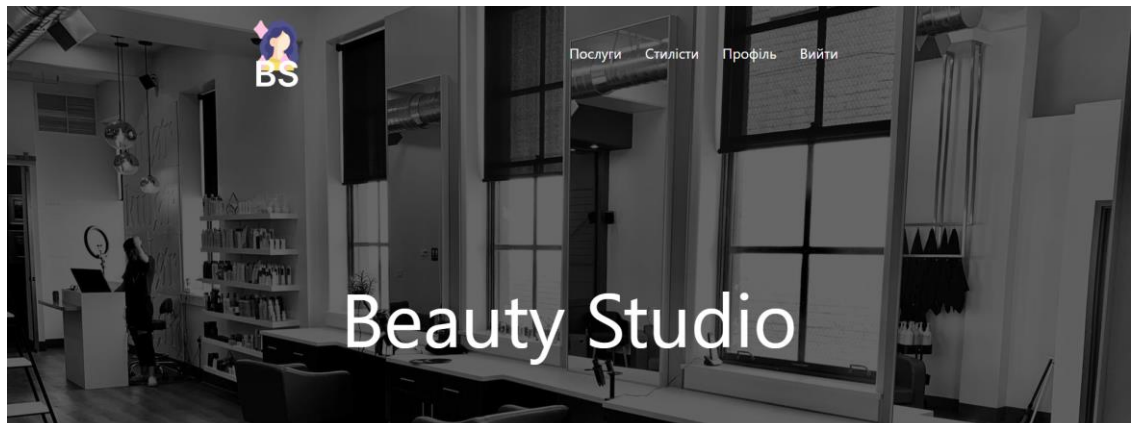


Рисунок 4.39 – Відображення головної сторінки сайту

Джерело: побудовано автором (знімок з екрану)



Запис на послугу

Оберіть послугу:
Стрижка

Оберіть стиліста:
Анна Петрова -> Стрижка

Оберіть дату сеансу:
ДД.ММ.РРРР

[Створити запис](#)

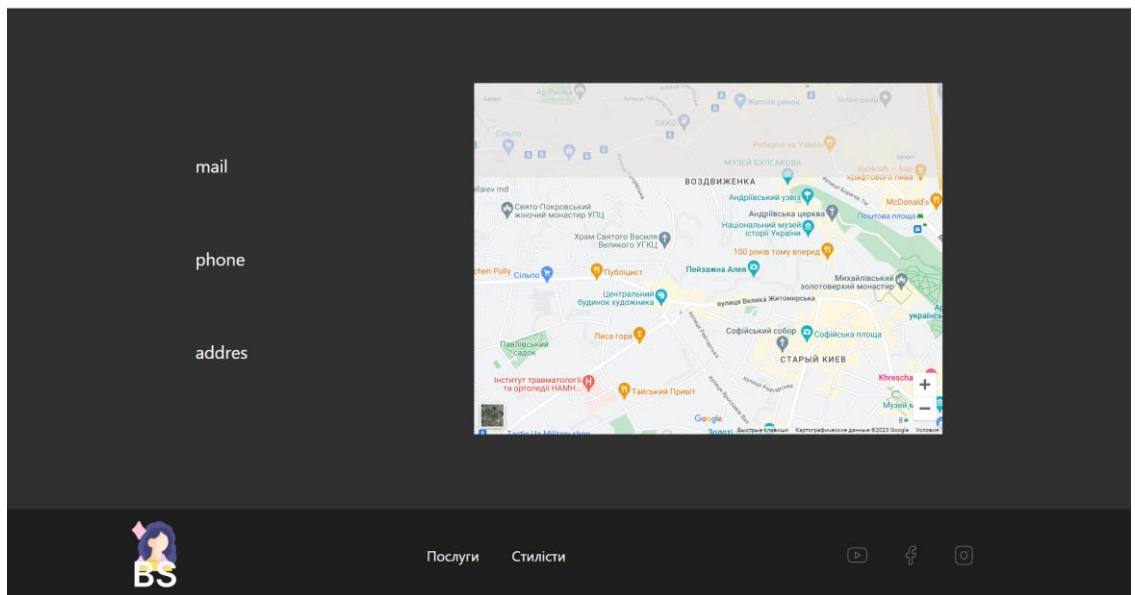


Рисунок 4.40 – Відображення сторінки запису
Джерело: побудовано автором (знімок з екрану)

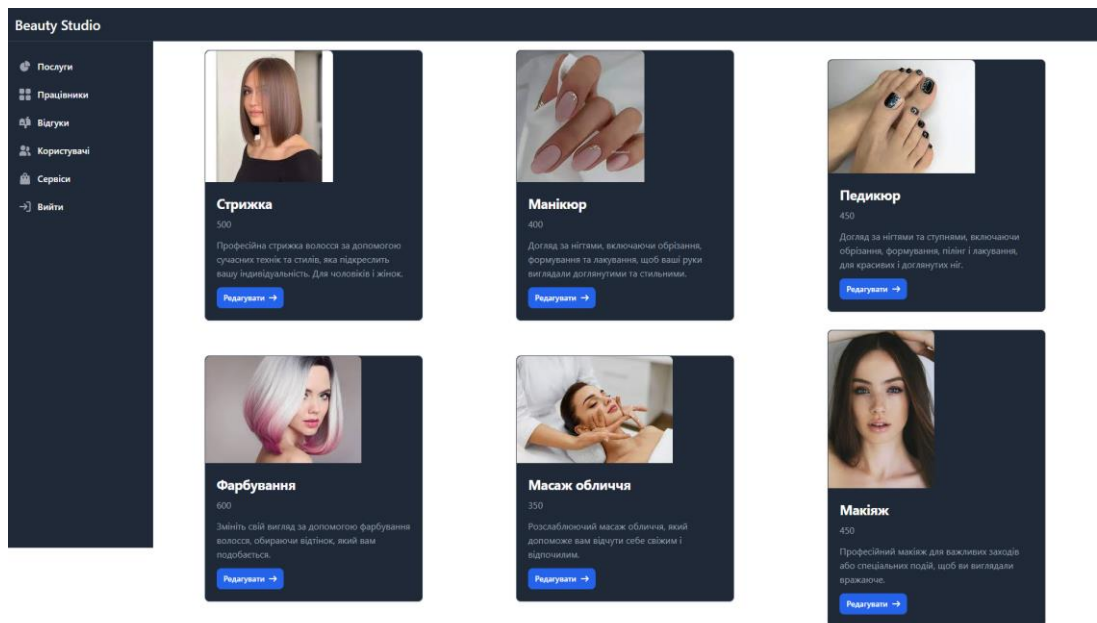


Рисунок 4.41 – Відображення адмін сторінки сервісів

Джерело: побудовано автором (знімок з екрану)

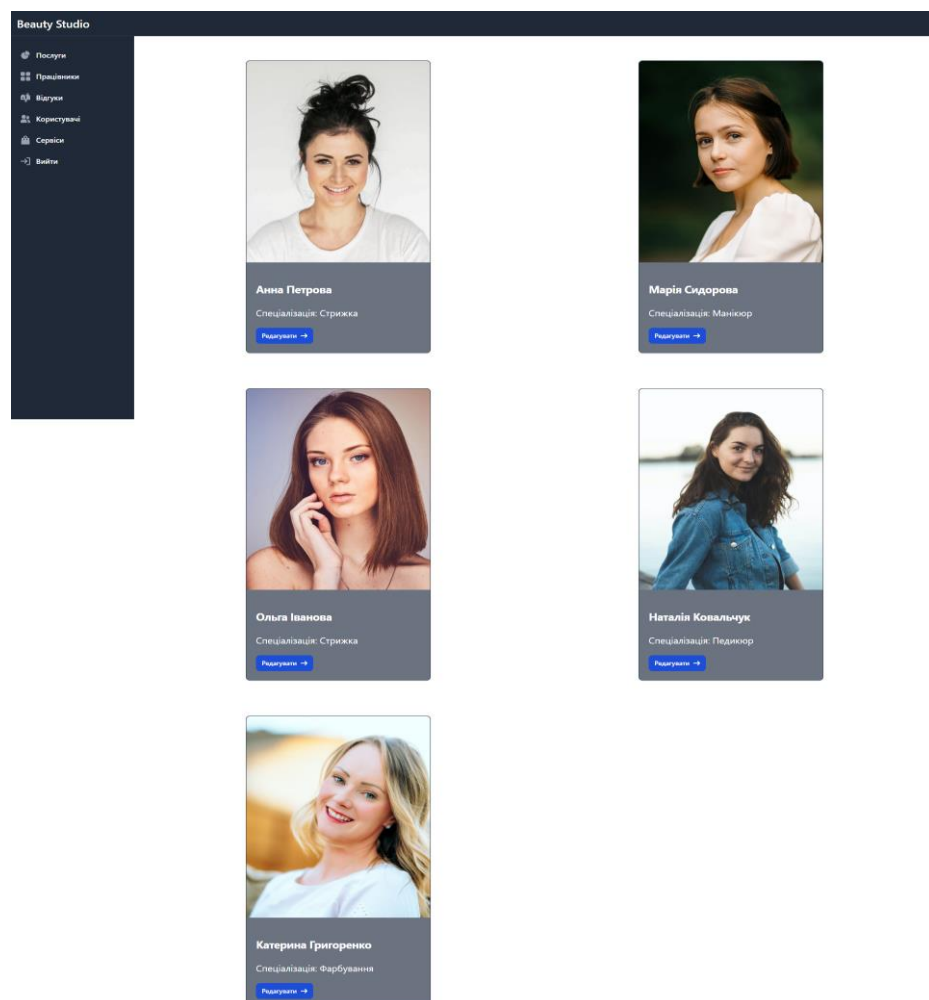


Рисунок 4.42 – Відображення адмін сторінки працівників

Джерело: побудовано автором (знімок з екрану)

4.5 Тестування web-орієнтованої системи

Після реалізації web-орієнтованої системи було проведено функціональне тестування, де перевірено роботу головних функції сайту. Були протестовані кнопки й посилання, а саме переходи на соціальні сторінки салону, а також перевірено на коректність роботи меню. Не виявлено помилок при записі на послугу з обрання часу та спеціаліста та відміні цього запису та протестована функція створення відгуку. При спробі авторизуватися та реєструватися не виникало проблем. Перевірено правильність виконання запитів, а саме занесення, вилучення та оновлення даних.

Також відбулася перевірка контенту: переглянута граматика та орфографія, розміщення заголовків, відповідність фотографій. Провівся перегляд на наявність зайвих елементів.

Після функціонального тестування було проведено нефункціональне тестування, а саме перевірка сайту на кросбраузерність. Web-орієнтована система правильно функціонує та відображається в усіх популярних браузерах: Google Chrome, Firefox, Internet Explorer і т.д.

ВИСНОВКИ

У ході виконання дипломного проекту з теми "Web-орієнтована система підтримки діяльності салону краси" були досягнуті значущі результати, які визначаються вирішенням поставлених завдань та розв'язаною проблемою. Головна мета дослідження полягала у впровадженні сучасних технологій та розробці системи, яка оптимізує управління салоном краси та підвищує якість обслуговування клієнтів.

Результати дослідження дозволили чітко визначити проблеми традиційного управління салоном краси, такі як ускладненість процесів бронювання, обліку послуг та взаємодії з клієнтами. Розроблена система інтегрує онлайн-календар бронювання, зручний інтерфейс для управління базами даних клієнтів та аналітичні інструменти для ефективного аналізу попиту та маркетингових кампаній.

Важливим аспектом є позитивний вплив системи на користувацький досвід. Зменшення часу, необхідного для резервації послуг, та забезпечення гнучкості у виборі часів та послуг допомогли підвищити задоволення клієнтів від візитів у салон.

Дослідження також вказало на необхідність постійного вдосконалення та розширення функціоналу системи. У висновках рекомендується активно працювати над додатковими інструментами для аналізу попиту, впровадження новітніх маркетингових стратегій та забезпечення безпеки та конфіденційності даних клієнтів.

Отримані результати є якісним та кількісним підтвердженням ефективності розробленої системи для салонів краси та свідчать про потенційний позитивний вплив на практику управління в даній галузі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kaur G. Css: What Is CSS? Independently Published, 2022.
2. Vickler A. Javascript: Javascript Basics for Beginners. Independently Published, 2021.
3. Queinnec C. Javascript. Technologies logicielles Architectures des systèmes. 2017. URL: <https://doi.org/10.51257/a-v2-h3120> (дата звернення: 10.10.2023).
4. Welling L., Thomson L. PHP and MySQL Web development, 2003.
5. Берко А. Ю., Верес О. М. Організація баз даних: практичний курс: Навч. посіб. для студ. Нац. ун-т «Львів. політехніка». Л., 2003. 149 с.
6. Відео урок про фреймворк Laravel. URL: <https://www.youtube.com/watch?v=oKRC8b2ZEn8> (дата звернення: 18.11.2023).
7. Вельбицький, І. В. Технологія програмування. Київ: Техніка, 1984.
8. Гайна Г. А. Основи проектування баз даних : навч. посіб. для студ. вищ. навч. закл., що навчаються за напрямом підгот. 0804 "Комп'ютерні науки". Київ : Кондор, 2008. 200 с.
9. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. Х: ХНАМГ, 2010. 222 с.
10. Документація бази даних MS SQL Server [Електронний ресурс] – Режим доступу до ресурсу: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019>
11. Документація бази даних MySQL. URL: <https://dev.mysql.com/doc/> (дата звернення: 05.11.2023).
12. Документація веб сервісу phpMyAdmin. URL: <https://docs.phpmyadmin.net/uk/latest/> (дата звернення: 01.11.2023).
13. Документація фреймворку Laravel. URL: <https://laravel.com/docs/10.x/documentation> (дата звернення: 09.11.2023).

14. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем. Навчальний посібник : ХНУМГ ім. О. М., 2017.

15. Ковалюк Т.В. Основи програмування, 2005.

16. Корисна інформація про бази даних. URL: <https://wiki.cuspu.edu.ua/index.php> (дата звернення: 01.10.2023).

17. Корисна інформація про бази даних. URL: <https://apeps.kpi.ua/shco-take-basa-danykh> (дата звернення: 03.10.2023).

18. Корисна інформація про бази даних. URL: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення: 04.10.2023).

19. Корисна інформація про бази даних. URL: <https://www.ukraine.com.ua/blog/programming/sql-baza-dannih-dlya-chego-prednaznachena-baza-dannih.html> (дата звернення: 06.10.2023).

20. Корисна інформація про фреймворк Laravel. URL: <https://probotiuk.com/blog/sho-take-laravel> (дата звернення: 09.10.2023).

21. Корисна інформація про мову програмування php. URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-php/> (дата звернення: 11.10.2023).

22. Корисна інформація про мову програмування php. URL: http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/43_PHP/index.html (дата звернення: 13.10.2023).

23. Куленко М. Я. Основи графічного дизайну: підручник для студентів вищих навч. закладів / Михайло Куленко; МОНУ; Київський нац. ун-т будівництва і архітектури. – 2-ге вид., виправл. та доп. – Київ: Кондор, 2007. – 492 с.

24. Пасічник О. В. Веб-дизайн: підруч. для студ. вищ. навч. закл. М-во освіти і науки України, Л. : Магнолія 2006, 2010, 519 с.

25. Рибак А.І. Системні методи розробки і оцінки проектів. Збірник наукових праць, 2003.

26. Сайт салону краси Backstage. URL: <https://backstage.ua/> (дата звернення: 26.09.2023).

27. Сайт салону краси Wizard. URL: <https://wizard-salon.com.ua/uk/> (дата звернення: 28.09.2023).

28. Сайт салону краси Mira. URL: <https://mirasalon.com.ua/> (дата звернення: 29.09.2023).

29. Трофименко О. Г. Організація баз даних : навч. посібник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. 2-ге вид. виправ. і доповн. – Одеса : Фенікс, 2019. – 246 с.

30. Цеслів О. В. WEB-програмування : навч. посібник / О. В. Цеслів; М-во освіти і науки, молоді та спорту України, Нац. техн. ун-т України “Київ. політехн. ін-т”. – Київ: НТУУ “КПІ”, 2011. – 296, с.

ДОДАТОК А

Планування робіт

А.1 Ідентифікація мети ІТ-проекту

Мета ІТ-проекту полягає в розробці та впровадженні Web-орієнтованої системи підтримки діяльності салону краси з метою оптимізації управління процесами, поліпшення обслуговування клієнтів та підвищення конкурентоспроможності на ринку. Головні завдання включають в себе створення зручного та ефективного інструменту для бронювання послуг, обліку клієнтів, а також інтеграцію з мережами соціальних мереж та іншими інструментами маркетингу.

Таблиця А.1 – Результат деталізації методом SMART

Джерело: побудовано автором

Specific (конкретна)	Створити Web-орієнтовану систему підтримки діяльності салону краси
Measurable (вимірювана)	Розробити Web-орієнтовану систему за 3 місяці, користуючись мінімальною кількістю ресурсів
Achievable (досяжна, узгоджена)	Для створення Web-орієнтованої системи потрібні знання мов HTML, CSS, PHP, JavaScript, баз даних MySQL
Relevant (реалістична)	Створена Web-орієнтована система допоможе переглядати послуги салону та робити запис
Time-framed (обмежена в часі)	Розробка Web-орієнтованої системи має часове обмеження

А.2 Планування змісту структури робіт ІТ-проекту

Структура робіт ІТ-проекту буде організована наступним чином:

1. Аналіз предметної області:
 - вивчення особливостей сучасних салонів краси;
 - визначення вимог та потреб користувачів.
2. Розробка технічного завдання:
 - визначення функціональності системи;
 - встановлення технічних вимог до програмної частини.
3. Проектування системи:
 - розробка структурно-функціональної моделі;
 - розробка бази даних та моделей даних;
 - визначення інтерфейсу користувача.
4. Реалізація системи:
 - розробка back-end частини з використанням Laravel;
 - розробка front-end частини з використанням HTML, CSS, JavaScript;
 - тестування та виправлення помилок.

На рисунку А.2 представлено організаційну структуру планування проекту.

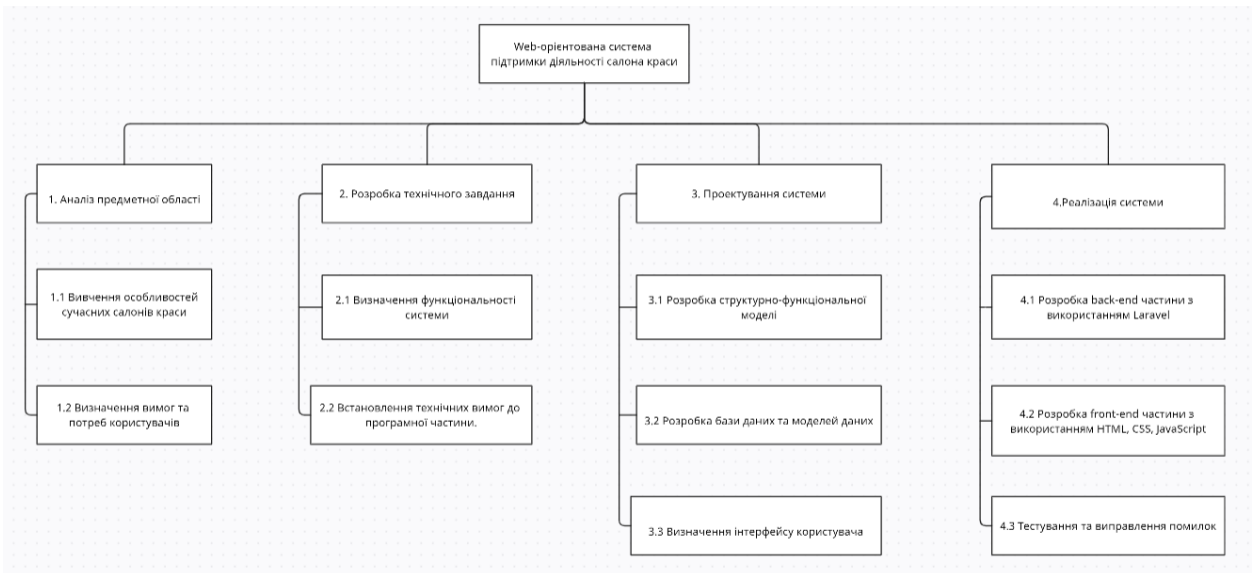


Рисунок А.1 – WBS-структура робіт проекту

Джерело: побудовано автором

Список виконавців, що функціонують представлено на рисунку А.2.

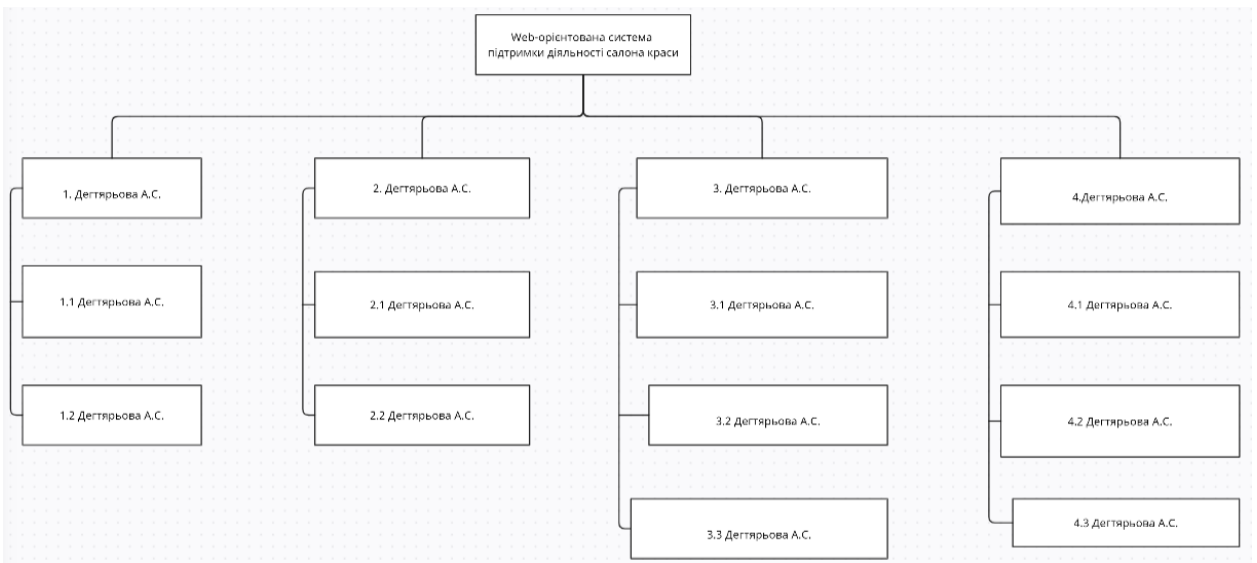


Рисунок А.2 – OBS-структура робіт проекту

Джерело: побудовано автором

А.3 Побудова календарного графіку виконання кваліфікаційної роботи

Календарний графік проходження кваліфікаційної роботи розпочинається з етапу аналізу предметної області. Цей етап триватиме приблизно 7 дні, починаючи з 25 вересня і закінчуючи 2 жовтня.

Далі настає етап постановки задачі дослідження, розпочнеться в жовтні та триватиме приблизно 8 днів.

Планування робіт з розроблення програмного продукту, що є практичним результатом виконання кваліфікаційної роботи магістра заплановано на період з 10 жовтня по 16 жовтня.

Проектування, моделювання об'єкту дослідження триватиме приблизно 1 тиждень.

Практична реалізація Web-орієнтованої системи починається 24 жовтня та закінчується 1 грудня.

Фінальний етап – написання розділів пояснювальної записки та надання керівнику на перевірку.

Календарний графік проходження кваліфікаційної роботи представлено на рисунку А.3.

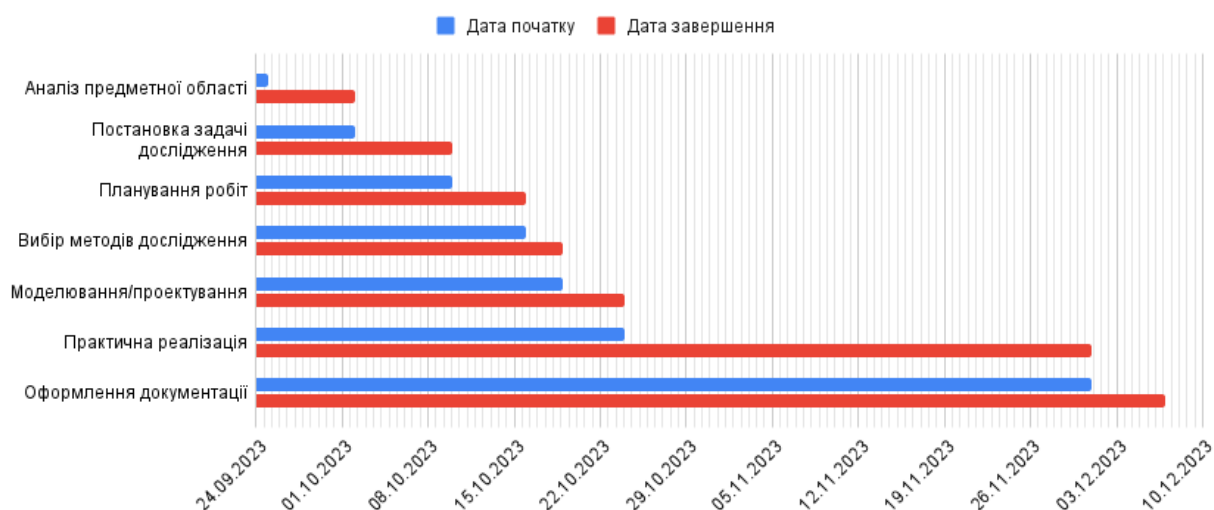


Рисунок А.3 – Календарний графік

Джерело: побудовано автором

А.4 Планування ризиків проекту

В ході розробки плану ризиків для ІТ-проекту "Web-орієнтована система підтримки діяльності салону краси" ідентифікуються та оцінюються можливі загрози та приймаються стратегії управління ними.

Таблиця А.4 – Ризики проекту

Джерело: побудовано автором

Ризик	Назва ризику	Опис ризику
1	Зміни вимог	Внесення змін у вимоги може виникнути внаслідок непорозумінь або зміни потреб замовника.
2	Технічні труднощі	Розробка може зіткнутися з технічними проблемами або несподіваними труднощами, зокрема пов'язаними із вибором технологій чи інтеграцією.
3	Затримка в розробці	Затримка через непередбачені обставини або низьку продуктивність виконавця.
4	Проблема з програмним забезпеченням	Несправність у вигляді порушення роботи програмного забезпечення.
5	Відсутність резервних копій	Невиконання процесу резервного копіювання. Відсутність збереження резервних копій на різних носіях.
6	Виконання додаткових функцій	Реалізація додаткових непотрібних функцій.
7	Людський фактор	Зниження продуктивності через неуважність.
8	Зміна терміну виконання робіт	Зміна календарного плану виконання робіт.
9	Недостатня кваліфікація виконавця	Брак досвіду та знань, що впливає на продуктивність роботи

Таблиця А.5 – Реєстрація ризиків

Джерело: побудовано автором

Ризик	Вірогідність виникнення	Ступінь впливу	Значення ризику
1	3	3	8
2	2	2	5
3	2	2	4
4	1	3	8
5	1	2	6
6	1	1	4
7	2	2	3
8	2	1	5
9	2	1	4

Таблиця А.6 – Матриця впливу

Джерело: побудовано автором

Вірогідність виникнення	Матриця впливу		
3			1
2	8, 9	2, 3, 7	
1	6	5	4
Ступінь впливу	1	2	3

Таблиця А.7 – Матриця впливу

Джерело: побудовано автором

Ризик	Назва ризику	Реакція на ризик
1	Зміни вимог	Узгоджувати кожний етап із замовником.

Продовження таблиці А.7

2	Технічні труднощі	Проаналізувати технології перед реалізацією проекту.
3	Затримка в розробці	Створити календарний план та дотримуватись його.
4	Проблема з програмним забезпеченням	Налаштувати програмне забезпечення, в іншому випадку замінити його.
5	Відсутність резервних копій	Налаштувати автоматичне копіювання. Зберігати резервні копії на носіях.
6	Виконання додаткових функцій	Створити та узгодити план із замовником та чітко дотримуватися його.
7	Людський фактор	Робити на кожному етапі перевірки, для запобігання помилок.
8	Зміна терміну виконання робіт	Переговорити із замовником та зменшити кількість додаткових функцій.
9	Недостатня кваліфікація виконавця	Пройти курси для підвищення кваліфікації.

ДОДАТОК Б

Лістинг програмного коду

Лістинг Б.1 – Код файлу web.php

```
<?php
use Illuminate\Support\Facades\Route;
/*
|-----|
|                               Web                               Routes
|-----|
|
| Here is where you can register web routes for your
| application. These
| routes are loaded by the RouteServiceProvider and all of them
| will
| be assigned to the "web" middleware group. Make something
| great!
|
*/
//                               Main                               page                               route
Route::get('/', 'App\Http\Controllers\MainController@index' )-
>name('home');

//                               auth                               routes
Route::get('/login',
'App\Http\Controllers\AuthController@showLoginForm' )-
>name('login');
Route::post('/login_process',
'App\Http\Controllers\AuthController@login' )-
>name('login_process');

Route::get('/logout',
'App\Http\Controllers\AuthController@logout' )->name('logout');
Route::get('/profile',
'App\Http\Controllers\AuthController@profile' )-
>name('profile');

Route::get('/register',
'App\Http\Controllers\AuthController@showRegisterForm' )-
>name('register');
Route::post('/register_process',
'App\Http\Controllers\AuthController@register' )-
>name('register_process');

//                               routes                               for                               services                               page
Route::get('/services',
'App\Http\Controllers\ServiceController@index' )-
>name('services.index');
```

```

Route::get('/services/{service}',
'App\Http\Controllers\ServiceController@show'           )-
>name('services.show');

//           routes           for           stylist           page
Route::get('/stylists',
'App\Http\Controllers\StylistController@index'         )-
>name('stylists.index');
Route::get('/stylists/{stylist}',
'App\Http\Controllers\StylistController@show'         )-
>name('stylist.show');

//           routes           for           appointments           page
Route::get('/appointments',
'App\Http\Controllers\AppointmentController@index'     )-
>name('appointments.index');
Route::get('/appointments/create',
'App\Http\Controllers\AppointmentController@create'    )-
>name('appointment.create');
Route::delete('/appointments/{appointment}',
'App\Http\Controllers\AppointmentController@destroy'   )-
>name('appointment.delete');

Route::get('send', '\App\Http\Controllers\mailController@send');

```

Лістинг Б.2 – Код файлу admin.blade.php

```

<!doctype                                     html>
<html                                           lang="en">
<head>
  <meta                                           charset="UTF-8">
  <meta                                           name="viewport"
        content="width=device-width,           user-scalable=no,
initial-scale=1.0,           maximum-scale=1.0,           minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible"           content="ie=edge">
  @vite('resources/css/app.css')
  @vite('resources/js/app.js')
  <title>@yield('title')</title>
</head>
<body>
  <div>
    <nav class="fixed top-0 z-50 w-full bg-white border-b
border-gray-200           dark:bg-gray-800           dark:border-gray-700">
      <div class="px-3 py-3 lg:px-5 lg:pl-3">
        <div class="flex items-center justify-between">
          <div class="flex items-center justify-
start">
            <button data-drawer-target="logo-
sidebar" data-drawer-toggle="logo-sidebar" aria-controls="logo-
sidebar" type="button" class="inline-flex items-center p-2 text-
sm text-gray-500 rounded-lg sm:hidden hover:bg-gray-100
focus:outline-none focus:ring-2 focus:ring-gray-200 dark:text-
gray-400           dark: hover:bg-gray-700           dark: focus:ring-gray-600">

```

```

                                <span                                class="sr-only">Open
sidebar</span>
                                <svg                                class="w-6                                h-6"                                aria-
hidden="true"                                fill="currentColor"                                viewBox="0                                0                                20                                20"
xmlns="http://www.w3.org/2000/svg">
                                <path                                clip-rule="evenodd"                                fill-
rule="evenodd"                                d="M2 4.75A.75.75 0 012.75 4h14.5a.75.75 0 010
1.5H2.75A.75.75 0 012 4.75zm0 10.5a.75.75 0 01.75-.75h7.5a.75.75
0 010 1.5h-7.5a.75.75 0 01-.75-.75zM2 10a.75.75 0 01.75-
.75h14.5a.75.75 0 010 1.5H2.75A.75.75 0 012 10z"></path>
                                </svg>
                                </button>
                                <span                                class="self-center                                text-xl                                font-
semibold                                sm:text-2xl                                whitespace-nowrap                                dark:text-white">Beauty
Studio</span>
                                </div>
                                </div>
                                </div>
                                </div>
                                </nav>
                                <aside                                id="logo-sidebar"                                class="fixed                                top-0                                left-0                                z-40
w-64                                h-screen                                pt-20                                transition-transform                                -translate-x-full                                bg-
white                                border-r                                border-gray-200                                sm:translate-x-0                                dark:bg-gray-800
dark:border-gray-700"                                aria-label="Sidebar">
                                <div                                class="h-full                                px-3                                pb-4                                overflow-y-auto                                bg-
white                                dark:bg-gray-800">
                                <ul                                class="space-y-2                                font-medium">
                                <li>
                                <a
href="{{route('admin.services.index')}}"                                class="flex                                items-
center                                p-2                                text-gray-900                                rounded-lg                                dark:text-white                                hover:bg-
gray-100                                dark:hover:bg-gray-700                                group">
                                <svg                                class="w-5                                h-5                                text-gray-500
transition                                duration-75                                dark:text-gray-400                                group-hover:text-gray-
900                                dark:group-hover:text-white"                                aria-hidden="true"
xmlns="http://www.w3.org/2000/svg"                                fill="currentColor"
viewBox="0                                0                                22                                21">
                                <path                                d="M16.975 11H10V4.025a1 1
0 0 0-1.066-.998 8.5 8.5 0 1 0 9.039 9.039.999.999 0 0 0-1-
1.066h.002Z"/>
                                <path                                d="M12.5 0c-.157 0-
.311.01-.565.027A1 1 0 0 0 11 1.02V10h8.975a1 1 0 0 0 1-
.935c.013-.188.028-.374.028-.565A8.51 8.51 0 0 0 12.5 0Z"/>
                                </svg>
                                <span                                class="ml-3">Послуги</span>
                                </a>
                                </li>
                                <li>
                                <a                                href="#"                                class="flex                                items-center                                p-2
text-gray-900                                rounded-lg                                dark:text-white                                hover:bg-gray-100
dark:hover:bg-gray-700                                group">
                                <svg                                class="flex-shrink-0                                w-5                                h-5

```

```
text-gray-500 transition duration-75 dark:text-gray-400 group-
hover:text-gray-900 dark:group-hover:text-white" aria-
hidden="true" xmlns="http://www.w3.org/2000/svg"
fill="currentColor" viewBox="0 0 18 18">
    <path d="M6.143 0H1.857A1.857
1.857 0 0 0 0 1.857v4.286C0 7.169.831 8 1.857 8h4.286A1.857
1.857 0 0 0 8 6.143V1.857A1.857 1.857 0 0 0 6.143 0Zm10 0h-
4.286A1.857 1.857 0 0 0 10 1.857v4.286C10 7.169 10.831 8 11.857
8h4.286A1.857 1.857 0 0 0 18 6.143V1.857A1.857 1.857 0 0 0
16.143 0Zm-10 10H1.857A1.857 1.857 0 0 0 0 11.857v4.286C0
17.169.831 18 1.857 18h4.286A1.857 1.857 0 0 0 8 16.143v-
4.286A1.857 1.857 0 0 0 6.143 10Zm10 0h-4.286A1.857 1.857 0 0 0
10 11.857v4.286c0 1.026.831 1.857 1.857 1.857h4.286A1.857 1.857
0 0 0 18 16.143v-4.286A1.857 1.857 0 0 0 16.143 10Z"/>
    </svg>
    <span class="flex-1 ml-3 whitespace-
nowrap">Працівники</span>
</a>
</li>

<li>
    <a
href="{{route('admin.categories.index')}}" class="flex items-
center p-2 text-gray-900 rounded-lg dark:text-white hover:bg-
gray-100 dark:group-hover:text-white" aria-
hidden="true" xmlns="http://www.w3.org/2000/svg"
fill="currentColor" viewBox="0 0 18 20">
    <path d="M17 5.923A1 1 0 0 0 16
5h-3V4a4 4 0 1 0-8 0v1H2a1 1 0 0 0-1 .923L.086 17.846A2 2 0 0 0
2.08 20h13.84a2 2 0 0 0 1.994-2.153L17 5.923ZM7 9a1 1 0 0 1-2
0V7h2v2Zm0-5a2 2 0 1 1 4 0v1H7V4Zm6 5a1 1 0 1 1-2 0V7h2v2Z"/>
    </svg>
    <span class="flex-1 ml-3 whitespace-
nowrap">Категорії послуг</span>
</a>
</li>

<li>
    <a href="#" class="flex items-center p-2
text-gray-900 rounded-lg dark:text-white hover:bg-gray-100
dark:group-hover:text-white" aria-
hidden="true" xmlns="http://www.w3.org/2000/svg"
fill="currentColor" viewBox="0 0 20 20">
    <path d="m17.418 3.623-.018-
.008a6.713 6.713 0 0 0-2.4-.569V2h1a1 1 0 1 0 0-2h-2a1 1 0 0 0-1
1v2H9.89A6.977 6.977 0 0 1 12 8v5h-2V8A5 5 0 1 0 0 8v6a1 1 0 0 0
1 1h8v4a1 1 0 0 0 1 1h2a1 1 0 0 0 1-1v-4h6a1 1 0 0 0 1-1V8a5 5 0
0 0-2.582-4.377ZM6 12H4a1 1 0 0 1 0-2h2a1 1 0 0 1 0 2Z"/>
    </a>
</li>
```

```

                                </svg>
                                <span class="flex-1 ml-3 whitespace-
nowrap">Налаштування                                сайту</span>
                                </a>
                                </li>
                                <li>
                                <form action="{{route('admin.logout')}}"
method="post">
                                @csrf
                                <button type="submit" class="flex
items-center p-2 text-gray-900 rounded-lg dark:text-white
hover:bg-gray-100 dark:group-hover:bg-gray-700 group">
                                <svg class="flex-shrink-0 w-5 h-
5 text-gray-500 transition duration-75 dark:text-gray-400 group-
hover:text-gray-900 dark:group-hover:text-white" aria-
hidden="true" xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 18 16">
                                <path stroke="currentColor"
stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M1 8h11m0 0L8 4m4 4-4 4m4-11h3a2 2 0 0 1 2 2v10a2 2 0 0 1-2
2h-3"/>
                                </svg>
                                <span class="flex-1 ml-3
whitespace-nowrap">Вийти</span>
                                </button>
                                </form>
                                </li>
                                </ul>
                                </div>
                                </aside>
                                <div class="p-4 sm:ml-64 mt-14 text-black">
                                @yield('content')
                                </div>
                                </div>
</body>
</html>

```

Лістинг Б.3 – Код файлу admin.php

```

<?php
use Illuminate\Support\Facades\Route;

Route::get('/login',
'App\Http\Controllers\Admin\AuthController@index'
) -
>name('login');
Route::post('/login_process',

```

```
'App\Http\Controllers\Admin\AuthController@login'           )-
>name('login_process');
Route::post('/logout',
'App\Http\Controllers\Admin\AuthController@logout'         )-
>name('logout');

Route::get('/main',
'App\Http\Controllers\Admin\AuthController@main'           )-
>name('admin.main');

//services
Route::resource('services',\App\Http\Controllers\Admin\ServiceCo
ntroller::class);

//categories
Route::resource('categories',\App\Http\Controllers\Admin\Categor
yController::class);

//stylists
Route::resource('stylists',\App\Http\Controllers\Admin\StylistCo
ntroller::class);

//appointments
Route::resource('appointments',\App\Http\Controllers\Admin\Appoi
ntmentController::class);

//users
Route::resource('users',\App\Http\Controllers\Admin\UserControll
er::class);
```