МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток імідж-студії Глянець»

за напрямом підготовки 6.050101 «Комп'ютерні науки»

Виконавець роботи: студент групи ІТ-62 Рудник Марина Олексіївна

Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою _____«___» _____2020 р.

Науковий керівник _____ доц., Парфененко Ю.В.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії _____ Шифрін Д. М.
(підпис) (науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
Зав. секцією ІТП
_____ В. В. Шендрик
«__» _____ 2020 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Рудник Марина Олексіївна

1 Тема роботи *Web-додаток імідж-студії Глянець*

керівник роботи *Парфененко Юлія Вікторівна, доцент*,

затверджені наказом по університету від « 14 » травня 2020 р. № 0576-III

2 Строк подання студентом роботи « 1 » червня 2020 р.

3 Вхідні дані до роботи технічне завдання на розробку веб-додатку.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку імідж-студії Глянець, розробка web-додатку імідж-студії Глянець.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність розроблення web-додатку, постановка задачі, аналіз існуючих моделей, методів, технологій вирішення задачі, порівняння web-додатків-аналогів, функціональні вимоги до web-додатку, огляд існуючих програмних продуктів для вирішення поставленої задачі, контекстна діаграма процесу підтримки роботи імідж-студії «Глянець», діаграма декомпозиції контекстної діаграми, діаграма варіантів використання web-додатку, ER-діаграма таблиць бази даних, архітектура web-додатку, плагіни та модулі, демонстрація готового web-додатку, висновки, ІМА-2020.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання_____01.10.2019_____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Визначення актуальності проблеми	03.09.19 – 07.09.19	
2	Формування мети та задач	15.09.19 – 19.09.19	
3	Аналіз додатків-аналогів	25.09.19 – 30.09.19	
4	Вибір засобів реалізації	10.10.19 – 17.10.19	
5	Постановка цілей	21.10.19 – 02.11.19	
6	Складання технічного завдання	05.11.19 – 20.11.19	
7	Підготовка прототипу	25.11.19 – 06.12.19	
8	Створення макету дизайну web-додатку	10.12.19 – 20.12.19	
9	Верстка	01.01.20 – 20.01.20	
10	Робота над модулями для web-додатку	28.01.20 – 20.02.20	
11	Робота з контентом	05.03.20 – 15.03.20	
12	Розміщення контенту та каталогів з фото у web-додатку	07.04.20 – 26.04.20	
13	Перевірка працездатності web-додатку	01.05.20 – 15.05.20	
14	Завершення роботи	16.05.20 – 27.05.20	

Студент _____ Рудник М.О.
(підпис)

Керівник роботи _____ доц. Парфененко Ю.В.
(підпис)

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра: «Web-додаток імідж-студії Глянець».

Пояснювальна записка містить вступ, 3 розділи, висновки, додатки та список літератури, включає 79 сторінок, 10 таблиць, 46 ілюстрації, 23 джерела.

В першому розділі наведено результати аналізу предметної області. А саме розгляд застосування інформаційних технологій для організації діяльності імідж-студій, проведено огляд та аналіз існуючих web-додатків та описана постановка задачі для даної роботи.

Другий розділ описує проектування web-додатку імідж-студії «Глянець». Виконано структурно-функціональне моделювання процесу розробки web-додатку імідж-студії та моделювання варіантів використання розроблюваного продукту. Також було виконане проектування моделі бази даних MySQL.

Третій розділ включає в себе розробку web-додатку імідж-студії «Глянець». В розділі описано процес створення web-додатку, встановлення модулів та плагінів, а також редагування деяких функцій на файлів плагінів за для повного виконання вимог замовника.

Результатом проведеної роботи є розроблений web-додаток імідж-студії «Глянець».

Ключові слова: web-додаток, веб-розробка, CMS WordPress, контент, імідж-студія, Глянець, beauty-сфера, фотографи, студія.

ЗМІСТ

ВСТУП 6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 8
1.1 Застосування інформаційних технологій для організації діяльності імідж-студій 8
1.2 Огляд існуючих web-додатків для імідж-студій 10
1.3 Постановка задачі 13
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ІМІДЖ-СТУДІЇ ГЛЯНЕЦЬ 16
2.1 Структурно-функціональне моделювання процесу розробки web-додатку імідж-студії «Глянець» 16
2.2 Моделювання варіантів використання web-додатку імідж-студії «Глянець» 19
2.3 Проектування моделі бази даних 23
3 РОЗРОБКА WEB-ДОДАТКУ ІМІДЖ-СТУДІЇ «ГЛЯНЕЦЬ» 26
ВИСНОВОК 45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46
Додаток А. Технічне завдання 48
ТЕХНІЧНЕ ЗАВДАННЯ 48
Додаток Б. Планування робіт 58

ВСТУП

Наше життя сьогодні неможливо уявити без комп’ютера – важливого відкриття наукового прогресу. Дана новинка інформаційних технологій швидко увійшла в наше життя, стрімкими темпами замінюючи раніше винайдені здобуття людства. Такими ж самими темпами вона весь час змінюється і удосконалюється.

Технологія – це здатність робити щось досконало. Коли говорять про інформаційні технології – матеріалом вважають інформацію. Продуктом також являється інформація, але це абсолютно нова інформація про об'єкт, явище або процес.

Технології дуже активно просуваються в різні сфери життєдіяльності людей. В результаті користувачі очікують зручності, яку їм пропонують різноманітні IT-стартапи, і від сервісів краси і догляду за собою також.

Тому beauty-брендам доводиться все активніше впроваджувати інновації, що включають штучний інтелект, машинне навчання, 3D-моделювання для створення більш персоналізованих продуктів і рекомендацій. Цей процес триватиме і стане ще більш інтенсивним, оскільки бренди будуть не тільки використовувати існуючі технології і підходи, а й створювати свої.

Велике значення в сучасному світі має представництво імідж-студії у мережі Інтернет, тому що значна кількість імовірних клієнтів для студії перебуває саме там. Тому імідж-студія потрібує у власному web-додатку, який має відображати всі можливі послуги студії, рекламувати її. Для розробки веб-додатку імідж-студії необхідно зробити аналіз подібних за тематикою додатків, винести з них корисні елементи. Web-додаток студії має свою структуру, яка повинна робити його легким для використання та багатофункціональним для користувача.

Метою даного проекту є створення web-додатку для імідж-студії «Глянець» оскільки в сучасному житті майже кожна організація має свій web-додаток. Необхідним чинником існування, який дозволяє збільшити рекламну діяльність та розширити клієнтську базу є застосування інформаційних технологій.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз предметної області розробки web-додатків для імідж-студій, скласти технічне завдання та обрати засоби реалізації web-додатку;
- виконати моделювання роботи імідж-студії з використанням web-додатку;
- розробити web-додаток для імідж-студії та виконати його тестування.

Щодо практичного значення web-додатку: додаток має на меті інформувати потенційних клієнтів про послуги імідж-студії, майстрів-візажистів, перукарів, їх досвід роботи та портфоліо. Також можна буде переглянути роботи фотографа та доступні фотозони, здійснити онлайн-запис до майстра або фотографа.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Застосування інформаційних технологій для організації діяльності імідж-студій

Поширення мережі Інтернет відбулося практично у всі точки планети, а через розвиток IT-технологій – комп'ютер, який має доступ до глобальної мережі, можна зустріти практично в усіх будинках. Доступність Інтернету та його повсюдне поширення, дуже скоро призвели до того, що велика кількість людей значну частину вільного часу проводять в мережі. Для більшості людей Інтернет набув значення основного джерела інформації. Він почав поступово проходити на зміну друкованим ЗМІ та телебаченню, внаслідок чого, значимість друкованої та телевізійної реклами стала зменшуватись.

Аналізуючи те, що відбувається, сфера бізнесу не могла не реагувати на зміни. Підприємці, котрі змогли прогнозувати дане явище, почали швидко переносити деякі частини свого бізнесу в мережу. Поступово розуміння необхідності створення свого web-ресурсу зросло до масштабів світового ринку. Так, за відносно короткий проміжок часу, кількість різних комерційних web-додатків значно збільшилась.

Напевно немає жодної організації чи компанії, яка позиціонує себе серйозним представником свого роду діяльності та не має особистого представництва в мережі Інтернет, адже web-додаток – це не тільки обличчя власника, а ще й гарний спосіб просування свого бренду. Web-додаток надає можливість швидко та без особливих складнощів ознайомити потенційних клієнтів з власною організацією та з послугами, які вона надає.

Як джерело інформації, мережа Інтернет настільки витіснила телебачення, радіо і газети з повсякденного життя людства, що більш вдалого варіанту для реклами не варто й шукати. Щоб дізнатися відповідь на будь яке питання, люди вже не йдуть до бібліотеки, а роблять запит в пошукових

стрічках своїх браузерів і пошук студії, яка допоможе підготуватися до фотосесії та організувати саму фотосесію – далеко не виключення.

Актуальність розроблення web-додатку імідж-студії обґрунтовується подальшим розширенням клієнтської бази, активним ростом рівня обслуговування клієнтів. Завдяки охопленню великої частини цільової аудиторії очікується стрімке збільшення прибутків для власника студії.

Користувач розроблюваного web-додатку матиме змогу ознайомитися зі всіма майстрами студії, їх портфоліо та досвідом роботи, останніми новинами, списком усіх послуг та їх цін, а також здійснити онлайн-запис до бажаного майстра та слідкувати за статусом обробки замовлення. А, власне, студія зможе на постійній основі робити рекламу вигідних пропозицій та акцій від організації, розповсюджувати корисні статті про сферу краси та фотографії тощо. Дані функції облегшують процес роботи адміністраторів та імідж-студії в цілому з клієнтською базою.

Як результат, після виконання даного проекту буде створено та впроваджено web-додаток імідж-студії «Глянець», за допомогою якого зросте рівень конкурентоспроможності організації на ринку.

1.2 Огляд існуючих web-додатків для імідж-студій

Після аналізу предметної області та на початку роботи над web-додатком було проаналізовано існуючі web-додатки організацій конкурентів. Після чого було виявлено ряд аналогів. Розглянемо декілька web-додатків для студій краси, такі як: Maija.ua, Mirasalon.com.ua та Stella.in.ua.

Maija.ua - це web-додаток для мережі салонів «Майя» (рис. 1.1) [1]. Дизайн даного додатку виконаний у чорно-білих відтінках. На головній сторінці можна побачити актуальні процедури, відео проведення деяких з них та основні дані про салон. Також є онлайн-запис для клієнтів, але там не можливо обрати майстра та не відображається, який час для запису на процедуру вільний. В цілому, через простоту інтерфейсу додаток являється легким для сприйняття та використання.

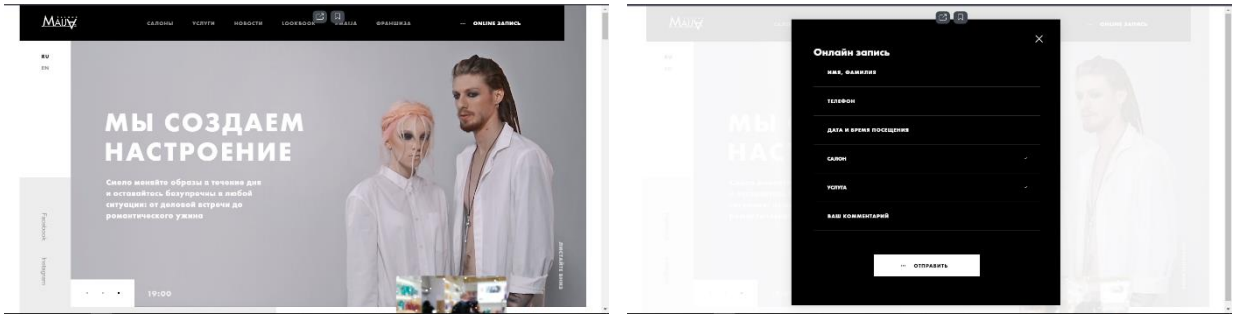

Рисунок 1.1 – Головна сторінка web-додатку Maija.ua

Mirasalon.com.ua – це web-додаток салону Міра (рис. 1.2) [2]. Додаток складається з головної сторінки та сторінки онлайн-запису, що не дуже практично. Перше що ми бачимо при вході в додаток – кнопку для онлайн-запису. Процедура онлайн-запису досить зручна, так як є можливість обрати процедуру, яку хочемо зробити, після чого з’явиться список майстрів та їх вільний час. Також є можливість змінити мову додатку.

На сторінці ще можна ознайомитись з послугами, які надає салон, основною інформацією та контактами компанії. Недоліком є відсутність меню додатку.

Рисунок 1.2 – Головна сторінка та сторінка запису web-додатку Mirasalon.com.ua

Stella.in.ua – web-додаток виконано в червоно-білих кольорах (рис. 1.3) [3]. Додаток досить зручний у використанні, оскільки має структуроване меню. Одним з пунктів меню є відгуки від клієнтів. Це дуже значна перевага, бо відразу можна зробити висновки про послуги та впевнитись у правильності вибору салону.

Також додаток наповнений великою кількістю корисних статей для клієнтів та інформацією про кожну процедуру, яку надає салон. Єдиним недоліком даного додатку є фактична відсутність онлайн запису. При натисканні на кнопку «Запис» додаток переходить на сторінку контактів салону.

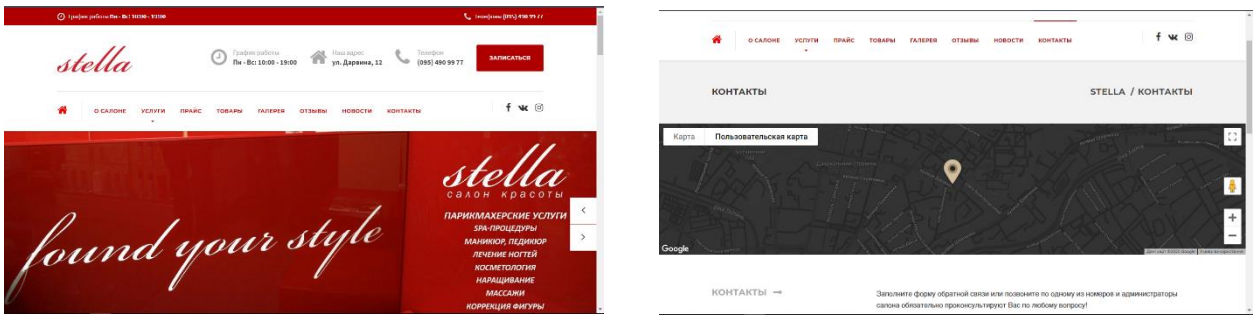

Рисунок 1.3 – Головна сторінка web-додатку Stella.in.ua

Перелік проблем програмних продуктів-аналогів наводиться у таблиці 1.1.

Таблиця 1.1 – Перелік недоліків існуючих програмних продуктів

№	Назва / Функція	Зміст	Недолік, який уточняється	Примітка: роль/ чий інтерес
1	Можливість вибору майстра при онлайн-записі	Web-додаток містить онлайн-запис, але тільки для одного майстра, а працюють декілька майстрів.	Обмеженість в виборі майстрів	Вузька область використання додатку
2	Наявність корисних статей	Додатки конкурентів майже не мають інформативних статей для клієнтів.	Користі інформатив-ні статі	Мала інформаційна складова
3	Наявність інформації про майстрів (візажистів, перукарів, фотографів)	Клієнти не мають можливості ознайомитись з майстрами, які працюють в студії та їх досвідом роботи.	Недостатньо інформації про майстрів	Необґрунтова-ний вибір майстра
4	Можливість відслідковування статусу замовлення	Додаток не має даної функції, через що не зрозуміло чи підтвердили запис.	Не можливо слідкувати за статусом замовлення	Непорозуміння при онлайн-записі
5	Можливість корегувати графік роботи	Деякі додатки мають сталий графік у онлайн записі, через що відбуваються помилки у записі клієнтів.	Сталий графік роботи	Непорозуміння у записі між клієнтами та майстрами

1.3 Постановка задачі

Мета даного проекту – розробка web-додатку імідж-студії «Глянець», задля рекламування послуг студії та спрощення роботи адміністраторів, а саме створення онлайн-запису до будь-кого з майстрів. Процес експлуатації web-додатку має призвести до подальшого розширення клієнтської бази та активного росту рівня обслуговування клієнтів. Завдяки охопленню великої частини цільової аудиторії очікується стрімке збільшення прибутків для власника бізнесу.

Web-додаток розроблюється для конкретної імідж-студії, яка висловила особисті вимоги до продукту, через це користуватися ним зможе тільки замовник. Даний додаток буде переважно працювати на клієнтів з міста Суми, тому що сфера краси не має змоги працювати на великих відстанях.

Даний web-додаток має виконувати наступні функції:

- ознайомлення клієнтів з майстрами та фотографами студії;
- перегляд портфоліо персоналу студії;
- інформування клієнтів про новини студії та можливі акції;
- перегляд всіх послуг, які надає студія та прайс-листа на них;
- онлайн-запис до майстрів або фотографа;
- наведення контактів студії;
- інформативні статті для клієнтів;
- адаптивність додатку під телефон та персональний комп’ютер ;
- реєстрація та вхід в особистий кабінет;
- можливість відслідковувати статус виконання замовлення.

Web-додаток має бути легким для сприйняття, зручним та зрозумілим для використання клієнтами. Щоб досягти заданих цілей, було вирішено створити верхнє горизонтальне меню, розроблено інтерфейс з використанням зображень великого розширення для кращого візуального сприйняття, а також адаптовано web-додаток для телефону та ПК.

Панель керування, яка буде використовуватись для адміністрування web-додатком, було створено за допомогою CMS WordPress. Дана функція послугувала однією з головних причин у виборі системи управління контентом для розробки web-додатку. За допомогою адміністративної панелі керування адміністратор зможе створювати, видаляти та редагувати контент та змінювати зовнішній вигляд додатку.

У додатку А наведено технічне завдання, де детально описано призначення та мету створення web-додатку, вимоги, склад і зміст робіт по розробці даного додатку. Планування робіт представлене у додатку Б.

Коли постала потреба у виборі засобів реалізації програмного продукту, було виведено переваги та недоліки декількох варіантів, але все таки вирішено обрати систему управління контентом WordPress, щоб зменшити час витрачений на розробку додатку[4].

Обрана система одна з найпопулярніших у світі. На ній працює більш ніж 30 відсотків web-додатків у мережі Інтернет, а також безліч шаблонів та плагінів для налаштування зовнішнього вигляду та функціоналу перебувають у вільному доступі [5]. Загалом, система WordPress розроблена за допомогою мови PHP, що спрощує під'єднання, за необхідністю, власноруч написаних плагінів. Також, як вже згадувалося раніше, значимою перевагою CMS WordPress є вже реалізована адміністративна панель та функція адаптивності, а саме ці пункти є у технічному завданні додатку. Через простоту панелі керування, адміністратору студії та замовнику не потрібно буде витрачати багато часу на її освоєння [6].

Реляційну систему управління базами даних MySQL було використано для реалізації збереження, бо саме вона надає можливість забезпечення потрібного рівня безпеки даних. Також система достатньо проста у використанні. У базі даних буде міститися велика кількість фотографій для портфоліо, а також багато записів та даних, які реалізовані через модуль онлайн-запису та авторизації. Варто відмітити, що при створенні web-додатку CMS WordPress відразу пропонує створити базу MySQL.

Для написання модулів онлайн-запису та форми зворотного зв’язку було обрано мову PHP, оскільки на сьогодні вона є дуже розповсюдженою. Ця мова програмування розроблена спеціально задля реалізації web-додатків. Код PHP одразу може бути використаний у файлі HTML [7].

Програмне забезпечення MySQL Workbench 8.0 було обрано для роботи з базою даних задля адміністрування. Дане ПЗ є уніфікованим візуальним інструментом для архітектури баз даних. Воно надає можливість моделювання даних та комплексні інструменти адміністрування, резервного копіювання [8].

Оскільки обраний інструментарій користується великим попитом у світі, то за необхідності в мережі Інтернет є багато корисних статей та літератури щодо роботи з ним. Також великим плюсом є те, що всі ці продукти працюють разом злагоджено та не викликають певних проблем при поєднанні.

2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ІМІДЖ-СТУДІЇ ГЛЯНЕЦЬ

2.1 Структурно-функціональне моделювання процесу розробки web-додатку імідж-студії «Глянець»

Для моделювання різних бізнес-процесів можна використати декілька різних методів, в основу яких покладено як структурі, так і об’єктно-орієнтовані підходи моделювання. Найпоширеніший – метод функціонального моделювання IDEF0. Головним принципом підходу є сукупність деяких правил, які слугують для створення функціональної моделі об’єкту з різних предметних областей. Модель IDEF0 надає точну специфікацію операцій, що відбуваються в бізнес-процесі та їх взаємозв’язки.

На рис. 2.1 наведено контекстну діаграму підтримки роботи імідж-студії «Глянець».

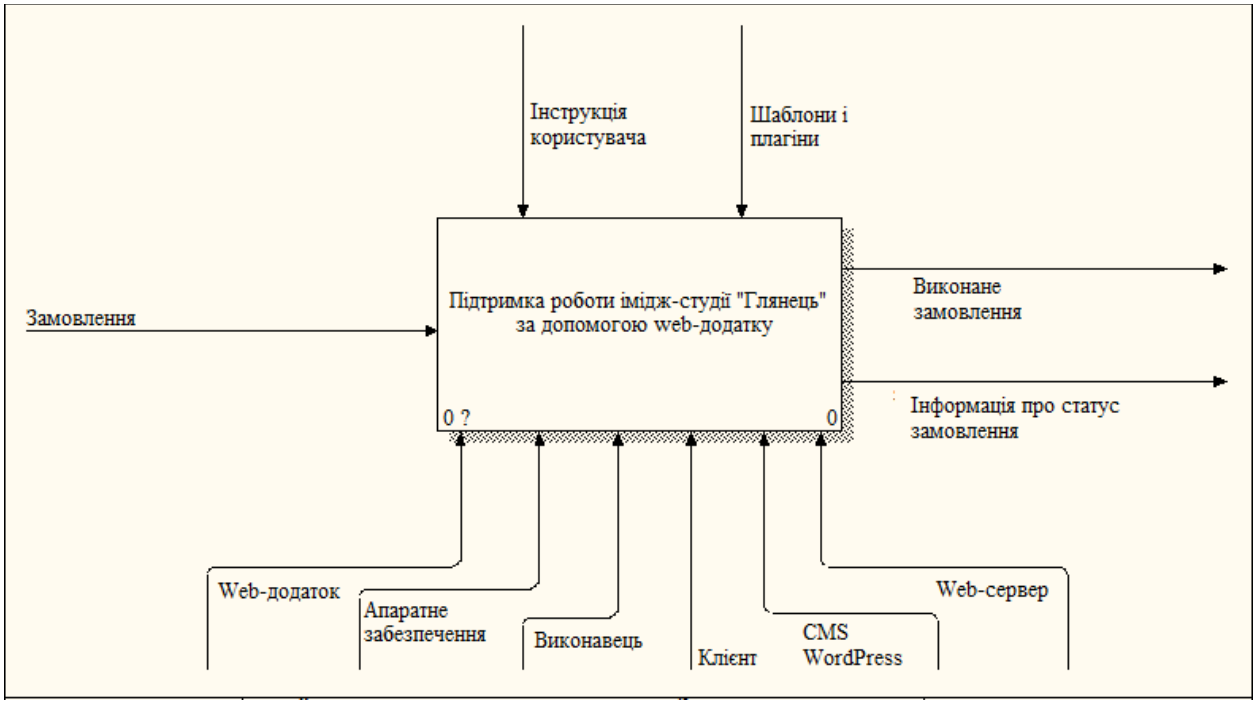

Рисунок 2.1 – Контекстна діаграма процесу підтримки роботи імідж-студії «Глянець»

Дана діаграма включає в себе функціональний блок, що містить функцію даної системи. Кожна зі сторін блоку має свою роль, а саме:

- Управління – верхня сторона блоку;
- Вхід – ліва сторона блоку. ;
- Вихід – права сторона блоку;
- Механізм – нижня сторона блоку [9].

Спрямована стрілка (інтерфейсна дуга) вказує на елемент системи, який обробляє функціональний блок або впливає на його функцію. Кожна стрілка має унікальне найменування. Залежно від сторін блоку, куди входить або виходить дуга, розрізняють наступні найменування: «Входить», «Механізм», «Керуюча» та «Виходить». Через те, що реалізація процесів проходить відповідно до певних правил та має видавати результат, функціональний блок повинен мати хоча б по одній керуючій та вихідній стрілці.

У випадку з даною діаграмою, функціональний блок відповідає за процес «Підтримка імідж-студії “Глянець”». На вхід подається Замовлення. На виході отримуємо Виконане замовлення та Інформацію про статус замовлення.

Управління процесом здійснюється за допомогою Технічного завдання та Шаблонів і плагінів, а механізмами слугують Виконавець, Клієнт, Апаратне забезпечення, CMS WordPress, Web-сервер та Web-додаток.

Другий етап – це декомпозиція контекстної діаграми. Вона розбиває основну функцію на деяку кількість підпроцесів та уточнює загальну функцію. В даному процесі стрілки з контекстної діаграми переходять в декомпозуючу. При цьому блоки підпроцесів пов’язані між собою послідовними діями, що ілюструє рух потоків інформації [10].

У розробленій діаграмі декомпозиції функціональний блок розкладається на наступні процеси:

– здійснити онлайн-запис;
– підтвердити запит клієнта на запис;
– змінити статус замовлення;

– отримати відгуки про виконання замовлення.

Після першого процесу матимемо сформований запит, який потім піде до другого процесу. На виході з другого процесу отримуємо прийняте замовлення, яке дає змогу зміни статусу виконання замовлення. Після виконання трьох процесів студія матиме змогу отримати відгуки про виконання замовлення. Отже, на виході отримуємо інформацію про статус замовлення та виконане завдання.

Декомпозиція контекстної діаграми зображена на рис.2.2.

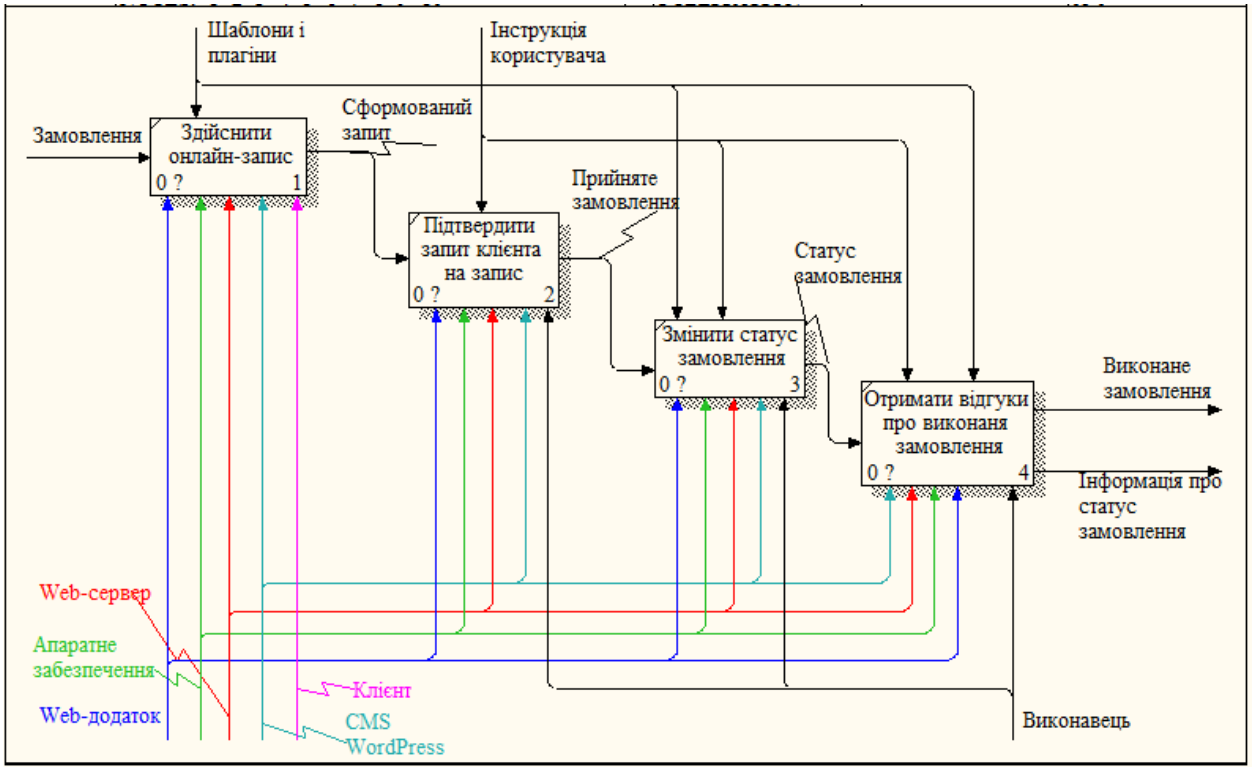

Рисунок 2.2 – Декомпозиція контекстної діаграми

До блоку «Здійснити онлайн-запис» направлена вхідна стрілка «Замовлення», стрілка управління «Шаблони і плагіни», стрілки механізмів «Клієнт», «Web-сервер», «Апаратне забезпечення», «CMS WordPress» та «Web-додаток». На виході маємо «Сформований запит».

До блоку «Підтвердити запит клієнта на запис» направлена вхідна стрілка «Сформований запит», стрілка управління «Технічне завдання»,

стрілки механізмів «Виконавець», «Web-сервер», «Апаратне забезпечення», «CMS WordPress» та «Web-додаток». На виході маємо «Прийняте замовлення».

До блоку «Змінити статус замовлення» направлена вхідна стрілка «Прийняте замовлення», стрілки управління «Шаблони і плагіни» та «Технічне завдання», стрілки механізмів «Виконавець», «Web-сервер», «Апаратне забезпечення», «CMS WordPress» та «Web-додаток». На виході маємо «Статус».

До блоку «Отримати відгуки про виконання замовлення» направлена вхідна стрілка «Статус», стрілки управління «Шаблони і плагіни» та «Технічне завдання», стрілки механізмів «Виконавець», «Web-сервер», «Апаратне забезпечення», «CMS WordPress» та «Web-додаток». На виході маємо «Виконане замовлення» та «Інформацію про статус замовлення».

2.2 Моделювання варіантів використання web-додатку імідж-студії «Глянець»

Як відомо, UML модель, а саме діаграма варіантів використання, являється вихідним концептуальним представленням системи в момент процесу розробки. Дана модель містить в собі акторів, варіанти використання (прецеденти) та зображує відносини між ними.

Головна мета побудови UML моделі – це представлення проектованої системи у вигляді множини акторів, котрі взаємодіють із системою завдяки варіантам використання.

Актором називають будь-яку систему, об’єкт чи суб’єкт, котрі взаємодіють з системою, що моделюється. Щодо варіанту використання – це специфікація тих функцій, що надаються актор системою [12].

Проектування UML моделі допомагає розробнику визначити загальні межі web-додатку, а також демонструє функціональну поведінку системи.

На даному етапі проектування web-додатку імідж-студії «Глянець», було розроблено UML діаграму варіантів використання, яка зображена на рис. 2.3.

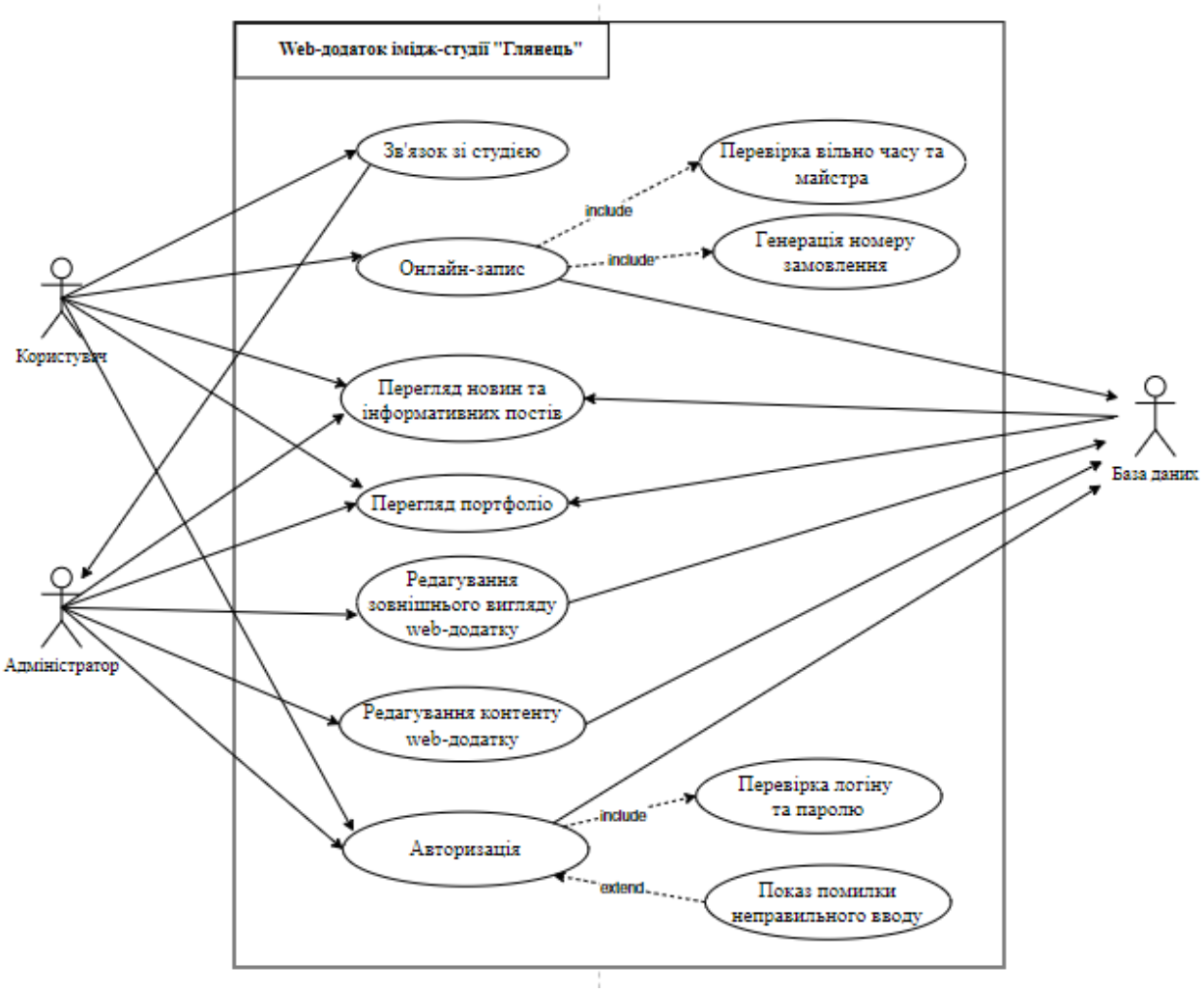

Рисунок 2.3 – UML діаграма варіантів використання web-додатку імідж студії «Глянець»

Виділено наступних акторів UML моделі для web-додатку імідж-студії:

Адміністратор – людина, котра відповідає за керування функціонуванням web-додатку, приймає заявки на запис, редагує контент тощо;

Користувач – клієнт, що використовуючи web-додаток має змогу оглянути всю розміщену інформацію та здійснити онлайн-запис;

База даних – СУБД, що містить в собі всю інформацію щодо web-додатку, зберігає нову та, за потреби, відповідає на запити.

Варіанти використання UML моделі web-додатку імідж-студії:

ВВ Зв’язок зі студією – ВВ дозволяє написати електронний лист на пошту студії.

ВВ Онлайн-запис – ВВ дозволяє клієнту записатись на процедуру, обравши час, майстра та залишити коментар, щодо замовлення.

ВВ Перегляд новин та інформативних постів. – ВВ дозволяє клієнту дізнатися про новини студії та ознайомитись з постами з корисною інформацією.

ВВ Перегляд портфоліо – ВВ дозволяє користувачу ознайомитись зі спектром робіт всіх майстрів студії.

ВВ Редагування зовнішнього вигляду web-додатку – ВВ надає можливість адміністратору змінювати тему або структура web-додатку.

ВВ Редагування наповнення web-додатку – ВВ надає можливість адміністратору змінювати контент, працювати з фотографіями та іншою інформацією, що знаходиться у web-додатку.

ВВ Слідкування за статусом замовлення – ВВ дозволяє користувачу дізнаватися про статус обробки та виконання його замовлення.

2.3 Розробка прототипів

Задля того, щоб всі нюанси та функціональність інтерфейсу задовольняли вимоги замовника, було розроблено прототипи деяких сторінок web-додатку.

Прототип допомагає наглядно проілюструвати всі задуми, а також зробити правки в інтерфейс з мінімальними зусиллями.

На рис. 2.4-2.6 наведено прототипи деяких сторінок web-додатку:

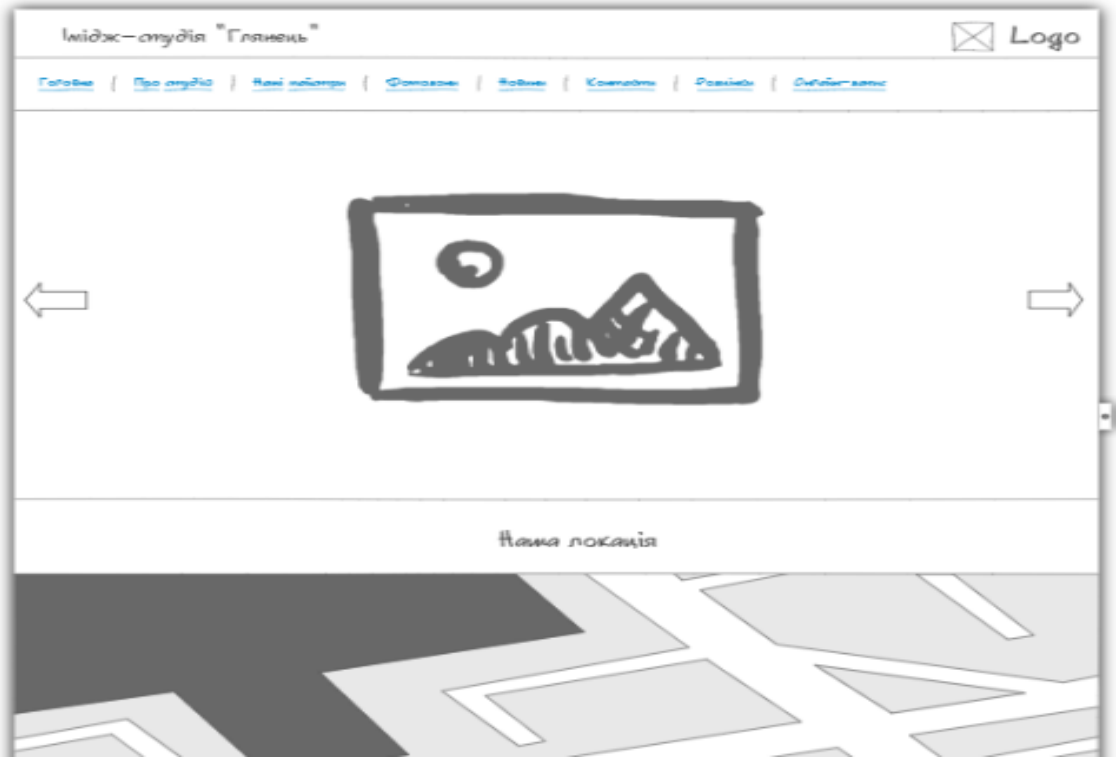

Рисунок 2.4 – Прототип головної сторінки web-додатку

Рисунок 2.5 – Прототип сторінки «Про студію»

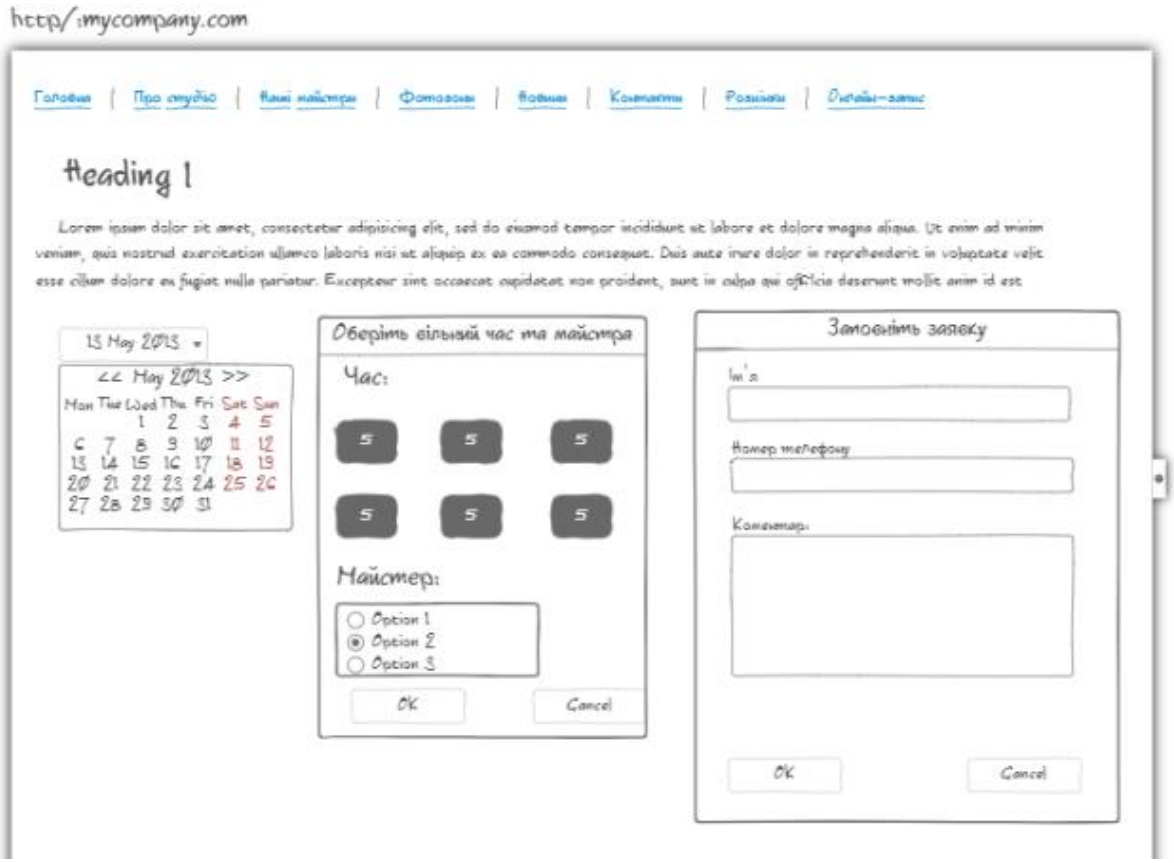

Рисунок 2.6 – Прототип сторінки онлайн-запису

2.4 Проектування моделі бази даних

Великою перевагою для створення web-додатку з використанням CMS WordPress являється автоматичне створення бази даних.

Реляційна база даних дозволяє легко знаходити потрібну інформацію, а також сортувати дані на основі будь-якого поля та генерувати звіти, які складаються лише з певних полів таблиці. Бази даних використовують таблиці для зберігання інформації. Стандартні поля та записи представлені у вигляді стовпців (полів) та рядків (записів) у таблиці.

За допомогою реляційної бази даних є можливість швидко порівняти інформацію через розташування даних у стовпцях. Модель реляційної бази даних використовує цю уніфікованість для створення абсолютно нових таблиць із необхідної інформації з існуючих таблиць. Іншими словами, вона

використовує зв'язок подібних даних для підвищення швидкості та універсальності бази даних.

Оскільки в даному випадку база даних створювалась автоматично, розглянемо деякі з таблиць.

Таблиця 2.1 – Таблиці бази даних

№ з/п	Назва таблиці	Призначення
1	wp_users	Збереження інформації про зареєстрованих користувачів.
2	wp_usermeta	Збереження інформації власного профілю користувача.
3	wp_posts	Збереження всіх записів, постів та рубрик.
4	wp_comments	Збереження інформації про відгуки.
5	wp_finaltiles_gallery_images	Збереження інформації про галереї та портфоліо.
6	wp_rm_forms	Інформація про форму реєстрації.

На рис. 2.7 можна переглянути ER-діаграму створеної бази даних.

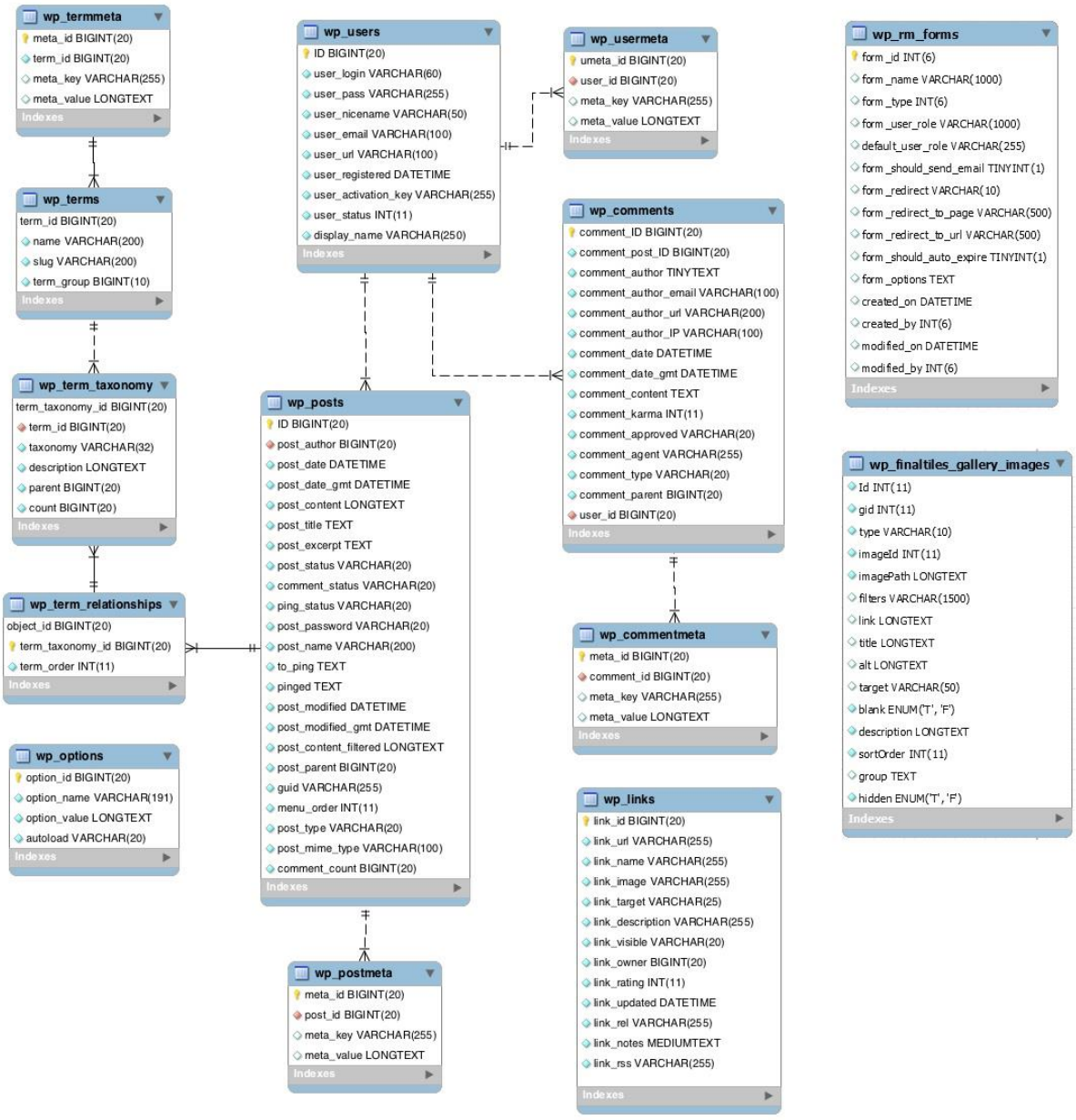

Рисунок 2.7 – ER-діаграма бази даних

3 РОЗРОБКА WEB-ДОДАТКУ ІМІДЖ-СТУДІЇ «ГЛЯНЕЦЬ»

3.1 Архітектура програмного додатку

Архітектура web-додатку – це спосіб структурування web-додатку, що може допомогти користувачам легко та швидко знаходити інформацію та здійснювати переходи.

Зручна архітектура web-додатків підсилює користувацький рівень web-додатку. Крім того, коли рівень користувача буде високим – рейтинг пошукових систем зросте. Користувачі проводять більше часу у web-додатку та посилаються на web-сторінки, що є важливими показниками того, що даний бренд створює якісний контент. Також варто відзначити, що надійна архітектура web-додатків допомагає пошуковим системам ефективно сканувати їх.

На рисунку 3.1 зображена архітектура web-додатку для імідж-студії «Глянець».

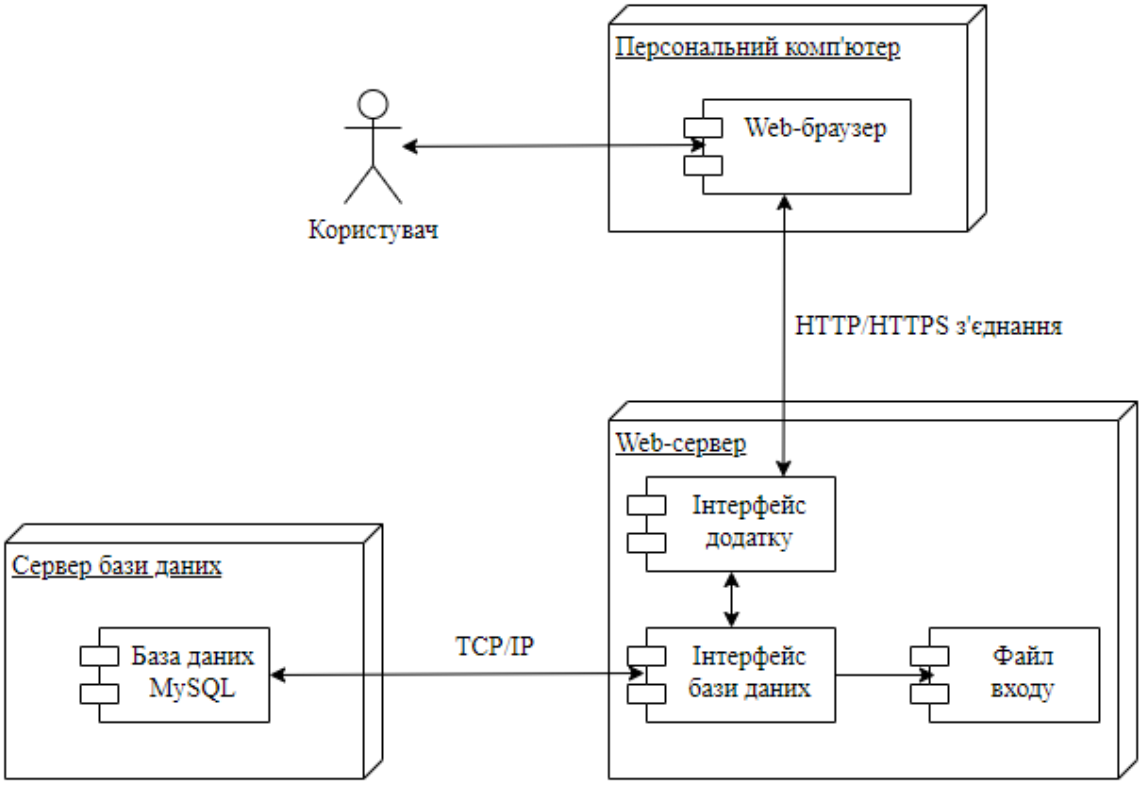

Рисунок 3.1 – UML діаграма розгортання

3.2 Встановлення та розробка компонентів web-додатку

Для розробки web-додатку вирішено використати систему управління контентом WordPress, оскільки після проведення аналізу було виявлено велику кількість переваг, а саме автоматичне створення бази даних MySQL при встановленні CMS, відсутність значних вимог до сервера, не високі системні вимоги, а також можливість перенесення на будь-який хостинг.

Першим кроком на початку роботи було вирішено обрати хостинг та встановити на ньому WordPress. Після цього системою буде запропоновано створити базу даних засобами PhpMyAdmin. База даних, яка була створена, має назву gloss_studio (рис. 3.2).

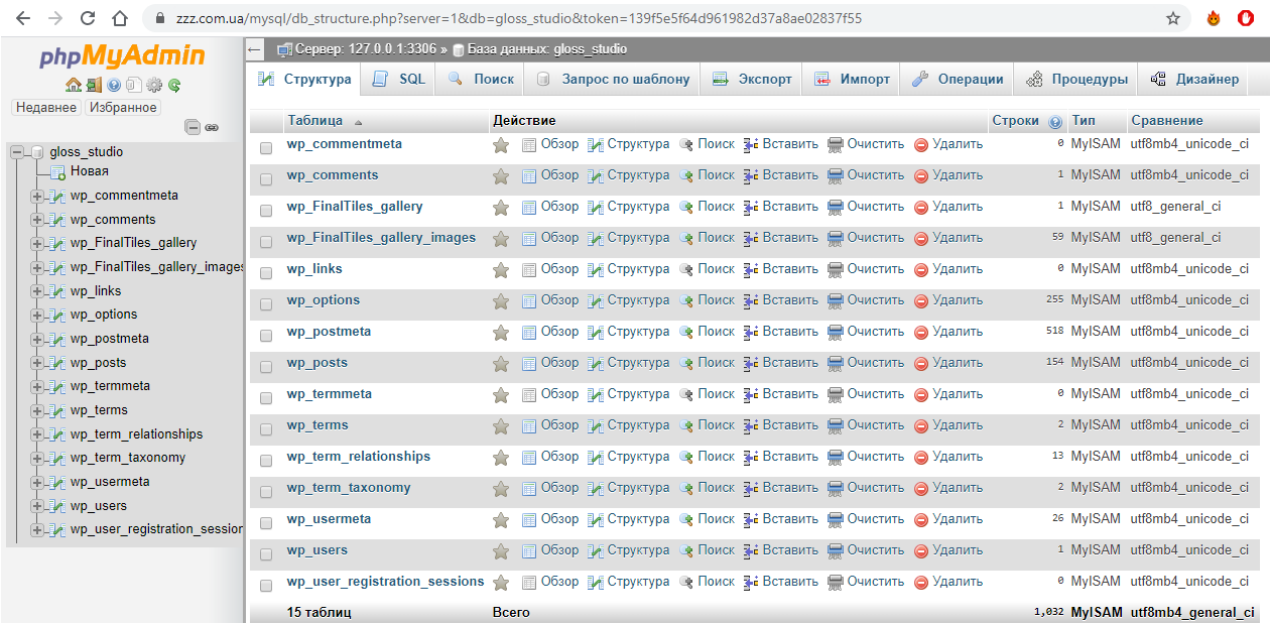

Рисунок 3.2 – Бази даних в сервісі PhpMyAdmin

Для того щоб підключити створену базу даних до CMS WordPress було використано програму FileZilla Client та в текстовому редакторі відредаговано файл wp-config.php, де треба прописати назву та пароль бази даних:

```
// ** MySQL settings - You can get this info from your web host ** //

/** The name of the database for WordPress */

define( 'DB_NAME', 'gloss_studio' );


/** MySQL database username */

define( 'DB_USER', 'marinarudnyk' );


/** MySQL database password */

define( 'DB_PASSWORD', 'Rudnyk12' );


/** MySQL hostname */

define( 'DB_HOST', 'mysql.zzz.com.ua' );
```

Рисунок 3.3 – Редагування файлу wp-config.php

Оскільки, серед функціональних вимог до web-додатку була наявність онлайн-запису, було вирішено за шаблон використати плагін Booking Wordpress Plugin, але він не виконує всіх потрібних функцій, тому деякі з файлів було відредаговано та дописано необхідні фрагменти коду. Через те, що плагін містив в собі можливість вибору дати та часу, але не можливо було обрати майстра, який буде виконувати процедуру, було дописано фрагмент коду, який за це відповідає в файл Assistant.php (рис. 3.4).

```
class SLN_Shortcode_SalonAssistant
{
    const NAME = 'salon_booking_assistant';

    private $plugin;
    private $attrs;

    function __construct(SLN_Plugin $plugin, $attrs)
    {
        $this->plugin = $plugin;
        $this->attrs = $attrs;
    }

    public static function init(SLN_Plugin $plugin)
    {
        add_shortcode(self::NAME, array(__CLASS__, 'create'));
    }

    public static function create($attrs)
    {
        SLN_TimeFunc::startRealTimezone();

        $obj = new self(SLN_Plugin::getInstance(), $attrs);

        $ret = $obj->execute();
        SLN_TimeFunc::endRealTimezone();
        return $ret;
    }
```

Рисунок 3.4 – Створення файлу Assistant.php

Також було відредаговано файл attendants.php, бо в даній частині також треба було вставити фрагмент коду, щоб був можливий вибір майстра (рис. 3.5).

```
$isChooseAttendantForMeDisabled = $plugin->getSettings()->isChooseAttendantForMeDisabled();

$tmp = '';
$i = 0;
foreach ($attendants as $attendant) {
    if ($attendant->hasServices($services)) {
        $errors = SLN_Shortcode_Salon_AttendantHelper::validateItem($bookingServices->getItems(), $ah, $attendant);

    if (!$i && $isChooseAttendantForMeDisabled) {
        $tmp .= SLN_Shortcode_Salon_AttendantHelper::renderItem($size, $errors, $attendant, null, true);
    } else {
        $tmp .= SLN_Shortcode_Salon_AttendantHelper::renderItem($size, $errors, $attendant);
    }

        $hasAttendants = true;
    $i++;
    }
}
if ($tmp && !$isChooseAttendantForMeDisabled) {
    $tmp = SLN_Shortcode_Salon_AttendantHelper::renderItem($size).$tmp;
}

?>
<div class="sln-attendant-list">
    <?php if ($tmp) : ?>
        <div class="row"><?php echo $tmp ?></div>
    <?php else: ?>
        <div class="alert alert-warning">
            <p><?php echo __(
                    'Нет мастеров для выбранного времени - выберите другое',
                    'salon-booking-system'
                ) ?></p>
        </div>
    <?php endif ?>
</div>
```

Рисунок 3.5 – Редагування файлу attendants.php

Оскільки статус замовлення був тільки «Підтверджено», було вирішено додати ще такі як «В очікуванні», «Відмінено», «Помилка» (рис. 3.6).

```
public static function init()

{

    self::$labels = array(

        self::PENDING   => __('В ожидании', 'salon-booking-system'),

        self::CANCELED  => __('Отменён', 'salon-booking-system'),

        self::CONFIRMED => __('Подтверждён', 'salon-booking-system'),

        self::ERROR     => __('Ошибка', 'salon-booking-system'),

    );

}
```

Рисунок 3.6 – Редагування файлу booking_status.php

Функціонал форми з можливістю залишити відгук про послуги студії також був обмеженим, так як можна було лише написати коментар та відправити його. Додана можливість відмітити рейтинг виконання послуги, а також відгук можна залишити тільки після того, як послуга буде виконана. Це було зроблено для уникнення спаму (рис. 3.7).

```
<div id="ratingModal" class="modal fade" role="dialog" tabindex="-1">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title"></h4>
            </div>
            <div class="modal-body">
                <div id="step1">
                    <p><?php _e('Добрый день','salon-booking-system');?> <?php echo $data['user_name'] ?>!</p>
                    <p><?php _e('Как вы оцените обслуживание нашей командой в этот раз?','salon-booking-system');?></p>
                    <p><textarea id="" placeholder="<?php _e('Пожалуйста, напишите нам несколько строк, чтобы понять, соответствует ли ваш опыт вашим ожиданиям','salon-booking-sys
                    <p>
                    <div class="rating" id="<?php echo $item['id']; ?>"></div>
                    <span><?php _e('Оцените нашу команду!','salon-booking-system');?></span>
                    </p>
                    <p>
                        <button type="button" class="btn btn-primary" onclick="slnMyAccount.sendRate();"><?php _e('Отправить отзыв','salon-booking-system');?></button>
                        <button type="button" class="btn btn-default" data-dismiss="modal"><?php _e('Отменить','salon-booking-system');?></button>
                    </p>
                </div>
                <div id="step2">
                    <p><?php _e('Спасибо за ваш отзыв. Это поможет нам улучшить наши услуги.','salon-booking-system');?></p>
                    <p><?php _e('Надеемся увидеть Вас ещё в имидж-студии ','salon-booking-system');?> <?php echo $data['gen_name']; ?></p>
                </div>
            </div>
            <div class="modal-footer"></div>
        </div>
    </div>
</div>
```

Рисунок 3.7 – Редагування файлу my_account_details.php

Ще була розроблена форма зворотнього зв’язку зі студією без використання плагінів. Для цього було відредаговано файл function.php з папки, де знаходиться використовувана тема (рис. 3.8).

```
add_shortcode( 'art_feedback', 'art_feedback' );
function art_feedback() {

    ob_start();
    ?>
    <form id="add_feedback">
        <input type="text" name="art_name" id="art_name" class="required art_name" placeholder="Имя" value=""/>

        <input type="email" name="art_email" id="art_email" class="required art_email" placeholder="Ваш E-Mail"

        <input type="text" name="art_subject" id="art_subject" class="art_subject" placeholder="Тема" value=""/:

        <textarea name="art_comments" id="art_comments" placeholder="Введите ваше обращение" rows="10" cols="30'

        <input type="checkbox" name="art_anticheck" id="art_anticheck" class="art_anticheck" style="display: no

        <input type="text" name="art_submitted" id="art_submitted" value="" style="display: none !important;"/>

        <input type="submit" id="submit-feedback" class="button" value="Отправить"/>
    </form>
    <?php

    return ob_get_clean();
}
```

Рисунок 3.8 – Редагування файлу function.php

Далі потрібно було налаштувати зовнішній вид форми в файлі style.css активної теми (рис. 3.9).

```
#add_feedback {
    margin: 20px 0 0;
    position: relative;
}

#art_name, #art_email, #art_subject, #art_comments {
    padding: 10px 5px;
    display: inline-block;
    width: 49.548%;
    border-radius: 3px;
    border: 1px solid #ddd;
    font-size: 0.9em;
}

#art_subject {
    width: 100%;
    margin: 5px 0;
}

#art_comments {
    width: 100%;
}
```

Рисунок 3.9 – Налаштування зовнішнього виду в файлі style.css

Наступним кроком було створено файл feedback.js та прописано в ньому функціонал форми (рис. 3.10).

```
jQuery(document).ready(function ($) {
    var add_form = $('#add_feedback');

    // Сброс значений полей
    $('#add_feedback input, #add_feedback textarea').on('blur', function () {
        $('#add_feedback input, #add_feedback textarea').removeClass('error');
        $('.error-name,.error-email,.error-comments,.message-success').remove();
        $('#submit-feedback').val('Отправить сообщение');
    });

    // Отправка значений полей
    var options = {
        url: feedback_object.url,
        data: {
            action: 'feedback_action',
            nonce: feedback_object.nonce
        },
        type: 'POST',
        dataType: 'json',
        beforeSubmit: function (xhr) {
            // При отправке меняем надпись на кнопке
            $('#submit-feedback').val('Отправляем...');
        },
```

Рисунок 3.10 – Створення файлу feedback.js

Завдяки тому, що було обрано використати систему управління контентом WordPress, адміністративна панель була вже створена системою (рис. 3.11), що значно скорочує час створення web-додатку.

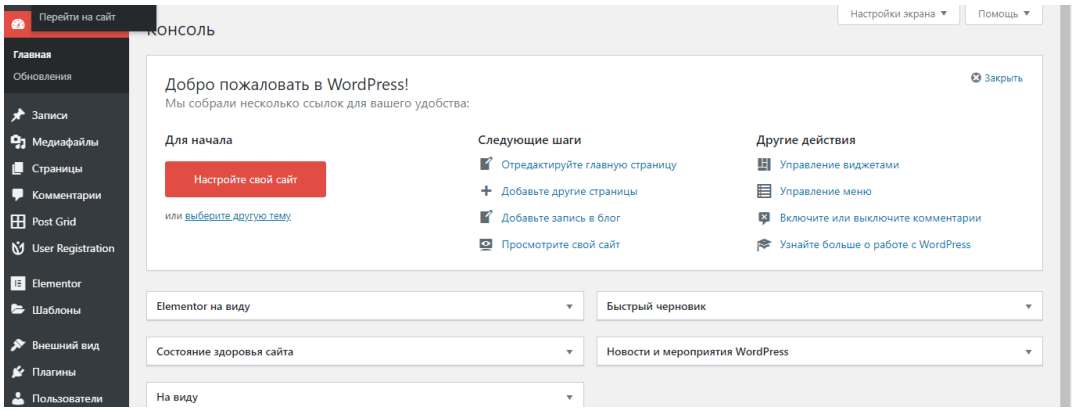

Рисунок 3.11 – Адміністративна панель

Також реалізована адаптивність web-додатку, що дає змогу додатку правильно відображатись на різних пристроях з різними характеристиками. На рисунку 3.12 показано адаптивність додатку під смартфон, що задовольняє вимоги замовника.

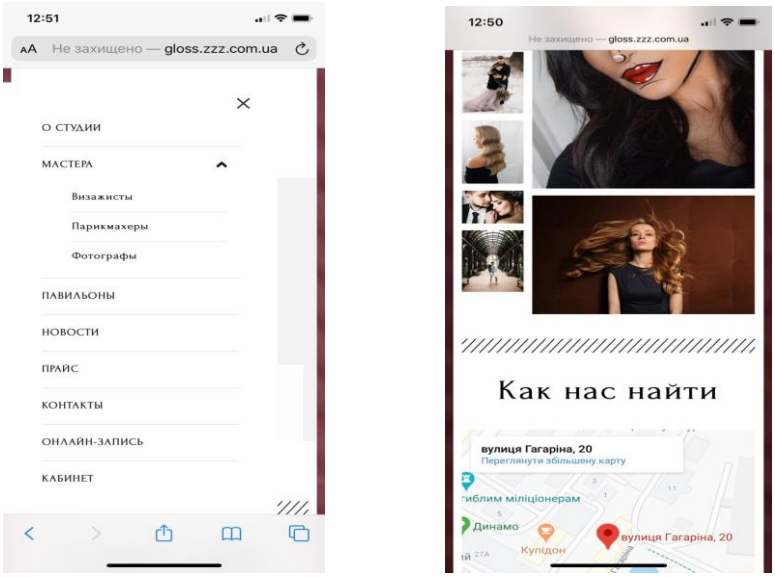

Рисунок 3.12 – Інтерфейс web-додатку на смартфоні

3.2 Результат реалізації web-додатку

Після того, як користувач вводить URL адресу додатку, а саме http:// http://gloss.zzz.com.ua/, користувач знаходиться на Головній сторінці web-додатку (рис. 3.13). На даній сторінці розміщений слайдер з новинами студії, галерея робіт всіх майстрів студії, а також карта відміткою місця розташування імідж-студії «Глянець» (рис. 3.14).

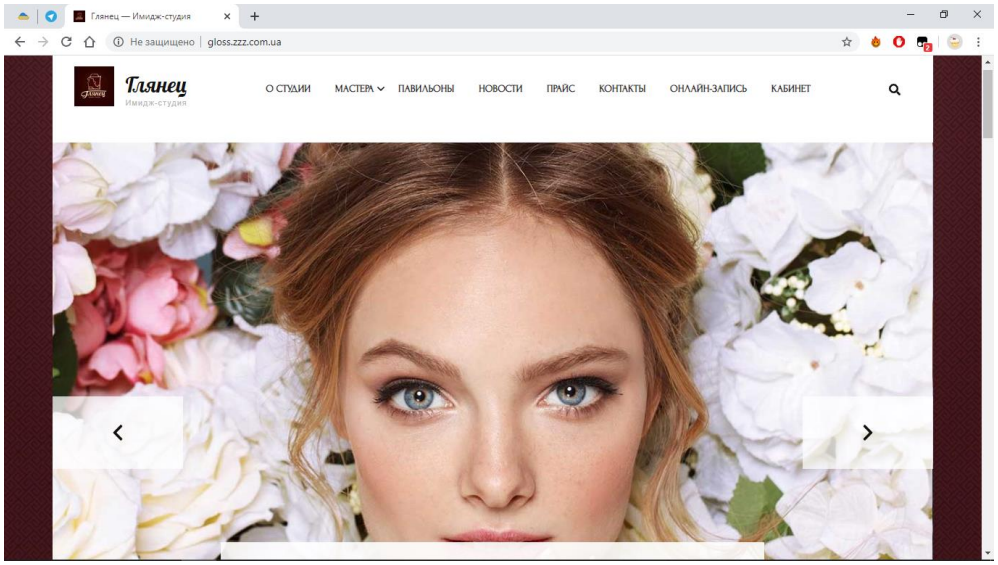

Рисунок 3.13 – Головна сторінка

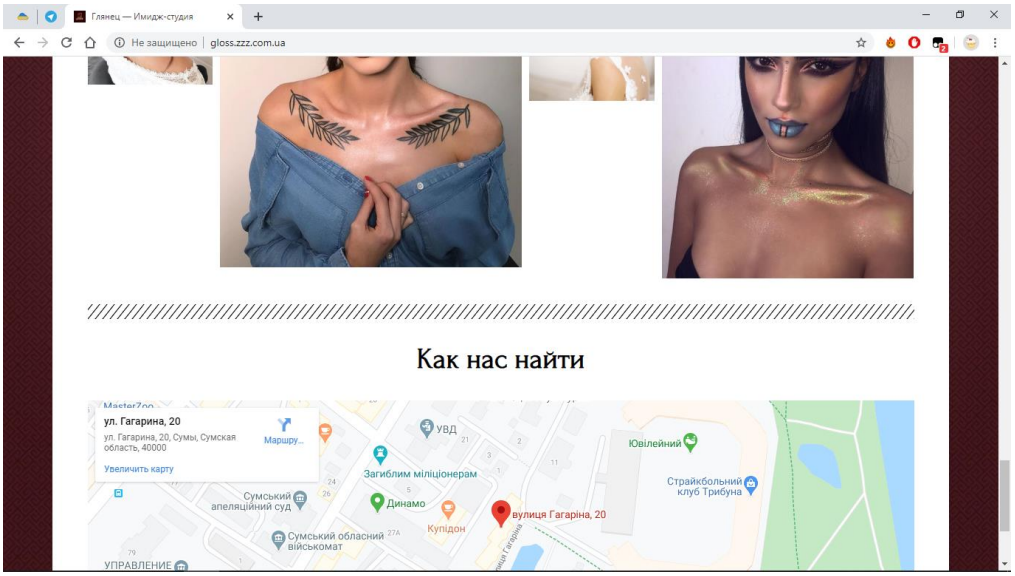

Рисунок 3.14 – Головна сторінка, розміщення студії

Наступна сторінка «О студии» (рис. 3.15), де користувач має змогу ознайомитись з описом, перевагами та фотографіями імідж-студії.

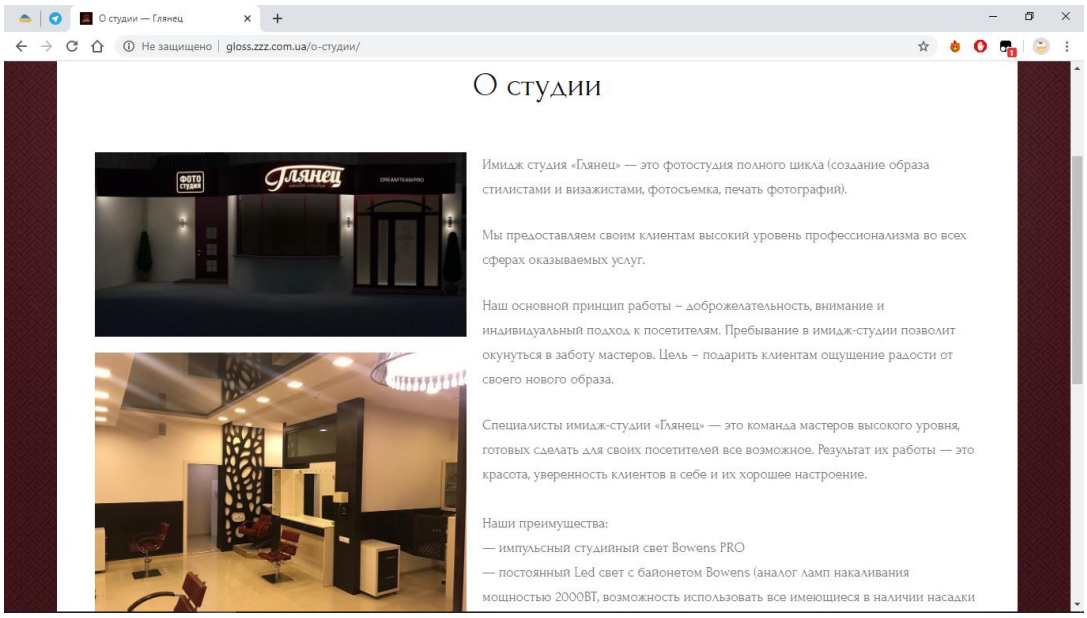

Рисунок 3.15 – Сторінка «О студии»

Далі сторінка «Мастера» (рис. 3.16). На ній розміщені фото майстрів та їх імена, щоб потенційний клієнт міг переглянути скільки майстрів працює в кожній категорії. Якщо в випадаючому меню обрати сторінку «Визажисты» (рис.3.17), користувач зможе ознайомитись з майстрами-візажистами, переглянути їх біографію та портфоліо робіт. При виборі сторінки «Парикмахеры» (рис. 3.18) знаходиться інформація про майстрів-перукарів, а на сторінці «Фотографы» (рис. 3.19) інформація про фотографів. Дані сторінки реалізовані в одному стилі, що дуже зручно для користувача, оскільки про кожного майстра інформація викладена за одним шаблоном.

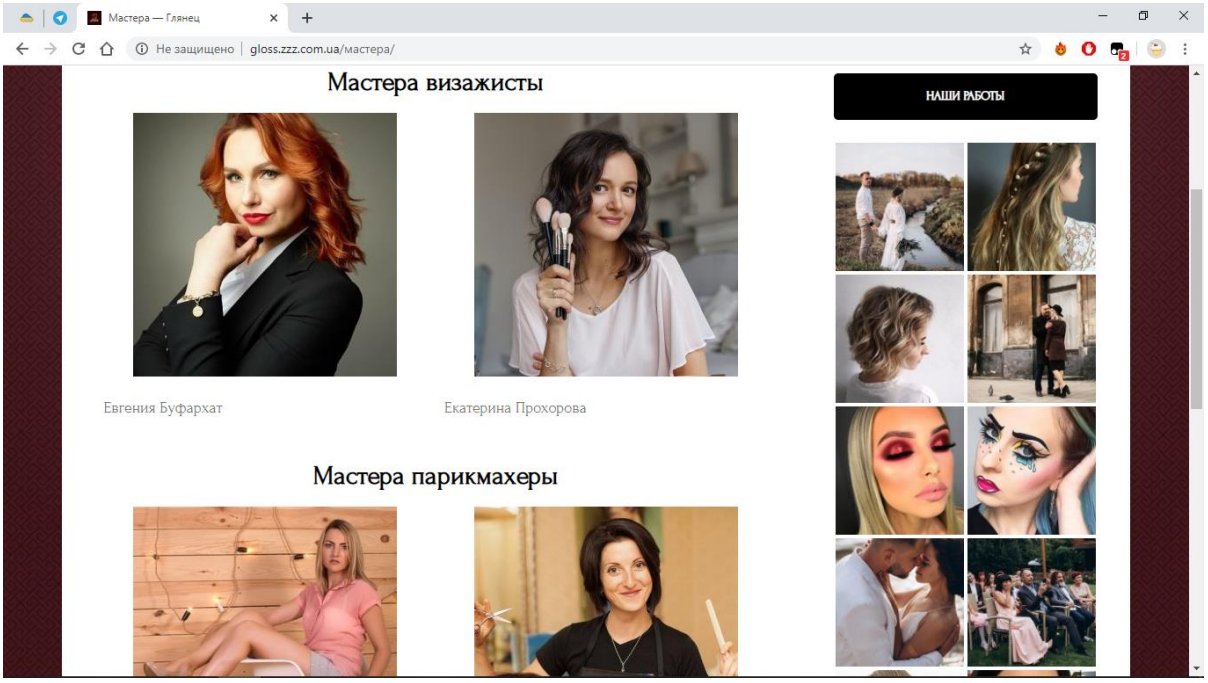

Рисунок 3.16 – Сторінка «Мастера»

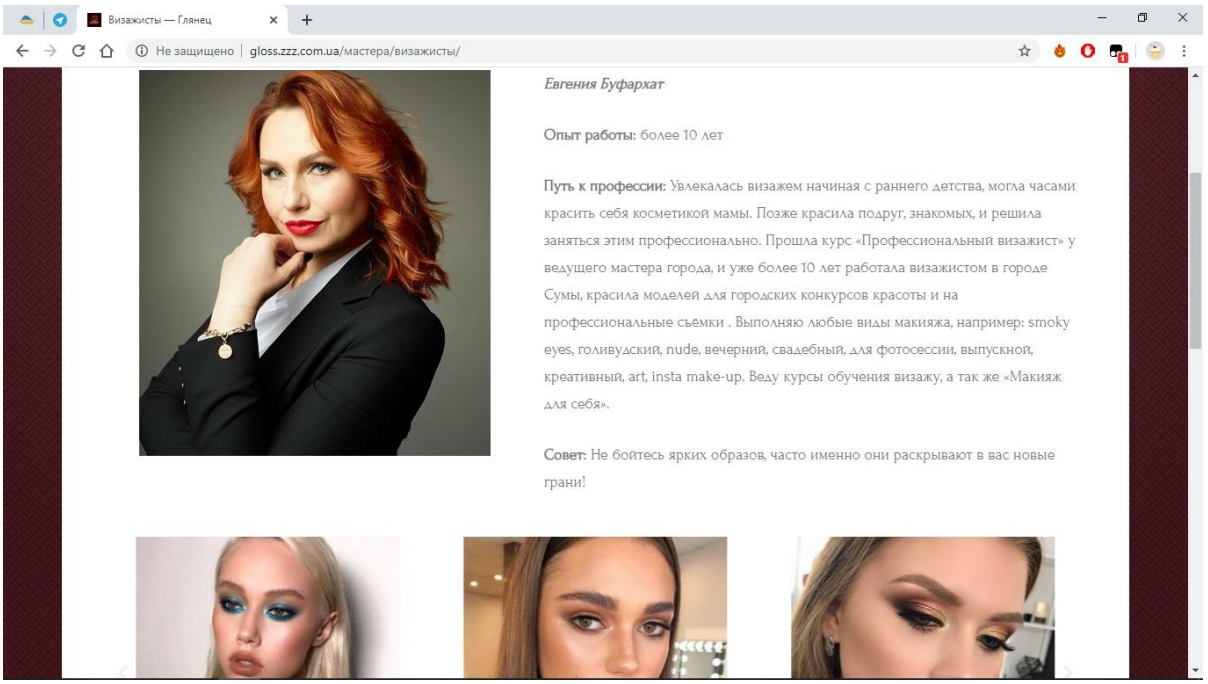

Рисунок 3.17 – Сторінка «Визажисты»

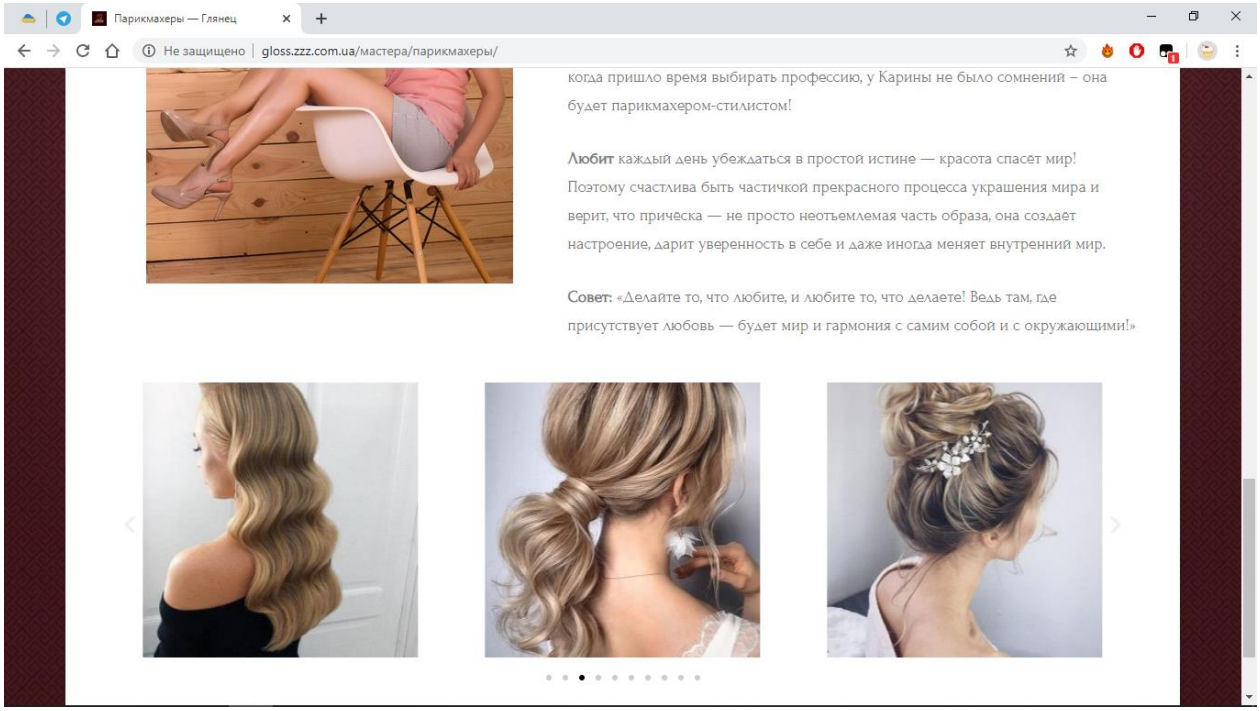

Рисунок 3.18 – Сторінка «Парикмахеры»

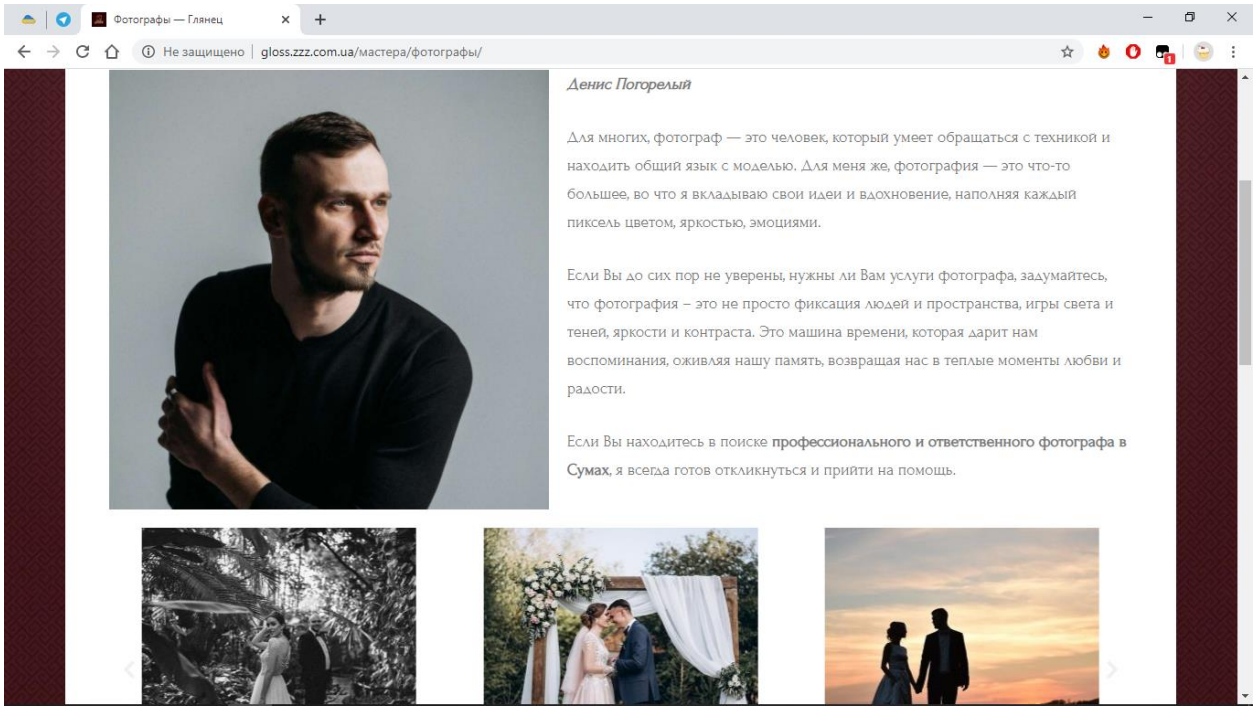

Рисунок 3.19 – Сторінка «Фотографы»

Оскільки імідж-студія «Глянець» має два зали для зйомок, то замовником було вирішено додати окрему сторінку «Павильоны» (рис. 3.20). На ній розміщена інформація про кожен з залів, їх особливості, та наведені фотографії, де фотографи працюють там.

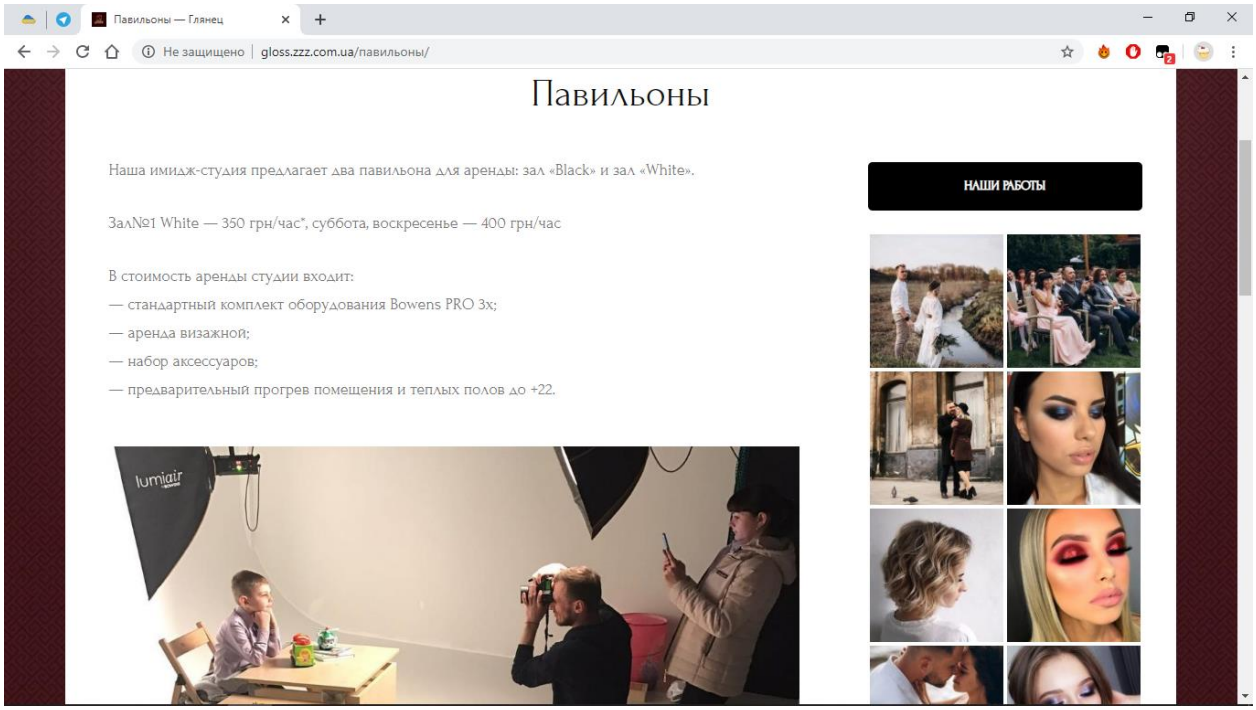

Рисунок 3.20 – Сторінка «Павильоны»

Сторінка «Новости» – сторінка для новин та інформативних статей, де відображаються записи, створювані адміністратором (рис 3.21). Записи відображаються на сторінці двома колонками, де міститься головне зображення та частина тексту. При натисканні користувач переходить на сторінку з повною статтею (рис. 3.22).

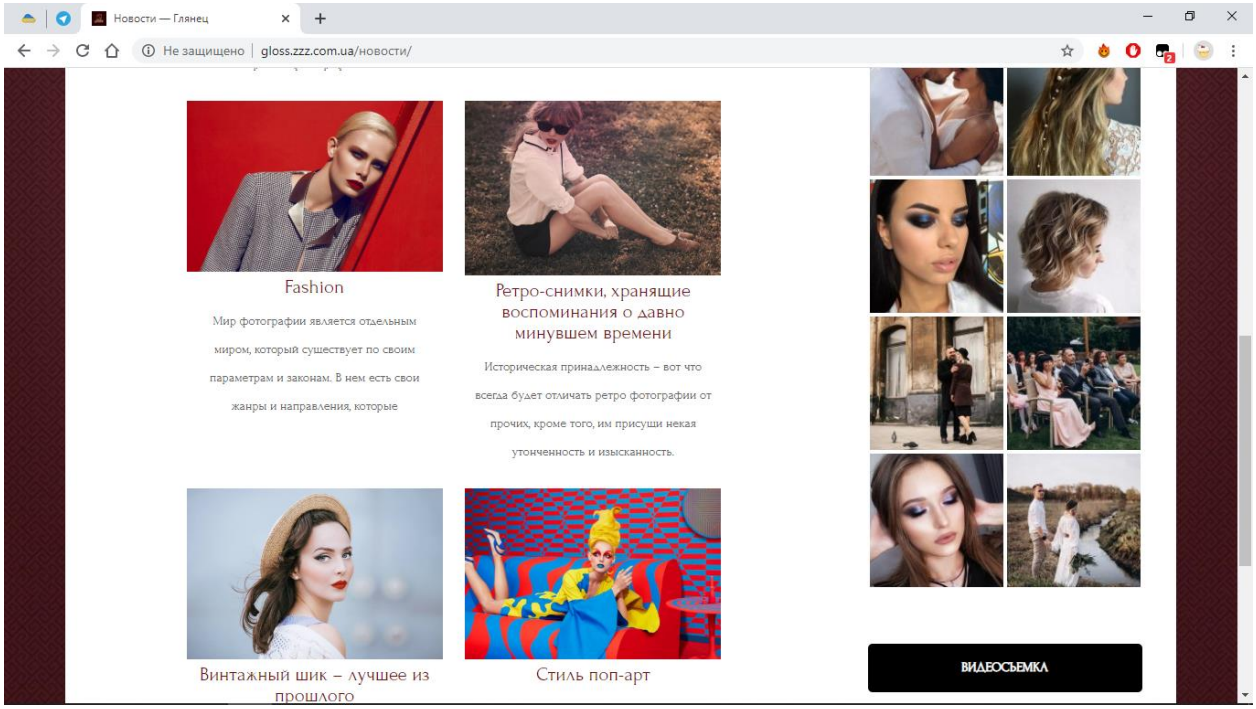

Рисунок 3.21 – Сторінка «Новости»

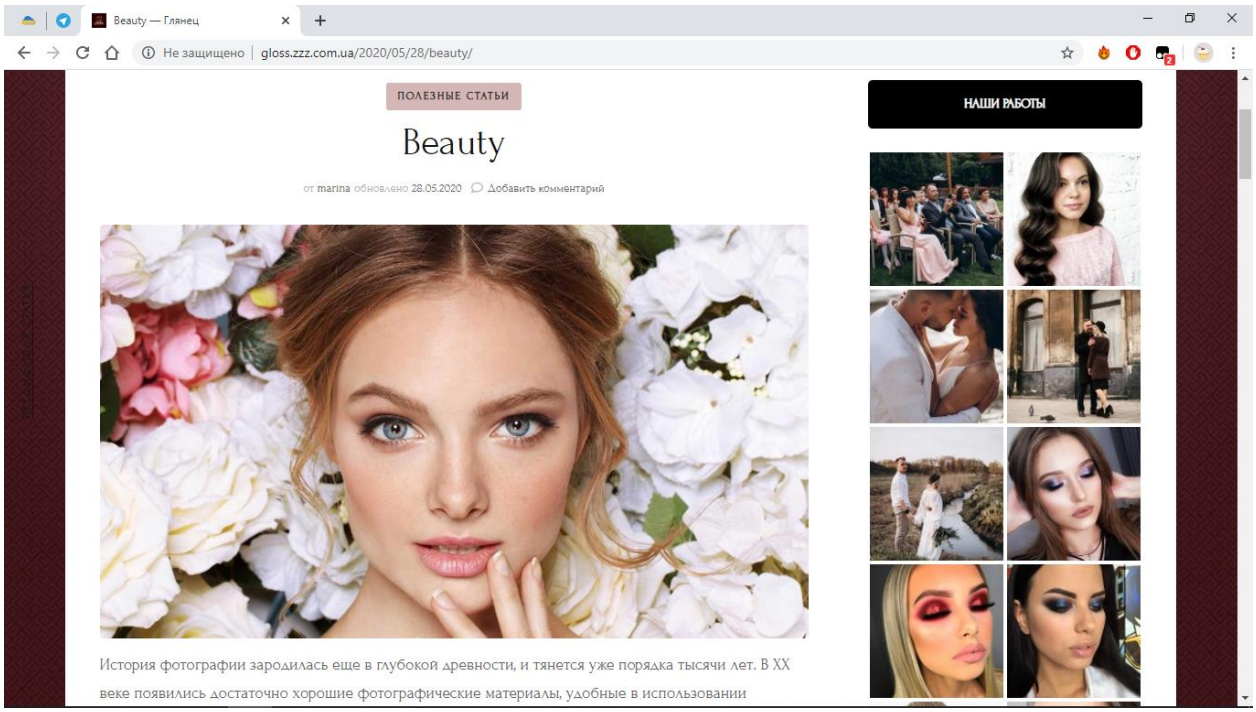

Рисунок 3.22 – Сторінка з повною статтею

Наступна сторінка «Прайс», де користувач має можливість ознайомитися з спектром послуг, які надає студія, та цінами та них. Сам прайс-лист був створений за допомогою програми Adobe Photoshop CC 2017. Сторінка «Прайс» зображена на рисунку 3.23.

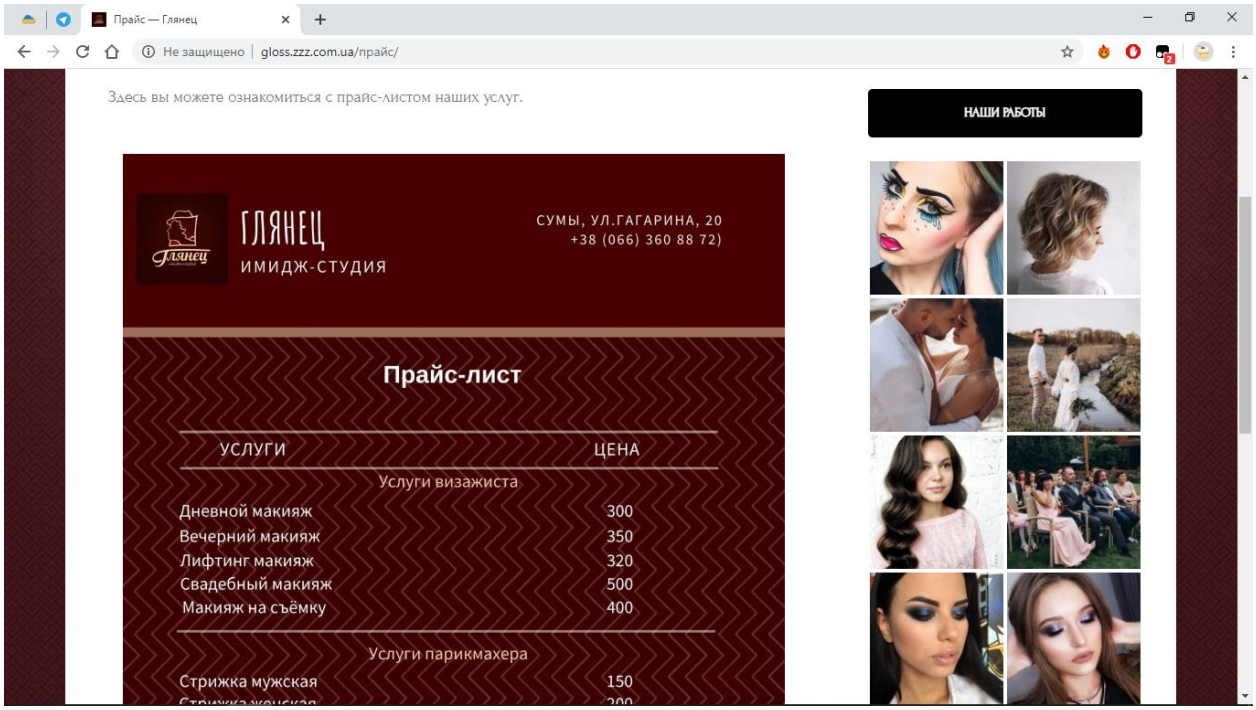

Рисунок 3.23 – Сторінка «Прайс»

Для зручного зв’язку користувача зі студією була створена сторінка «Контакты» (рис. 3.24). На даній сторінці відображені номера телефону імідж-студії, адреса, а також реалізована форма зворотного зв’язку. На рисунку 3.25 зображено лист, надісланий з форми.

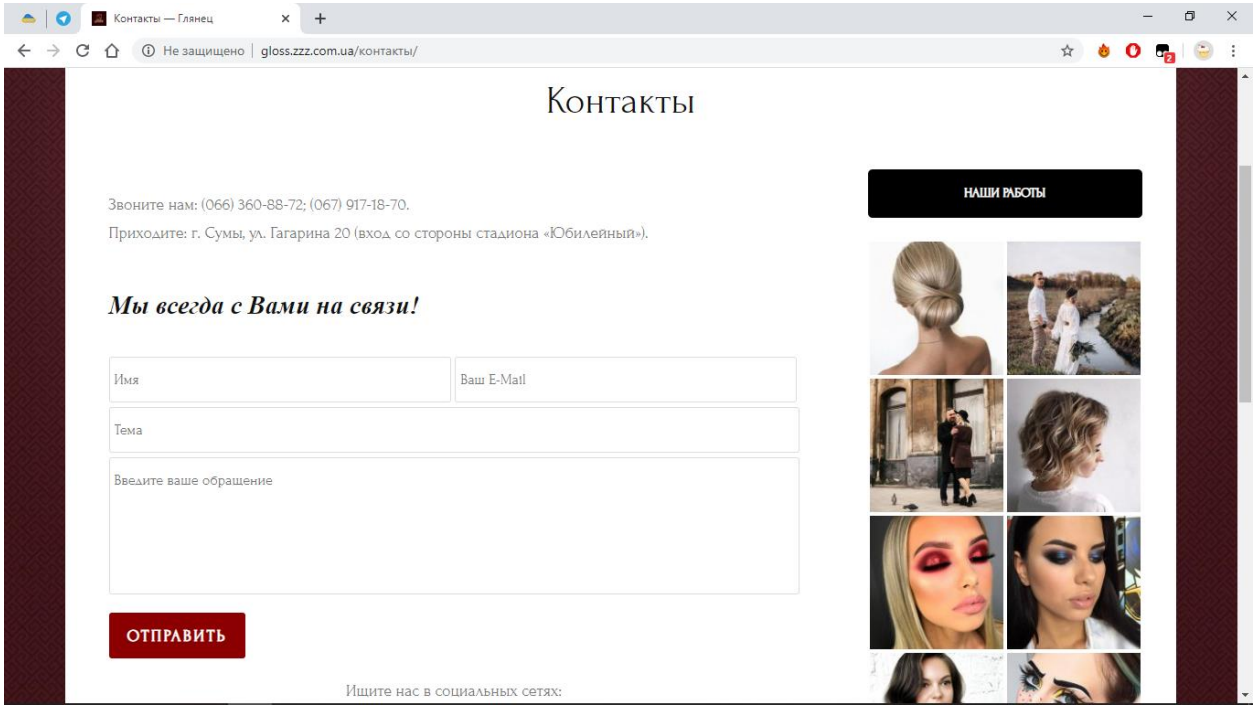

Рисунок 3.24 – Сторінка «Контакты»

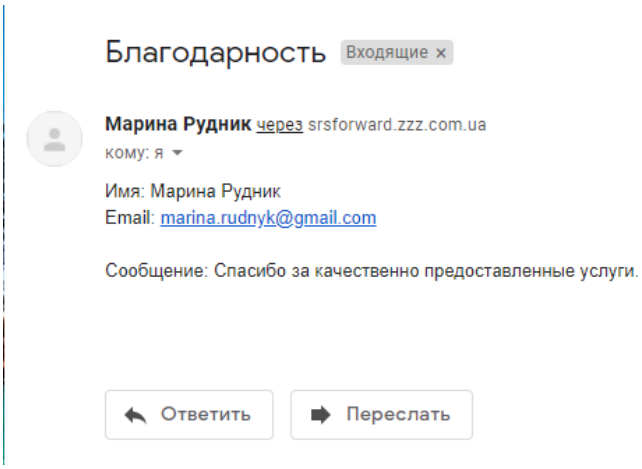

Рисунок 3.25 – Лист з форми

Наступний пункт меню – «Онлайн-запись». На даній сторінці користувач бачить інформацію про вхід в додаток, якщо він не авторизований, то на місці панелі буде форма входу (рис. 3.26). Але якщо користувач ще не зареєстрований, то по натисканню на відповідну кнопку, він переходить на

сторінку реєстрації (рис. 3.27). Також тут клієнт може здійснити онлайн-запис (рис. 3.28). Користувач сам обирає послугу, майстра, зручну дату та час.

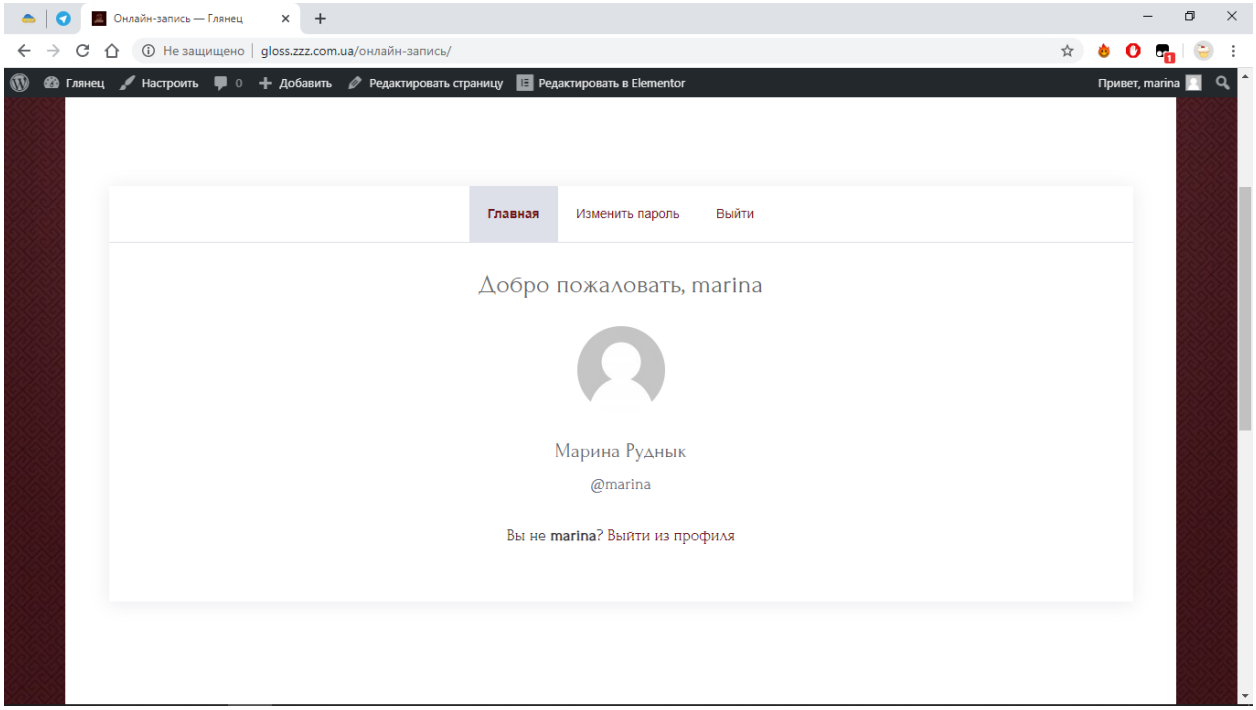

Рисунок 3.26 – Інформація про користувача

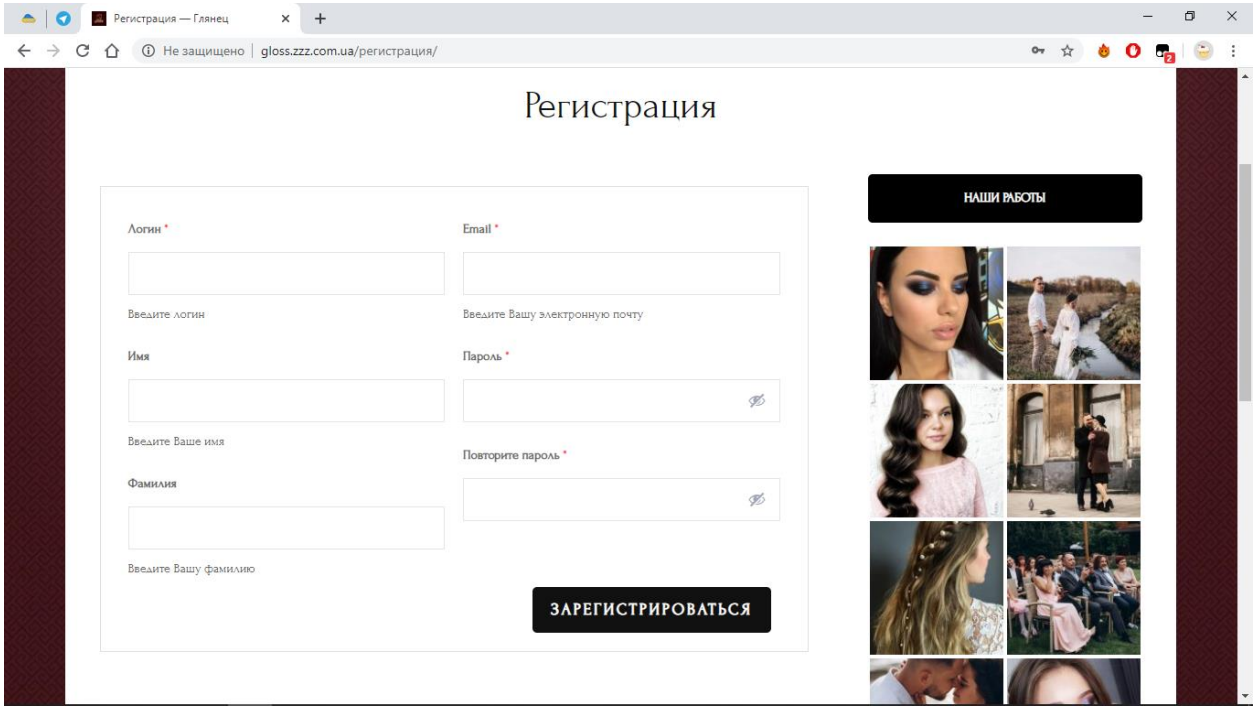

Рисунок 3.27 – Сторінка реєстрації

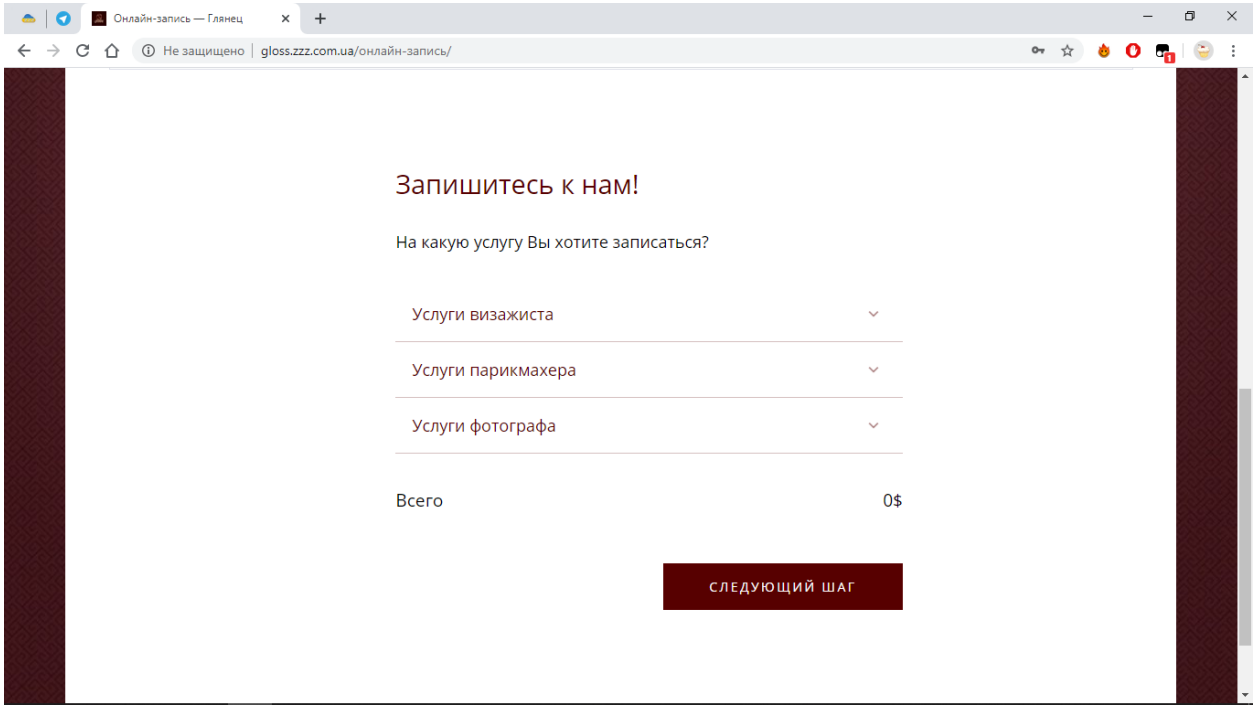

Рисунок 3.28 – Онлайн-запис

Після здійснення запису до імідж-студії користувач автоматично переходить на сторінку з подякою від студії (рис. 3.29).

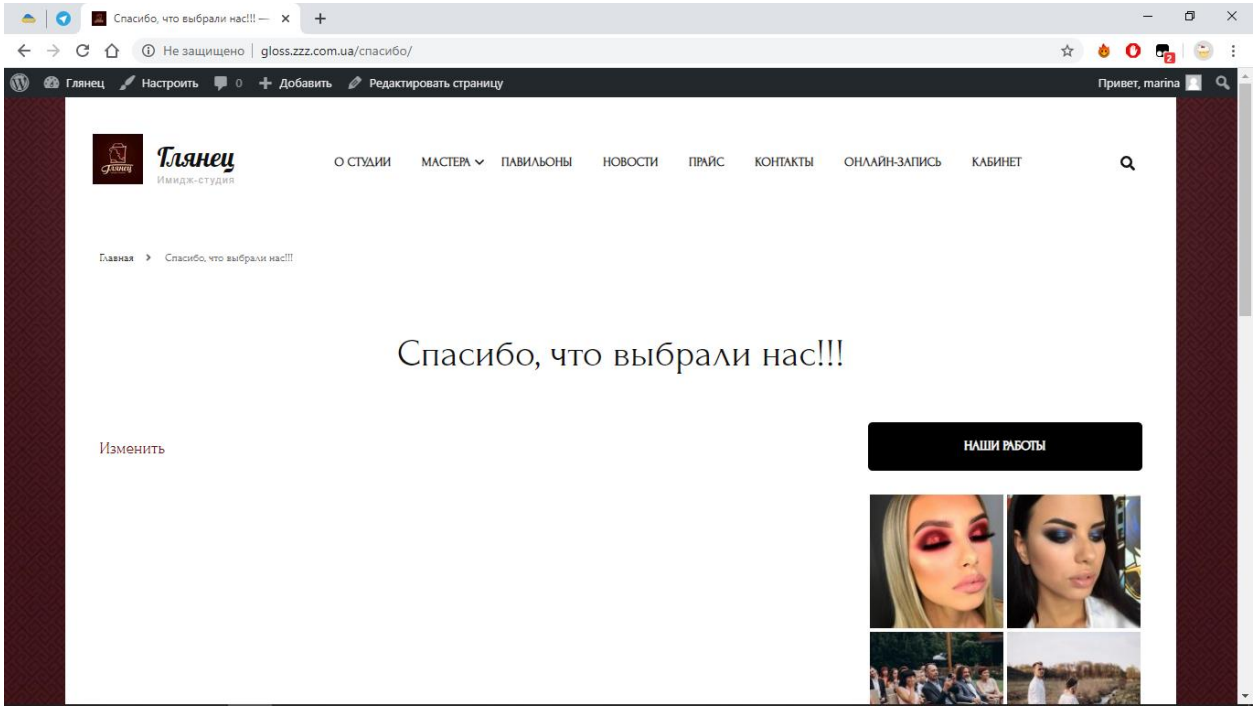

Рисунок 3.29 – Сторінка з подякою

Після того, як користувач здійснить онлайн-запис, замовлення набуває статусу «В ожидании» та потрапляє на панель адміністратора. В свою чергу адміністратор може прийняти, відхилити замовлення або змінити дату та час (рис. 3.30).

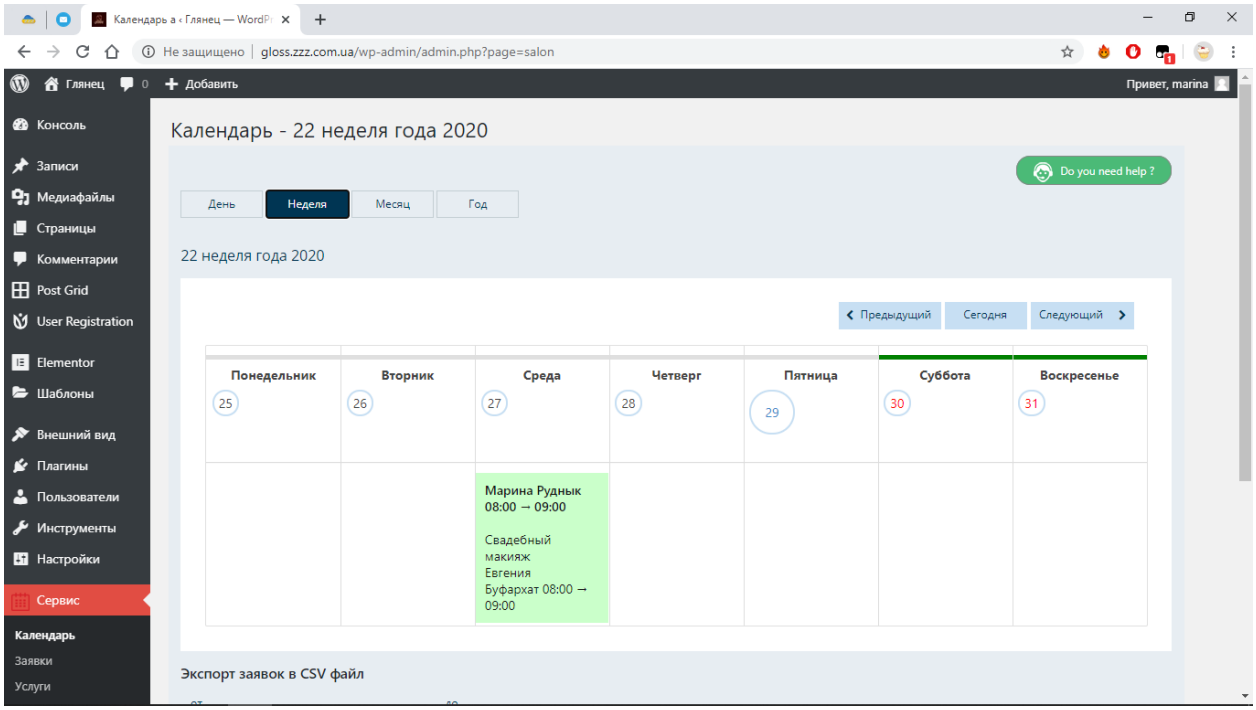

Рисунок 3.30 – Список замовлень на адміністративній панелі

Останній пункт меню – «Кабинет» (рис. 3.31). Тут користувач може переглядати інформацію про записи на послуги в студію. Серед цієї інформації: номер запису, дата та час, послуга, майстер, який буде виконувати послугу, сума та статус обробки запису. Також після того як час виконання послуги завершиться, буде доступна функція «Залишити відгук» (рис. 3.32).

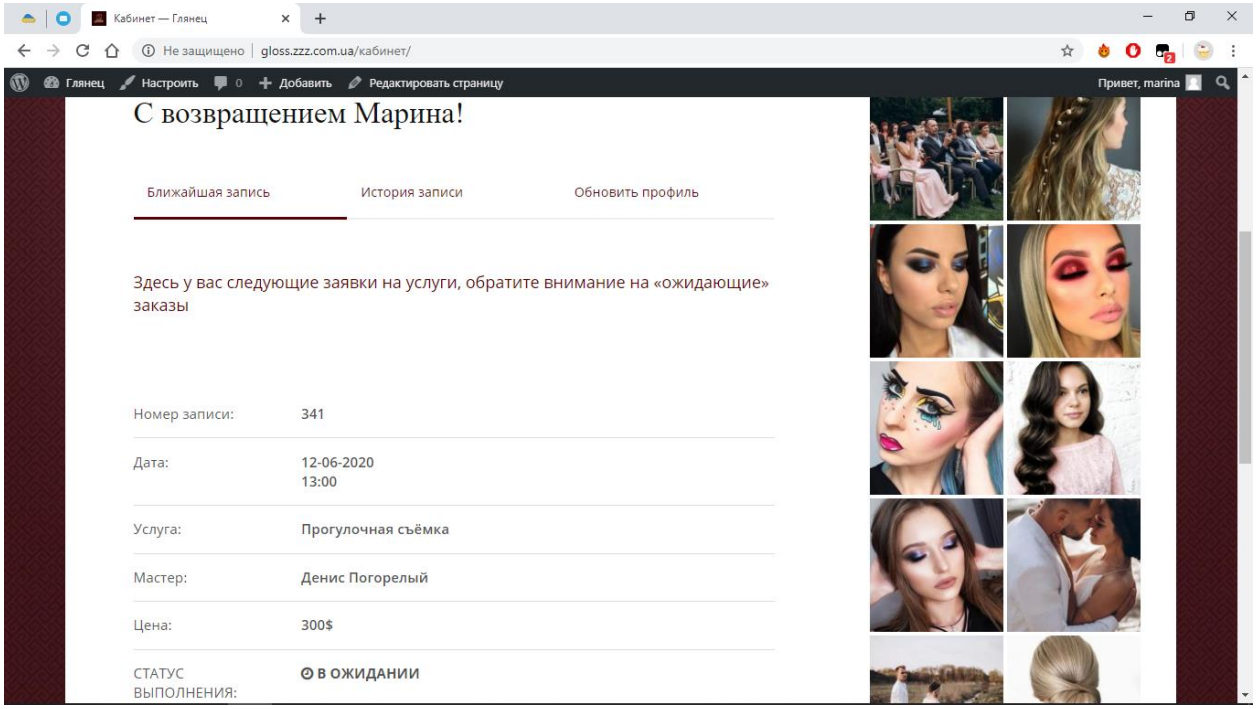

Рисунок 3.31 – Сторінка «Кабинет»

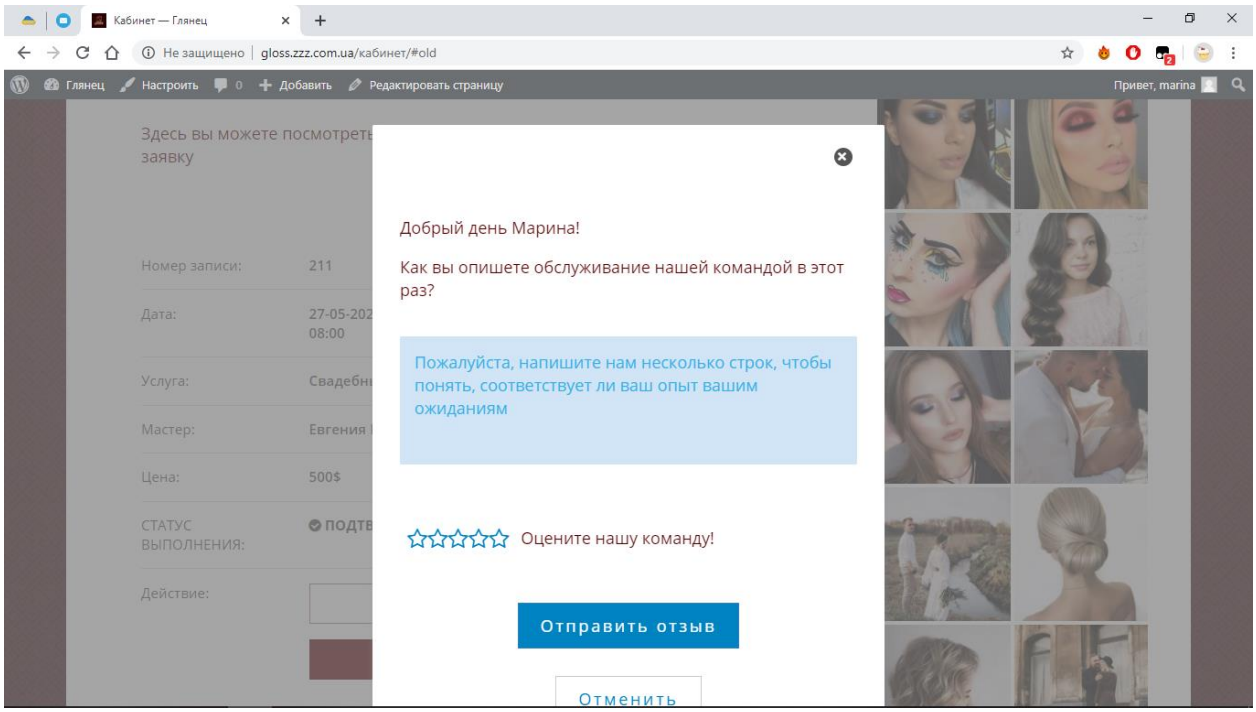

Рисунок 3.32 – Відгук

ВИСНОВОК

В результаті виконання кваліфікаційної роботи бакалавра було розроблено web-додаток імідж-студії «Глянець», який відповідає всім вимогам описаним в Технічному завданні.

Актуальність розроблення web-додатку імідж-студії обґрунтовується подальшим розширенням клієнтської бази, активним ростом рівня обслуговування клієнтів. Завдяки охопленню великої частини цільової аудиторії можливе стрімке збільшення прибутків для власника бізнесу.

Визначено основні функції, які повинен виконувати web-додаток, серед яких ознайомлення клієнтів з beauty-майстрами та фотографами студії, перегляд портфоліо робіт та онлайн-запис до майстрів або фотографа.

Було обрано інструментарій для реалізації web-додатку. Даними інструментами стали CMS WordPress, мова програмування PHP та СУБД MySQL для роботи з базою даних. Відповідно до структурно-функціонального моделювання було побудовано модель процесу підтримки роботи імідж-студії за допомогою створюваного web-додатку.

Основним функціоналом web-додатку імідж-студії «Глянець» є:

- ознайомлення клієнтів з майстрами та фотографами студії;
- перегляд портфоліо персоналу студії;
- інформування клієнтів про новини студії та можливі акції;
- перегляд всіх послуг, які надає студія та прайс-листа на них;
- онлайн-запис до майстрів або фотографа;
- наведення контактів студії;
- інформативні статті для клієнтів;
- адаптивність додатку під телефон та персональний комп’ютер ;
- реєстрація та вхід в особистий кабінет;
- можливість відслідковувати статус виконання замовлення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Maija [Електронний ресурс] – Режим доступу до ресурсу: http:// maija.ua/
2. Mirasalon [Електронний ресурс] – Режим доступу до ресурсу: https://www. mirasalon.com.ua/
3. Stella [Електронний ресурс] – Режим доступу до ресурсу: http://stella.in.ua/
4. Ultimate Guide: How to Make a Website in 2020 [Електронний ресурс] – Режим доступу до ресурсу: https://www.wpbeginner.com/guides/
5. WordPress Tutorial [Електронний ресурс] – Режим доступу до ресурсу: https://www.hostinger.com/tutorials/wordpress/
6. Какие плюсы и минусы WordPress? [Електронний ресурс] – Режим доступу до ресурсу: https://www.templatemonster.com/ru/faq/plus-i-minus-wordpress/
7. PHP: Что такое PHP? [Електронний ресурс] – Режим доступу до ресурсу: https://www.php.net/manual/ru/intro-whatis.php
8. Nexcess [Електронний ресурс] – Режим доступу до ресурсу: https://help.nexcess.net/74113-siteworx/what-is-phpmyadmin
9. Методология IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: https://itteach.ru/bpwin/metodologiya-idef0
10. Методологія idef0 [Електронний ресурс] – Режим доступу до ресурсу: https://studfiles.net/preview/5203249/page:2/
11. What is the use case? [Електронний ресурс] – Режим доступу до ресурсу: https://searchsoftwarequality.techtarget.com/definition/use-case
12. UML Use Case Diagram Tutorial [Електронний ресурс] – Режим доступу до https://www.lucidchart.com/pages/uml-use-case-diagram
13. Build an e-commerce application using Laravel and Vue [Електронний ресурс] – Режим доступу до ресурсу: https://blog.pusher.com/ecommerce-laravel-vue-part-1/

14. Мова програмування PHP [Електронний ресурс] – Режим доступу до ресурсу: http://sevidi.ru/php/phppage4.php

15. Розробка та просування сайту [Електронний ресурс] – Режим доступу до ресурсу: impulse-design.com.ua

16. К. Пэтчетт, М.Райт CGI/Perl: створення програм для Web BHV-Киев, 2000

17. С.Н. Коржинский настільна книга Web-майстра: ефективне застосування CSS, HTML і JavaScript. КноРус, 2000

18. JavaScript: The Definitive Guide S.Spainhour, R.Eckstein Webmaster in Nutshell. 1nd Edition O'Reilly, 1999

19. Л. Паттерсон.: Використання HTML 4.0. Ясно, коротко, надійно Диалектика, 1999

20. Хольцнер Стивен. Библия программиста. Ajax. — М.: Діалектика, 2009. — С. 556. — ISBN 978-5-8459-1502-3.

21. Дейв Крейн, Джордон Сонневельд., Бер Бибо.:Ajax на практиці. — М.: Вільямс, 2007. — ISBN 978-5-8459-1327-2.

22. MySQL — система управления базами данных [Електронний ресурс] – режим доступу: https://web-creator.ru/articles/mysql

23. Диаграммы вариантов использования [Електронний ресурс] – режим доступу: https://itteach.ru/rational-rose/diagrammi-variantov-ispolzovaniya/

Додаток А. Технічне завдання

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Web-додаток для імідж-студії “Глянець”»

Суми 2020

1. Призначення й мета створення web-додатку

1.1 Призначення web-додатку

Інформаційна система має надавати повноцінну інформацію клієнтам про імідж-студію «Глянець», рекламувати її та полегшити запис до майстрів студії.

1.2 Мета створення web-додатку

Збільшення кількості клієнтів імідж-студії «Глянець», за рахунок використання web-додатку.

1.3 Цільова аудиторія

До цільової аудиторії web-додатку можна віднести практично всіх людей, що зацікавлені у послугах візажиста, перукаря або фотографа, завдяки широкому списку послуг, які надає імідж-студія.

2 Вимоги до web-додатку

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури й функціонування web-додатку

Web-додаток має бути доступним в мережі Інтернет під доменним іменем gloss.zzz.com.ua. Web-додаток повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2 Вимоги до персоналу

Від персоналу не має вимагатися особливих технічних навичок для підтримки й експлуатації web-додатку, окрім загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером, а також потрібно бути ознайомленим з інтерфейсом CMS WordPress та мати основні навички роботи з нею.

2.1.3 Вимоги до збереження інформації

Уся інформація надана у web-додатку буде зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним.

Відповідно до прав доступу до інформації у web-додатку, усіх користувачів можна поділити на відвідувачів та адміністратора.

Відвідувачі можуть переглядати усі сторінки web-додатку, ознайомитись з видами послуг та замовити фотосесію або послуги візажиста, перукаря.

Адміністратор може редагувати зовнішній вигляд та наповнення web-додатку.

Панель адміністратора створюється автоматично завдяки використанню системи управління контентом WordPress, а доступ до неї здійснюється за унікальним логіном і паролем.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

Структура web-додатку являє собою набір сторінок, які також є пунктами головного меню.

Такими розділами є:

Головна – на сторінці зображені головне меню та карусель з акціями та пропозиціями від студії.

Про студію – основна інформація про імідж-студію.

Наші майстри – інформація та портфоліо кожного з майстрів студії.

Розцінки – інформація про ціну кожного виду послуг.

Павільйони – інформація про фотозони студії.

Новини – останні, цікаві події пов’язані з діяльністю студії.

Контакти – список всіх можливих видів зв’язку з фотографом.

Онлайн-запис – запис до будь-якого з майстрів студії.

Кабінет – особистий кабінет користувача.

2.2.2 Навігація

Відповідно до бажаного замовником дизайну web-додатку, для навігації, у шапці буде створена система контент меню. Меню необхідне для швидкого переміщення користувача по усім доступним сторінкам. Меню буде відображатися на всіх сторінках, щоб відвідувач міг в будь-який момент часу перейти на будь-яку сторінку web-додатку.

2.2.3 Наповнення web-додатку (контент)

Для управління контентом web-додатку буде використана система CMS WordPress.

Заповнення та редагування контенту web-додатку має бути зроблено через панель керування, використовуючи інформацію з бази даних.

Всю інформацію для наповнення web-додатку має надавати імідж-студія, включаючи всі портфоліо майстрів та інформацію про їх роботу.

2.2.4 Загальні вимоги

Стиль web-додатку має бути сучасним, приємним для сприйняття, у якості основних кольорів було запропоновано використати червоні та білі відтінки, так як це кольори, які переважають в стилістиці студії.

Основою мають бути фотографії гарної якості, виконані фотографом студії, web-додаток має бути інтуїтивно зрозумілим для використання.

Розташування елементів на головній сторінці web-додатку схематично показано на рисунку А.1.

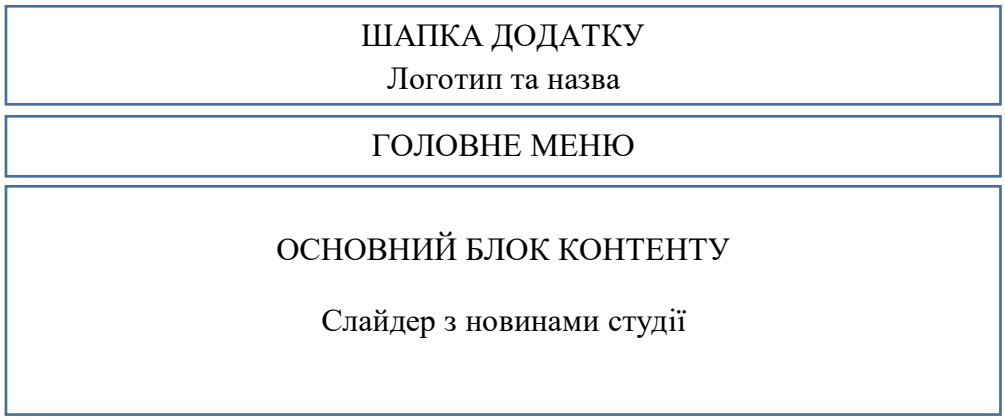

Рисунок А.1 – Схема головної сторінки

2.2.5 Система навігації (карта web-додатку)

Для повного розуміння структури web-додатку та зв’язку між його розділами та підрозділами було створено карту web-додатку.

Карта web-додатку зображена на рисунку А.2.

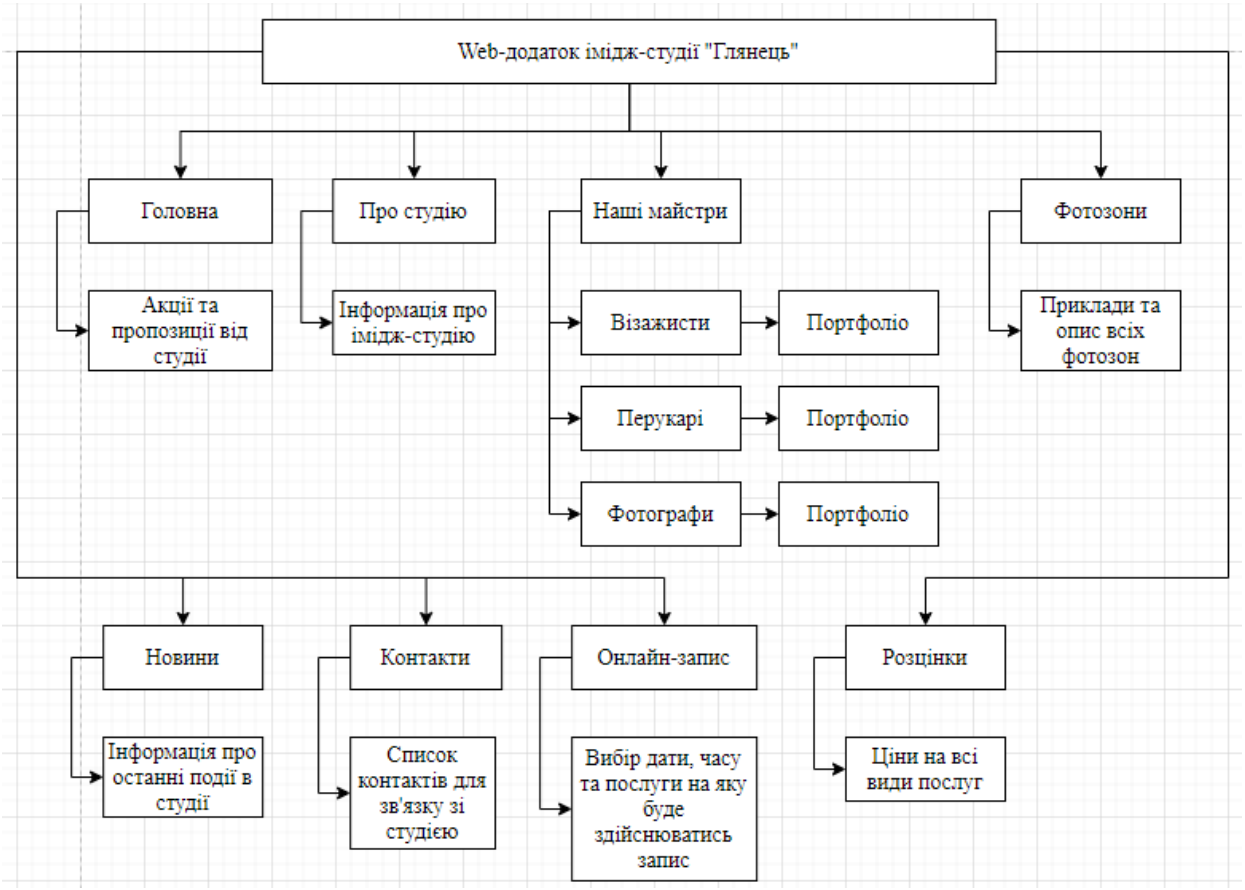

Рисунок А.2 – Карта web-додатку

2.3 Функціональні вимоги

Панель адміністратора повинна надавати можливість заповнювати та редагувати контент web-додатку. Мають бути реалізовані модулі зворотного зв’язку та онлайн-запису.

2.3.1 Функціональні можливості розділів

На головній сторінці будуть представлені наступні елементи:

- Головне меню;
- Кнопка онлайн-запису;
- Показ новин студії.

2.3.2 Типові навігаційні й інформаційні елементи

- Шапка web-додатку;
- Головне меню;
- Основне поле контенту;
- Сайдбар;
- Слайдер з новинами.

2.3.3 Потреби користувача

В цьому розділі представлено потреби та бажання виражені користувачем, на які очікується відповідь запропонованої системи, а також стисле представлення взаємодії на високому рівні між цільовою системою та різними учасниками.

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Онлайн-запис до майстра студії з вибором зручної дати та часу	Клієнт
UN-02	Перегляд портфоліо робіт майстрів	Клієнт
UN-03	Перегляд можливих фотозон студії	Клієнт
UN-04	Інформація про кожного з майстрів (досвід роботи, досягнення і т.д.)	Клієнт
UN-05	Перегляд акцій та пропозицій від студії	Клієнт
UN-06	Можливість зворотного зв’язку	Клієнт
UN-07	Редагування даних	Адміністратор
UN-08	Перегляд інформації про студію	Клієнт
UN-09	Перегляд новин імідж-студії	Клієнт
UN-10	Розрахунок орієнтовної вартості замовлення	Клієнт

2.3.4 Системні вимоги

Системні вимоги – це твердження, які зазвичай складаються розробником про те, як система повинна відповідати вимогам користувача.

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність модуля запису	M	Надає можливість клієнту здійснити онлайн-запис
SR-02	Каталог робіт	M	Формує портфоліо майстрів

SR-03	Каталог павільйонів	S	Формує можливість перегляду павільйонів
SR-04	База даних із замовленнями	M	Надає можливість відображати зайняті дати та дізнатися адміністратору про існуючі замовлення
SR-05	База даних з контентом	M	Відповідає за заповнення web-додатку контентом
SR-06	База даних з цінами на послуги	M	Надає можливість подальшого відображення цін на всі види послуг
SR-07	База даних з акціями	C	Відповідає за відображення акцій та пропозицій від імідж-студії
SR-08	Наявність модуля зворотного зв’язку	S	Надає можливість клієнту зв’язатися з адміністратором студії
SR-09	Панель адміністратора	M	Відповідає за подальше наповнення та редагування контенту адміністратором

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбувається з використанням:

- WordPress
- PHP 7.4.4
- MySQL 8.0

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додаток має бути виконаний на російській мові.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

3 Склад і зміст робіт зі створення web-додатку

Докладний опис етапів роботи зі створення web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка цілей необхідних для досягнення певного результату	1 день
2	Складання технічного завдання	3 дні
3	Підготовка прототипу	2 дні
4	Створення макету дизайну web-додатку	2 дні
5	Верстка	3 дні
6	Робота над модулями для web-додатку	2 дні
7	Робота з контентом	1 день
8	Розміщення контенту та каталогів з фото у web-додатку	1 день
9	Перевірка працездатності web-додатку	1 день
10	Завершення роботи	1 день
	Загальна тривалість робіт	18 днів

4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Для того, щоб web-додатком могли користуватися клієнти та відвідувачі необхідно розмістити його у мережі Інтернет, тому необхідно придбати доменне ім’я та місце на хостингу. На хостинг переноситься web-додаток і наповнення бази даних з подальшою їх доробкою. Для коректного переносу web-додатку на хостинг необхідно, щоб параметри хостинга відповідали вимогам, зазначеним у ТЗ.

Додаток Б.
Планування робіт

1. Планування змісту структури робіт.

Для планування змісту структури робіт головною є WBS-структура (Work Breakdown Structure) − розбиття всієї роботи, яку необхідно виконати для досягнення цілей проекту, на більш дрібні операції і дії , в ієрархічному порядку. Декомпозиція триває до етапу, на якому кожна робот цілком зрозуміла та може бути оцінена і спланована. Таке розбиття робіт дає можливість детально спланувати проект та розподілити персональну відповідальність виконавців. WBS діаграма зображена на рис. Б.1.

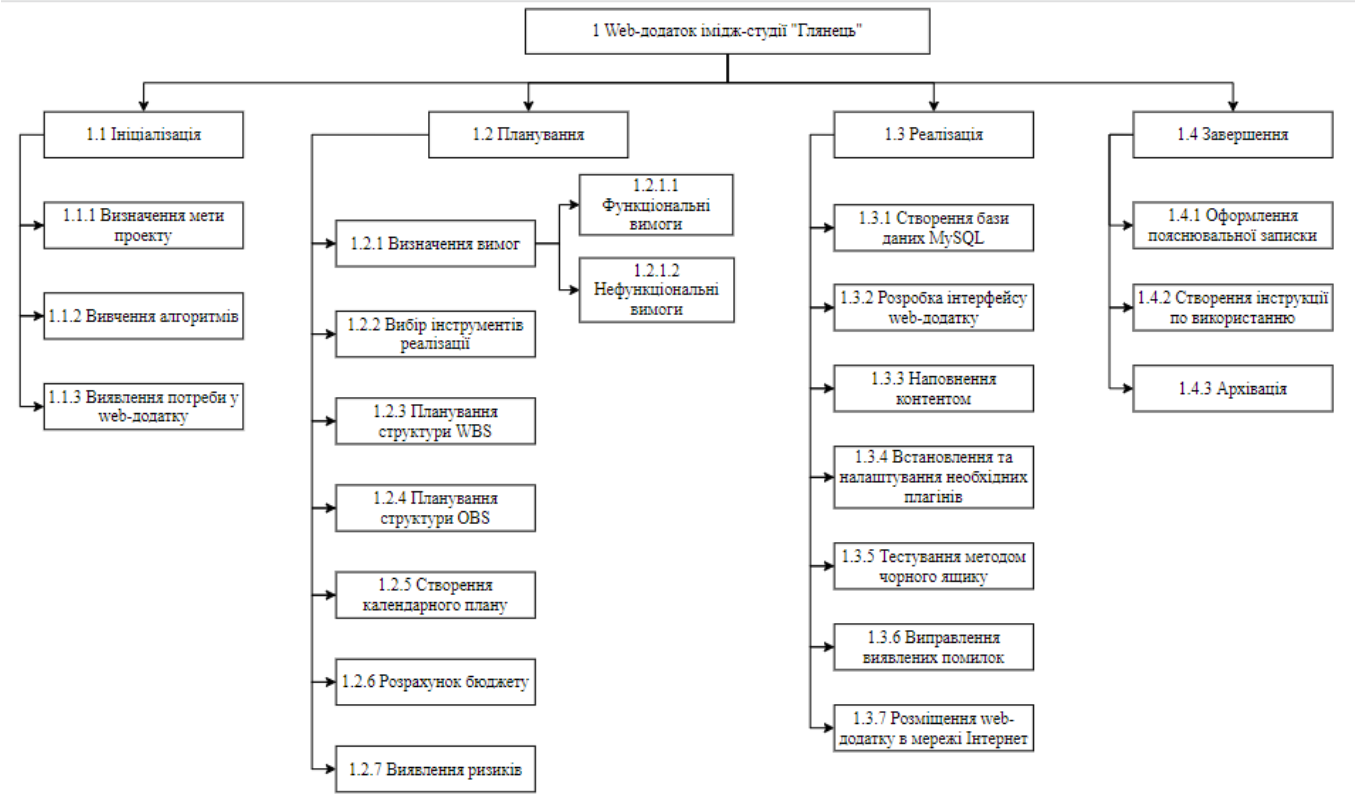

Рисунок Б.1 –WBS-структура системи

2. Планування структури виконання для впровадження готової інформаційної системи

OBS-структура (Organizational structure) – визначає розподіл повноважень і відповідальності всередині організації і не стосується відносин проектних груп чи учасників з батьківськими організаціями.

Будується OBS аналогічно робочій структурі, а саме:

- на першому рівні відображається організаційна структура як єдиний елемент;
- на другому і нижчих рівнях триває декомпозиція структури на основні організаційні елементи.

OBS діаграма зображена на рис. Б.2.

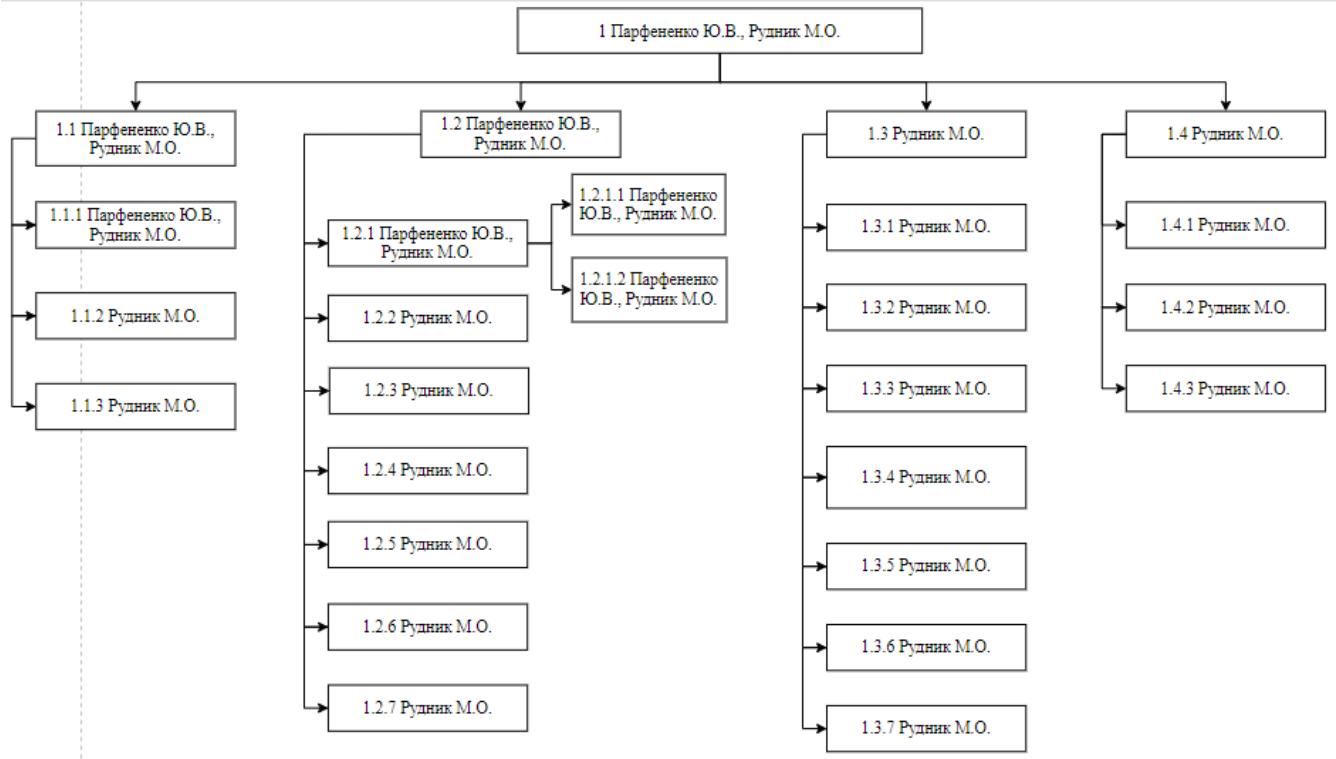

Рисунок Б.2 – OBS-структура системи

3. Побудова матриці відповідальності

Матриця відповідальності забезпечує опис структури відповідальності за виконання робіт. Це форма опису розподілу відповідальності за реалізацію робіт проекту, із зазначенням ролі кожного з підрозділів в їх виконанні.

Матриця відповідальності побудована на основі WBS та OBS структур та зображена у таблиці Б.1.

Таблиця Б.1 – Матриця відповідальності

WBS\OBS	Рудник М.О.	Парфененко Ю.В.
1.1.1 Визначення мети проекту		
1.1.2 Вивчення аналогів		
1.1.3 Виявлення потреби у web-додатку		
1.2.1 Функціональні вимоги		
1.2.2 Нефункціональні вимоги		
1.2.3 Планування структури WBS		
1.2.4 Планування структури OBS		
1.2.5 Створення календарного плану		
1.2.6 Розрахунок бюджету		
1.2.7 Виявлення ризиків		
1.3.1 Створення бази даних MySQL		
1.3.2 Розробка інтерфейсу web-додатку		
1.3.3 Наповнення унікальним контентом		
1.3.4 Встановлення та налаштування необхідних плагінів		
1.3.5 Тестування методом чорного ящика		
1.3.6 Виправлення виявлених помилок		
1.3.7 Розміщення web-додатку в мережі Інтернет		
1.4.1 Оформлення пояснювальної записки		
1.4.2 Створення інструкції з використання		
1.4.3 Архівація		

4. Розробка PDM мережі

PDM мережа спрямована на розрахунок часових ресурсів необхідних для виконання кожної окремої задачі та проекту в цілому. Також PDM мережа дає змогу визначити критичний та мінімальний час для завершення проекту.

Дана мережа була побудована за допомогою програми MS Project 2016. Мережу зображено на рис. Б.3

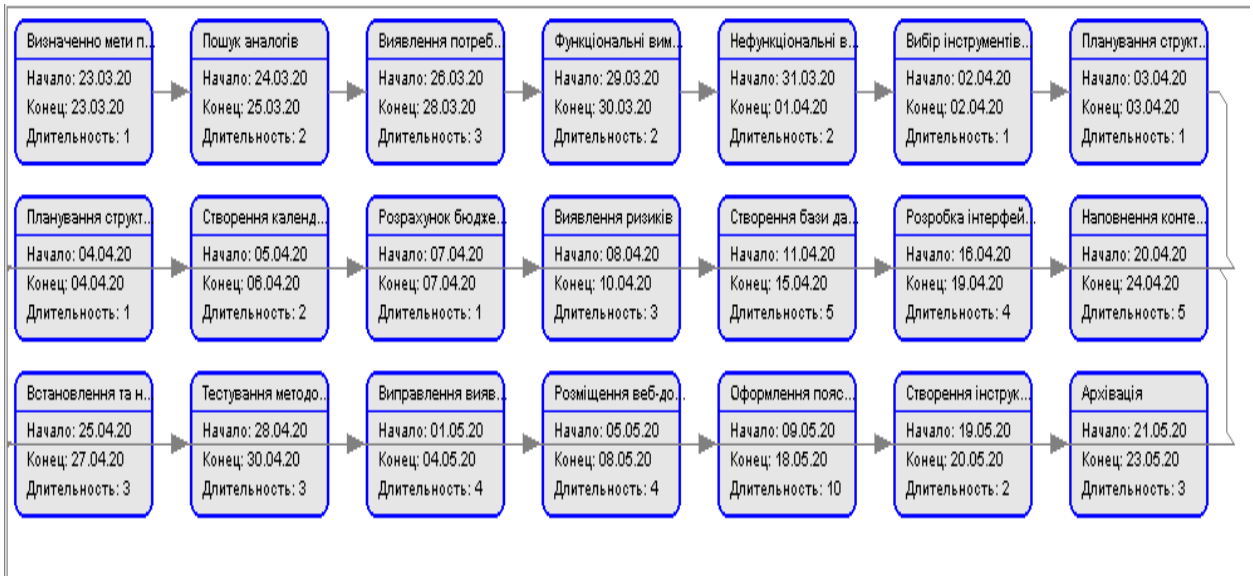

Рисунок Б.3 – PDM-мережа

5. Побудова календарного графіку виконання інформаційної системи

Календарний графік зображено у вигляді діаграми Ганта за допомогою програми MS Project 2016.

Діаграма Ганта – найпоширеніший з засобів планування та управління проектом, являє собою відрізки, розміщені на горизонтальній шкалі часу. Кожен такий відрізок відповідає окремій підзадачі. Дана діаграма допомагає контролювати дати початку і закінчення проекту, а також показує, яка робота виконується в будь-який момент часу.

Одиницею вимірювання часу в даному випадку є доба, проте реальна тривалість виконання робіт складає 2-3 години на добу. Також при створенні графіка було враховано вихідні та святкові дні.

Діаграма Ганта зображена на рис. Б.4.

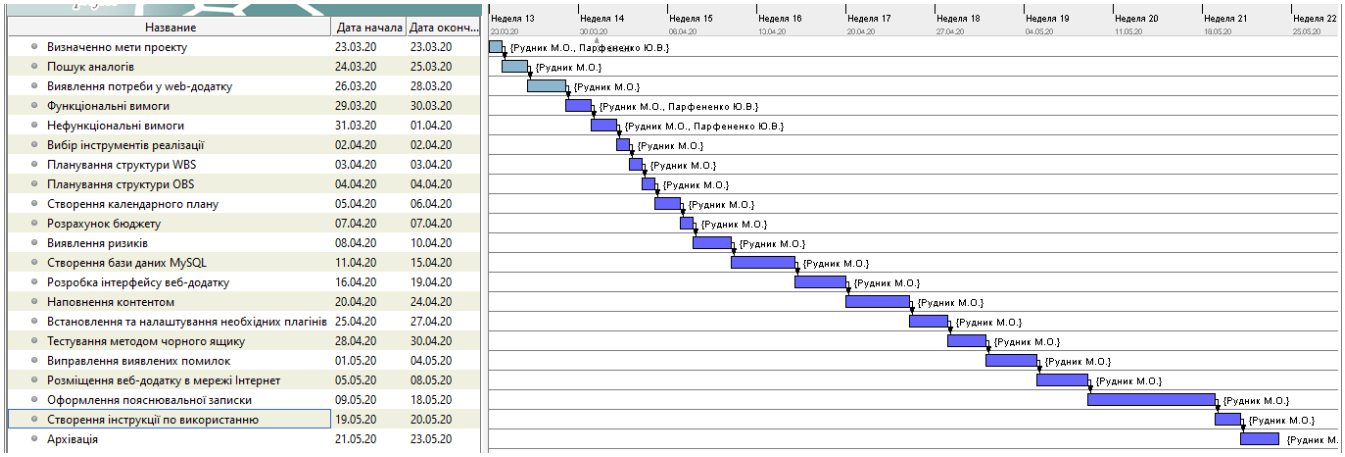

Рисунок Б.4 – Діаграма Ганта

6. Управління ризиками інформаційної системи

Для успішної реалізації проекту необхідно передбачити всі ризики, виконати їх оцінку, та передбачати варіанти усунення в разі їх виникнення.

Були виділені такі ризики як:

- R1 – зміна ТЗ на етапі розробки;
- R2 – недотримання календарного плану;
- R3 – відсутність в мережі необхідних плагінів;
- R4 – хвороба розробника;
- R5 – некоректне тестування;
- R6 – проблеми з розміщенням додатку в мережі Інтернет.

Після цього була визначена ймовірність появи і значимість впливу можливих ризиків, а також оцінена критичність кожного ризику за п’ятибальною шкалою.

Результати занесено до таблиць Б.2 та Б.3.

Таблиця Б.2 – Ймовірність виникнення ризиків

Ймовірність виникнення	R1	R2	R3	R4	R5	R6	Оцінка
Слабо ймовірний							1
малоймовірний							2
ймовірний							3
дуже ймовірний							4
майже можливий							5

Таблиця Б.3 – Втрати при виникненні ризиків

Значимість впливу	R1	R2	R3	R4	R5	R6	Оцінка
мінімальна							1
низька							2
середня							3
висока							4
максимальна							5

На підставі отриманих оцінок відбувається класифікація ризиків на: прийнятні, виправдані та недопустимі.

Результат зображено у вигляді таблиці Б.4.

Таблиця Б.4 − Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 6$	R1, R4, R6
2	Виправдані	$7 \leq R \leq 13$	R5
3	Недопустимі	$14 \leq R \leq 25$	R2, R3

На основі створених таблиць, була побудована матриця впливу (ймовірностей і наслідків) ризиків, де білим кольором позначені прийнятні ризики, світло-сірим позначені виправдані, а найтемніший колір відповідає за недопустимі.

Матриця ймовірностей і наслідків ризиків зображена у вигляді таблиці Б.5.

Таблиця Б.5 – Матриця впливу

Ймовірність			
			R3
	R6	R5	R2
	R4	R1	
Вплив			

На даній таблиці чітко видно два критичних ризики:

- R2 недотримання календарного плану;
- R3 відсутність в мережі необхідних плагінів;

Щодо першого ризику, він може виникнути в разі проявів інших ризиків, таких, як зміна ТЗ, хвороба тощо. Вирішити дану проблему можна буде за рахунок скорочення термінів виконання окремих робіт. Другий ризик, спираючись на ТЗ, досить ймовірний, якщо він виникне, то необхідно буде писати необхідні плагіни самостійно.

Додаток В.
Основні файли коду реалізації

Файл Assistant.php

```
class SLN_Shortcode_SalonAssistant
{
    const NAME = 'salon_booking_assistant';

    private $plugin;
    private $attrs;

    function __construct(SLN_Plugin $plugin, $attrs)
    {
        $this->plugin = $plugin;
        $this->attrs = $attrs;
    }

    public static function init(SLN_Plugin $plugin)
    {
        add_shortcode(self::NAME, array(__CLASS__, 'create'));
    }

    public static function create($attrs)
    {
        SLN_TimeFunc::startRealTimezone();

        $obj = new self(SLN_Plugin::getInstance(), $attrs);

        $ret = $obj->execute();
        SLN_TimeFunc::endRealTimezone();
        return $ret;
    }

    public function execute()
    {
```

```
            $attendants = false;
            $display = false;
            if(!empty($this->attrs['attendants'])){
                $attendants = explode(',',$this->attrs['attendants']);
            }
            if(!empty($this->attrs['display'])){
                $display = explode(',',$this->attrs['display']);
            }
            $repo = $this->plugin->getRepository(SLN_Plugin::POST_TYPE_ATTENDANT);

            $criteria = $attendants ? array(
                '@wp_query' => array('post__in' => $attendants)
            ) : array();

            $criteria = apply_filters('sln_attendants_shortcode_get_attendants_criteria', $criteria, $this->attrs);

            $attendants = $repo->sortByPos($repo->get($criteria));
            $data = array('attendants' => $attendants);
            $data['styled'] = !empty($this->attrs['styled']) && $this->attrs['styled']=== 'true';
            if(!empty($this->attrs['columns']) && intval($this->attrs['columns'])) $data['columns'] =  intval($this->attrs['columns']);
            $data['display'] = $display;
            $data['booking_url'] = get_the_permalink($this->plugin->getSettings()->getPayPageId());
            return $this->render($data);
        }

        protected function render($data = array())
        {
            return $this->plugin->loadView('shortcode/salon_assistant', compact('data'));
        }
}
```

Файл attendants.php

```
$ah = $plugin->getAvailabilityHelper();
$ah->setDate($plugin->getBookingBuilder()->getDateTime());
$bookingServices = SLN_Wrapper_Booking_Services::build($bb->getAttendantsIds(), $bb->getDateTime());
$hasAttendants = false;
$style = $step->getShortcode()->getStyleShortcode();
$size = SLN_Enum_ShortcodeStyle::getSize($style);


$services = $bb->getServices();
foreach ($services as $k => $service) {
   if (!$service->isAttendantsEnabled()) {
      unset($services[$k]);
   }
}

$isChooseAttendantForMeDisabled = $plugin->getSettings()->isChooseAttendantForMeDisabled();

$tmp = '';
$i = 0;
foreach ($attendants as $attendant) {
   if ($attendant->hasServices($services)) {
      $errors = SLN_Shortcode_Salon_AttendantHelper::validateItem($bookingServices->getItems(), $ah, $attendant);

         if (!$i && $isChooseAttendantForMeDisabled) {
            $tmp .= SLN_Shortcode_Salon_AttendantHelper::renderItem($size, $errors, $attendant, null, true);
         } else {
            $tmp .= SLN_Shortcode_Salon_AttendantHelper::renderItem($size, $errors, $attendant);
         }
```

```
            $hasAttendants = true;
                $i++;
        }
    }
    if ($tmp && !$isChooseAttendantForMeDisabled) {
        $tmp = SLN_Shortcode_Salon_AttendantHelper::renderItem($size).$tmp;
    }

    ?>
    <div class="sln-attendant-list">
        <?php if ($tmp) : ?>
            <div class="row"><?php echo $tmp ?></div>
        <?php else: ?>
            <div class="alert alert-warning">
                <p><?php echo __(
                        'Нет мастеров для выбранного времени - выберите другое',
                        'salon-booking-system'
                    ) ?></p>
            </div>
        <?php endif ?>
</div>
```

Файл my_account_details_table.php

```
    public static $noTimeStatuses = array(
        self::ERROR,
        self::CANCELED,
    );

    public static function toArray()
    {
        return self::getLabels();
    }

    public static function getLabel($key)
    {
```

```
        $labels = self::getLabels();
        return isset($labels[$key]) ? $labels[$key] : $labels[self::ERROR];
    }

    public static function getColor($key)
    {
        return isset(self::$colors[$key]) ? self::$colors[$key] : self::$colors[self::ERROR];
    }

    public static function getIcon($key)
    {
        return isset(self::$icons[$key]) ? self::$icons[$key] : self::$icons[self::ERROR];
    }

    public static function init()
    {
        self::$labels = array(
            self::PENDING   => __('В ожидании', 'salon-booking-system'),
            self::CANCELED  => __('Отменён', 'salon-booking-system'),
            self::CONFIRMED => __('Подтверждён', 'salon-booking-system'),
            self::ERROR     => __('Ошибка', 'salon-booking-system'),
        );
    }
}
```

Файл my_account_details.php

```
<div role="tabpanel" class="tab-pane sln-salon-my-account-tab-pane" id="profile">
    <?php include '_salon_my_account_profile.php'; ?>
</div>

<div id="ratingModal" class="modal fade" role="dialog" tabindex="-1">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
```

```
                                        <button type="button" class="close" data-dismiss="modal">&times;</button>
                                        <h4 class="modal-title"></h4>
                                    </div>
                                    <div class="modal-body">
                                        <div id="step1">
                                            <p><?php _e('Добрый день','salon-booking-system');?> <?php echo $data['user_name'] ?>!</p>
                                            <p><?php _e('Как вы опишете обслуживание нашей командой в этот раз?','salon-booking-system');?></p>
                                            <p><textarea id="" placeholder="<?php _e('Пожалуйста, напишите нам несколько строк, чтобы понять, соответствует ли ваш опыт вашим ожиданиям','salon-booking-system');?>"></textarea></p>
                                            <p>
                                            <div class="rating" id="<?php echo $item['id']; ?>"></div>
                                            <span><?php _e('Оцените нашу команду!','salon-booking-system');?></span>
                                            </p>
                                            <p>
                                                <button type="button" class="btn btn-primary" onclick="slnMyAccount.sendRate();"><?php _e('Отправить отзыв','salon-booking-system');?></button>
                                                <button type="button" class="btn btn-default" data-dismiss="modal"><?php _e('Отменить','salon-booking-system');?></button>
                                            </p>
                                        </div>
                                        <div id="step2">
                                            <p><?php _e('Спасибо за ваш отзыв. Это поможет нам улучшить наши услуги.','salon-booking-system');?></p>
                                            <p><?php _e('Надеемся увидеть Вас ещё в имидж-студии ','salon-booking-system');?> <?php echo $data['gen_name']; ?></p>
                                        </div>
                                    </div>
                                    <div class="modal-footer"></div>
```

```
                    </div>
                </div>
        </div>
```

Файл functions.php

```
        add_shortcode( 'art_feedback', 'art_feedback' );
        function art_feedback() {

            ob_start();
            ?>
            <form id="add_feedback">
                <input type="text" name="art_name" id="art_name" class="required art_name" placeholder="Имя" value=""/>

                <input type="email" name="art_email" id="art_email" class="required art_email" placeholder="Ваш E-Mail" value=""/>

                <input type="text" name="art_subject" id="art_subject" class="art_subject" placeholder="Тема" value=""/>

                <textarea name="art_comments" id="art_comments" placeholder="Введите ваше обращение" rows="10" cols="30" class="required art_comments"></textarea>

                <input type="checkbox" name="art_anticheck" id="art_anticheck" class="art_anticheck" style="display: none !important;" value="true" checked="checked"/>

                <input type="text" name="art_submitted" id="art_submitted" value="" style="display: none !important;"/>

                <input type="submit" id="submit-feedback" class="button" value="Отправить"/>
            </form>
            <?php
```

```
	return ob_get_clean();
}

add_action( 'wp_enqueue_scripts', 'art_feedback_scripts' );
function art_feedback_scripts() {

	wp_enqueue_script( 'jquery-form' );

	wp_enqueue_script(
		'feedback',
		get_stylesheet_directory_uri() . '/js/feedback.js',
		array( 'jquery' ),
		1.0,
		true
	);

	wp_localize_script(
		'feedback',
		'feedback_object',
		array(
			'url'   => admin_url( 'admin-ajax.php' ),
			'nonce' => wp_create_nonce( 'feedback-nonce' ),
		)
	);

}

add_action( 'wp_ajax_feedback_action', 'ajax_action_callback' );
add_action( 'wp_ajax_nopriv_feedback_action', 'ajax_action_callback' );

function ajax_action_callback() {

	// Массив ошибок
	$err_message = array();
```

```
		if ( ! wp_verify_nonce( $_POST['nonce'], 'feedback-nonce' ) ) {
			wp_die( 'Данные отправлены с неправильного адреса' );
		}

		if ( false === $_POST['art_anticheck'] || ! empty( $_POST['art_submitted'] ) ) {
			wp_die( 'Это спам' );
		}

		if ( empty( $_POST['art_name'] ) || ! isset( $_POST['art_name'] ) ) {
			$err_message['name'] = 'Пожалуйста, введите ваше имя.';
		} else {
			$art_name = sanitize_text_field( $_POST['art_name'] );
		}

		if ( empty( $_POST['art_email'] ) || ! isset( $_POST['art_email'] ) ) {
			$err_message['email'] = 'Пожалуйста, введите адрес вашей электронной
почты.';
		} elseif ( ! preg_match( '/^[[:alnum:]][a-z0-9_.-]*@[a-z0-9.-]+\.[a-z]{2,4}$/i',
$_POST['art_email'] ) ) {
			$err_message['email'] = 'Адрес электронной почты некорректный.';
		} else {
			$art_email = sanitize_email( $_POST['art_email'] );

		}
		if ( empty( $_POST['art_subject'] ) || ! isset( $_POST['art_subject'] ) ) {
			$art_subject = 'Сообщение с сайта';
		} else {
			$art_subject = sanitize_text_field( $_POST['art_subject'] );
		}

		if ( empty( $_POST['art_comments'] ) || ! isset( $_POST['art_comments'] ) ) {
			$err_message['comments'] = 'Пожалуйста, введите ваше сообщение.';
		} else {
			$art_comments = sanitize_textarea_field( $_POST['art_comments'] );
		}
```

```
		if ( $err_message ) {

			wp_send_json_error( $err_message );

		} else {

			$email_to = 'mariha.dubovik@gmail.com';

			if ( ! $email_to ) {
				$email_to = get_option( 'admin_email' );
			}

			$body = "Имя: $art_name \nEmail: $art_email \n\nСообщение: $art_comments";
			$headers = 'From: ' . $art_name . ' <' . $email_to . '>' . "\r\n" . 'Reply-To: ' . $email_to;

			wp_mail( $email_to, $art_subject, $body, $headers );
			$message_success = 'Сообщение отправлено. В ближайшее время наша команда свяжеся с Вами.';
			wp_send_json_success( $message_success );
		}
		wp_die();
}
```

Файл style.css

```
	#add_feedback {
	   margin: 20px 0 0;
	   position: relative;
	}

	#art_name, #art_email, #art_subject, #art_comments {
	   padding: 10px 5px;
	   display: inline-block;
```

```
    width: 49.548%;
    border-radius: 3px;
    border: 1px solid #ddd;
    font-size: 0.9em;
}

#art_subject {
    width: 100%;
    margin: 5px 0;
}

#art_comments {
    width: 100%;
}

#add_feedback .button {
    border: none;
    padding: 10px 20px;
    color: #fff;
    font-size: 1em;
    display: inherit;
    margin: 10px 0 0 0;
    border-radius: 3px;
    background-color: #8B0000;
}

#add_feedback input[type="text"]:focus, #add_feedback input[type="password"]:focus,
#add_feedback input[type="email"]:focus, #add_feedback textarea:focus {
    color: #444;
    box-shadow: 0 0 3px rgba(68, 68, 68, 0.2);
}

.error-text {
    background: #F59E9E;
    padding: 15px 0px;
```

```
    text-align: center;
    color: #fff;
}

.error-name, .error-email, .error-comments {
    display: inline-block;
    font-size: 11px;
    position: absolute;
    top: -30px;
    color: white;
    border: 1px solid red;
    padding: 5px 10px;
    line-height: 1.1;
    background: red;
    box-shadow: 0 0 3px 0px rgba(0, 0, 0, 0.3);
}

.error-name:after, .error-email:after, .error-comments:after {
    content: '';
    position: absolute;
    left: 20px;
    bottom: -10px;
    border: 5px solid transparent;
    border-top: 5px solid red;
}

.error-comments {
    top: 16%;
    left: 0;
}

.error-name {

    left: 0;
}
```

```
.error-email {
    right: 4%;
}

.message-success {
    background: rgba(0, 128, 0, 0.5);
    color: #8B0000;
    padding: 20px;
    text-align: center;
    margin: 20px auto;
    border-radius: 3px;
}

#add_feedback .error {
    border: 1px solid red;

}

#add_feedback .required:after {
    content: '*';
    position: absolute;
    left: 20px;
    bottom: -10px;
}

#art_name:focus::-moz-placeholder,
#art_name:focus:-moz-placeholder,
#art_name:focus:-ms-input-placeholder,
#art_email:focus::-webkit-input-placeholder,
#art_email:focus::-moz-placeholder,
#art_email:focus:-moz-placeholder,
#art_email:focus:-ms-input-placeholder,
#art_comments:focus::-webkit-input-placeholder,
#art_comments:focus::-moz-placeholder,
```

```
        #art_comments:focus:-moz-placeholder,
        #art_comments:focus:-ms-input-placeholder,
        #art_subject:focus::-webkit-input-placeholder,
        #art_subject:focus::-moz-placeholder,
        #art_subject:focus:-moz-placeholder,
        #art_subject:focus:-ms-input-placeholder {
            color: transparent
}
```

Файл feedback.js

```
        jQuery(document).ready(function ($) {
            var add_form = $('#add_feedback');

            // Сброс значений полей
            $('#add_feedback input, #add_feedback textarea').on('blur', function () {
                $('#add_feedback input, #add_feedback textarea').removeClass('error');
                $('.error-name,.error-email,.error-comments,.message-success').remove();
                $('#submit-feedback').val('Отправить сообщение');
            });

            // Отправка значений полей
            var options = {
                url: feedback_object.url,
                data: {
                    action: 'feedback_action',
                    nonce: feedback_object.nonce
                },
                type: 'POST',
                dataType: 'json',
                beforeSubmit: function (xhr) {
                    // При отправке меняем надпись на кнопке
                    $('#submit-feedback').val('Отправляем...');
                },
                success: function (request, xhr, status, error) {
```

```
                if (request.success === true) {
                    // Если все поля заполнены, отправляем данные и меняем надпись на кнопке
                    add_form.after('<div class="message-success">' + request.data + '</div>').slideDown();
                    $('#submit-feedback').val('Отправить сообщение');
                } else {
                    // Если поля не заполнены, выводим сообщения и меняем надпись на кнопке
                    $.each(request.data, function (key, val) {
                        $('.art_' + key).addClass('error');
                        $('.art_' + key).before('<span class="error-' + key + '">' + val + '</span>');
                    });
                    $('#submit-feedback').val('Что-то пошло не так...');

                }
                // При успешной отправке сбрасываем значения полей
                $('#add_feedback')[0].reset();
            },
            error: function (request, status, error) {
                $('#submit-feedback').val('Что-то пошло не так...');
            }
        };
        // Отправка
        add_form.ajaxForm(options);
});
```