

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 004.9:687.5

«ПОГОДЖЕНО»

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Декан факультету
інформаційних технологій

Завідувач кафедри комп'ютерних наук

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

_____ 202_ р.

_____ 202_ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему «ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АНАЛІЗУ ОБРОБКИ
ЗАМОВЛЕНЬ В СФЕРІ КРАСИ ТА ЗДОРОВ'Я»**

Спеціальність 121 «Інженерія програмного забезпечення

Освітня програма «Програмне забезпечення інформаційних систем»

Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми

Доцент кафедри комп'ютерних наук _____

Голуб Б.Л.

Керівник магістерської кваліфікаційної роботи

Старший викладач _____

Ящук Д.Ю.

Виконав

_____ Москальчук С.Ю.

ЗМІСТ

ВСТУП	6
<hr/>	
1. СУЧАСНИЙ СТАН ТА АКТУАЛЬНІСТЬ ДИПЛОМНОЇ РОБОТИ	8
<hr/>	
1.1. АКТУАЛЬНІСТЬ ТЕМИ ДИПЛОМНОЇ РОБОТИ	8
1.2. АНАЛІЗ ІСНУЮЧИХ РОЗРОБОК	9
1.3. ЦІЛІ ТА ЗАВДАННЯ ДИПЛОМНОЇ РОБОТИ	10
Висновок до розділу 1	11
2. ПРОЕКТУВАННЯ ТА МОДЕЛЮВАННЯ ВЕБСАЙТУ	12
<hr/>	
2.1 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ РОБОТИ САЙТУ	12
2.2 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ САЙТУ	13
2.3 РОЗРОБКА АЛГОРИТМУ РОБОТИ САЙТУ	20
Висновок до розділу 2	29
3 РОЗРОБКА БАЗИ ДАНИХ І ВЕБСАЙТУ	30
<hr/>	
3.1 АНАЛІЗ ОБРАНОГО СЕРЕДОВИЩА ПРОГРАМУВАННЯ	30
3.2 АНАЛІЗ ОБРАНИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ВЕБСАЙТУ	31
3.3 РОЗРОБКА БАЗИ ДАНИХ	32
3.3.1 КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ БАЗИ ДАНИХ	32
3.3.2 ЛОГІЧНЕ ПРОЕКТУВАННЯ БАЗИ ДАНИХ	36
3.3.3 СТВОРЕННЯ ТАБЛИЦЬ ТА ЗВ'ЯЗКІВ МІЖ НИМИ	38
3.4 ПРОГРАМНА РЕАЛІЗАЦІЯ САЙТУ	40
3.4.1 ВИБІР ТЕХНОЛОГІЧНОГО СТЕКУ	40
3.4.2 АРХІТЕКТУРА ВЕБСАЙТУ	44
3.4.3 КЛЮЧОВІ ФУНКЦІОНАЛЬНОСТІ ТА МОЖЛИВОСТІ	45
3.4.4 РОЗРОБКА ІНТЕРФЕЙСУ САЙТУ	46
3.4.5 БЕЗПЕКА І ЗАХИСТ ДАНИХ	47
3.4.6 ТЕСТУВАННЯ І ВІДЛАДКА	47
3.4.7 ДЕПЛОЙМЕНТ І ПУБЛІКАЦІЯ	48
Висновок до розділу 3	48
4 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ АПРОБАЦІЇ РОЗРОБЛЕНОГО ВЕБСАЙТУ	49
<hr/>	
4.1 ПРОЦЕС ТЕСТУВАННЯ ВЕБСАЙТУ	49
4.2 РОБОТА З ОСНОВНОЮ ІНФОРМАЦІЄЮ САЙТУ	50

	4
4.2.1 ВХІД НА САЙТ ЯК КЛІЄНТ	52
4.2.2 ВХІД НА САЙТ ЯК ЗАКЛАД	55
4.2.3 ВХІД НА САЙТ ЯК СПЕЦІАЛІСТ	59
4.3 АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОЗРОБЛЕНОГО ВЕБСАЙТУ	64
Висновок до розділу 4	65
<u>ВИСНОВКИ</u>	67
<u>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</u>	68
<u>ДОДАТОК А</u>	69
<u>ДОДАТОК Б</u>	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

РЕФЕРАТ

Дипломна робота на тему "Програмне забезпечення системи аналізу обробки замовлень в сфері краси та здоров'я" складається з 75 сторінок, 47 ілюстрацій, 1 таблиці та 2 додатків. В роботі було використано __ літературних джерел.

Об'єктом дослідження є створення програмного забезпечення для системи аналізу обробки замовлень в галузі краси та здоров'я. Дослідження включає в себе аналіз потреб цільової аудиторії та розробку вебсайту для зручного управління замовленнями.

Методи дослідження: В роботі використано методи аналізу вимог, проектування програмного забезпечення, розробки вебсайту та тестування.

Метою дипломної роботи є створення веб-застосунку, який дозволить спростити та автоматизувати процес обробки замовлень для фахівців у сфері краси та здоров'я, а також полегшити процес замовлення послуг для клієнтів.

Отримані результати: В результаті дослідження було розроблено програмне забезпечення, яке дозволяє фахівцям створювати профілі, приймати та відхиляти замовлення, вести облік фінансів та взаємодіяти з клієнтами. Система також забезпечує зручний інтерфейс для клієнтів для замовлення послуг та спілкування з фахівцями.

Наукова новизна: Розроблене програмне забезпечення відзначається науковою новизною, оскільки воно пропонує новий підхід до управління замовленнями в сфері краси та здоров'я, спрощуючи процес як для фахівців, так і для клієнтів. Робота має прикладну значимість, оскільки розроблене програмне забезпечення може бути використане в реальних умовах для поліпшення обробки замовлень та зручності користувачів.

Отже, дипломна робота висвітлює процес розробки програмного забезпечення для системи аналізу обробки замовлень в сфері краси та здоров'я, що має наукову новизну та великий потенціал в покращенні галузі.

ABSTRACT

The diploma thesis on the topic "Software for Order Processing Analysis System in the Beauty and Health Industry" consists of 75 pages, 47 illustrations, 1 table, and 2 appendices. The work utilized __ literary sources.

The object of the research is the development of software for an order processing analysis system in the beauty and health industry. The research includes an analysis of the needs of the target audience and the development of a website for efficient order management.

Research methods: The work employs methods of requirement analysis, software design, website development, and testing.

The purpose of the diploma thesis is to create a web application that simplifies and automates the order processing for professionals in the beauty and health sector, as well as streamlining the process of ordering services for clients.

Results obtained: As a result of the research, software has been developed that allows professionals to create profiles, accept or reject orders, manage finances, and interact with clients. The system also provides a user-friendly interface for clients to order services and communicate with professionals.

Scientific novelty: The developed software offers scientific novelty as it introduces a new approach to order management in the beauty and health industry, simplifying the process for both professionals and clients. The work has practical significance since the software can be utilized in real-life scenarios to improve order processing and user convenience.

In conclusion, the diploma thesis covers the development of software for an order processing analysis system in the beauty and health industry, which brings scientific novelty and holds great potential for industry improvement.

ВСТУП

Сучасний ринок краси та здоров'я відзначається стрімким розвитком та постійним зростанням попиту на продукти та послуги в цій галузі. Потреби споживачів стають все більш різноманітними, а конкуренція серед підприємств, які надають послуги та продукти в цій сфері, стає дедалі жорсткішою. Одним із способів підтримки успішності на цьому ринку є використання сучасних інформаційних технологій та програмного забезпечення для ефективного управління бізнес-процесами.

Предметом дослідження є програмне забезпечення системи пошуку спеціалістів за даними критеріями та аналізу обробки замовлень в сфері краси та здоров'я, яке дозволить оптимізувати роботу підприємств та підвищити їхню конкурентоспроможність. Об'єктом для дослідження та створення веб-сайту було обрано сервіс онлайн-пошуку спеціалістів у сфері Н&В за власними критеріями «Clinics Up».

Метою дослідження є розробка та апробація програмного забезпечення, яке спростить та покращить процес пошуку і обробки замовлень на підприємствах краси та здоров'я. Крім того, визначені наступні цілі дослідження:

- Аналіз сучасних тенденцій у сфері краси та здоров'я та існуючих систем обробки замовлень.
- Розробка програмного забезпечення для системи аналізу обробки замовлень, яке відповідає потребам підприємства цієї галузі.
- Проведення апробації розробленого програмного забезпечення на реальних підприємствах краси та здоров'я для оцінки його ефективності.

Методи дослідження включають аналіз літератури, опитування фахівців та користувачів, розробку програмного забезпечення, тестування та статистичний аналіз результатів апробації.

Науковою новизною дослідження є розробка програмного забезпечення, спеціалізованого для пошуку послуг та фахівців за критеріями. Також створення для спеціалістів сайту-портфоліо для залучення нових клієнтів. Ресурс для поширення інформації про свій заклад та спеціалістів, щоб знайти нових клієнтів та співробітників.

Результати дослідження апробовані на реальному підприємстві сфери краси та здоров'я, що підтверджує їхню практичну застосовність.

Структура дипломної роботи буде наступною: у розділі 1 розглянуто сучасний стан та актуальність дослідження. Розділ 2 присвячений проектуванню та моделюванню веб-сайту. У розділі 3 розглянуто розробка бази даних і веб-сайту. Розділ 4 включає тестування та результати апробації програмного забезпечення. Висновок містить узагальнення дослідження та рекомендації щодо використання розробленого програмного забезпечення в практиці.

1. СУЧАСНИЙ СТАН ТА АКТУАЛЬНІСТЬ ДИПЛОМНОЇ РОБОТИ

1.1. Актуальність теми дипломної роботи

Сфера краси та здоров'я вже давно є важливою складовою нашого повсякденного життя. Споживачі активно використовують послуги салонів краси, студій масажу, фітнес-центрів, а також купують косметику та товари для догляду за собою. З цим зростанням популярності галузі виникає попит на нові та ефективні інструменти для пошуку та вибору послуг та товарів. Сучасний ринок краси та здоров'я став динамічним та дуже конкурентним середовищем, де підприємства постійно шукають способи покращити свою конкурентоспроможність та задовольнити зростаючий попит споживачів. Пошук та обробка інформації про спеціалістів у цій галузі, а також ефективне ведення обліку та аналізу замовлень є критично важливими процесами для оптимізації діяльності підприємств сфери краси та здоров'я.

Серед головних факторів, які підкреслюють актуальність даного дослідження, можна виділити наступні:

- Зростання популярності сфери краси та здоров'я.

Попит на послуги в цій галузі надалі зростає, що вимагає оптимізованого та швидкого пошуку та вибору спеціалістів.

- Швидка інформатизація галузі.

Інформаційні технології стають необхідною складовою для забезпечення ефективної роботи підприємств сфери краси та здоров'я.

- Зростання конкуренції.

З кожним роком кількість підприємств у цій галузі зростає, і конкуренція стає більш жорсткою. Пошук та приваблення клієнтів стає завданням першочергового значення.

Дослідження створення програмного забезпечення для системи пошуку спеціалістів та аналізу обробки замовлень в сфері краси та здоров'я є важливим кроком у вирішенні актуальних проблем цієї галузі і відповідає сучасним потребам та вимогам ринку.

1.2. Аналіз існуючих розробок

На сьогоднішній день існує декілька інформаційних систем та платформ, які спрямовані на пошук спеціалістів та фахівців у сфері краси та здоров'я. Деякі з них спеціалізуються на пошуку конкретних послуг (наприклад, стрижки, манікюр, масаж), тоді як інші надають можливість знаходити фахівців з різноманітних областей.

Основні переваги і обмеження існуючих систем.

- **Онлайн-рейтинги та відгуки.** Багато систем надають можливість користувачам залишати відгуки та оцінки для спеціалістів, що допомагає іншим користувачам вибирати послуги на основі досвіду інших. Це сприяє більшій прозорості та довірі.
- **Пошук за геолокацією.** Багато платформ надають можливість знаходити фахівців, які розташовані в найближчому районі або за обраною відстанню. Це дозволяє споживачам швидко знайти спеціалістів у своєму регіоні.
- **Можливість онлайн-запису.** Деякі системи дозволяють користувачам не лише шукати фахівців, але й записуватися на послуги онлайн, що робить процес набагато зручнішим та швидшим.
- **Відсутність інтегрованих рішень.** Більшість існуючих систем обмежені певними функціями, і вони не надають повноцінні інтегровані рішення для всіх аспектів пошуку фахівців та обробки замовлень.
- **Недостатні аналітичні можливості.** Багато платформ не надають інструментів для аналізу даних та вибудовування стратегічних рішень на основі цих даних.

Аналіз існуючих розробок показує, що на ринку існують різноманітні системи та платформи для пошуку спеціалістів у сфері краси та здоров'я. Кожна з них має свої переваги та обмеження. Деякі системи спрямовані на зручний пошук та відгуки користувачів, інші надають можливість онлайн-запису, а треті фокусуються на обробці замовлень та аналітиці. Однак багато з них не надають повного спектру інтегрованих рішень, існують обмеження щодо функціональності.

1.3. Цілі та завдання дипломної роботи

В рамках розробки платформи для пошуку спеціалістів у сфері краси та здоров'я визначено наступні цілі:

- Розробка зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, який дозволить їм швидко та зручно знаходити необхідних спеціалістів та послуги.
- Розробка системи для ефективного пошуку та фільтрації спеціалістів за різними критеріями, такими як локація, спеціалізація, рейтинг та інші.
- Розробка системи для користувачів, яка дозволить їм залишати відгуки та оцінки про спеціалістів, що сприятиме вибору найкращих фахівців.
- Розробка системи, яка дозволить підприємствам у сфері краси та здоров'я ефективно приймати та обробляти замовлення, планувати графіки роботи спеціалістів та управляти цінами.
- Розробка інструментів для аналізу даних про замовлення, користувачів та фахівців з метою надання підприємствам можливості приймати інформовані стратегічні рішення.
- Забезпечення безпеки та захисту особистих даних користувачів та підприємств, що забезпечить довіру користувачів до платформи.

- Проведення тестування та апробації платформи на реальних користувачах та підприємствах з метою оцінки її ефективності та виправлення можливих недоліків.

- Розробка плану впровадження платформи та підтримки її подальшої роботи для забезпечення стабільності та ефективності.

Здійснення цих цілей дозволить успішно реалізувати основну мету дипломної роботи - створити ефективну та корисну платформу для пошуку спеціалістів у сфері краси та здоров'я, яка відповідає потребам користувачів та підприємств цієї галузі.

Висновок до розділу 1

У розділі 1 дипломної роботи було представлено загальний огляд теми та актуальність проблеми. Зокрема, було розглянуто розвиток інформаційних технологій, що значно вплинули на різні галузі, включаючи сферу краси та здоров'я. Спостерігається зростання популярності послуг у цій галузі, а отже, з'являється потреба в сучасних рішеннях для покращення співпраці між клієнтами та спеціалістами.

Також, було відзначено важливість створення інформаційних систем, які спрямовані на оптимізацію бізнес-процесів, полегшення обліку, підвищення якості послуг та зручності для користувачів. Це створює підґрунтя для розробки веб-сайту, який об'єднує спеціалістів та клієнтів у цій галузі.

Зважаючи на наведені факти та аналіз сучасних потреб, розробка веб-сайту для оптимізації взаємодії між спеціалістами та клієнтами в сфері краси та здоров'я є актуальною та важливою задачею. Вибір технологічного стеку та архітектурних рішень буде визначальним для успішної реалізації цього проекту.

2. ПРОЕКТУВАННЯ ТА МОДЕЛЮВАННЯ ВЕБСАЙТУ

2.1 Розробка структурної схеми роботи сайту

Розробка структурної схеми сайту є важливим етапом у створенні платформи для пошуку спеціалістів у сфері краси та здоров'я. Цей розділ описує загальну структуру та організацію роботи сайту, яка включає в себе різні функціональні блоки та їх взаємодію. Нижче подана структурна схема роботи сайту:



Рис. 1.1 Діаграма варіантів використання

Під час проектування сайту створено схему взаємодії користувачів з сайтом (Рис 1.1).

Діаграма відображає взаємовідносини можливих акторів з системою, та комунікативні зв'язки інформаційної та апаратної складової з середовищем зовнішнім.

Вказані актори, що наведені в діаграмі варіантів використання, які повинні враховуватися в процесі проектування та розробки сайту. Відносяться такі

сутності до акторів: «Системний адміністратор», «Адміністратор сайту», «Користувач», який поділяється на «Клієнт», «Спеціаліст» та «Заклад».

Прецеденти «Системного адміністратора» такі як: підтвердження реєстрації користувачів, та створення категорій послуг та їх вартості.

Актор «Адміністратор сайту» має такі використання сайту як: вхід до системи, запис клієнтів, перегляд записів та бази даних клієнтів, виписка чеків, формування прайс-листу послуг спеціалістів.

«Спеціаліст» та «Заклад» має можливість використовувати такі прецеденти: вхід до системи, запис клієнтів та перегляд записів.

До «Клієнту» відносяться такі способи використання сайту як: вхід до системи, перегляд інформації на сайті та зворотній зв'язок.

Керування сайтом користувачем включає в себе взаємодію користувача з інтерфейсом сайту для виконання різних завдань та операцій. Передбачає забезпечення легкості взаємодії із сайтом та надання можливостей для виконання різноманітних завдань. Важливо забезпечити зручність інтерфейсу та доступність всіх функцій для користувачів, щоб забезпечити їх задоволеність та позитивний досвід використання сайту.

2.2 Розробка функціональної схеми сайту

У цьому розділі буде представлена функціональна схема роботи сайту для платформи пошуку спеціалістів у сфері краси та здоров'я. Розробка функціональної схеми роботи сайту полягає в створенні детального плану, який описує, як сайт функціонує та як користувачі взаємодіють з різними частинами сайту.

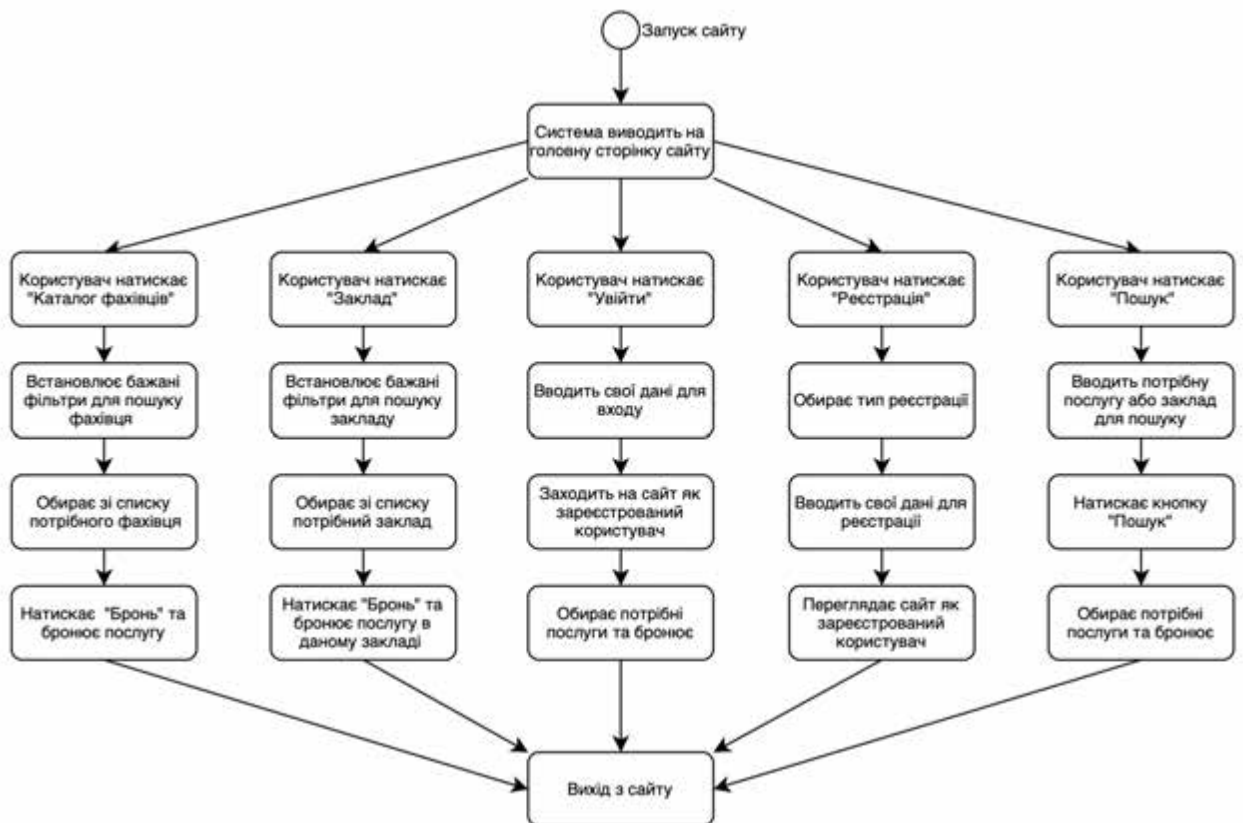


Рис. 1.2 – Функціональна схема для користування сайту користувачем

Зі схеми (Рис 1.2) можна побачити яким чином клієнт може використовувати сайт. Спочатку користувач повинен перейти за посиланням, щоб ознайомитися з усією інформацією даного сайту. При потребі клієнт може забронювати послугу. Для цього існує п'ять варіантів. Щоб скористатися одним із них, треба натискати на відповідні кнопки. Перший варіант – це користувач натискає «Каталог фахівців». Після чого встановлює бажані фільтри для пошуку потрібного фахівця та обирає його зі списку запропонованих. На профілі фахівця переглядає відгуки та всю інформацію про нього. І може забронювати послугу. Другий варіант схожий на перший, але спочатку користувач обирає «Заклад» і повторює попередній алгоритм. Щоб забронювати послугу користувачу необхідно спочатку обрати «Увійти» чи «Реєстрація» для того, щоб ввести свої дані та перейти на сторінку бронювання. Наступний варіант – це «Пошук», цим варіантом може скористатися як зареєстрований користувач, так і незареєстрований. Після чого він вводить потрібну послугу або заклад для пошуку. Обирає та бронює послугу.

Даний сайт призначений не тільки для клієнтів, а й для спеціалістів, то на рисунку 1.3 наведено схему для користування спеціалістом.

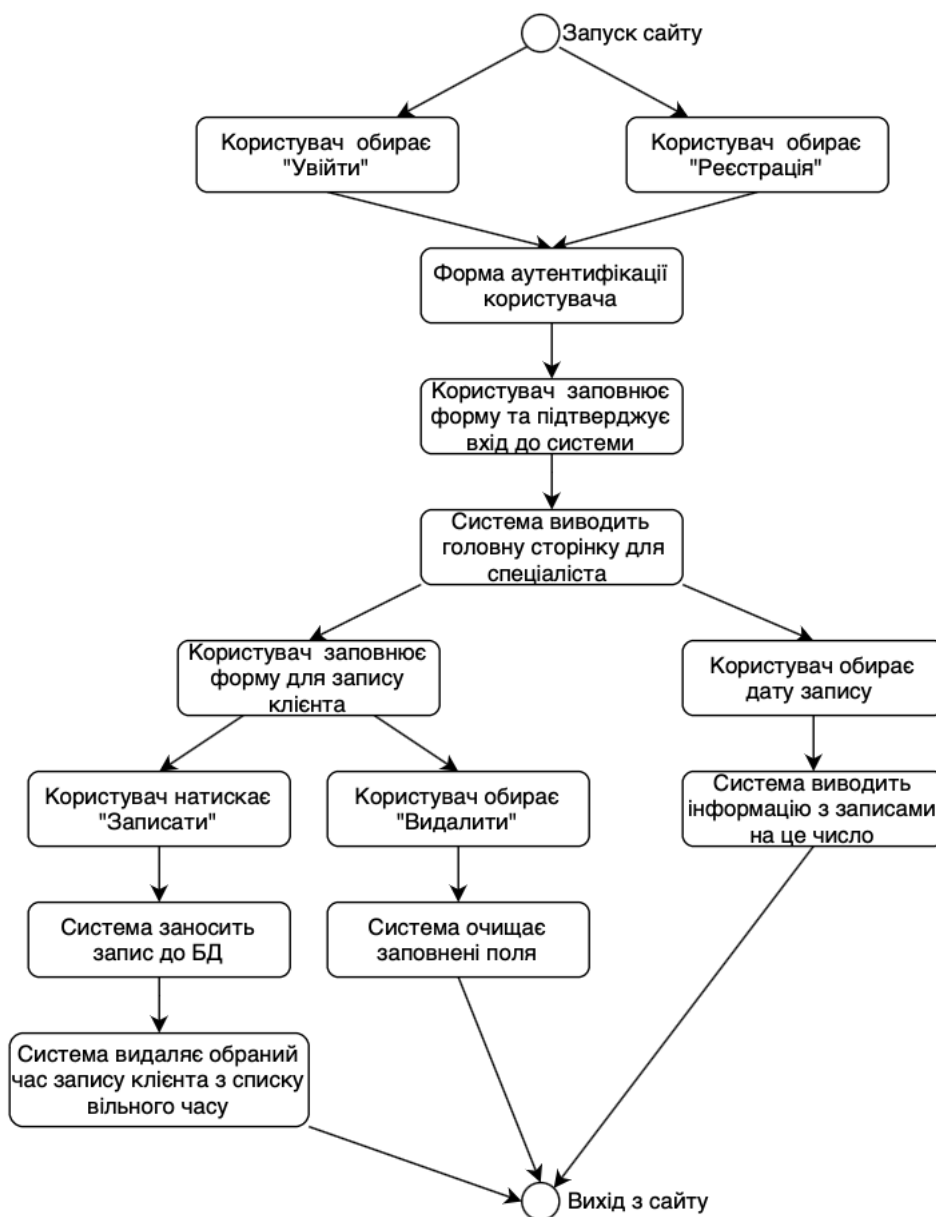


Рис. 1.3 – Функціональна схема для користування сайту спеціалістом

По функціональній схемі для користування сайту спеціалістом бачимо, що після запуску сайту для спеціаліста є два варіанти: «Увійти» або «Реєстрація». Далі заповнення форми аутентифікації користувача та підтвердження входу до системи. Система виводить на головну сторінку для спеціаліста. Потім є два варіанти. Перший – користувач обирає дату запису після чого система виводить інформацію з записом на це число. Другий – користувач натискає «Записатися»

і система заносить запис до БД. Система видаляє обраний час запису клієнта з списку вільного часу. Якщо користувач натискає «Видалити», то система очищає заповнені поля.

Багато функцій виконує адміністратор сайту, тому для спрощення його роботи вони були автоматизовані. В повноваження адміністратора входить як робота з інформацією про клієнта, так і з інформацією про спеціалістів. Тому, для зручності, функціональну схему було розділено на дві – схема, на якій зображено роботу над інформацією з клієнтами (рисунок 1.4) та друга – схема з роботою над інформацією для спеціалістів (рисунок 1.5). Так як функції для запису клієнтів та перегляд записів є спільними як для спеціаліста, так і для адміністратора, тому, щоб не повторюватися у схему їх не внесено[12].

На рисунку 1.4 бачимо функціональну схему для роботи над інформацією з клієнтами. Спочатку користувач входить на сторінку для адміністратора, заповнює форму для аутентифікації. Після чого система виводить його на головну сторінку сайту, де є різні функції для запису клієнтів та перегляду записів, які співпадають з функціями як і у спеціалістів (рисунок 1.3). Адміністратор обирає «Виписати чек» і його виводить на сторінку для виписки чеку, де адміністратор обирає того спеціаліста, в якого був клієнт. Після відкривається весь список клієнтів, які були в спеціаліста саме в обраний день. Після того, як адміністратор обирає клієнта для якого треба роздрукувати чек і в нього відкривається документ у форматі pdf. Є інший варіант використання – адміністратор обирає «Клієнти». Система виводить всіх клієнтів та форму для редагування. Адміністратор може обрати різні функції, які може відредагувати – такі як, «Додати», «Оновити» та «Видалити». Для того, щоб їх використати, необхідно заповнити форму та натиснути на потрібну кнопку[6].

Функціональна схема забезпечує адміністратору повний контроль над системою, дозволяючи виконувати адміністративні завдання та забезпечувати безпеку та надійність роботи системи.

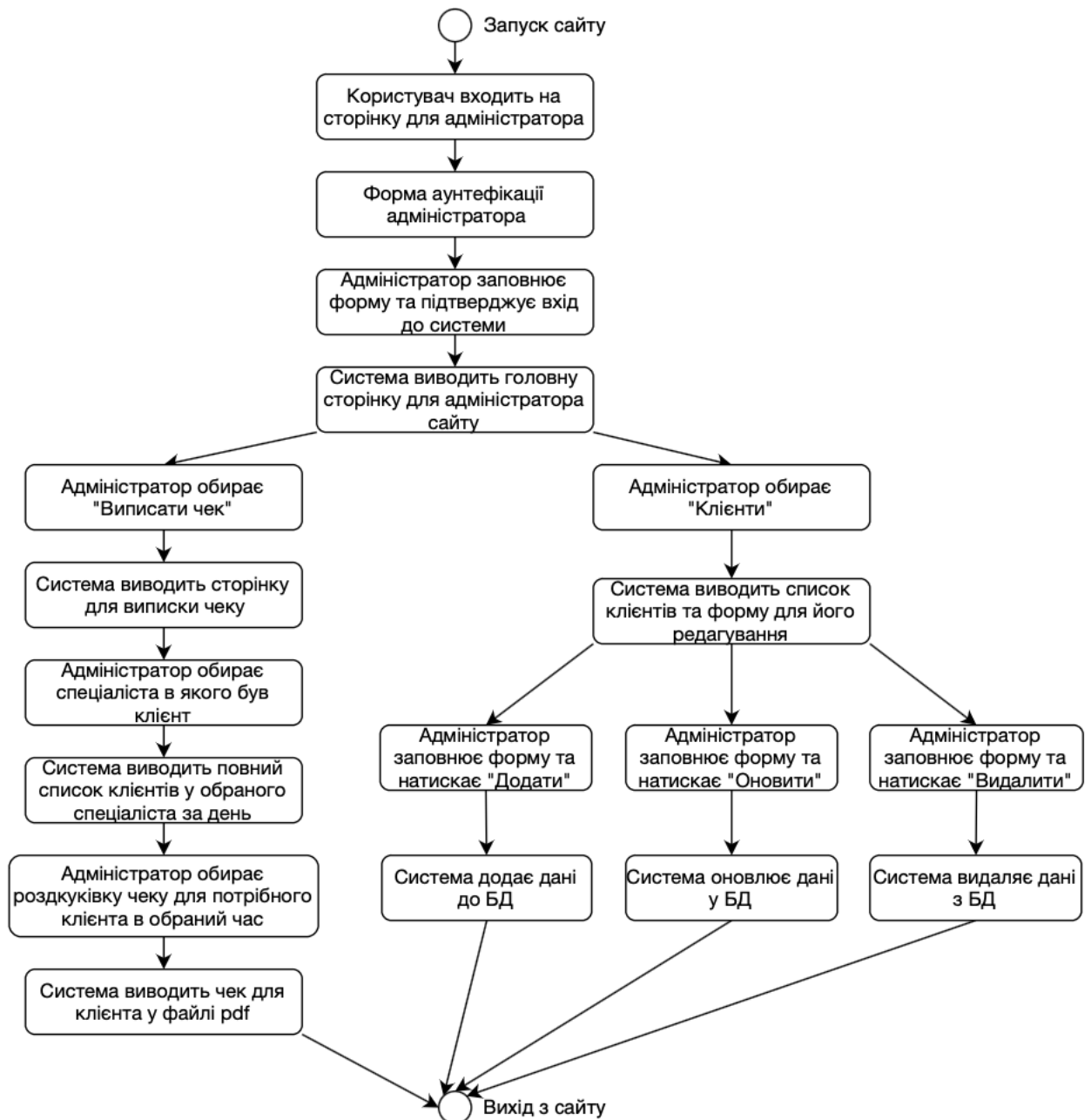


Рис. 1.4 – Функціональна схема для користування сайту адміністратором

На рисунку 1.5 бачимо функціональну схему для користування сайту адміністратором із інформацією про спеціалістів та закладів. Як і раніше, спочатку користувач входить на сторінку для адміністратора, заповнюючи форму аунтефікації увівши логін та пароль. Після чого може обрати дві функції: «Прайс лист» та «Дохід». Якщо адміністратор обирає «Прайс лист», то система виводить документ з тарифами на послуги у форматі pdf. Якщо адміністратор обирає «Дохід», то система виводить сторінку для розрахунку доходу. На цій сторінці адміністратор може обрати спеціаліста та місяць і

система підрахує та виведе документ у форматі pdf з доходом окремого спеціаліста. Є інша функція – адміністратор обирає не окремого спеціаліста, а «Загальний дохід», де система підраховує і виводить доходи компанії за потрібний період.

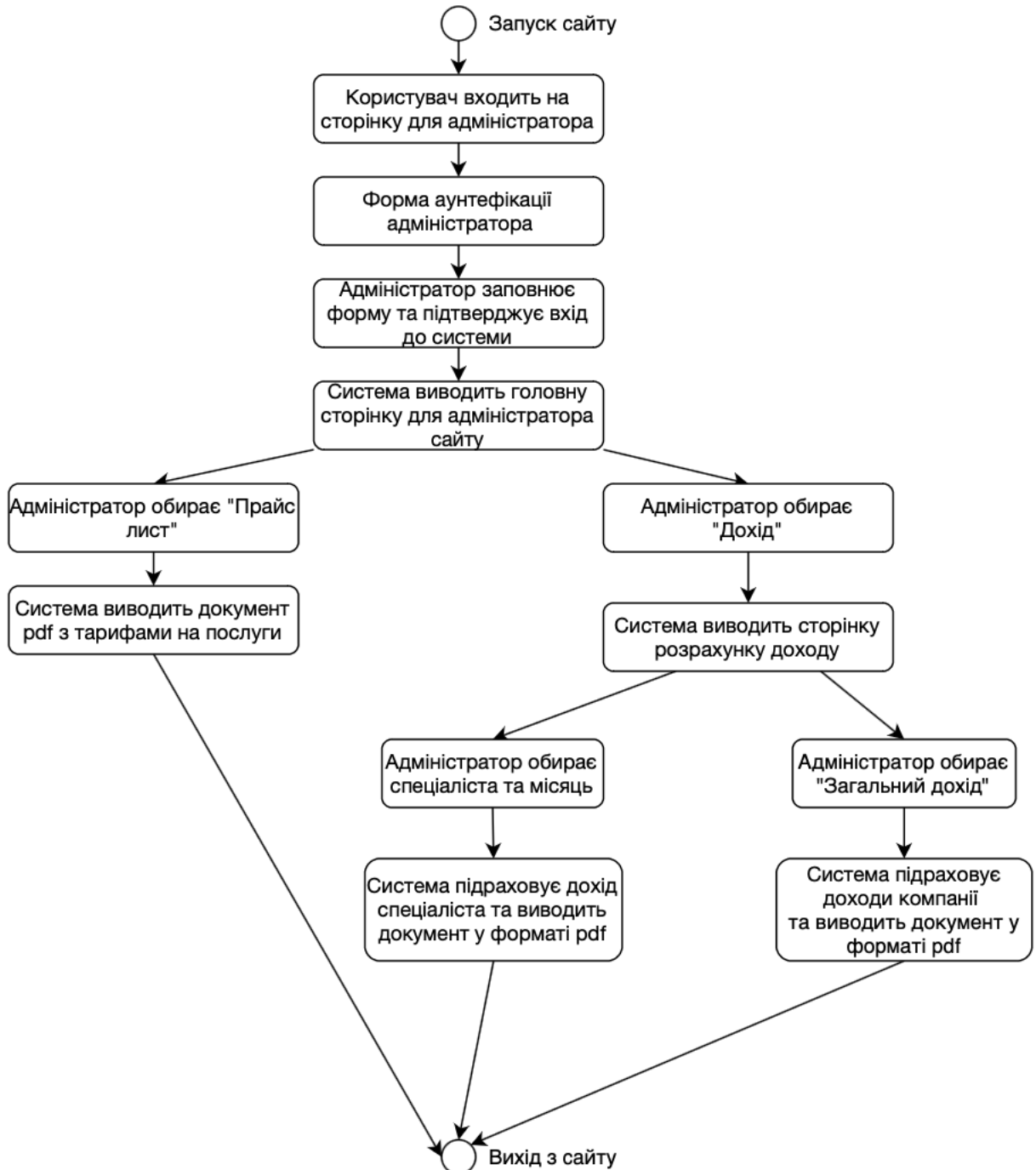


Рис. 1.5 – Функціональна схема для користування сайту адміністратором

При використуванні сайту «Clinics Up» активними учасниками є спеціалісти та клієнти салону, які взаємодіють із інтерфейсом сайту та

виконують певні дії. Сайт відповідає на ці дії та надає результати користувачам. Ці результати сприяють зручному пошуку необхідної інформації, полегшують управління салоном та дозволяють швидко генерувати необхідні звіти.

У розділі "Розробка функціональної схеми сайту" була детально розглянута структура та функціональні можливості веб-сайту системи пошуку спеціалістів у галузі краси та здоров'я. Згідно з проведеним аналізом та розробкою, можна зробити наступні висновки:

Чітко визначені функціональні можливості. Функціональна схема сайту надає чіткий огляд можливостей системи, включаючи авторизацію, реєстрацію, керування користувачами, модерацію контенту, керування замовленнями, системні налаштування, повідомлення, статистику та підтримку. Кожна функція має свою роль у забезпеченні ефективності та зручності користування.

Логічна взаємодія елементів системи. Функціональна схема відображає логічну взаємодію між різними компонентами системи, яка дозволяє користувачам та адміністраторам ефективно керувати системою та виконувати потрібні завдання.

Забезпечення безпеки та контролю: Система включає функції модерації контенту та управління користувачами, що сприяє забезпеченню безпеки та якості користування сайтом.

Можливість майбутнього розширення: Функціональна схема передбачає можливість майбутнього розширення функціоналу сайту, що дозволить системі адаптуватися до змінних потреб користувачів та ринку.

В цілому, розроблена функціональна схема є ключовим кроком у розробці веб-сайту та системи для пошуку спеціалістів у сфері краси та здоров'я. Вона забезпечує ясність та структурованість системи, полегшуючи її розробку та забезпечуючи зручне користування для всіх сторін.

2.3 Розробка алгоритму роботи сайту

Усі програми мають власну послідовність операцій, яку називають алгоритмом.

Алгоритм - це набір конкретних інструкцій, які описують послідовність дій, які повинен виконати виконавець, щоб досягти бажаного результату у вирішенні завдання за скінченну кількість дій. Це система правил, які регулюють виконання конкретного процесу та допомагають досягнути мети за обмежений час. Для того, щоб краще зрозуміти ці інструкції, часто використовують графічні схеми, відомі як блок-схеми[7].

Сучасний світ заснований на інформаційних технологіях, і веб-сайти відіграють ключову роль у забезпеченні доступу до цієї інформації та послуг. Розробка та оптимізація алгоритмів роботи веб-сайту стає критично важливою, оскільки вони визначають спосіб, яким користувачі взаємодіють із системою, як система обробляє їхні запити та як забезпечує комфортне користування.

У цьому розділі нашої дипломної роботи ми глибоко дослідимо процес розробки алгоритму роботи веб-сайту, зосереджуючись на створенні оптимальної послідовності дій для досягнення високої продуктивності та зручності користування. Ми розглянемо важливі аспекти, такі як вимоги до системи, взаємодія з користувачами, обробка запитів та безпека даних, і розробимо алгоритм, який гармонійно об'єднує всі ці складові в єдиний функціональний механізм. Розробка алгоритму роботи веб-сайту - це крок вперед у забезпеченні якості та зручності користування, і він має велике значення для досягнення успіху в галузі розвитку онлайн-платформ.

Цей розділ роботи є необхідним етапом в розробці будь-якого веб-сайту, оскільки він визначає те, як сайт буде взаємодіяти з користувачами та обробляти їхні запити. Правильно розроблений алгоритм дозволяє покращити продуктивність та функціональність сайту, забезпечуючи надійну та зручну платформу для користувачів. У цьому розділі ми розкриємо всі аспекти розробки

алгоритму та надамо читачам глибше розуміння важливості цього процесу в сучасному інтернет-середовищі.

Оскільки наша система має прямий зв'язок з базою даних, давайте проілюструємо це, наводячи приклади алгоритмів для деяких основних команд, які використовуються при виконанні завдань системи.

Алгоритм дії команди SELECT, представлений у вигляді схеми (Рис. 2.1), в мові запитів SQL (Structured Query Language) при роботі з базою даних полягає в виділенні конкретних даних або записів з бази даних, які відповідають вказаним умовам. Відбір даних за допомогою команди SELECT може бути розглянутий у наступних кроках:

1. **Сформування команди SELECT:** Першим кроком є написання команди SQL "SELECT", за якою слідує поля, які вас цікавлять, із таблиці, з якої потрібно витягти дані.
2. **Відправка команди до бази даних:** Сформовану команду SELECT відправляють до системи управління базами даних (СУБД).
3. **Виконання запиту:** СУБД виконує команду SELECT, перевіряючи всі записи в таблиці, щоб знайти ті, які відповідають умовам, вказаним у запиті.
4. **Повернення результатів:** Після виконання запиту СУБД повертає результати, які включають всі вибрані поля для записів, які задовольняють вказаним умовам.
5. **Обробка результатів:** Результати запиту можна обробити у програмі або веб-сервері, які взаємодіють з базою даних. Вони можуть відобразити результати на веб-сторінці або використовувати їх для подальшого аналізу або виведення користувачу.

Цей алгоритм дії команди SELECT використовується для ефективного вибору та отримання потрібних даних з бази даних під час роботи з інформацією.

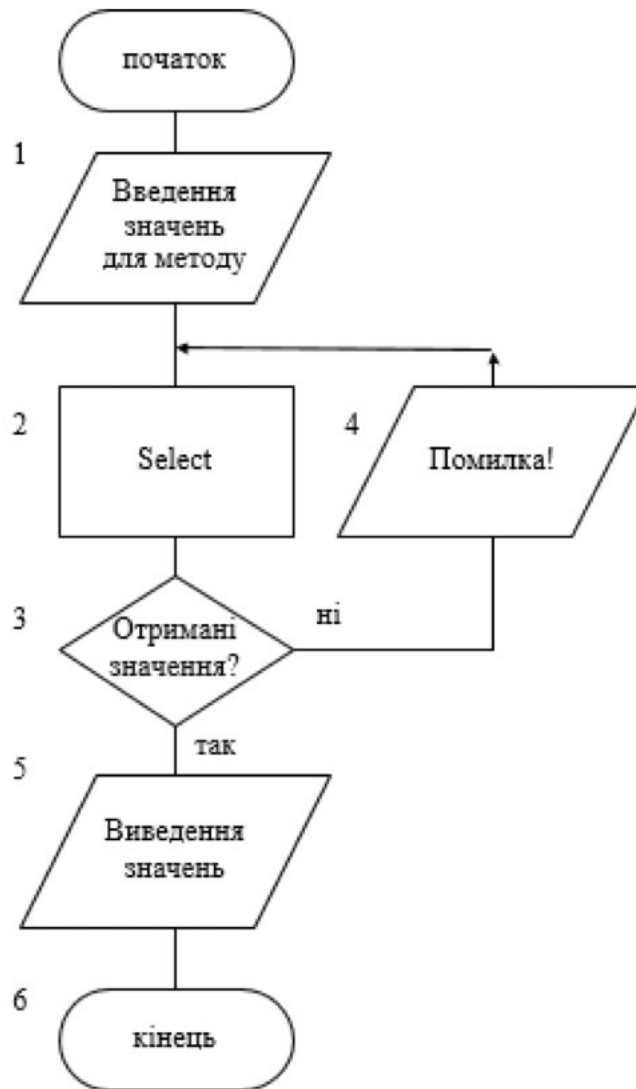


Рис. 2.1 – Алгоритм команди «SELECT»

Алгоритм дії команди "DELETE", представлений у вигляді схеми (Рис. 2.2), в мові запитів SQL (Structured Query Language) при роботі з базою даних передбачає видалення записів із таблиці бази даних. Основні кроки алгоритму видалення записів з бази даних виглядають так:

1. **Сформування команди DELETE:** Перший крок полягає в написанні команди SQL "DELETE" разом із зазначенням таблиці, з якої потрібно видалити записи, та встановлення умов, за якими видалення повинно відбутися.
2. **Відправка команди DELETE до бази даних:** Сформовану команду DELETE відправляють до системи управління базами даних (СУБД).

3. **Виконання запиту:** СУБД виконує команду DELETE та перевіряє всі записи в таблиці, щоб знайти ті, які відповідають вказаним умовам для видалення.

4. **Видалення записів:** Знайдені записи, які відповідають умовам, видаляються із таблиці. Вони більше не доступні для подальшого використання або виведення.

5. **Повернення результатів (необов'язково):** Команда DELETE зазвичай не повертає якісь результати, оскільки її основна функція - видалення записів, а не отримання даних.

Алгоритм команди DELETE дозволяє впоратися із забезпеченням актуальності та чистоти даних у базі даних, а також використовується для видалення зайвої інформації або даних, які більше не потрібні. Надійність та обережність при використанні цієї команди мають велике значення, оскільки видалені дані зазвичай неможливо відновити.

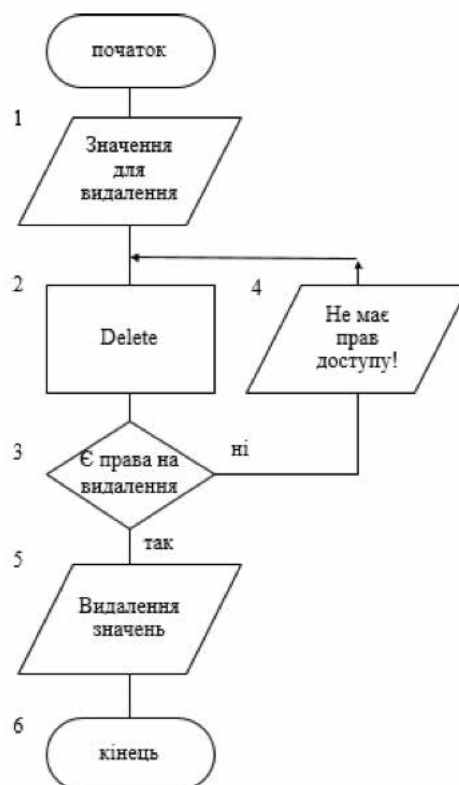


Рис. 2.2 – Алгоритм команди «DELETE»

Алгоритм дії підключення до бази даних (БД), представлений у вигляді схеми (рисунок 2.3), в програмі може бути описаний наступними кроками:

1. **Визначення параметрів підключення:** Спочатку визначаються параметри, необхідні для підключення до БД. Це включає ім'я сервера БД, ім'я користувача, пароль, назву самої БД, а також інші параметри, такі як порт чи метод аутентифікації.
2. **Ініціалізація підключення:** Програма ініціалізує підключення до БД, використовуючи визначені параметри. Це може бути здійснене шляхом виклику відповідного методу чи функції для обраної мови програмування.
3. **Підключення до сервера БД:** Програма встановлює фізичне підключення до сервера бази даних, використовуючи вказані параметри. В цей момент програма зв'язується з сервером і намагається аутентифікуватися.
4. **Аутентифікація користувача:** Сервер БД перевіряє введені ім'я користувача та пароль. Якщо вони вірні, користувач отримує доступ до бази даних. У випадку невірних облікових даних зазвичай генерується помилка.
5. **Виконання запитів та робота з даними:** Після успішного підключення до БД, програма може виконувати запити до БД, витягувати, оновлювати, вставляти або видаляти дані за допомогою SQL-запитів.
6. **Завершення роботи та роз'єднання:** По завершенню роботи з базою даних програма повинна закрити підключення, щоб звільнити ресурси сервера БД. Це важливо для підтримання надійності та ефективності сервера.
7. **Обробка помилок та винятків:** Під час роботи з БД можуть виникати помилки, такі як недоступність сервера, неправильні дані, конфлікти тощо. Програма повинна передбачити обробку цих ситуацій і відповідну поведінку, включаючи виведення повідомлень про помилку або реагування на них.

Цей алгоритм є загальним описом процесу підключення до бази даних у програмному кодї. Кожна мова програмування може мати свої особливості та

бібліотеки для роботи з базами даних, але загальний принцип залишається схожим.



Рис. 2.3 – Алгоритм підключення до БД

Оскільки програмний код додатку має значний обсяг та включає в себе велику кількість функцій, нижче буде наведений опис алгоритмів декількох важливих функцій, які виконує сам додаток. Зокрема, це алгоритм перевірки логіну та паролю при спробі входу в систему, що є однією з ключових операцій у взаємодії з клієнтом (Рис. 2.4), а також алгоритм перевірки наявності співробітника у системі під час реєстрації нового користувача (рисунок 2.5).

На рисунку 2.4 зображено алгоритм перевірки логіну та паролю при вході в систему можна описати наступними кроками:

1. **Введення даних користувачем:** Користувач вводить свій логін (ідентифікатор) та пароль в спеціальні поля на веб-сторінці або додатку.

2. **Відправка запиту на сервер:** Після введення логіна та паролю, користувач натискає кнопку "Увійти" або подібний елемент інтерфейсу. Додаток або веб-сайт відправляє запит на сервер.

3. **Перевірка існування користувача:** Сервер отримує запит та перевіряє, чи існує користувач із введеним логіном в базі даних.

4. **Порівняння паролів:** Якщо користувач існує, сервер знаходить збережений в базі даних пароль, який був пов'язаний із цим логіном під час реєстрації. Потім сервер порівнює введений користувачем пароль із тим, що зберігся в базі даних.

5. **Авторизація:** Якщо введений пароль відповідає збереженому паролю для цього логіну, сервер вважає користувача авторизованим і дозволяє доступ до облікового запису.

6. **Неспівпадіння паролів або невірний логін:** Якщо введений пароль не відповідає збереженому паролю або логін невірний, сервер видає повідомлення про помилку та можливість повторно ввести дані.

7. **Відновлення пароля:** У випадку забутого пароля користувач може мати можливість відновити його, отримавши листа з посиланням на відновлення пароля на вказану електронну пошту.

Цей алгоритм забезпечує безпеку входу в систему шляхом перевірки ідентифікатора та паролю користувача, а також враховує можливість відновлення паролю в разі забуття.

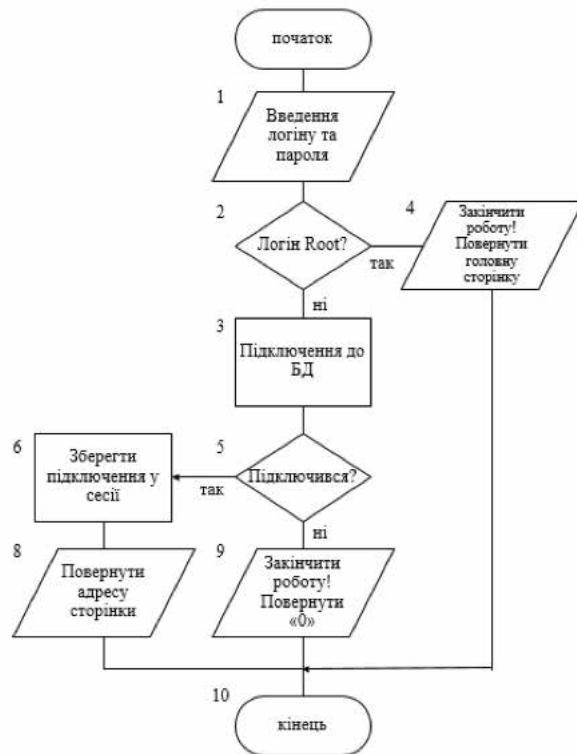


Рис. 2.4 – Алгоритм перевірки логіну та паролю

Алгоритм перевірки існування співробітника в системі при реєстрації нового користувача може бути наступним чином (Рис. 2.5):

1. **Введення даних користувачем:** При реєстрації нового користувача, той вводить свої особисті дані, такі як ім'я, прізвище, адреса електронної пошти та інші обов'язкові або додаткові дані.
2. **Відправка запиту на сервер:** Після введення всіх необхідних даних, користувач натискає кнопку «Зареєструватися» або подібний елемент інтерфейсу, що ініціює відправку запиту на сервер.
3. **Перевірка існування співробітника:** Сервер отримує запит та перевіряє, чи існує користувач із введеними даними (наприклад, за адресою електронної пошти) в базі даних серед співробітників.
4. **Умова перевірки:** Якщо сервер не знаходить запису із введеними даними серед співробітників, то вважається, що такого користувача не існує, і реєстрація може продовжитися.

5. **Конфлікт існування:** Якщо сервер виявляє співпадіння введених даних із вже існуючим співробітником, то може бути видача повідомлення про конфлікт, і користувач повинен виправити введені дані.

6. **Продовження реєстрації:** Якщо перевірка показала, що такого співробітника не існує, реєстрація може продовжитися, і дані користувача додаються до бази даних як новий обліковий запис.

7. **Збереження даних користувача:** Сервер зберігає введені дані нового користувача в базі даних та встановлює йому права доступу до системи.

8. **Відправка підтвердження:** Після успішної реєстрації може бути відправлене повідомлення користувачеві, підтверджуючи його реєстрацію та надаючи необхідну інформацію для входу в систему.

Цей алгоритм дозволяє переконатися, що при реєстрації нового користувача в системі не виникає конфліктів із вже існуючими співробітниками і що дані користувача зберігаються правильно в базі даних.

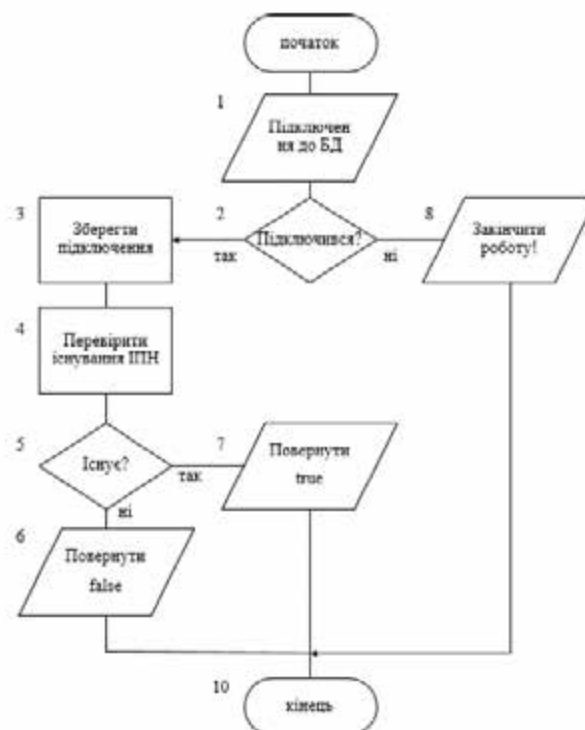


Рис. 2.5 – Алгоритм перевірки існування співробітника

Висновок до розділу 2

В розділі 2 нашої дипломної роботи було проведено проектування та моделювання веб-сайту, яке є ключовим етапом у розробці інформаційних систем. Процес проектування розпочався з визначення функціональних та нефункціональних вимог до веб-сайту, включаючи основні функціональності, інтерфейс користувача, безпеку та продуктивність.

Було проведено аналіз сучасних підходів до розробки веб-сайтів, визначено основні складові технологічного стеку та вибрано їх для використання у нашому проекті. Детально було описано архітектуру веб-сайту, включаючи клієнтську та серверну сторони, а також базу даних.

Робота включала в себе розробку інтерфейсу веб-сайту, де було враховано сучасні дизайн-тенденції та практики щодо зручності користувачів. Проаналізовано можливості для вдосконалення користувацького досвіду та забезпечення ефективної навігації.

Завершальним кроком було проведення моделювання бази даних, де визначено сутності, зв'язки між ними та атрибути. Важливою частиною була розробка схеми бази даних, яка дозволить ефективно зберігати і керувати інформацією.

3 РОЗРОБКА БАЗИ ДАНИХ І ВЕБСАЙТУ

3.1 Аналіз обраного середовища програмування

Для розробки даного вебсайту обрано два середовища програмування, такі як, WebStorm та PHPStorm.

WebStorm - це інтегроване середовище розробки (IDE), розроблене для веб-розробників, що вирізняється своєю спеціалізацією на розробці веб-додатків та веб-сайтів. Ця платформа надає розробникам розширені засоби та можливості для ефективного програмування веб-додатків. Однією з ключових особливостей WebStorm є підтримка різних мов програмування, включаючи JavaScript, HTML, CSS, TypeScript та інші.

Інтегровані інструменти для автодоповнення коду допомагають розробникам писати код швидше та уникати синтаксичних помилок. Вбудований відлагоджувач спрощує виявлення та виправлення помилок в JavaScript-кодi, забезпечуючи більш зручну відладку веб-додатків.

WebStorm підтримує популярні веб-фреймворки та бібліотеки, такі як Angular, React, Vue.js, і надає розробникам інструменти для зручної роботи з ними. Можливості рефакторингу дозволяють розробникам оптимізувати структуру коду та підтримувати його в хорошому стані.

Також підтримує системи керування версіями, зокрема Git, і надає інтерфейс для ефективного керування версіями коду. Це сприяє колаборації в команді та відстеженню змін у проекті.

Усі ці можливості роблять WebStorm потужним інструментом для розробки веб-додатків і забезпечують розробникам продуктивне та комфортне середовище для веб-розробки.

PHPStorm - це інтегроване середовище розробки (IDE), яке розроблене спеціально для PHP-розробників. Ця платформа надає розробникам широкий набір інструментів і можливостей для розробки веб-додатків і веб-сайтів на мові програмування PHP.

PHPStorm спрощує розробку веб-додатків, забезпечуючи продуктивність і зручність для розробників. Його основною метою є полегшення процесу програмування на мові PHP та забезпечення високоякісного коду. Від автодоповнення коду та відлагодження до підтримки фреймворків і систем керування версіями, PHPStorm надає інструменти для швидкої та ефективної розробки веб-додатків.

Ця IDE також підтримує різні інші технології, що дозволяє розробникам працювати з різними стеками та розширювати можливості розробки. Загалом, PHPStorm створений з урахуванням потреб PHP-розробників і надає їм інструменти для успішної розробки веб-додатків на цій мові програмування.

3.2 Аналіз обраних технологій для створення вебсайту

Технології, які були використані при розробці вебсайту, включають SSR (Server-Side Rendering), SPA (Single Page Application), STATIC APP (статичні веб-додатки) та ASP.NET. Кожна з цих технологій відіграє свою роль у створенні вебсайту.

SSR (Server-Side Rendering): SSR використовується для створення веб-сторінок на сервері та відправки готового HTML-коду на клієнтський браузер. Це дозволяє швидко відобразити вміст сторінки, що поліпшує швидкодію та покращує SEO, оскільки пошукові системи можуть індексувати вміст.

SPA (Single Page Application): SPA є моделлю розробки, де весь веб-додаток завантажується один раз, а потім змінюється динамічно, без перезавантаження сторінки. Ця технологія дозволяє створювати інтерактивні додатки з швидкими переходами між сторінками.

STATIC APP (статичні веб-додатки): Статичні додатки генеруються перед розгортанням і відомі як "передзавантажені" додатки. Вони ідеально підходять для веб-сайтів зі статичним контентом, який рідко змінюється. Вони можуть забезпечити високу швидкодію завдяки кешуванню та ефективному використанню CDN.

ASP.NET: Це фреймворк для розробки веб-додатків, який надає інструменти для створення динамічних та високопродуктивних веб-додатків. Він використовується для створення серверних частин веб-додатків та обробки запитів від клієнтів.

	SSR	SPA	STATIC APP	ASP.NET
Багатофункціональність	+	+	+	+
Продуктивність	+	+	+/-	+/-
Простота використання	+/-	+/-	+	+/-
Наявність доступних програмних бібліотек	+	+	+/-	-
Розподіл дизайну та логіки	-	-	+	+

Табл. 3.1 – Порівняння сучасних технологій розробки вебсайтів

У комбінації цих технологій, вебсайт може бути розроблений з урахуванням потреб проекту, забезпечуючи як високу швидкість, так і інтерактивність. Використання SSR може покращити швидкість та SEO, SPA - забезпечити інтерактивність, STATIC APP - полегшити кешування статичного контенту, а ASP.NET - обробку серверних запитів та взаємодію з базою даних.

3.3 Розробка бази даних

3.3.1 Концептуальне проектування бази даних

Перший крок у процесі розробки бази даних називається концептуальним моделюванням. У цьому етапі ми створюємо абстрактну модель даних для певної частини організації. Під час розробки концептуальної моделі ми спочатку визначаємо, які основні об'єкти даних існують (сутності) і як вони пов'язані між собою[3].

Вивчивши предметну область, можна ідентифікувати наступні ключові елементи:

- client (клієнти);

- employee (спеціалісти) ;
- register (запис);
- specialty (спеціальність);
- users (користувачі);
- service (сервіс).

Можна пояснити, що між сутностями "Клієнти" та "Запис" існує 1:М зв'язок, де кожен запис (запис на прийом до майстра) може бути пов'язаний з багатьма клієнтами. Однак кожен клієнт ідентифікується унікальним номером, тобто кожен номер клієнта посилається тільки на одного клієнта.

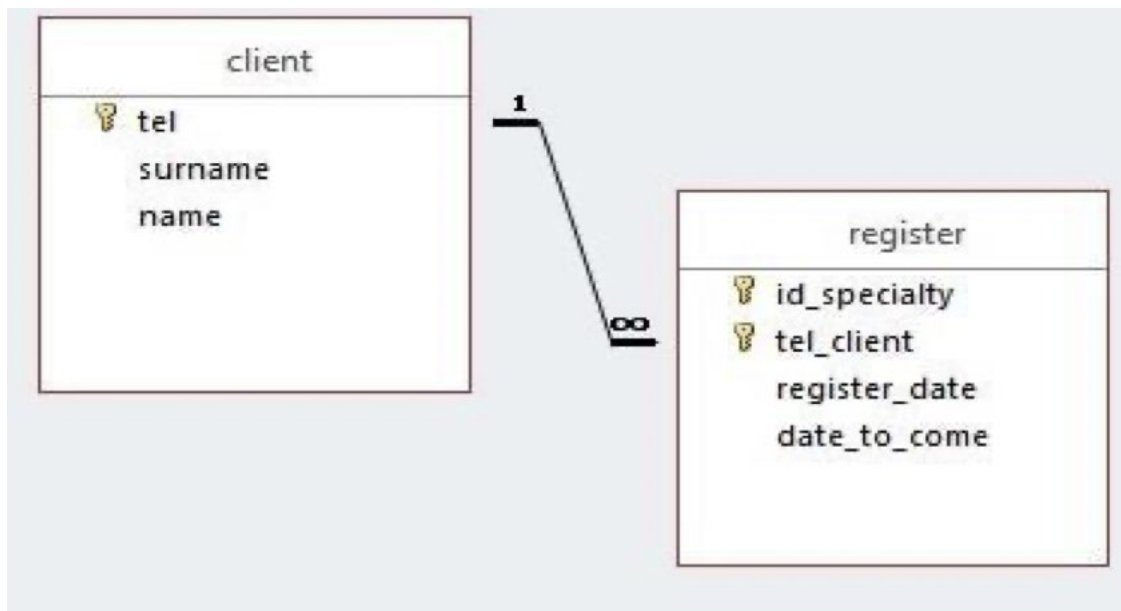


Рис. 3.1 – Зв'язок сутностей Клієнт та Запис

Між сутностями "Спеціаліст" та "Спеціальність" існує зв'язок 1:М, де кожен співробітник ідентифікується унікальним податковим номером, тобто на одного співробітника припадає тільки одна спеціальність. Проте однакова спеціальність може бути призначена багатьом співробітникам.

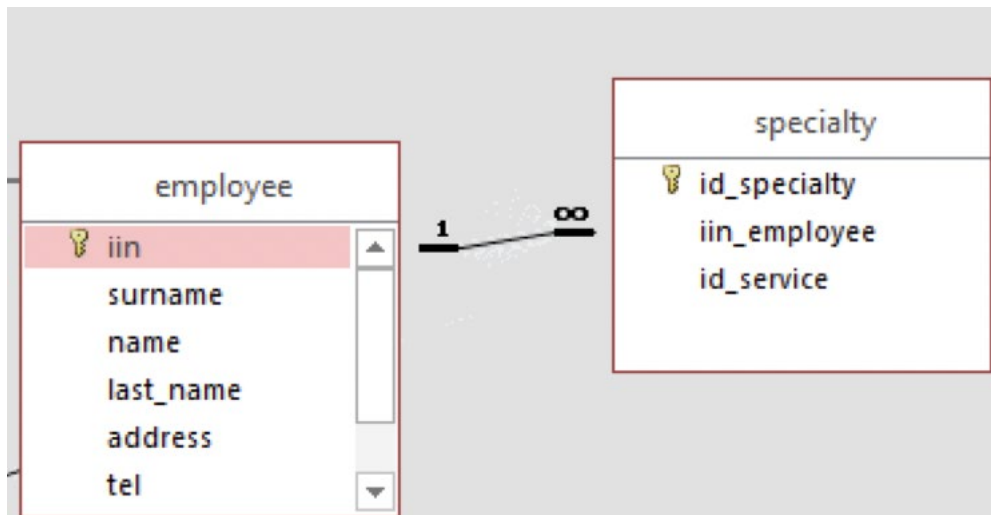


Рис. 3.2 – Зв'язок сутностей Спеціаліст та Спеціальність

Між сутністю "Сервіс" та "Спеціальність" існує 1:М зв'язок, оскільки кожен сервіс ідентифікується за унікальним ім'ям, тобто для кожного імені сервісу є тільки один відповідний запис. Проте різні майстри можуть пропонувати низку різних сервісів, тому зв'язок між майстрами та послугами є одному до багатьох.

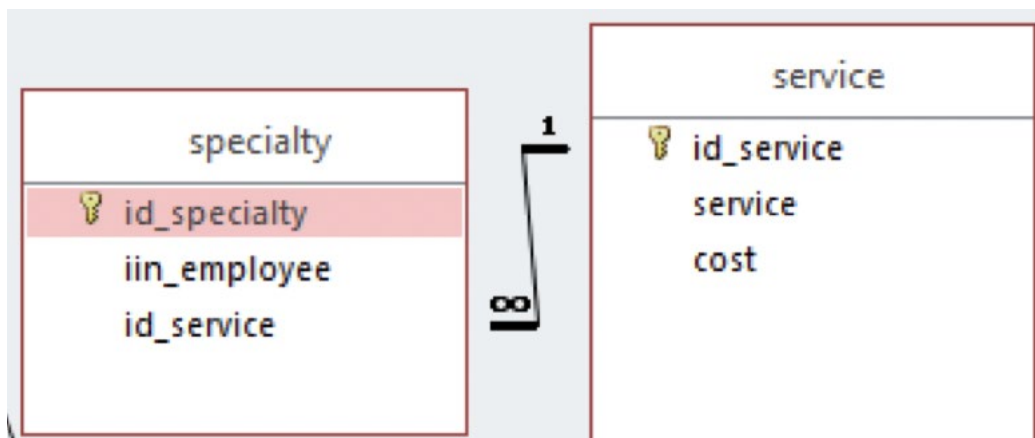


Рис. 3.3 – Зв'язок сутностей Сервіс та Спеціальність

Можна пояснити, що між сутностями "Спеціаліст" та "Користувач" існує однозначний зв'язок, оскільки кожен співробітник має свій унікальний обліковий запис (логін) у системі. Іншими словами, кожному співробітнику в системі відповідає лише один користувач, і навпаки.

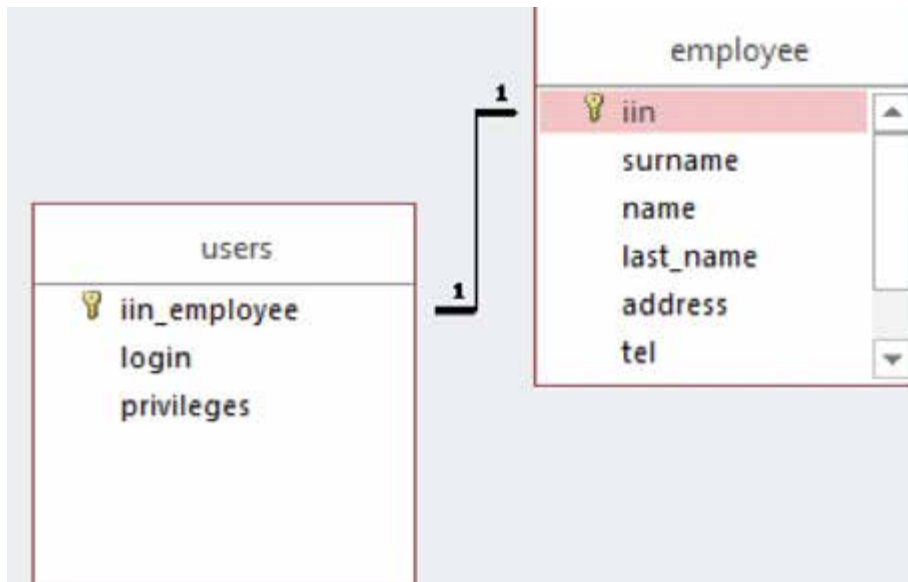


Рис. 3.4 – Зв'язок сутностей Спеціаліст та Користувач

Між сутністю "Спеціальність" та "Запис" існує однозначний зв'язок, оскільки для кожної конкретної спеціальності існує лише один запис. Проте цю саму спеціальність можна використовувати багато разів, оскільки вона може бути пов'язана з численними записами.

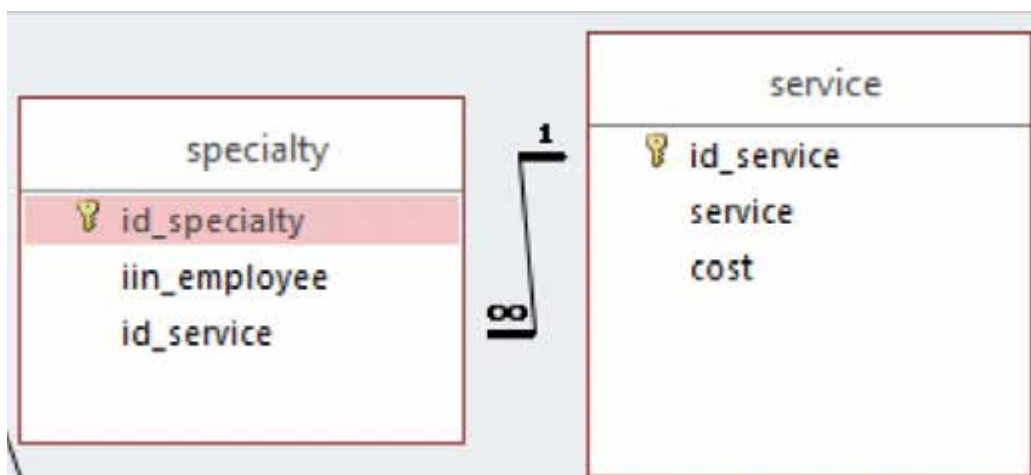


Рис. 3.5 – Зв'язок сутностей Запис та Спеціальність

Отже, отримана ER-модель бази даних представлена на рисунку 3.6

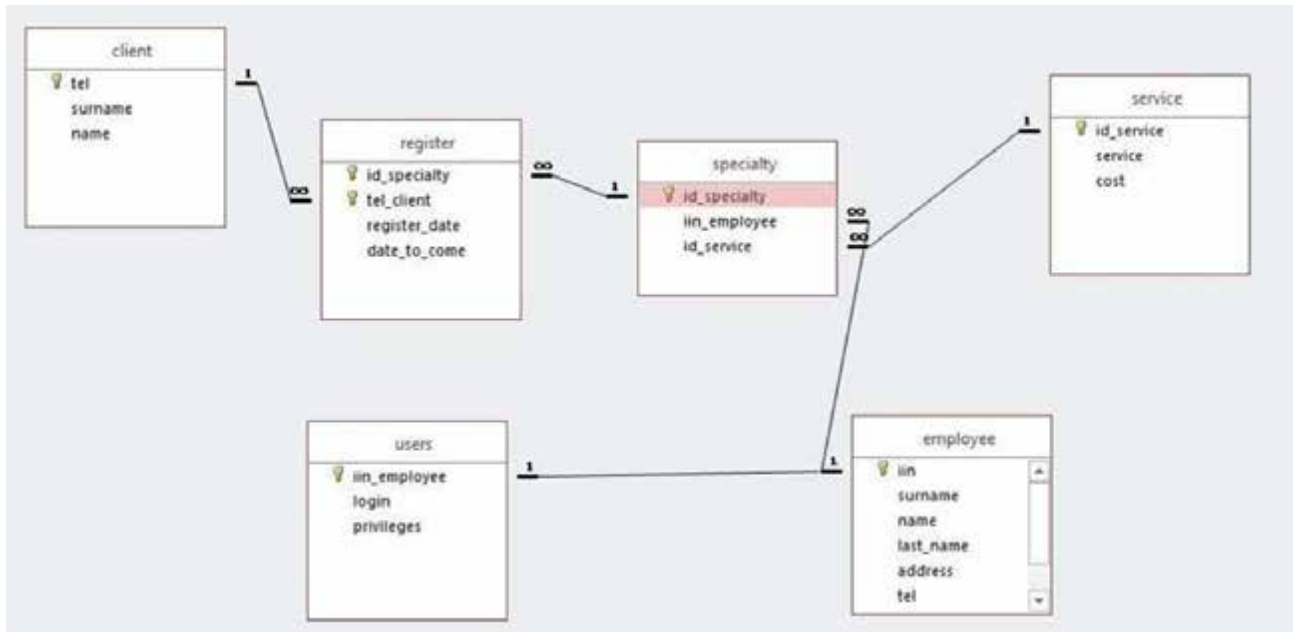


Рис. 3.6 – Результиуюча ER – модель

3.3.2 Логічне проектування бази даних

Логічне проектування бази даних - це етап, на якому концептуальна модель перетворюється в логічну модель, з урахуванням вибору конкретної системи управління базами даних (СУБД). Головне завдання логічного проектування полягає у створенні логічної схеми, яка відповідає обраній СУБД. Оскільки більшість сучасних СУБД є реляційними, то концептуальну модель бази даних слід відобразити на реляційну модель.

За допомогою ER-моделі бази даних було визначено, що проектувана база даних повинна включати такі таблиці:

- client (клієнти);
- employee (спеціалісти) ;
- register (запис);
- specialty (спеціальність);
- users (користувачі);
- service (сервіс).

Інформація про характеристики атрибутів таблиць, які були описані раніше, представлена на рисунках від 3.7 до 3.12.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 tel	varchar(255)	utf8mb4_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2 surname	varchar(255)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 name	varchar(255)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё

Рис. 3.7 – Список атрибутов сущности Клиенти

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 iin	varchar(12)	utf8mb4_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2 surname	varchar(20)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 name	varchar(20)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	4 last_name	varchar(20)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	5 address	varchar(80)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	6 tel	varchar(20)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	7 birth_date	date			Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	8 employee_type	varchar(30)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё

Рис. 3.8 – Список атрибутов сущности Специалисты

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id_specialty	int(10)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2 tel_client	varchar(20)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 register_date	date			Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	4 date_to_come	timestamp			Нет	0000-00-00 00:00:00			Изменить Удалить Ещё

Рис. 3.9 – Список атрибутов сущности Запис

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id_specialty	int(10)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2 iin_employee	varchar(12)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 id_service	int(10)			Да	NULL			Изменить Удалить Ещё

Рис. 3.10 – Список атрибутов сущности Специальность

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id_specialty	int(10)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2 iin_employee	varchar(12)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 id_service	int(10)			Да	NULL			Изменить Удалить Ещё

Рис. 3.11 – Список атрибутов сущности Користувачі

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id_service	int(10)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2 service	varchar(80)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 cost	int(10)			Да	NULL			Изменить Удалить Ещё

Рис. 3.12 – Список атрибутов сущности Сервіс

Під час логічного проектування бази даних була створена логічна модель, виходячи з концептуальної моделі, яка відповідає реляційній моделі даних. Усі

таблиці перебувають у третій нормальній формі (3НФ), всі атрибути в кожній таблиці повністю залежать від її ключа, і в таблицях відсутні дубльовані дані, а також немає транзитивних та часткових функціональних залежностей.

3.3.3 Створення таблиць та зв'язків між ними

Мова SQL не тільки призначена для обробки даних, але і дозволяє виконувати всі можливі операції з базами даних і таблицями, включаючи створення та управління таблицями. Існують два способи створення таблиць:

Багато Систем Управління Базами Даних (СУБД) мають візуальний інтерфейс для інтерактивного створення та керування таблицями.

Також можна маніпулювати таблицями за допомогою SQL-операторів. Зауважимо, що при використанні інтерактивного інструментарію СУБД фактично генерується SQL-код автоматично, і користувач не завжди бачить ці команди напряду (це схоже на запис макросу в Excel, коли макрорекодер фіксує ваші дії і перетворює їх у команди VBA, що виконуються безпосередньо)[2].

У середовищі СУБД MySQL таблиці створюються вручну шляхом використання SQL-оператора CREATE TABLE. Для створення таблиці за допомогою цього оператора потрібно вказати наступні дані:

- Ім'я таблиці, яке визначається після ключового слова CREATE TABLE.
- Інформацію про стовпці таблиці, включаючи їх назви та визначення, розділені комами.
- Деякі СУБД можуть також вимагати вказання місця розташування таблиці.

Приклад створення таблиці наданий нижче та інші таблиці створюються відповідно:

```
create table register(id_specialty int unsigned,tel_client
varchar(20), register_date date, date_to_come timestamp, primary
key(id_specialty, date_to_come),foreign key(id_specialty)
references specialty(id_specialty) on delete cascade on update
cascade,foreign key(tel_client) references client(tel) on update
cascade on delete cascade);
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id_specialty	int(10)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2 tel_client	varchar(20)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	3 register_date	date			Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	4 date_to_come	timestamp			Нет	0000-00-00 00:00:00			Изменить Удалить Ещё

Рис. 3.13 – Вихідна таблиця Запис (register)

Спершу ми вказуємо назву нової таблиці, а потім в дужках перераховуємо всі стовпці, які ми хочемо створити. Важливо, щоб назви стовпців були унікальними всередині цієї таблиці. Після назви кожного стовпця ми вказуємо тип даних, який відповідає цьому полю. Крім того, ми вказуємо, чи дозволено цьому полю мати пусті значення (NULL або NOT NULL), і обираємо поле, яке буде первинним ключем (Primary key) [13].

Мова SQL також дає можливість задати значення за замовчуванням для кожного поля. Іншими словами, якщо користувач не надає значення для певного поля, СУБД автоматично встановлює значення за замовчуванням. Значення за замовчуванням визначається за допомогою ключового слова DEFAULT при визначенні стовпців за допомогою оператора CREATE TABLE.

Зовнішні ключі допомагають встановити зв'язки між таблицями. Вони призначені для стовпців в залежній таблиці, які посилаються на один із стовпців у головній таблиці. Зазвичай зовнішні ключі посилаються на первинний ключ у головній таблиці.

Основний формат встановлення зовнішнього ключа на рівні таблиці виглядає наступним чином:

```
[CONSTRAINT імя_обмеження]
```

```
FOREIGN KEY (стовпець1, стовпець2, ... стовпець N)
```

```
REFERENCES головна_таблиця (стовпець1, стовпець2, ... стовпець N)
```

```
[ON DELETE дія]
```


[ON UPDATE дія]

Для створення обмеження зовнішнього ключа, спочатку вказується стовпець у таблиці, який буде представляти зовнішній ключ. Потім після ключового слова "REFERENCES" вказується ім'я таблиці, з якою встановлюється зв'язок. Після цього, в дужках вказується ім'я стовпця в цій пов'язаній таблиці, на який буде посилатися зовнішній ключ.

Далі визначаються вирази "ON DELETE" і "ON UPDATE", які вказують, що відбудеться при видаленні або оновленні записів в головній таблиці.

У цьому проекті для створення нашої системи всі таблиці були створені вручну, без використання автоматичних методів або інших засобів.

3.4 Програмна реалізація сайту

Розглянемо детально процес програмної реалізації нашого вебсайту для пошуку спеціалістів у сфері краси та здоров'я. Ми розкриємо основні технічні аспекти, використані технології та інструменти, а також розглянемо ключові функції та можливості нашого вебсайту.

3.4.1 Вибір технологічного стеку

Першим етапом у розробці нашого вебсайту був вибір технологічного стеку. Для створення динамічного та інтерактивного вебсайту, який має надавати користувачам зручний інтерфейс для пошуку спеціалістів, було вирішено використовувати такі ключові технології:

1. **Frontend.** Вибір технологічного стеку для фронтенд-розробки - це крок у процесі розробки веб-сайту, який визначає, якими інструментами та технологіями буде будуватися клієнтська частина сайту. Для оптимального вибору технологічного стеку, розглянемо кожну частину, яка входить до нього:

HTML (HyperText Markup Language): HTML є основною мовою для створення структури веб-сторінок. Вона використовується для створення елементів, таких як заголовки, параграфи, таблиці та форми. HTML5 - остання

версія HTML - має багато розширень і покращень, які дозволяють створювати більше інтерактивних та захопливих веб-сторінок.

CSS (Cascading Style Sheets): CSS використовується для визначення зовнішнього вигляду веб-сторінок, таких як кольори, розміри, шрифти і розміщення. Важливо використовувати CSS для забезпечення послідовного та привабливого дизайну.

SCSS (Sass): Sass - це препроцесор CSS, який надає додаткові функції, які полегшують розробку. Він дозволяє використовувати змінні, вкладені стилі, міксини та багато інших корисних можливостей. Використання SCSS допомагає зробити CSS більш читабельним та підтримуваним.

Gulp: Gulp - це інструмент для автоматизації рутинних задач в розробці, таких як компіляція SCSS в CSS, оптимізація зображень та конкатенація файлів. Він допомагає зекономити час та покращити продуктивність розробника.

Webpack: Webpack - це модульний збирач, який допомагає керувати завантаженням ресурсів на сторінку. Він включає в себе можливості збирання та оптимізації JavaScript, CSS, зображень та інших ресурсів. Webpack дозволяє розробникам створювати ефективні та швидкі веб-сторінки.

Отже, вибір технологічного стеку для фронтенд-розробки, який включає HTML, CSS, SCSS, Gulp та Webpack, є раціональним. HTML і CSS є стандартними мовами для створення структури та дизайну веб-сторінок, а використання препроцесора SCSS полегшує розробку та підтримку коду. Gulp і Webpack допомагають автоматизувати рутинні задачі та збірку ресурсів, що зробить розробку більш продуктивною та ефективною.

2. **Backend.** Визначення технологічного стеку для розробки бекенду веб-сайту є ключовим етапом у процесі створення проекту. Для докладного розгляду були відібрані наступні технології:

PHP: PHP - це серверна мова програмування, яка широко використовується для розробки веб-додатків. Вона є однією з найпопулярніших мов для створення бекенду і має багатий екосистему фреймворків, що

полегшують розробку та підтримку проєктів. PHP дозволяє взаємодіяти з базами даних, обробляти запити користувачів і створювати динамічний контент.

Laravel: Laravel - це один з найпопулярніших PHP-фреймворків, призначений для розробки веб-додатків. Він надає розробникам інструменти для створення швидких, надійних та масштабованих додатків. Laravel має вбудовану підтримку маршрутизації, автентифікації, кешування, роботи з базами даних та багато інших корисних функцій.

PHP є однією з найпопулярніших мов програмування для розробки бекенду, і велика спільнота розробників забезпечує підтримку та надійність мови. Laravel надає багато готових інструментів для розробки веб-додатків, що дозволяє швидко створювати робочі рішення з меншими зусиллями. Laravel дозволяє легко масштабувати ваш додаток, що важливо для подальшого розвитку та зростання аудиторії. Laravel має вбудовані засоби для забезпечення безпеки, такі як автентифікація, авторизація, захист від атак на введення даних і т. д. Laravel надає розробникам зручний та ефективний інтерфейс для роботи, а також докладну документацію.

Вибір PHP та Laravel допомагає забезпечити ефективну розробку та високу якість вашого веб-додатку, що є важливим для досягнення успіху в сфері краси та здоров'я.

3. **База даних.** Вибір технологічного стеку для бази даних є ключовим етапом при розробці веб-сайту для сфери краси та здоров'я. У даній роботі розглядається використання MySQL як системи управління базами даних (СУБД). Ось детальний розгляд цього вибору:

MySQL - це одна з найпопулярніших і відкритих СУБД, яка використовується на всьому світі. Вона служить для зберігання та керування даними у веб-додатках. Основні переваги MySQL включають високу швидкість, надійність та широкий функціонал. Деякі з головних переваг MySQL також включають безкоштовність, активну спільноту розробників і підтримку різних операційних систем. Є відкритою системою, що означає, що ви можете

користуватися його базовою версією безкоштовно. Це важливо для стартапів та малих підприємств, оскільки дозволяє зекономити кошти на ліцензіях. MySQL славиться своєю високою продуктивністю та швидкістю виконання запитів до бази даних. Вона дозволяє ефективно обробляти велику кількість даних та багатокористувацьких об'ємів. Існує велика спільнота розробників, яка активно підтримує MySQL. Це означає, що ви зможете знайти багато матеріалів, документацію та форуми для вирішення проблем та отримання порад. Має вбудовані засоби безпеки, такі як можливість налаштування прав доступу користувачів до бази даних, що допомагає захистити ваші дані від несанкціонованого доступу. Оскільки PHP є обраною мовою для розробки веб-сайту, MySQL ідеально підходить для використання разом із PHP, що дозволяє зручно взаємодіяти між серверною та базовою частиною додатку. MySQL добре масштабується і може використовуватися для проєктів різних розмірів, від невеликих сайтів до великих веб-порталів.

4. **Хостинг та деплоймент.** Вибір технологічного стеку для хостингу та деплоювання веб-сайту включає в себе рішення, як DigitalOcean. Далі детально розглянуто вибір цієї технології для веб-проєкту:

DigitalOcean: DigitalOcean - це хмарний хостинг-провайдер, який спеціалізується на віртуальних приватних серверах (VPS). Ця платформа надає розробникам ізольоване середовище, де вони можуть розгорнути свої веб-додатки та бази даних. Основні переваги DigitalOcean включають:

- **Простоту використання:** DigitalOcean має дружній інтерфейс, який дозволяє користувачам без проблем створювати та керувати віртуальними серверами.

- **Гнучкість:** Ви можете вибирати різні конфігурації серверів відповідно до потреб вашого веб-проєкту, змінюючи обсяги CPU, RAM і сховища.

- Швидкість та продуктивність: DigitalOcean надає надзвичайно високу швидкість завдяки SSD-сховищу та швидким мережевим з'єднанням, що робить його ідеальним для веб-сайтів з високою завантаженістю.

- Документація та спільнота: Є велика кількість ресурсів та документації, яка допомагає розробникам з різними рівнями досвіду використовувати DigitalOcean.

DigitalOcean відомий своєю конкурентоспроможною ціною. Це важливо для стартапів та малих підприємств, оскільки дозволяє зекономити кошти на інфраструктурі. Дозволяє швидко масштабувати ваші сервери в залежності від потреб вашого веб-сайту. Є можливість легко додавати або зменшувати потужність серверів. Надає зручні інструменти для керування вашими серверами, включаючи можливість моніторити ресурси та легко налаштовувати засоби безпеки. Надає можливість встановити файрволи, VPN та інші засоби для забезпечення безпеки та приватності даних. Має центри обробки даних в різних частинах світу, що дозволяє вибрати локацію серверів для оптимізації доступності та завантаженості веб-сайту. DigitalOcean дозволяє легко спільно працювати над проєктом, надаючи засоби керування доступом для команди розробників.

Використання DigitalOcean для хостингу та деплоювання вашого веб-сайту забезпечить надійний та продуктивний ресурс для вашого веб-проєкту в сфері краси та здоров'я.

3.4.2 Архітектура вебсайту

Архітектура нашого вебсайту включає два основні компоненти: клієнтську та серверну сторони.

Клієнтська сторона: Вебсайт побудований як статичний додаток (Static App) з використанням технологій HTML, CSS та JavaScript, де для взаємодії з сервером використовується фреймворк jQuery та фреймворк Laravel.

Використання такого підходу дозволило створити інтерфейс, який реагує на дії користувачів без перезавантаження сторінки.

Серверна сторона: Backend нашого вебсайту побудований на мові програмування PHP з використанням фреймворку Laravel. Цей компонент обробляє запити від клієнтів, взаємодіє з базою даних та забезпечує логіку роботи системи.

База даних: Для зберігання даних про спеціалістів та клієнтів використовується система управління базами даних MySQL. Структура бази даних була ретельно спроектована для відображення всіх взаємозв'язків між сутностями та забезпечення консистентності даних.

Автентифікація і авторизація: Вебсайт використовує систему автентифікації та авторизації для користувачів. Кожен користувач має особистий обліковий запис, і для входу в систему потрібно вказати логін та пароль. Після входу користувачам доступні різні функціональні можливості, залежно від їхніх прав.

3.4.3 Ключові функціональності та можливості

Основні ключові функціональності та можливості вебсайту "Система аналізу обробки замовлень в сфері краси та здоров'я" включають:

- **Реєстрація та Вхід:** Користувачі можуть створити облікові записи, вказавши необхідну інформацію. Зареєстровані користувачі можуть увійти в систему за допомогою логіна та пароля.
- **Профілі користувачів:** Кожен користувач має власний профіль, де він може редагувати свої особисті дані. Можливість завантаження та оновлення фото профілю.
- **Пошук спеціалістів:** Користувачі можуть шукати спеціалістів за різними параметрами, такими як місцезнаходження, послуги, рейтинг та інше.
- **Запис на прийом:** Клієнти можуть обирати дату та час для запису до обраного спеціаліста. Можливість перегляду графіку роботи спеціалістів.

- **Керування замовленнями:** Користувачі можуть переглядати та відслідковувати статуси своїх замовлень.
- **Рейтинг та відгуки:** Користувачі можуть залишати відгуки та оцінки для спеціалістів, що збагачує систему інформацією про якість послуг.
- **Адміністративний доступ:** Адміністратори системи мають спеціальний доступ для керування користувачами, спеціалістами та замовленнями. Можливість блокування користувачів або редагування даних.
- **Аналітика та звіти:** Система надає можливість аналізу статистики, включаючи кількість замовлень, популярні послуги, активність користувачів і т.д. Генерація звітів для покращення ефективності бізнесу.
- **Автентифікація та безпека:** Захист даних користувачів за допомогою автентифікації і авторизації. Захист від потенційних загроз безпеки, таких як SQL-ін'єкції та CSRF-атаки.
- **Мобільна сумісність:** Вебсайт має адаптивний дизайн і оптимізований для відображення на мобільних пристроях.
- **Сповіщення:** Система може надсилати сповіщення користувачам щодо підтвердження запису або інших важливих подій.
- **Система підтримки:** Користувачі можуть звертатися до системи підтримки для вирішення питань або проблем.
- **Апаратна інфраструктура:** Система розміщена на серверах хостинг-постачальника Digital Ocean для надійності та продуктивності.

Ця функціональність та можливості забезпечують повноцінне функціонування вебсайту та задовольняють потреби як клієнтів, так і спеціалістів у сфері краси та здоров'я.

3.4.4 Розробка інтерфейсу сайту

Розробка інтерфейсу для дипломної роботи включає в себе створення користувацького досвіду, який є зручним та привабливим для користувачів. Він повинен бути інтуїтивно зрозумілим та легким у використанні. Дизайн інтерфейсу детально вивчається та розробляється, враховуючи функціональність сайту, адаптивність для різних пристроїв та загальну естетику. Крім того, інтерфейс містить:

- Меню та навігацію для легкого переміщення по сторінках.
- Форми для реєстрації, входу та редагування профілю користувачів.
- Відображення інформації про спеціалістів та їх послуг, включаючи відгуки та рейтинг.
- Особисті кабінети для клієнтів та спеціалістів з можливістю керування замовленнями та додавання фото профілю.
- Можливість залишати відгуки та оцінки, які відображаються на профілях спеціалістів.
- Систему підтримки для звернень та надсилання запитів.
- Функціональність для введення логіну та пароля при вході в систему, забезпечуючи безпеку користувачів.

Увесь інтерфейс ретельно розробляється та тестується, забезпечуючи зручний та ефективний користувацький досвід для клієнтів та спеціалістів, які відвідують веб-сайт.

3.4.5 Безпека і захист даних

Один з пріоритетів у розробці вебсайту була безпека даних. Ми впровадили ряд заходів для захисту конфіденційної інформації користувачів, включаючи шифрування паролів, обмеження доступу до адміністративних функцій, регулярну перевірку на наявність вразливостей та вдосконалення системи автентифікації.

3.4.6 Тестування і відладка

Перед впровадженням вебсайту в експлуатацію, ми провели широкомасштабне тестування для виявлення і усунення помилок. Тестування включало в себе функціональне тестування, тестування користувацького інтерфейсу, а також навантажувальне тестування для оцінки продуктивності та стабільності вебсайту.

3.4.7 Деплоймент і публікація

Після успішного завершення розробки та тестування вебсайту, ми розгорнули його на сервері Digital Ocean. Зараз вебсайт доступний для користувачів через інтернет, і ми проводимо постійний моніторинг та підтримку для забезпечення безперебійної роботи.

Висновок до розділу 3

Розділ 3 нашої роботи був присвячений розробці бази даних і вебсайту для системи аналізу обробки замовлень в галузі краси та здоров'я. Протягом цього етапу було вирішено низку важливих завдань та досягнуто ключових результатів.

У контексті розробки бази даних, була створена структура бази даних на основі системи управління базами даних MySQL. База даних була спроектована так, щоб відображати всі необхідні сутності та їх взаємозв'язки. Це забезпечує консистентність даних і дозволяє зберігати та взаємодіяти з інформацією ефективно.

Щодо розробки вебсайту, було використано сучасні технології, такі як HTML, CSS, JavaScript, і фреймворк Laravel для створення клієнтської та серверної сторін веб-додатку. Це дозволило створити динамічний інтерфейс та ефективно обробляти дані та замовлення.

На цьому етапі розробки вдалося створити інтерактивний вебсайт, який сприяє зручній обробці замовлень та забезпечує користувачів зручним інтерфейсом для оформлення замовлень і спілкування з фахівцями.

Отже, розділ 3 демонструє успішну розробку бази даних та вебсайту для системи аналізу обробки замовлень в галузі краси та здоров'я. Ці результати є важливими для подальшого розвитку системи та покращення процесів у цій галузі.

4 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ АПРОБАЦІЇ РОЗРОБЛЕНОГО ВЕБСАЙТУ

4.1 Процес тестування вебсайту

Розділ "Тестування розробленого веб-сайту" є критичним у процесі розробки імплементованого веб-проекту. В цьому розділі проводиться докладне тестування всіх компонентів та функціональностей веб-сайту, щоб переконатися в його надійності, ефективності та зручності в користуванні. Процес тестування включає в себе кілька етапів, які описуються далі.

Функціональне тестування: Перший крок - це перевірка основних функціональностей веб-сайту. Кожна функція, розроблена на стороні клієнта і сервера, повинна бути протестована на коректність роботи. Це включає в себе перевірку реєстрації та входу користувачів, пошуку та перегляду інформації, роботу з формами та взаємодію з базою даних. Окрема увага приділяється обробці помилок і валідації даних.

Тестування взаємодії інтерфейсу користувача: У цьому етапі перевіряється взаємодія користувача з інтерфейсом веб-сайту. Це охоплює перевірку правильності роботи всіх кнопок, посилань, меню та інших елементів інтерфейсу. Важливо впевнитися, що користувач може легко навігувати сайтом та виконувати всі дії без труднощів.

Тестування сумісності та адаптивності: Веб-сайт повинен бути доступним для користувачів на різних пристроях та веб-браузерах. Тому важливо перевірити, як сайт відображається на різних роздільностях екрану та в різних браузерах. Також слід переконатися, що сайт працює коректно на мобільних пристроях та планшетах.

Тестування продуктивності: Важливо оцінити продуктивність веб-сайту, перевірити час завантаження сторінок і відгук сервера на запити. Може бути проведено тестування з великою кількістю одночасних користувачів, щоб визначити, як сайт веде себе під навантаженням.

Тестування безпеки: Веб-сайт повинен бути захищеним від потенційних загроз безпеці, таких як атаки SQL-ін'єкції, перехоплення сесій, атаки на переповнення буфера тощо. Проводиться аудит безпеки та перевірка на вразливості.

Тестування на завади та відновлення: В разі непередбачених помилок або аварій, важливо переконатися, що веб-сайт може автоматично відновити роботу та зберегти дані. Проводяться тести на відновлення системи в разі непередбачуваних ситуацій.

Тестування сумісності з базою даних: Перевірка на відповідність даних в базі даних даним на веб-сайті, тестування запитів до бази даних та забезпечення цілісності даних.

Тестування відповідності вимогам та користувацькому досвіду: Перевірка, чи веб-сайт відповідає вимогам, визначеним на початку проекту. Також важливо перевірити користувацький досвід та впевнитися, що веб-сайт надає користувачам те, що вони очікують.

Завершальним етапом тестування є підготовка звіту з результатами та виправлення виявлених помилок та недоліків. Після успішного завершення тестування веб-сайт готовий до релізу і використання користувачами.

4.2 Робота з основною інформацією сайту

Основна інформація на веб-сайті відіграє важливу роль у забезпеченні користувачів необхідною та корисною інформацією. Ця інформація може бути різноманітною в залежності від типу сайту та його цілей.

Щоб побачити всю інформацію на сайті переходимо на головну сторінку сайту (Рис. 4.1)

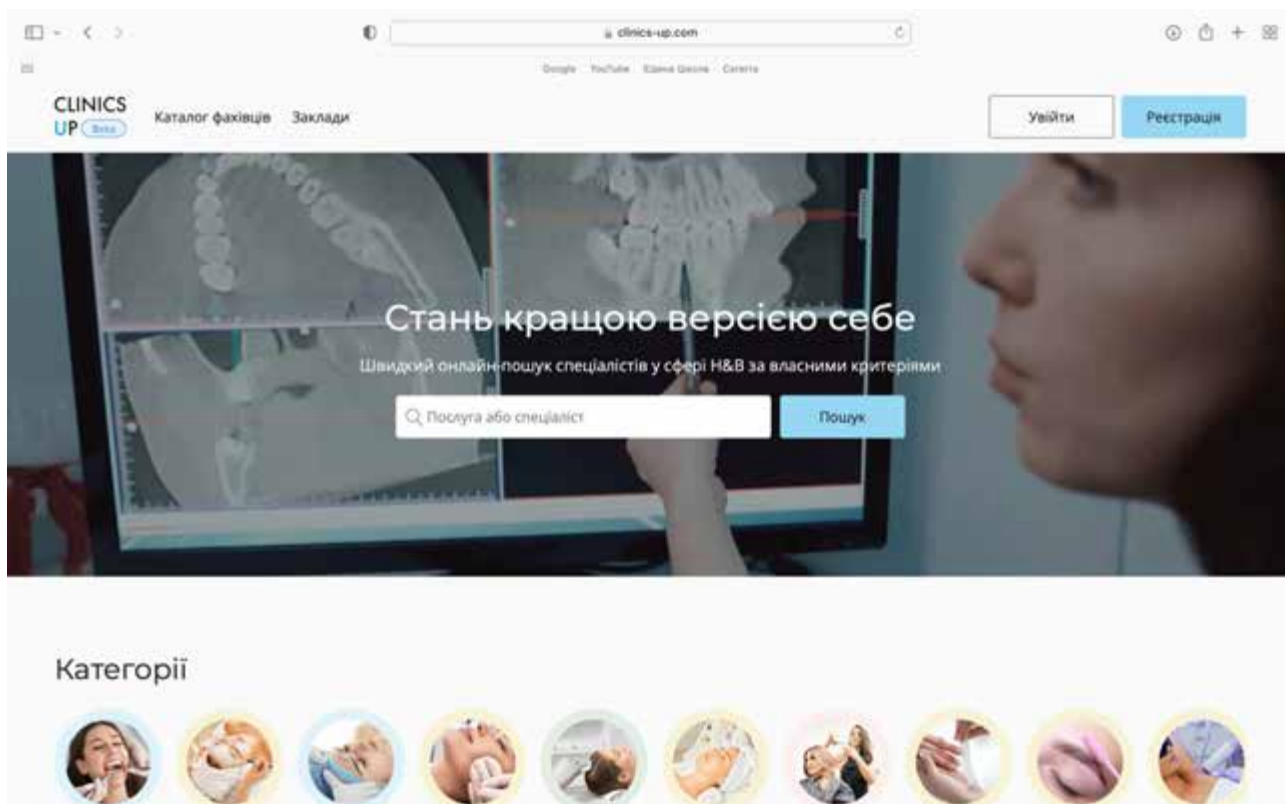


Рис. 4.1 – Головна сторінка сайту «ClinicsUp»

Далі переходимо на сторінку реєстрації та обираємо тип реєстрації: «Я клієнт», «Я спеціаліст», «Заклад» (Рис.4.2). Розглянемо реєстрацію для клієнта:

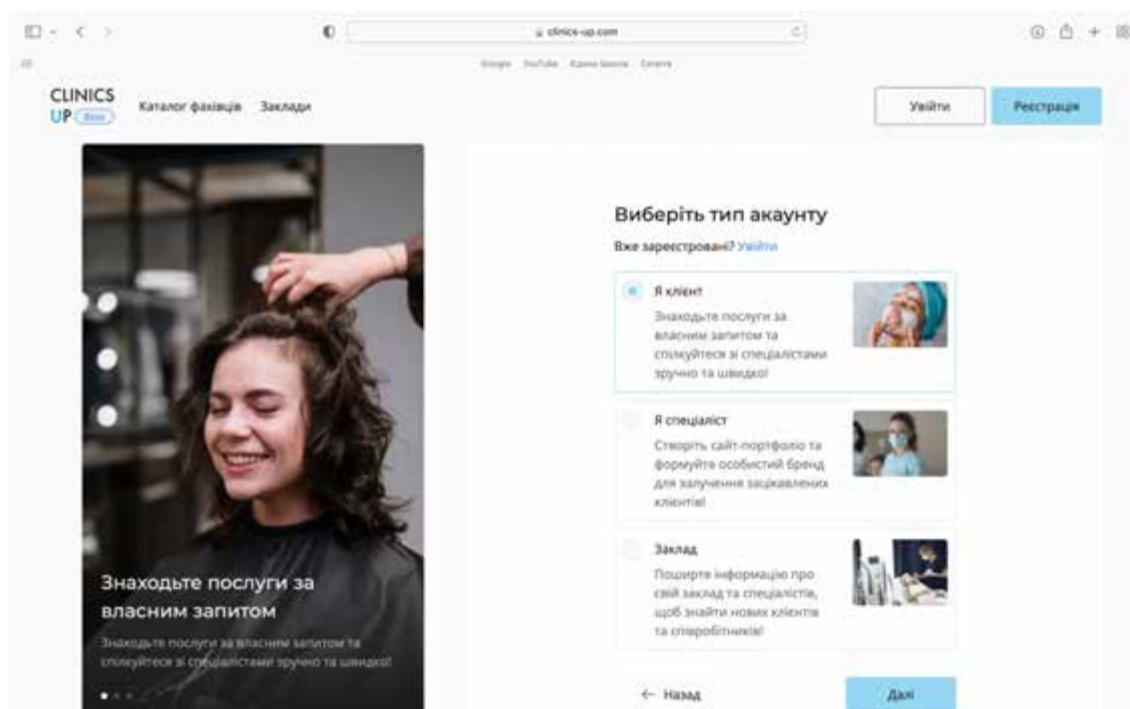


Рис. 4.2 – Типи реєстрації

4.2.1 Вхід на сайт як клієнт

Розглянемо реєстрацію та функції на сайті для клієнта.

1. Обираємо тип «Я клієнт» та натискаємо «Далі» і нас перекидає на сторінку форми (Рис.4.3)

The screenshot shows a web browser window with the URL 'clincs-up.com'. The page features a navigation bar with 'CLINICS UP' and 'Каталог фахівців' and 'Заклади'. A 'Увійти' button is in the top right. The main content area is a registration form titled 'Реєстрація' with the subtext 'Вже зареєстровані? Увійти'. It includes a 'Sign in with Google' button, followed by input fields for 'Ім'я', 'Прізвище', 'Номер телефону' (with a '+380' prefix), and 'Email'. A 'Далі' button is at the bottom right, and a 'Назад' button is at the bottom left. A banner on the left side of the page reads 'Знаходьте послуги за власним запитом'.

Рис. 4.3 – Сторінка форми

2. Після заповнення форми з інформацією про клієнта натискаємо «Далі» і відкривається вікно введення паролю (Рис. 4.4).

This screenshot shows the same registration form as in Figure 4.3, but it has progressed to the password creation step. The 'Ім'я', 'Прізвище', 'Номер телефону', and 'Email' fields are now filled. The 'Далі' button is highlighted in blue. Below the 'Email' field, there is a line of text: 'Створюючи профіль, ви погоджуєтесь з Умовами використання'. The 'Пароль' and 'Повторіть пароль' fields are now the primary focus, with 'Далі' button at the bottom right.

Рис. 4.4 – Вікно створення паролю

3. Коли пароль створено, то нас автоматично перекидає на сторінку кабінету, де ми можемо знайти спеціаліста чи заклад декілька кліків (Рис. 4.5)

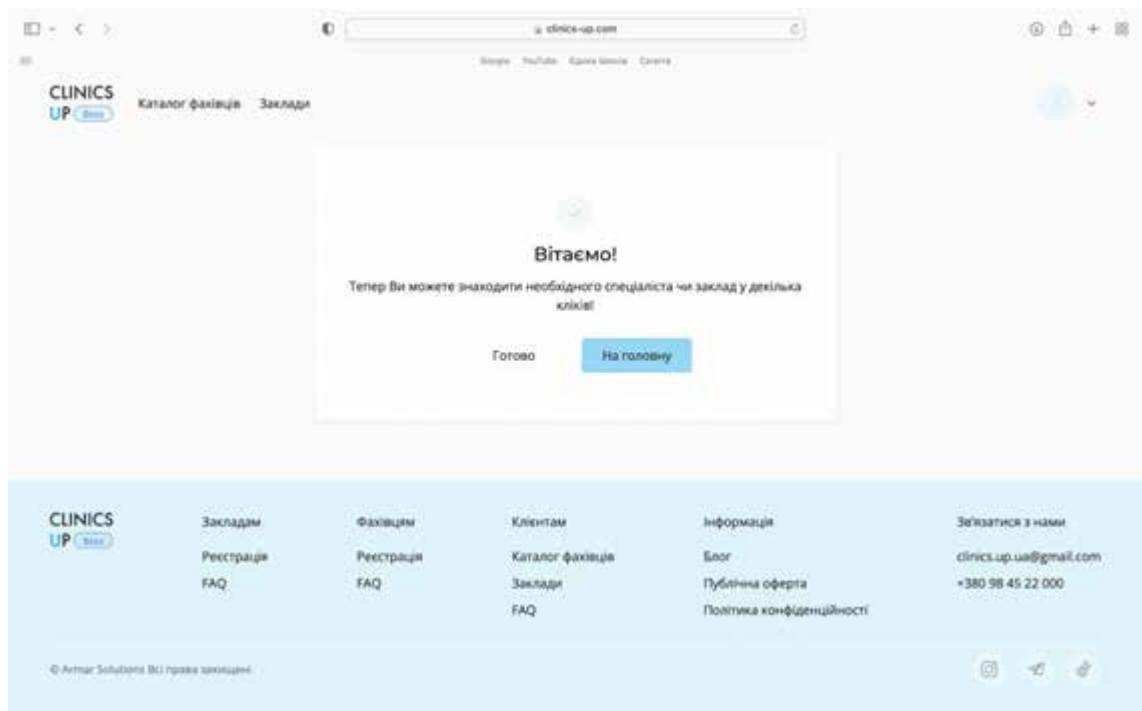


Рис. 4.5 – Особистий кабінет клієнта

4. Перейшовши на «Каталог фахівців» ми можемо обрати потрібного фахівця для себе за різними критеріями (Рис. 4.6)

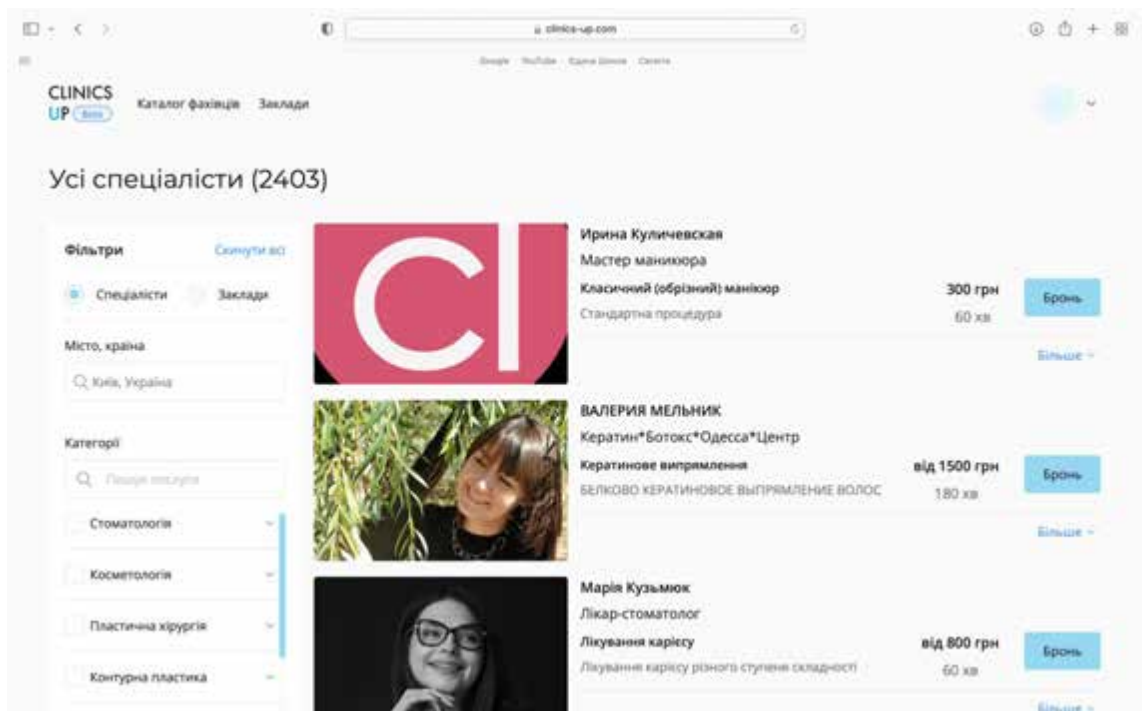


Рис. 4.6 – Каталог фахівців

5. Обираємо спеціаліста та можемо забронювати необхідну послугу на певний день та час (Рис. 4.7)

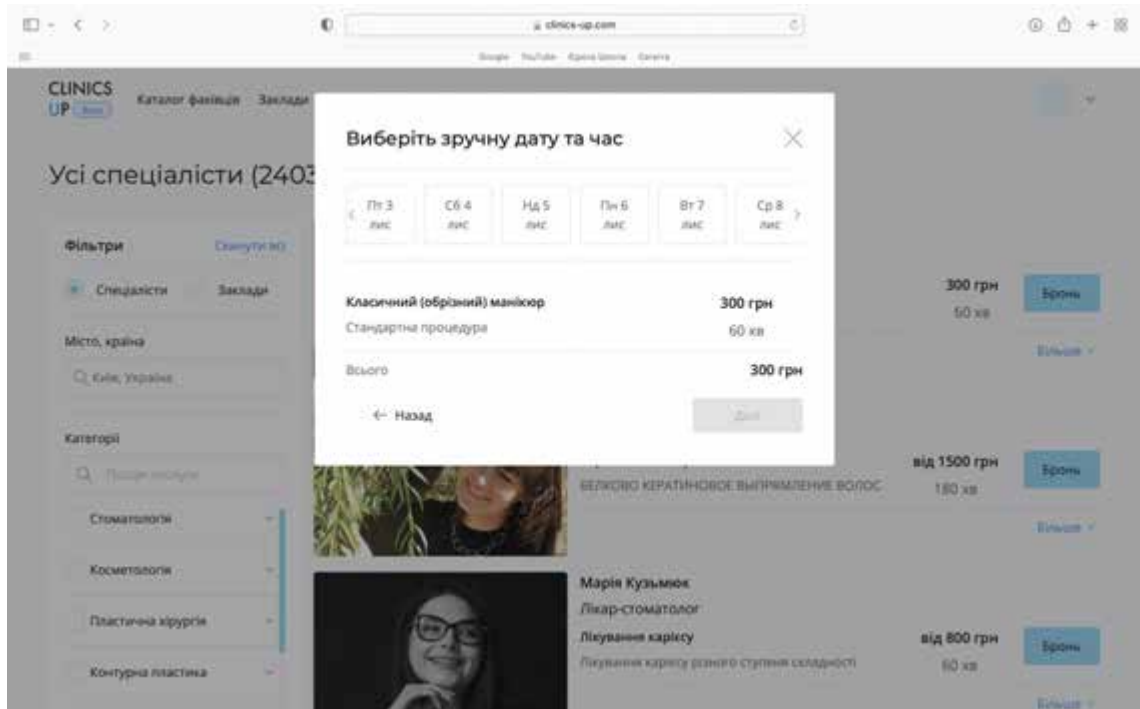


Рис. 4.7 – Вікно запису клієнта до спеціаліста

Підтвердивши бронювання на пошту нам приходить лист з усією інформацією про запис до даного спеціаліста.

Реєстрація для спеціаліста та закладу полягає в тому ж алгоритмі, що й реєстрація клієнта, саме з цієї причини ми не будемо детально розглядати ці кроки реєстрації для спеціаліста та закладу.

Якщо користувач авторизован в системі, то він може натиснути «Увійти» та ввести свої дані, що перейти до сторінки бронювання послуг та перегляду сайту (Рис. 4.8)

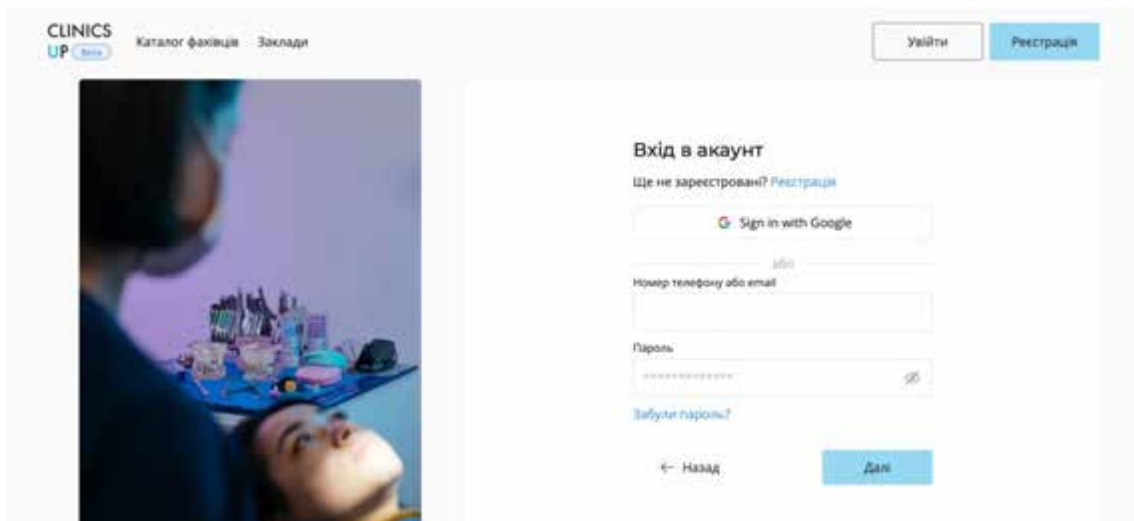


Рис. 4.8 – Вікно входу в акаунт

4.2.2 Вхід на сайт як заклад

Коли користувач, який представляє "Заклад", заходить на сайт, він має доступ до ряду можливостей та функціональностей, призначених саме для закладів. Ось деякі з них:

1. **Реєстрація (Рис. 4.9)/Вхід в особистий кабінет (Рис. 4.10):** Заклад може мати окремий обліковий запис на сайті, який дозволяє зареєструватися або увійти в особистий кабінет. Це дає змогу керувати інформацією про свій заклад.

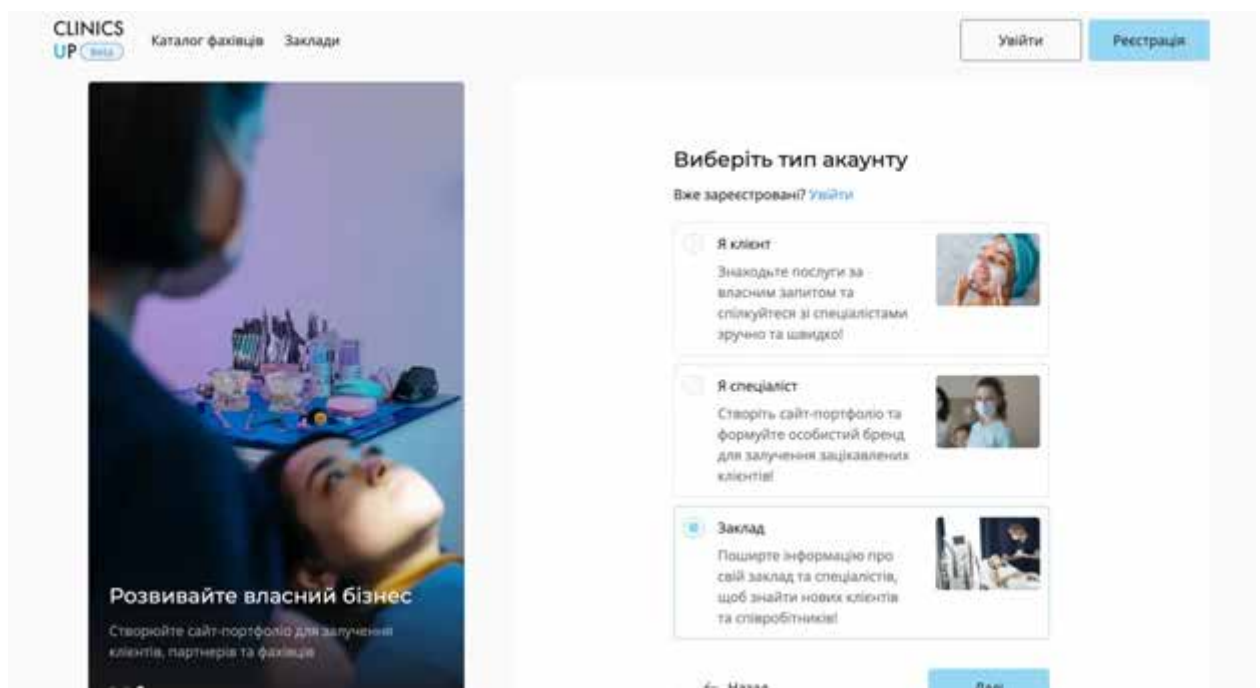


Рис. 4.9 – Вікно реєстрації для закладу

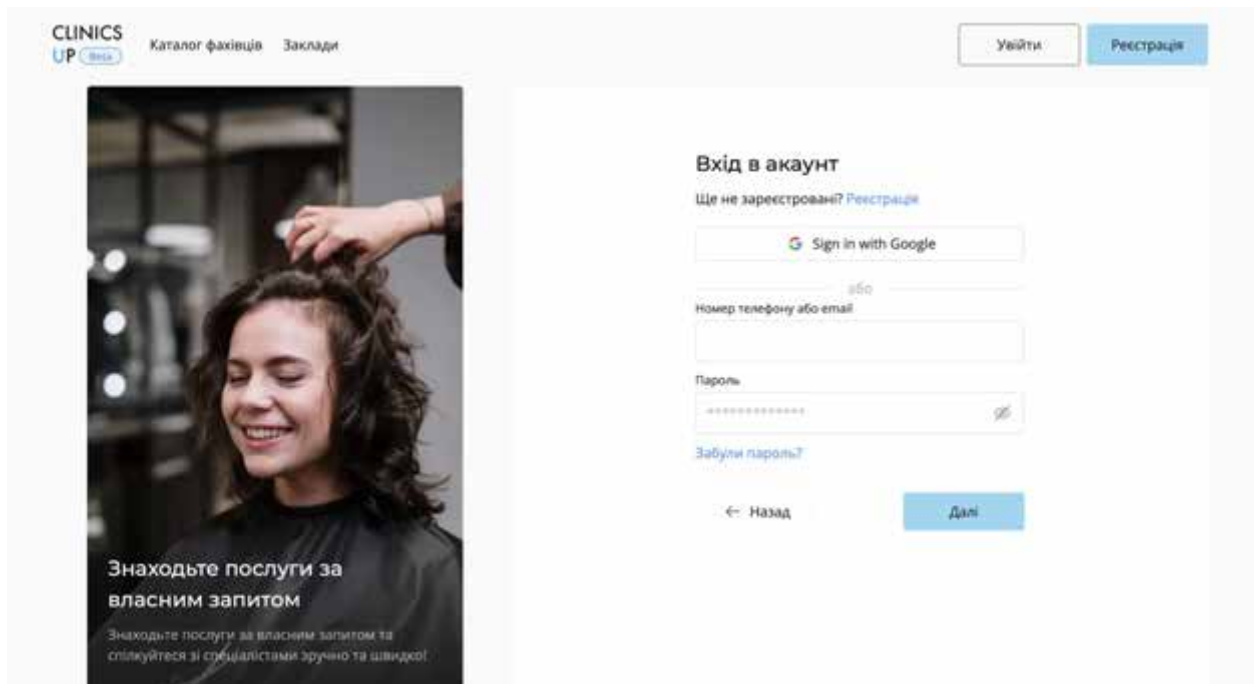


Рис. 4.10 – Вікно входу в особистий кабінет

Редагування інформації про заклад (Рис.4.11): Після входу в особистий кабінет, представник закладу може редагувати та оновлювати інформацію про заклад, таку як контактні дані, режим роботи, адресу, фотографії тощо.

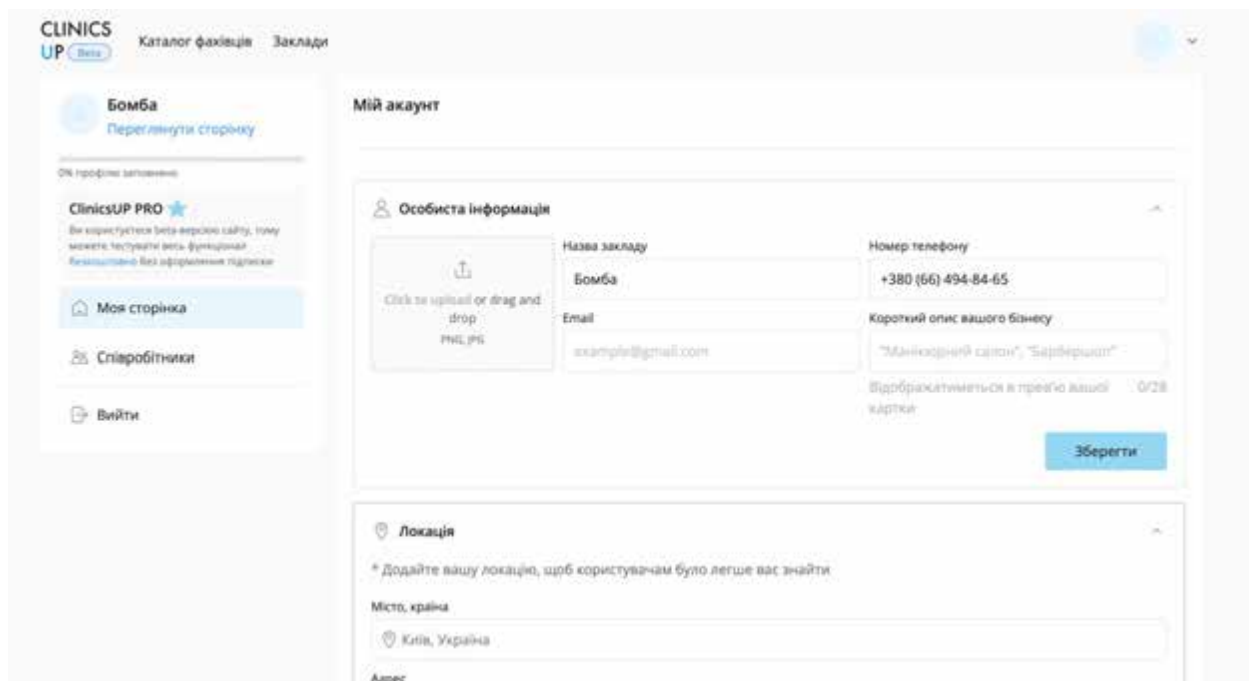


Рис. 4.11 – Редагування інформації про заклад

Додавання послуг та спеціалістів (Рис.4.12): Заклад може додавати інформацію про надані послуги, персонал (спеціалістів), їхні кваліфікації та графіки роботи.

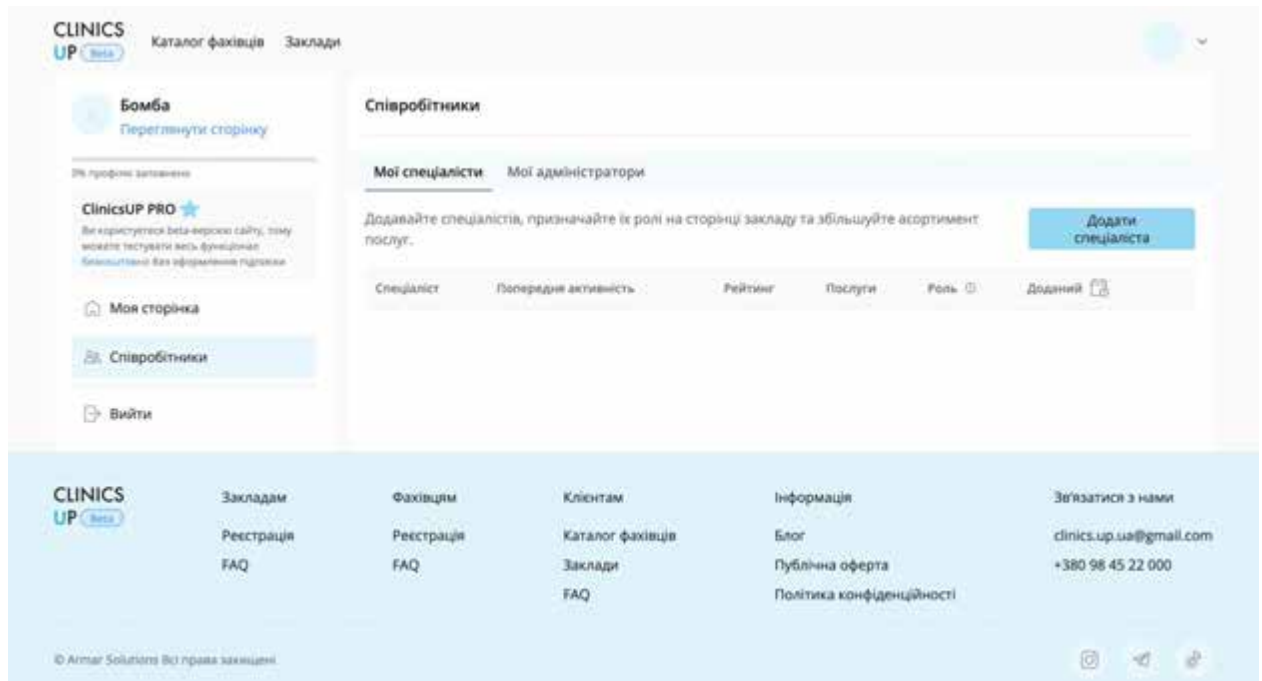


Рис. 4.12 – Додавання послуг та спеціалістів

Запис клієнтів (Рис. 4.13): Заклад може приймати записи від клієнтів на свої послуги. Це може бути реалізовано через інтерактивний графік, де клієнти можуть обирати доступні години та спеціалістів.

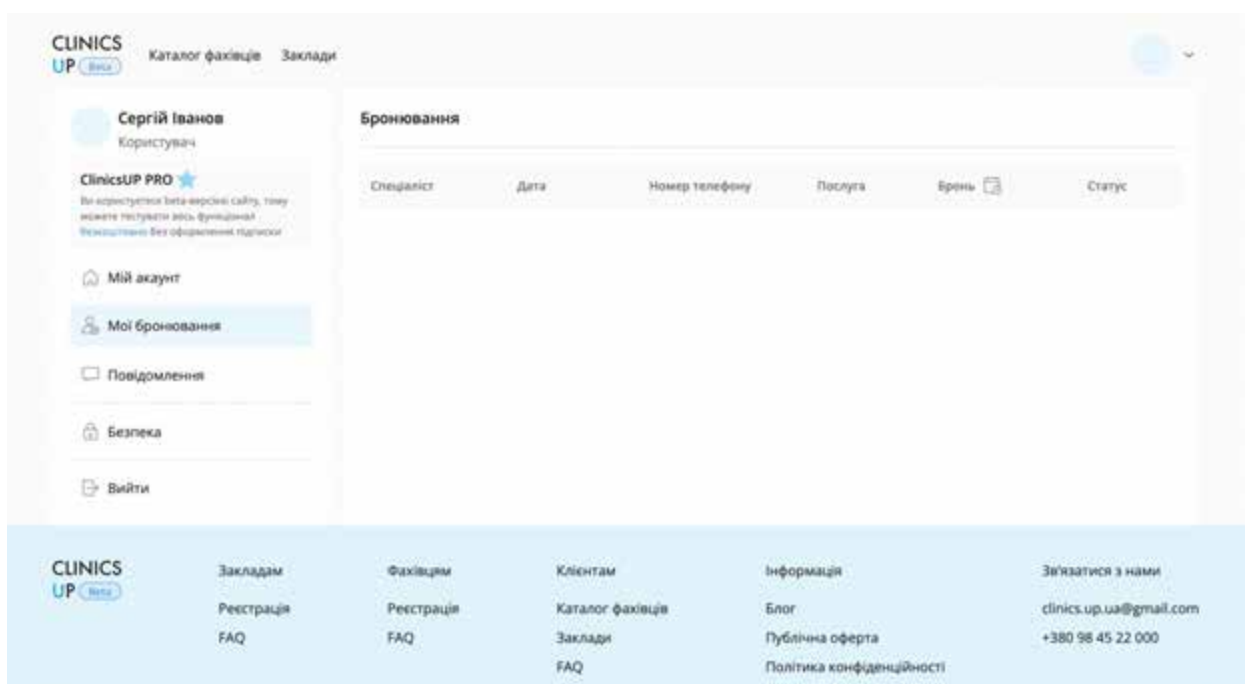


Рис. 4.13 – Запис клієнтів

Відгуки та рейтинг (Рис. 4.14): Заклад може переглядати відгуки від клієнтів та відповідати на них. Високий рейтинг та позитивні відгуки сприяють покращенню репутації закладу.

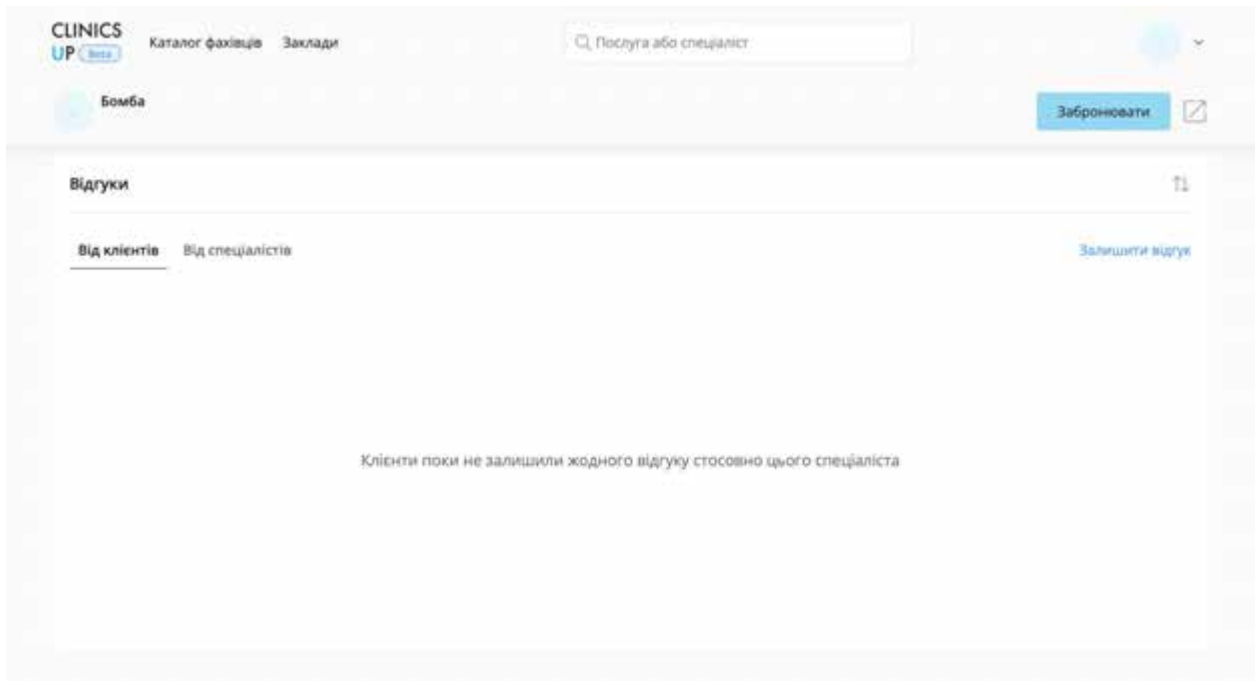


Рис. 4.14 – Відгуки та рейтинг

Адміністрування облікових записів співробітників (Рис. 4.15): На сайті є реєстрація співробітників, представник закладу може керувати доступом до акаунтів свого персоналу та надавати їм відповідні ролі.

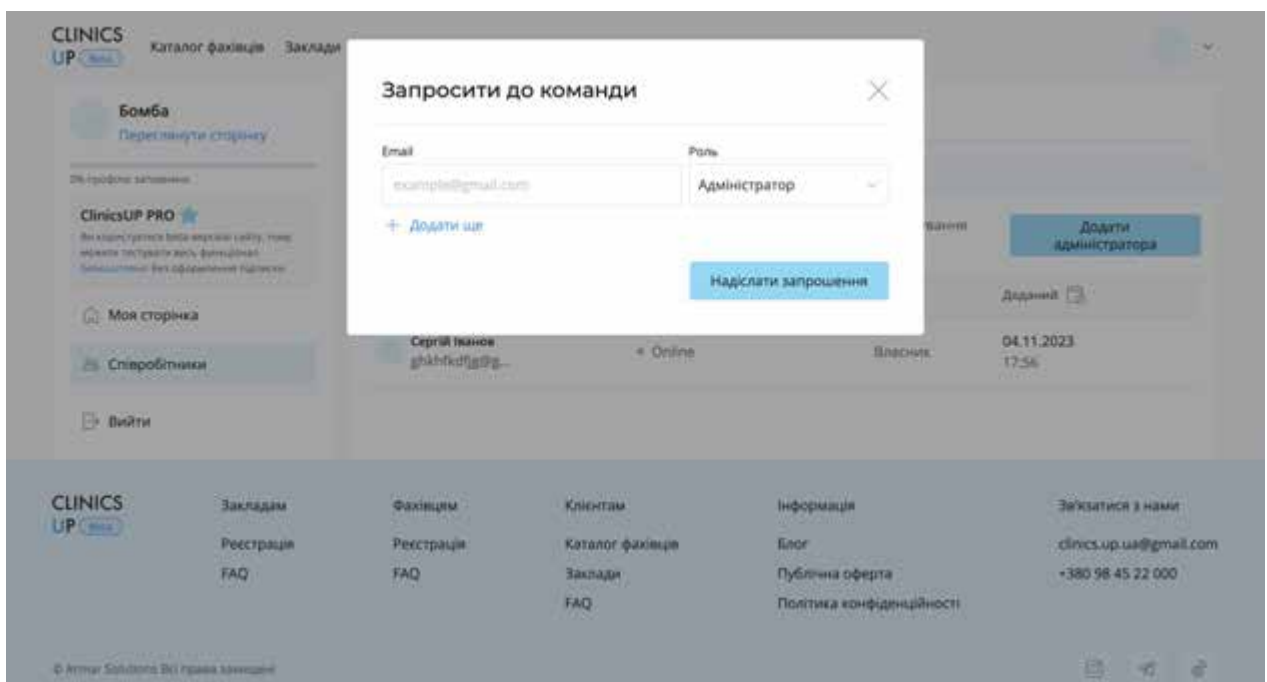


Рис. 4.15 – Адміністрування облікових записів співробітників

Сповіщення і листи (Рис. 4.16): Заклад може отримувати сповіщення та листи від клієнтів, а також розсилати інформаційні повідомлення про акції, події та оновлення.

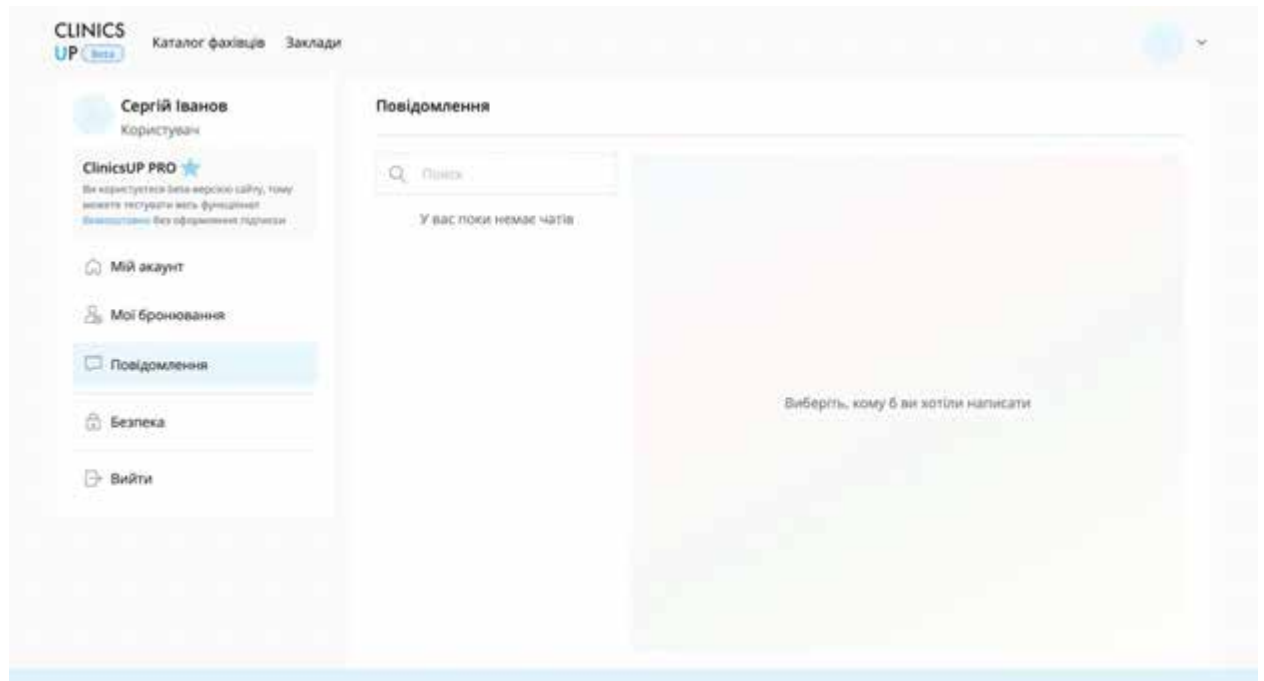


Рис. 4.16 – Сповіщення та листи

Це загальні можливості для користувача типу "Заклад".

4.2.3 Вхід на сайт як спеціаліст

Користувач, який заходить на сайт як "Спеціаліст" має наступні можливості:

Авторизація (Рис. 4.17): Спеціаліст може увійти в систему, використовуючи свій особистий логін та пароль. Це дозволяє отримати доступ до особистого облікового запису.

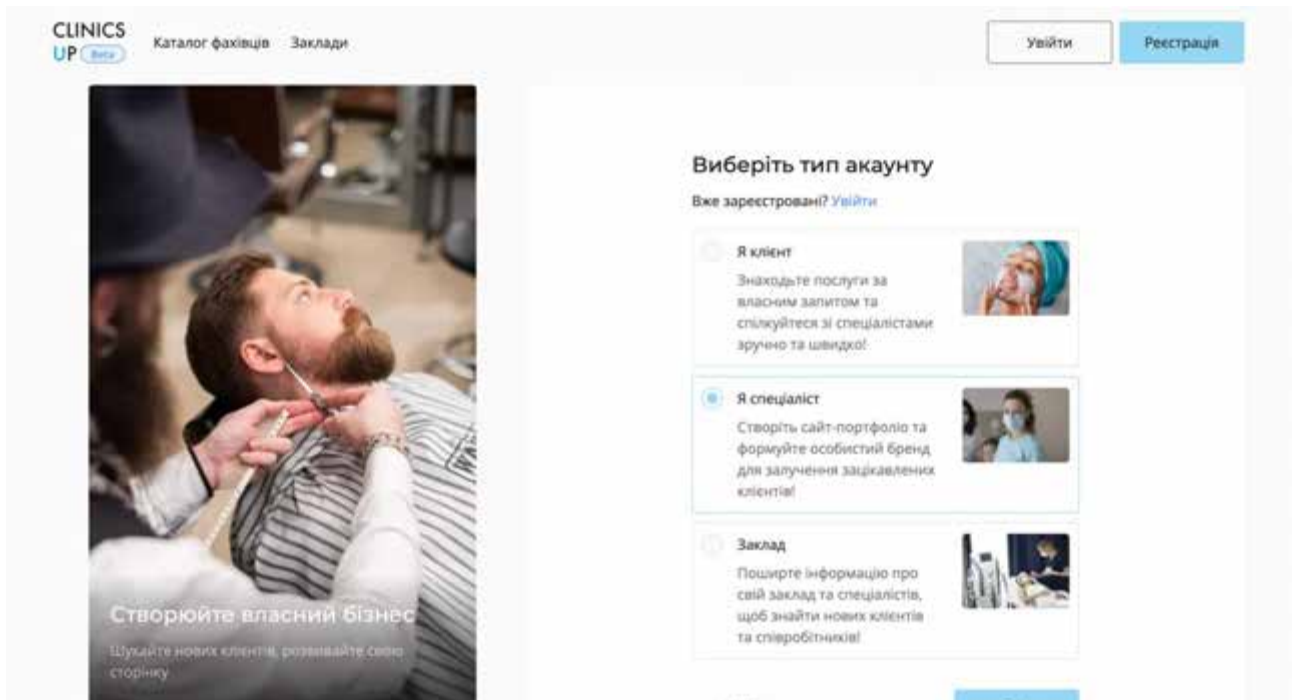


Рис. 4.17 – Авторизація спеціаліста

Профіль спеціаліста (Рис. 4.18): Після входу в систему, спеціаліст може переглядати та редагувати свій особистий профіль. Це включає в себе додавання або зміну особистих даних, фотографій, інформації про послуги, робочих годин, та інше.

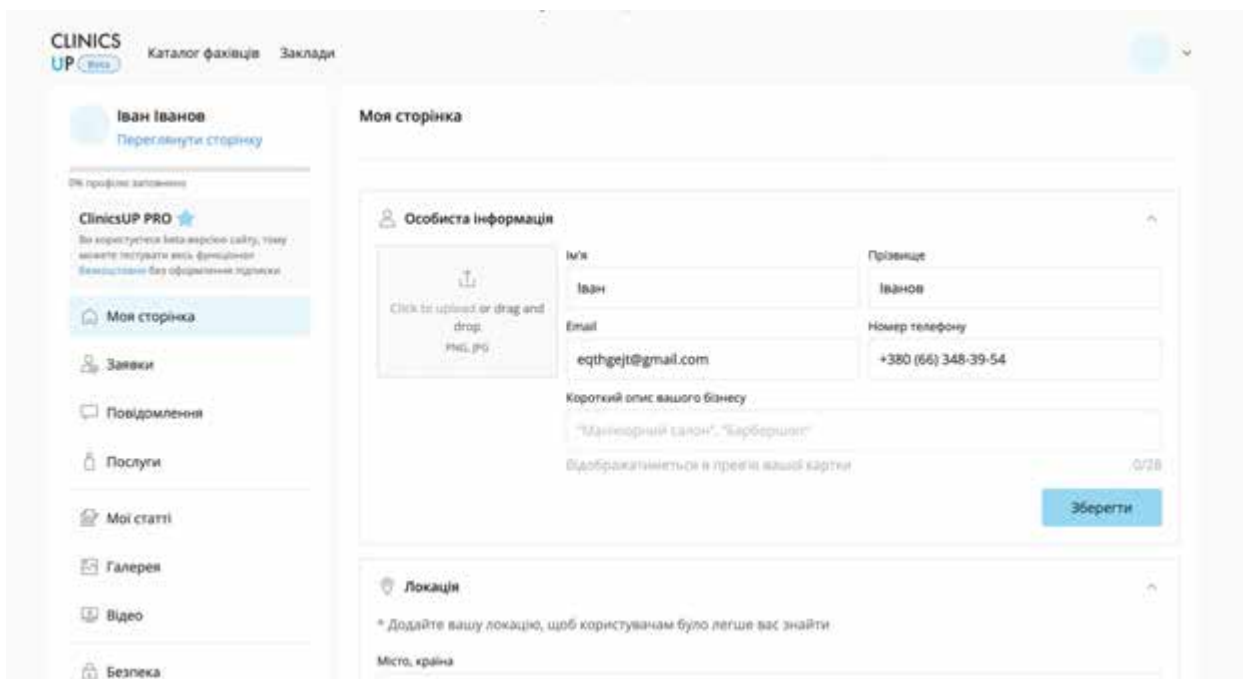


Рис. 4.18 – Профіль спеціаліста

Керування замовленнями (Рис. 4.19): Спеціаліст може переглядати замовлення, які надійшли від клієнтів. Він може приймати або відхиляти замовлення, а також встановлювати дати та часи прийому клієнтів.

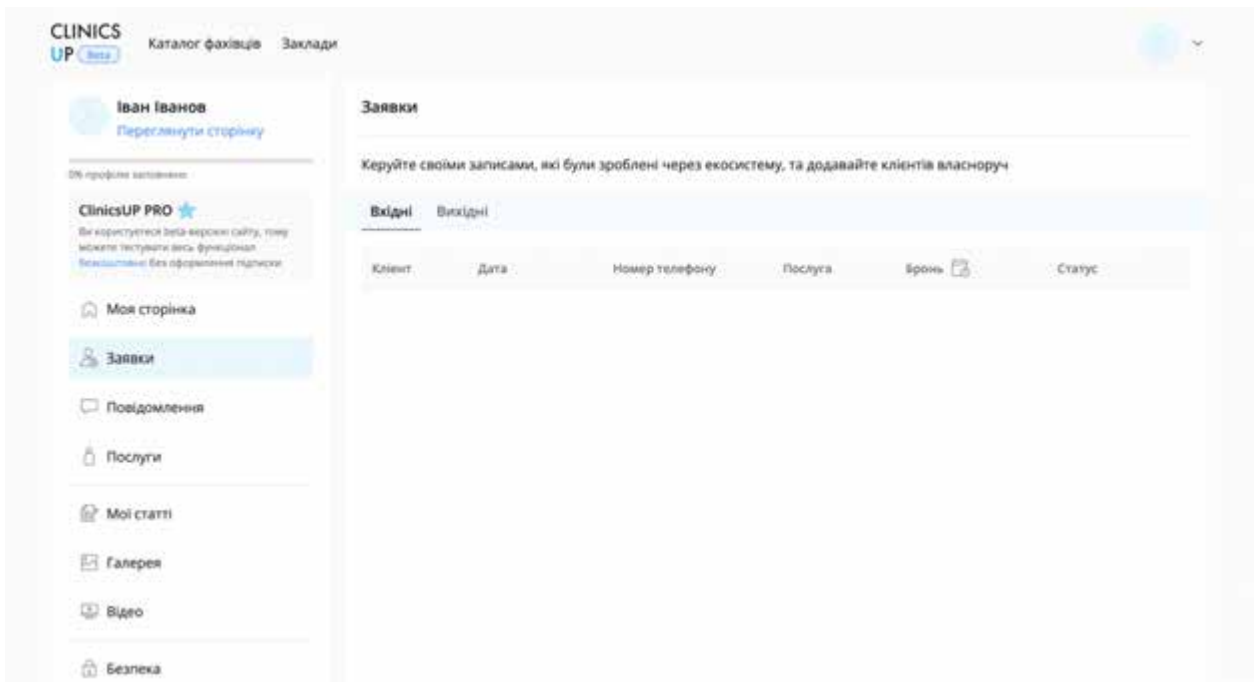


Рис. 4.19 – Керування замовленнями

Комунікація з клієнтами (Рис. 4.20): Спеціаліст може спілкуватися з клієнтами через систему, надсилаючи повідомлення та інформацію про послуги.

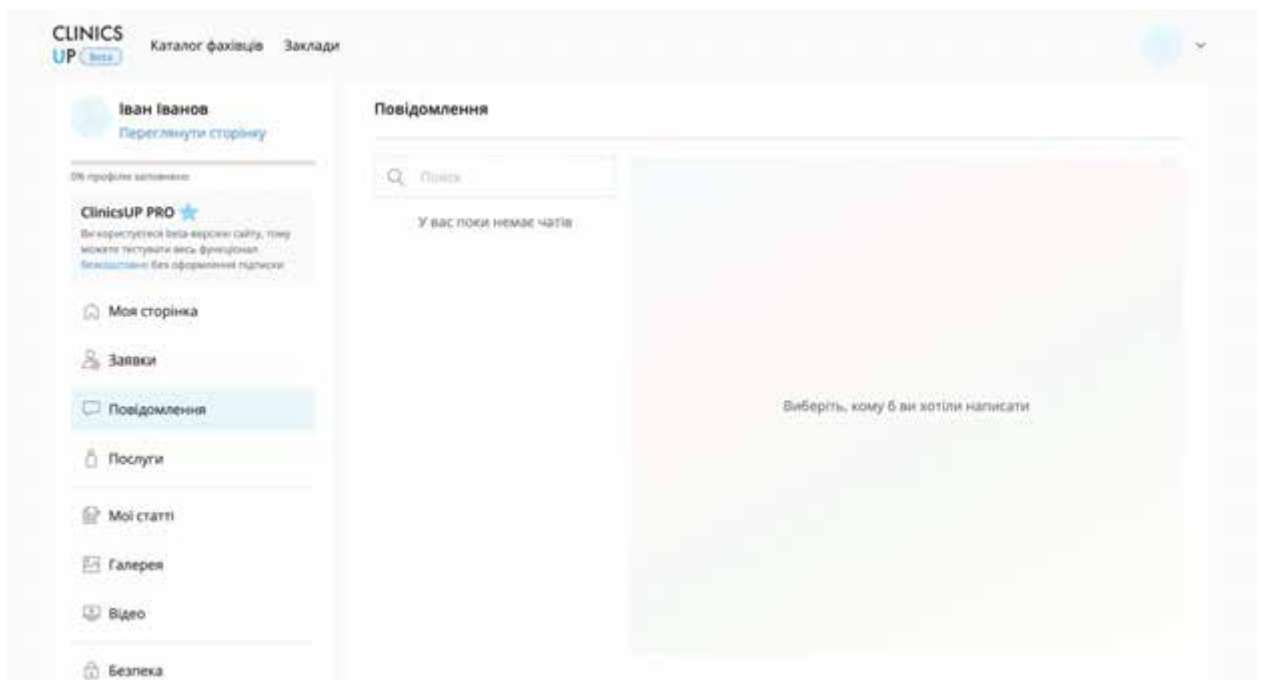


Рис. 4.20 – Комунікація з клієнтами

Послуги (Рис. 4.21): Користувач-спеціаліст може переглядати та редагувати інформацію про послуги, які він надає. Він може додавати нові послуги, вказуючи назву, опис, ціну та інші важливі атрибути. Також, користувач може оновлювати та видаляти існуючі послуги за необхідності.

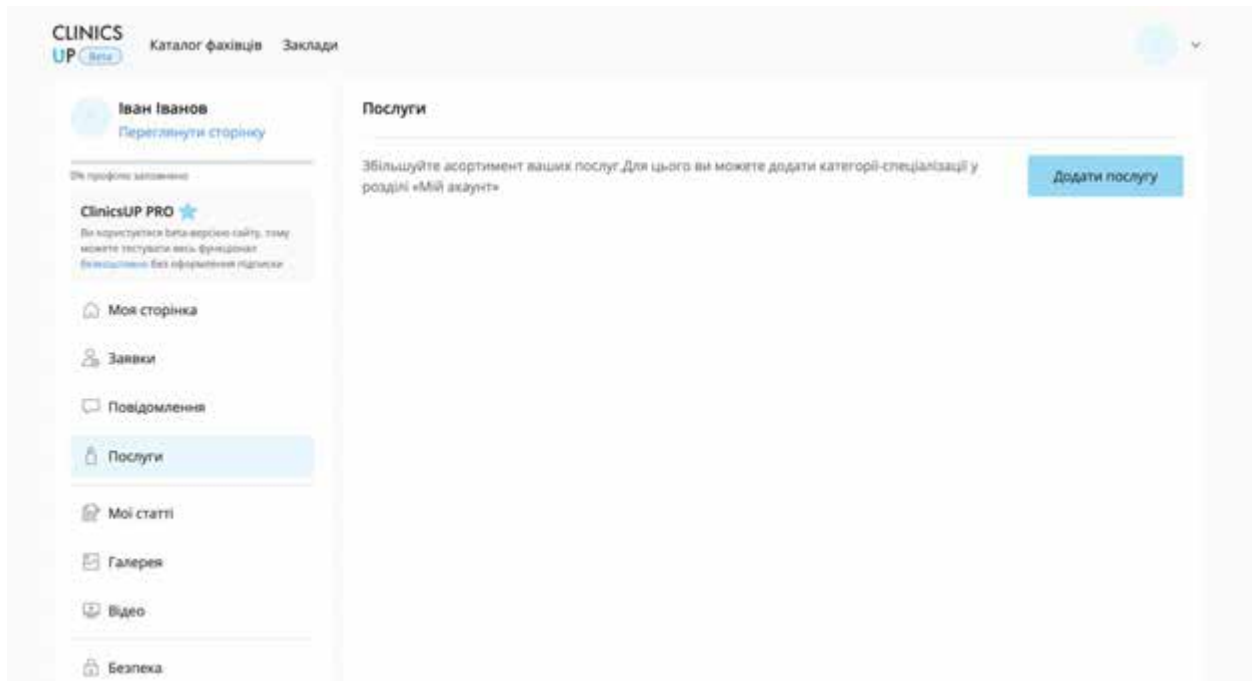


Рис. 4.21 – Послуги

Мої статті (Рис. 4.22): Користувач може створювати та редагувати статті, пов'язані з його сферою діяльності. Кожна стаття має назву, опис, текст статті та зображення.

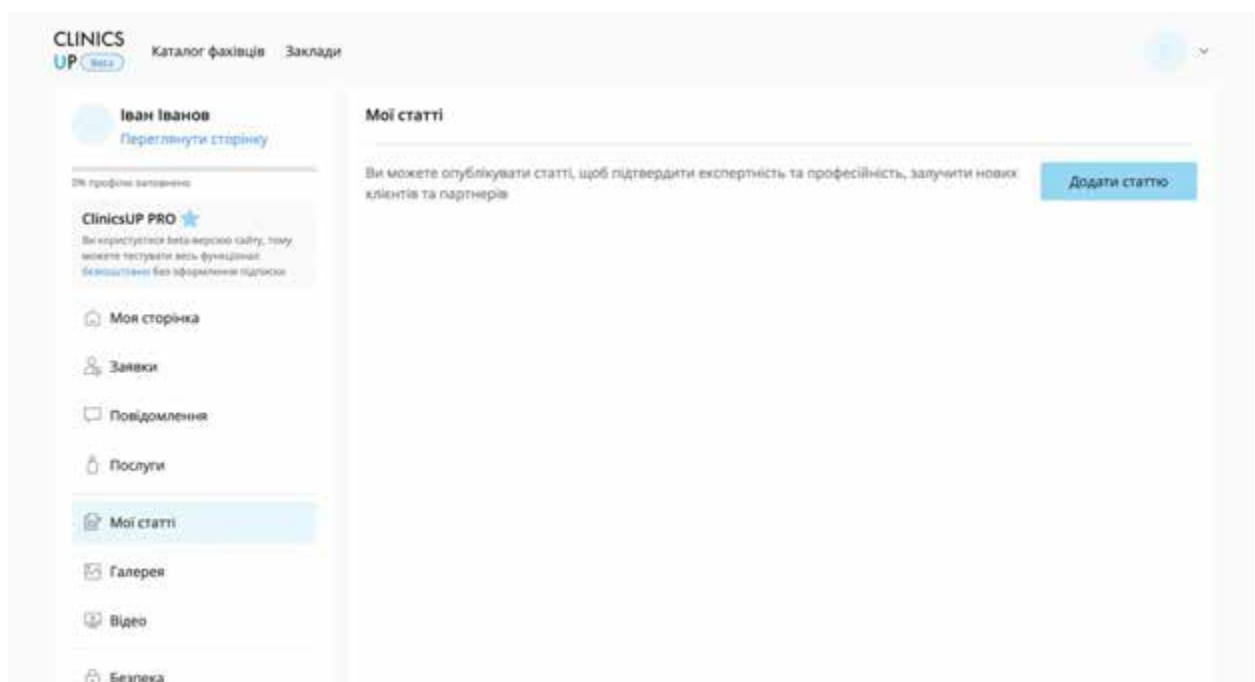


Рис. 4.22 – Мої статті

Галерея (Рис. 4.23): Спеціаліст може завантажувати свої фотографії в галерею. Це може бути зображення робіт, результати професійної діяльності, фотографії з подій або будь-які інші зображення, пов'язані зі своєю роботою. Користувач може додавати опис до кожного зображення та редагувати їхні дані.

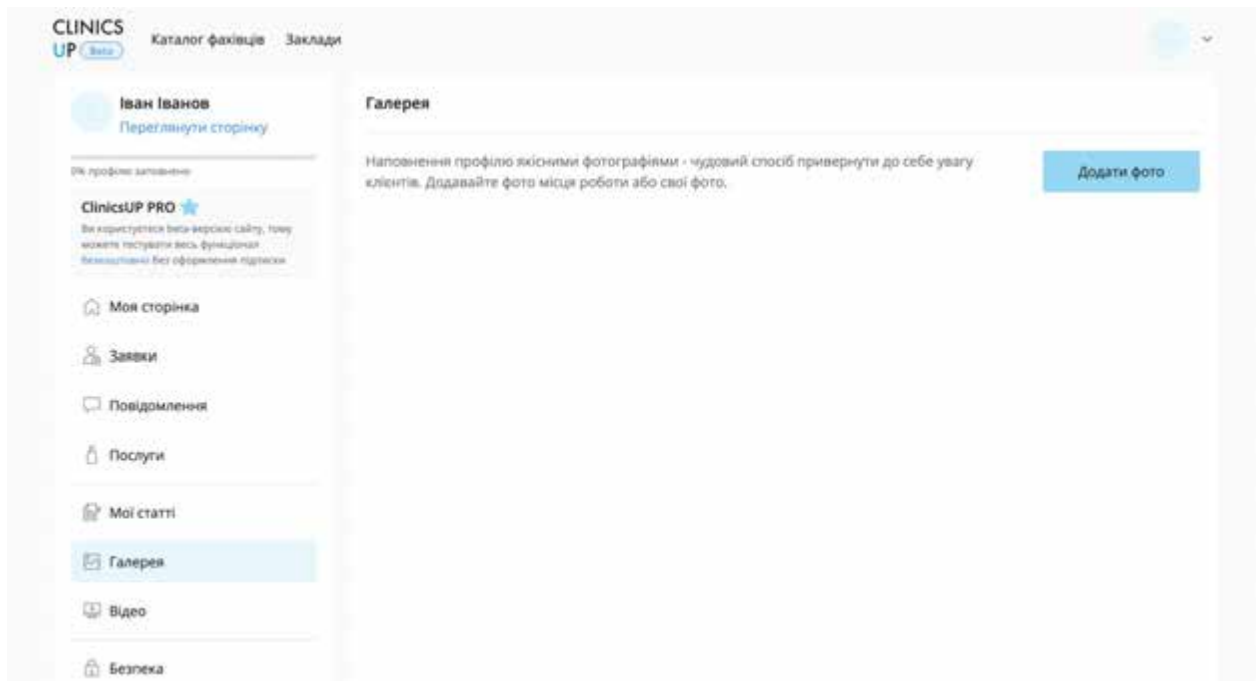


Рис. 4.23 – Галерея

Відео (Рис. 4.24): Користувач-спеціаліст може додавати відео, пов'язані зі своєю діяльністю. Він може вкладати посилання на відеоролики з популярних відеохостингів, наприклад, YouTube. Ці відео можуть демонструвати роботу спеціаліста, інструкції, огляди продуктів або будь-який інший контент, який може бути корисним для його клієнтів.

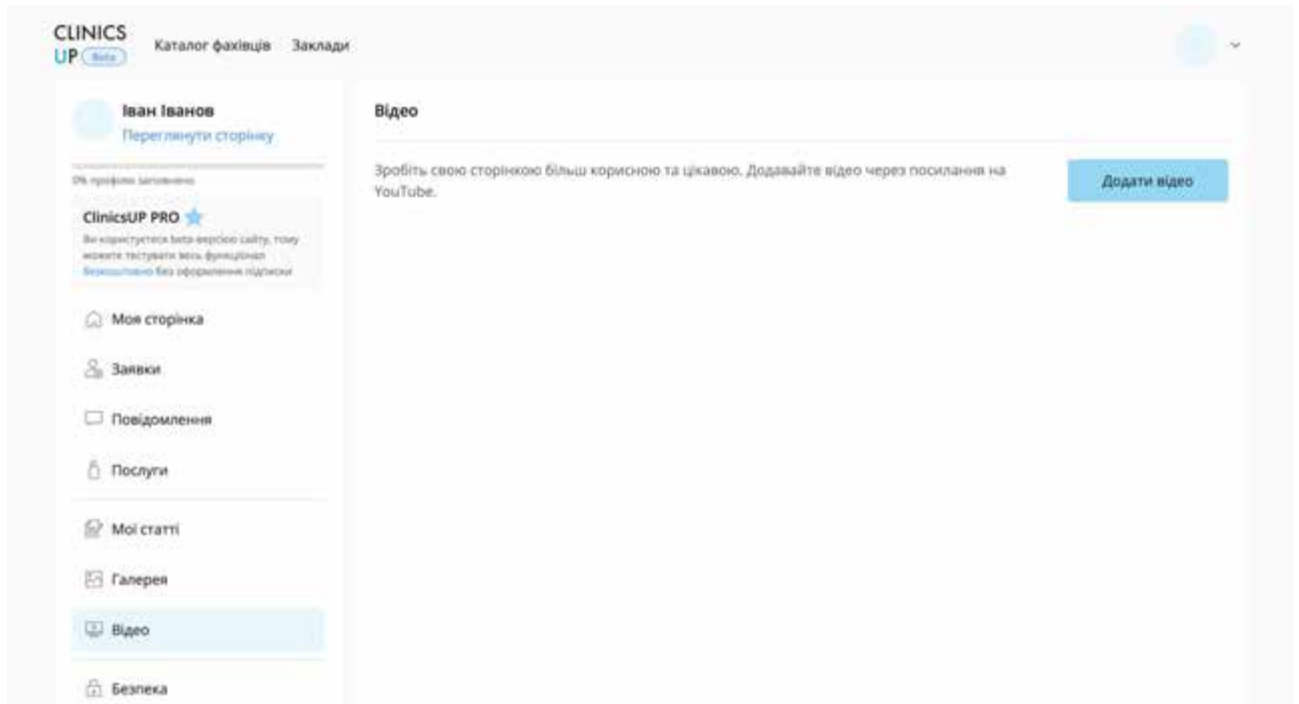


Рис. 4.24 – Відео

Ці можливості дозволяють спеціалістам зручно та ефективно управляти своєю роботою та взаємодіяти з клієнтами через вебсайт.

4.3 Апробація результатів розробленого вебсайту

Результати апробації розробленої системи "Програмне забезпечення системи аналізу обробки замовлень в сфері краси та здоров'я" підтвердили успішність та ефективність проекту. Основні результати апробації включають:

Функціональність: Всі функції системи пройшли успішне тестування і відповідають поставленим вимогам. Користувачі змогли використовувати систему для обробки замовлень, керування спеціалістами та клієнтами, а також для ведення обліку та аналізу.

Інтерфейс користувача: Інтерфейс системи був підданий тестуванню на зручність та користувацьку дружність. Користувачі відзначили зручність та інтуїтивність інтерфейсу.

Продуктивність: Система проявила високу продуктивність та стабільність навіть при великих навантаженнях. Час завантаження сторінок та відгук системи відповідає прийнятним стандартам.

Безпека: Під час тестування були виявлені і виправлені вразливості, що гарантує безпеку даних користувачів та надійність системи.

Автентифікація і авторизація: Система успішно здійснює процеси автентифікації та авторизації користувачів, забезпечуючи конфіденційність та контроль доступу до функцій.

Рекомендації щодо покращень: Після апробації були сформульовані рекомендації щодо подальшого вдосконалення системи, включаючи можливості розширення функціональності та покращення інтерфейсу.

Загальною висновком апробації є те, що розроблена система відповідає вимогам та готова до впровадження в сфері краси та здоров'я. Вона дозволить оптимізувати та полегшити процес обробки замовлень та керування персоналом.

Висновок до розділу 4

У розділі 4 нашої дипломної роботи було проведено важливий етап - тестування та апробацію розробленого вебсайту для системи аналізу обробки замовлень в галузі краси та здоров'я. Цей етап включав в себе низку важливих завдань і етапів, і результати важливі для підтвердження функціональності та ефективності системи.

Тестування вебсайту було проведено для перевірки коректності роботи всіх його функцій і можливостей. В ході тестування було виявлено і виправлено низку помилок і неточностей, що дозволило покращити якість вебсайту та користувацький досвід.

Апробація розробленого вебсайту була проведена на практиці, в реальних умовах використання системи фахівцями з галузі краси та здоров'я. Під час апробації були отримані важливі відгуки та рекомендації від користувачів, що сприяли подальшому вдосконаленню системи.

Отримані результати свідчать про успішне тестування та апробацію розробленого вебсайту. Система виявилася ефективною та зручною для

користувачів, і вже знайшла застосування в реальних умовах в галузі краси та здоров'я.

Отже, розділ 4 підтверджує готовність та придатність розробленого вебсайту для практичного використання, а також вказує на успішність проведених тестів та апробації.

ВИСНОВКИ

В моїй дипломній роботі було проведено дослідження та розробка програмного забезпечення для системи аналізу обробки замовлень в галузі краси та здоров'я. Робота включала в себе детальний аналіз потреб цільової аудиторії, проектування та розробку вебсайту, створення бази даних, а також тестування та апробацію системи.

Головною метою роботи було створення зручного та ефективного інструмента для фахівців з галузі краси та здоров'я для обробки та прийому замовлень, а також для користувачів, які шукають послуги у цій галузі. Система була розроблена з врахуванням найсучасніших технологій та практик в галузі веб-розробки.

В результаті проведеного дослідження та розробки було розроблено програмне забезпечення, яке успішно пройшло тестування та апробацію. Система дозволяє фахівцям створювати профілі, приймати та відхиляти замовлення, вести облік фінансів та взаємодіяти з клієнтами. Користувачам надається зручний інтерфейс для замовлення послуг та спілкування з фахівцями.

Отримані результати свідчать про наукову новизну та прикладну значимість розробленого програмного забезпечення. Система може бути використана для поліпшення обробки замовлень та покращення обслуговування клієнтів у галузі краси та здоров'я.

У цій дипломній роботі було вивчено та впроваджено найсучасніші технології в галузі веб-розробки, що дає можливість використовувати їх у практичних цілях. Виконана робота має великий потенціал для подальшого розвитку та вдосконалення в майбутньому.

Отже, дипломна робота підтверджує важливість розробки програмного забезпечення для системи аналізу обробки замовлень у сфері краси та здоров'я та вказує на можливості вдосконалення цієї галузі за допомогою сучасних технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

ДОДАТОК Б