

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ВЕБДОДАТКУ ДЛЯ СТОМАТОЛОГІЧНОГО КАБІНЕТУ З
ВИКОРИСТАННЯМ ASP.NET MVC ТА БАЗОЮ ДАНИХ MS SQL
SERVER**

**DEVELOPMENT OF A WEB APPLICATION FOR A DENTAL OFFICE
USING ASP.NET MVC AND AN MS SQL SERVER DATABASE**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-41
Кравченко Денис Олександрович

Керівник:
Потейчук Михайло Іванович

Кваліфікаційну роботу
допущено до захисту

«10» 06 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

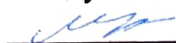
Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри



«12» 12 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Кравченюку Денису Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка вебдодатку для стоматологічного кабінету з використанням ASP.NET MVC та базою даних MS SQL Server»

Керівник роботи: асистент Потейчук М. І.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

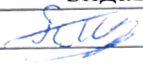
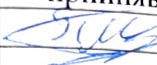
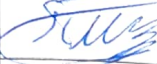
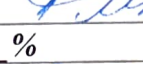
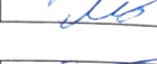
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: ASP.NET MVC, MS SQL Server, Visual Studio, LINQ to SQL, Magick.NET, HtmlSanitizer, Telegram Bot API, методичні вказівки до виконання кваліфікаційної роботи бакалавра

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз предметної області, огляд існуючих аналогів, проектування архітектури вебдодатку, обґрунтування вибору технологій, реалізація функціоналу, тестування, розробка бази даних, забезпечення безпеки

5. Перелік графічного матеріалу: 10 рисунків, 1 таблиця

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Потейчук М. І.		
Специфікація вимог до розробленої системи	Потейчук М. І.		
Розробка об'єкта проектування	Потейчук М. І.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	0,51 %		
Академічна доброчесність	Потейчук М. І.		

7. Дата видачі завдання «20» грудня 2024 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	виконано
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	виконано
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	виконано
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	виконано
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	виконано
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	виконано
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	до 10.06.2025 р.	виконано

Здобувач вищої освіти



Денис КРАВЧЕНЮК

Керівник кваліфікаційної роботи



Михайло ПОТЕЙЧУК

АНОТАЦІЯ

Кравченко Д. О. Розробка вебдодатку для стоматологічного кабінету з використанням ASP.NET MVC та базою даних MS SQL Server. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків та списку використаних джерел.

У першому розділі здійснено аналіз предметної області, досліджено програмні аналоги, обґрунтовано вибір засобів розробки і сформульовано завдання. У другому розділі представлено розробку архітектури вебдодатку, моделі бази даних і реалізацію ключових функцій. У третьому розділі описано процес реалізації коду, взаємодії з базою даних, механізми безпеки та тестування. У висновках підбито підсумки роботи, визначено напрями подальшого вдосконалення.

Ключові слова: вебдодаток, стоматологія, ASP.NET, MVC, MS SQL Server.

ABSTRACT

Kravchenyuk D. Development of a Web Application for a Dental Office Using ASP.NET MVC and an MS SQL Server Database. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions and a list of references.

The first chapter presents the analysis of the subject area, research of existing solutions, rationale for the choice of technologies and formulates the objectives. The second chapter describes the architecture of the application, database model, and implementation of core functionality. The third chapter explains the practical coding process, database interaction, implemented security mechanisms and testing. The conclusions summarize the results and outline prospects for further improvement.

Keywords: web application, dentistry, ASP.NET, MVC, MS SQL Server.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ У СФЕРІ СТОМАТОЛОГІЧНИХ ПОСЛУГ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми.....	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	12
Висновки до розділу 1	12
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	18
Висновки до розділу 2	24
РОЗДІЛ 3 РОЗРОБКА ВЕБДОДАТКУ СТОМАТОЛОГІЧНОГО КАБІНЕТУ ...	25
3.1 Практична реалізація об'єкта проектування	25
3.2 Тестування та налагодження системи	29
3.3 Розробка бази даних	31
3.4 Захист інформаційно-комп'ютерної системи	33
Висновки до розділу 3	34
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39

ВСТУП

Актуальність теми. У сучасних умовах цифровізації охорони здоров'я зростає потреба в автоматизованих рішеннях для ефективного управління медичною діяльністю. Однією з важливих ланок системи охорони здоров'я є стоматологічні кабінети, які надають високоспеціалізовані послуги населенню. Водночас більшість таких закладів, особливо приватного сектору, стикаються з проблемами неефективного обліку пацієнтів, складністю адміністрування записів, відсутністю єдиного цифрового середовища для роботи лікарів і персоналу. Ці проблеми потребують вирішення шляхом впровадження індивідуалізованих програмних рішень.

Відповідно, актуальність теми обумовлена необхідністю автоматизації управлінських та медико-облікових процесів у стоматологічному кабінеті. Більшість існуючих інформаційних систем не відповідають специфічним вимогам приватних стоматологій або потребують значних фінансових витрат на ліцензування. Тому розробка вебдодатку, адаптованого до конкретних потреб закладу, на основі відкритих технологій (зокрема ASP.NET MVC та MS SQL Server), є доцільною та своєчасною.

Світові тенденції свідчать про активне впровадження електронних медичних карт, систем для онлайн-запису, хмарних сервісів для управління пацієнтською базою. Це вимагає створення вебзастосунків, які працюють у режимі реального часу, забезпечують безпеку персональних даних і мають зручний інтерфейс для лікарів та пацієнтів.

Метою кваліфікаційної роботи бакалавра є розробка вебдодатку для стоматологічного кабінету з використанням ASP.NET MVC та бази даних MS SQL Server, який забезпечує зручне управління записами пацієнтів, обліком послуг і розкладом лікарів.

Об'єктом у кваліфікаційній роботі виступає процес керування та планування особистих завдань із використанням веборієнтованих програмних інструментів.

Предметом кваліфікаційної роботи бакалавра є процес розробки, моделювання та впровадження вебдодатку для автоматизації внутрішніх процесів стоматологічного кабінету.

Для досягнення поставленої мети були сформульовані такі завдання:

- 1) провести аналіз сучасних програмних рішень у сфері медичних інформаційних систем;
- 2) сформулювати вимоги до вебдодатку для стоматологічного кабінету та побудувати архітектурну модель системи;
- 3) обрати інструменти розробки;
- 4) розробити вебдодаток на основі ASP.NET MVC з використанням MS SQL Server, реалізувати базові функціональні модулі;
- 5) провести тестування і налагодження розробленої системи;
- 6) побудувати архітектурну модель бази даних та реалізувати її;
- 7) реалізувати базові механізми захисту.

Розробка є самостійним завершеним рішенням, що має практичну цінність для закладів охорони здоров'я, які прагнуть оптимізувати свою діяльність за допомогою сучасних вебтехнологій. Робота пов'язана зі сферою розробки інформаційних систем, зокрема тих, що орієнтовані на підтримку прийняття рішень та ведення обліку в медицині.

РОЗДІЛ 1

АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ У СФЕРІ СТОМАТОЛОГІЧНИХ ПОСЛУГ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сфері охорони здоров'я, зокрема в стоматологічній практиці, інформаційні системи поступово стають невід'ємною частиною надання якісних медичних послуг. Упровадження автоматизованих систем дає змогу не лише підвищити рівень обслуговування пацієнтів, а й оптимізувати адміністративну діяльність закладів. За даними сучасних досліджень (наприклад, PubMed, ScienceDirect, IEEE Xplore), основні тенденції у розробці медичних інформаційних систем пов'язані з інтеграцією електронної медичної документації (EMR/EHR), мобільного доступу, хмарних технологій, а також захисту персональних даних відповідно до стандартів, як-от HIPAA та GDPR.

У процесі аналізу сучасного стану автоматизації в галузі стоматологічних послуг особливу увагу привертають дослідження, що розглядають впровадження інформаційних технологій у медичну практику, зокрема – цифрові рішення, системи обліку та CRM-платформи. Однією з актуальних тенденцій є інтеграція штучного інтелекту у сферу стоматології. У роботі N. Ahmed та співавторів [1] здійснено систематичний огляд застосування технологій штучного інтелекту в галузі стоматології, з акцентом на прогнозування, діагностику та оптимізацію лікування. Автори підкреслюють, що цифрові інструменти значно покращують точність і ефективність прийняття рішень, однак успішність їх використання залежить від якості вхідних даних та рівня інтеграції в існуючі медичні системи.

У дослідженні S. B. Но та співавторів [2] розглядається питання цифрової трансформації стоматологічних клінік, зокрема розробка інформаційної системи управління клінікою. Автори надають методологічні рекомендації щодо підвищення продуктивності, зручності користування та автоматизації основних адміністративних процесів. Досвід зарубіжних клінік свідчить про те,

що найбільш ефективними виявляються системи з можливістю гнучкої адаптації до бізнес-логіки конкретного закладу.

Український контекст аналізується в публікації М. Козулі та О. Солдатко [3], де описано процес проектування CRM-системи для стоматологічної практики. Дослідження охоплює структуру інформаційних потоків у закладі охорони здоров'я та пропонує модель системи, яка враховує специфіку адміністрування в галузі стоматології. Практична цінність публікації полягає у використанні вітчизняного досвіду й акценті на необхідності врахування фактору персоніфікованого обслуговування.

У доповідях О. Гарматюк та А. Горохівського [4] представлено стислий огляд інновацій у сфері цифрової стоматології, зокрема йдеться про впровадження мобільних додатків, онлайн-запису та хмарних сервісів. Акцент зроблено на доступності послуг і зручності пацієнта як головних факторах ефективності цифрових рішень.

В. В. Білий та О. І. Денисенко [5] описують практичну реалізацію облікової системи клієнтів для стоматології, з базовим функціоналом введення, редагування та перегляду інформації про пацієнтів. Автори підкреслюють, що автоматизація рутинних операцій дозволяє звільнити ресурси адміністрації та покращити якість обслуговування.

Нарешті, у роботі А. В. Чередниченка та В. А. Чередниченка [6] досліджується загальна тенденція цифровізації бізнес-процесів у стоматології. Описано зміни в управлінні ресурсами, взаємодії з клієнтами та формуванні цифрових слідів обслуговування, що дозволяють краще аналізувати ефективність клініки.

Отже, аналіз наукових джерел свідчить про високу актуальність автоматизації в стоматологічній галузі, причому провідною тенденцією є створення гнучких, адаптивних і безпечних інформаційних систем, що дозволяють ефективно керувати пацієнтськими даними, записами та послугами. Це підтверджує доцільність розробки спеціалізованого вебдодатку, орієнтованого на потреби конкретного медичного закладу.

Також проведено аналіз аналогічних рішень. DentOffice – українська програма, орієнтована на ведення обліку пацієнтів, фінансову звітність, графік роботи лікарів [7]. Перевагами є локалізований інтерфейс, підтримка касових операцій. Недоліком є закрита архітектура, відсутність хмарного доступу та обмежена кастомізація. EasyPractice – SaaS-рішення для приватної практики з широким функціоналом (записи, повідомлення, платежі), доступне онлайн [8]. Однак орієнтоване переважно на ринок ЄС, має мовні бар'єри і платну модель. WebDental – вебплатформа для стоматологічних клінік із підтримкою мобільної версії, резервного копіювання, модулів HER [9]. Основним недоліком є висока вартість підключення та обслуговування.

У ході патентного пошуку за основними запитами виявлено низку запатентованих рішень, які, однак, фокусуються на окремих аспектах: візуалізації зубних формул, інтеграції з обладнанням або діагностичних модулях. Комплексне, доступне рішення з відкритою архітектурою, яке дозволяє легко модифікувати функціонал під конкретні потреби стоматологічного кабінету, фактично відсутнє на українському ринку.

Зважаючи на це, виникає необхідність створення вебдодатку, який би забезпечував базову функціональність для обліку пацієнтів, запису на прийом, адміністрування персоналу, а також міг масштабуватись у разі потреби. Його реалізація на основі ASP.NET MVC забезпечить структуровану архітектуру, розділення відповідальностей, гнучке керування даними через Entity Framework, а MS SQL Server дозволить надійно зберігати та обробляти дані з урахуванням вимог безпеки.

Загалом, огляд сучасного стану проблеми підтверджує недостатню доступність адаптивних, економічно доцільних рішень для малих стоматологічних закладів, що й обумовлює доцільність розробки нової інформаційної системи, орієнтованої на ефективну та безпечну автоматизацію процесів в межах приватного медичного бізнесу.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Метою кваліфікаційної роботи є розробка вебдодатку для стоматологічного кабінету, який забезпечує автоматизацію облікових і управлінських процесів закладу з використанням технологій ASP.NET MVC і MS SQL Server. Для досягнення поставленої мети необхідно вирішити наступні завдання:

- 1) провести аналіз сучасних програмних рішень у сфері медичних інформаційних систем;
- 2) сформулювати вимоги до вебдодатку для стоматологічного кабінету та побудувати архітектурну модель системи;
- 3) обрати інструменти розробки;
- 4) розробити вебдодаток на основі ASP.NET MVC з використанням MS SQL Server, реалізувати базові функціональні модулі;
- 5) провести тестування і налагодження розробленої системи;
- 6) побудувати архітектурну модель бази даних та реалізувати її;
- 7) реалізувати базові механізми захисту.

Ці завдання дозволяють комплексно охопити процес створення інформаційної системи для стоматологічного кабінету, зосередивши увагу на її функціональності, безпеці, зручності використання та адаптованості до реальних потреб галузі.

Висновки до розділу 1

У першому розділі було здійснено огляд сучасного стану проблеми автоматизації діяльності стоматологічних кабінетів. На основі аналізу літературних джерел та існуючих програмних продуктів виявлено, що більшість рішень не враховують специфічних потреб малих медичних закладів, є складними у впровадженні або мають надмірний функціонал, що ускладнює їх адаптацію.

Розглянуті аналоги, зокрема DentOffice, EasyPractice та WebDental, хоча й мають позитивні функціональні характеристики, не забезпечують достатньої гнучкості, масштабованості та доступності, що є критичним для приватних стоматологічних практик. У результаті було обґрунтовано доцільність створення нового вебдодатку, що дозволить ефективно вирішити завдання обліку, адміністрування і планування роботи в межах стоматологічного кабінету.

Сформульовані конкретні завдання для реалізації кваліфікаційної роботи охоплюють всі основні аспекти розробки інформаційної системи – від аналізу вимог до побудови архітектури, створення бази даних, реалізації вебінтерфейсу, тестування та захисту даних. Визначена мета роботи та обґрунтовано її актуальність у контексті розвитку цифрових медичних сервісів.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

На основі попереднього аналізу предметної області та цільової аудиторії було визначено функціональні та нефункціональні вимоги до вебдодатку для стоматологічного кабінету «LK Dental». Метою є створення зручної, надійної та адаптивної системи, яка забезпечує облік пацієнтів, управління записами, доступ до інформації про послуги, лікарів, цінову політику, а також дозволяє взаємодіяти з потенційними та наявними клієнтами.

Вимоги до функціональності системи:

- 1) онлайн-запис на прийом – з можливістю вибору лікаря, дати, години, послуги;
- 2) модуль «Послуги» – структурований каталог стоматологічних послуг з описами, цінами, інфографікою;
- 3) профілі лікарів – персональні сторінки із зазначенням спеціалізації, стажу, сертифікатів, фото;
- 4) відгуки пацієнтів – система коментарів, оцінок, відображення публічних відгуків;
- 5) блог – публікація новин, корисних статей, порад, SEO-оптимізований контент;
- 6) інформаційна сторінка про клініку – місія, історія, контактна інформація, години роботи, мапа.

Нефункціональні вимоги:

- 1) адаптивний дизайн, підтримка мобільних пристроїв;
- 2) захист форми запису від ботів, обмежений доступ до адміністративної частини, шифрування паролів;
- 3) оптимізація завантаження сторінок, кешування статичного контенту;
- 4) можливість розширення функціоналу без порушення архітектури.

Для реалізації вебдодатку було обрано архітектурну модель MVC (Model-View-Controller), що забезпечує чітке розділення логіки відображення, обробки запитів і роботи з даними [10]. Це дозволяє досягти модульності коду, зручності тестування та підтримки.

Вимоги формувались на основі аналізу функціоналу аналогічних систем (EasyPractice, WebDental), UX-досліджень взаємодії користувачів з медичними сайтами та опрацювання потреб адміністрації стоматологічного кабінету.

Для опису структури і взаємодії компонентів розроблено UML-діаграми. Use case-діаграма (рис. 2.1) показує сценарії взаємодії користувача з системою (пацієнт, адміністратор).

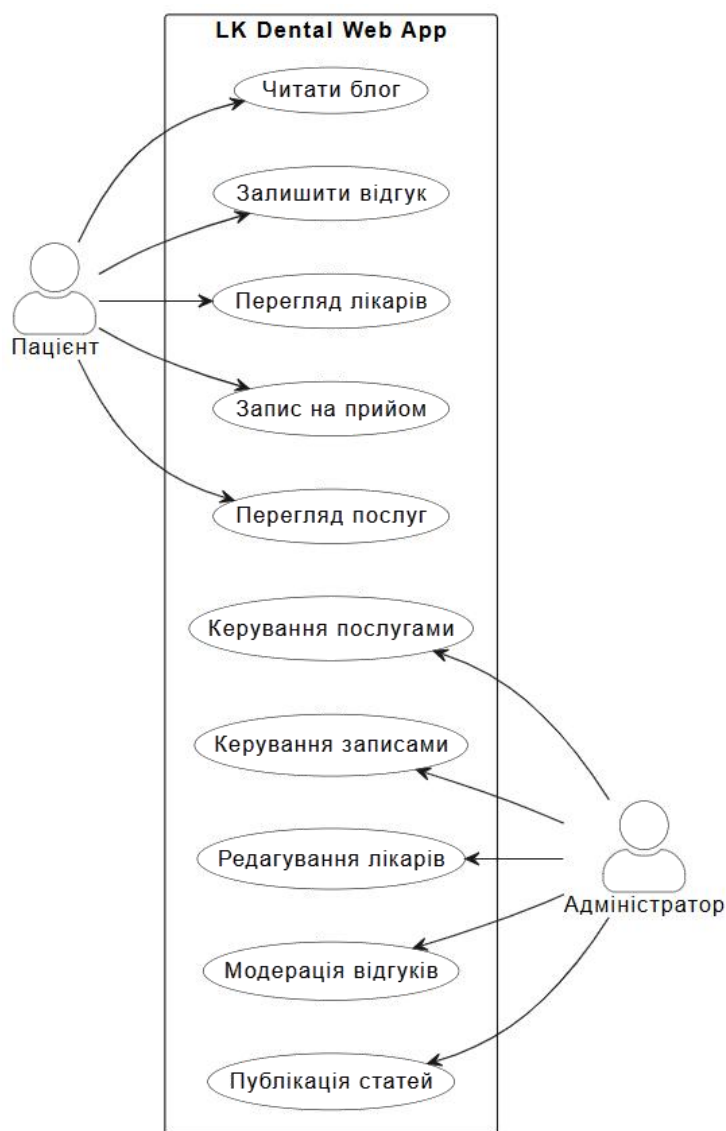


Рисунок 2.1 – Діаграма варіантів використання

Діаграма класів (рис. 2.2) демонструє моделі сутностей: «Пацієнт», «Послуга», «Запис», «Лікар», «Відгук», «Адміністратор».

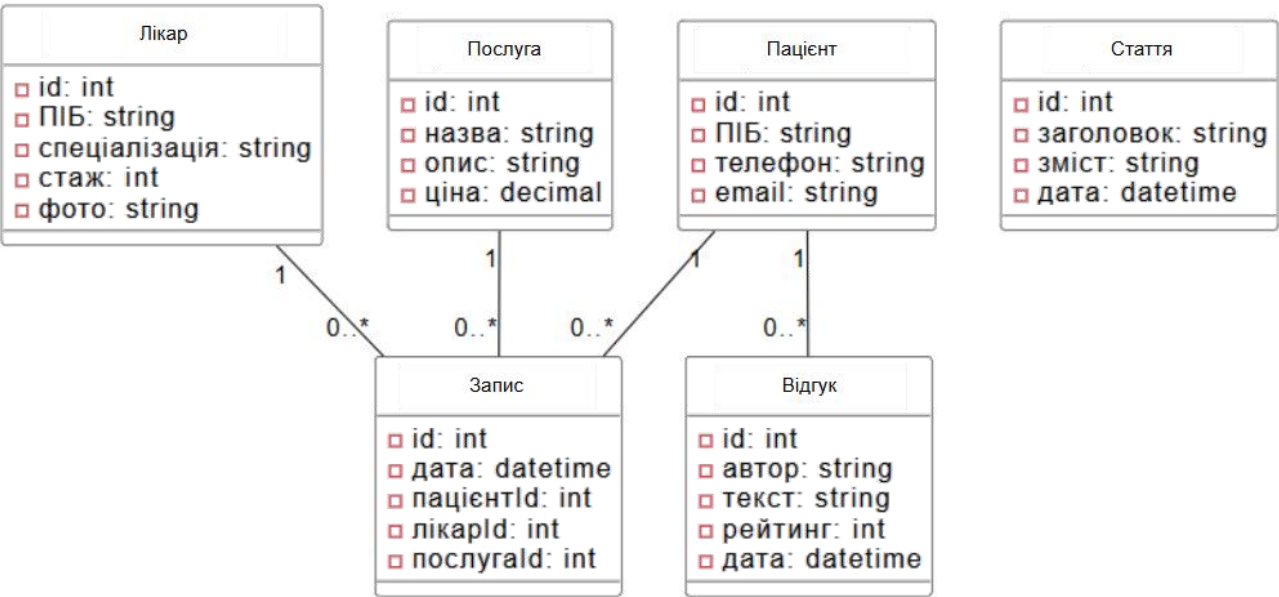


Рисунок 2.2 – Діаграма класів

Sequence-діаграма (рис. 2.3) демонструє послідовності дій користувача під час запису на прийом.

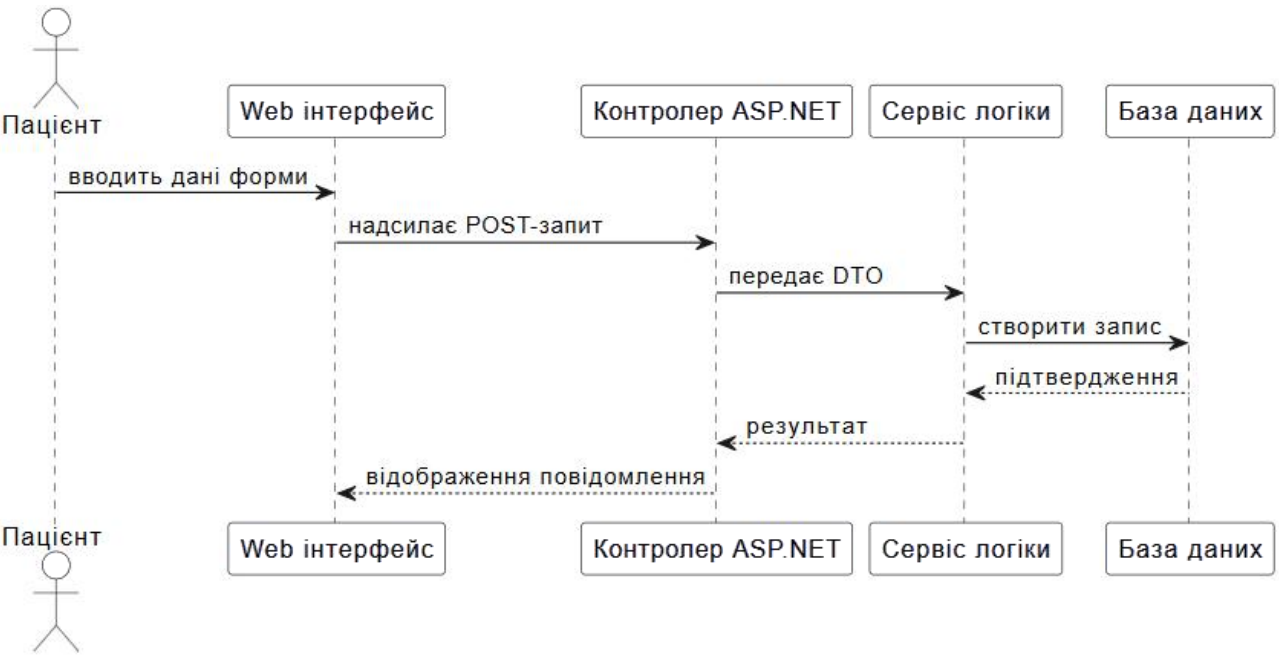


Рисунок 2.3 – Діаграма послідовності

Activity-діаграма (рис. 2.4) демонструє логіку запису – від заповнення форми до обробки сервером.

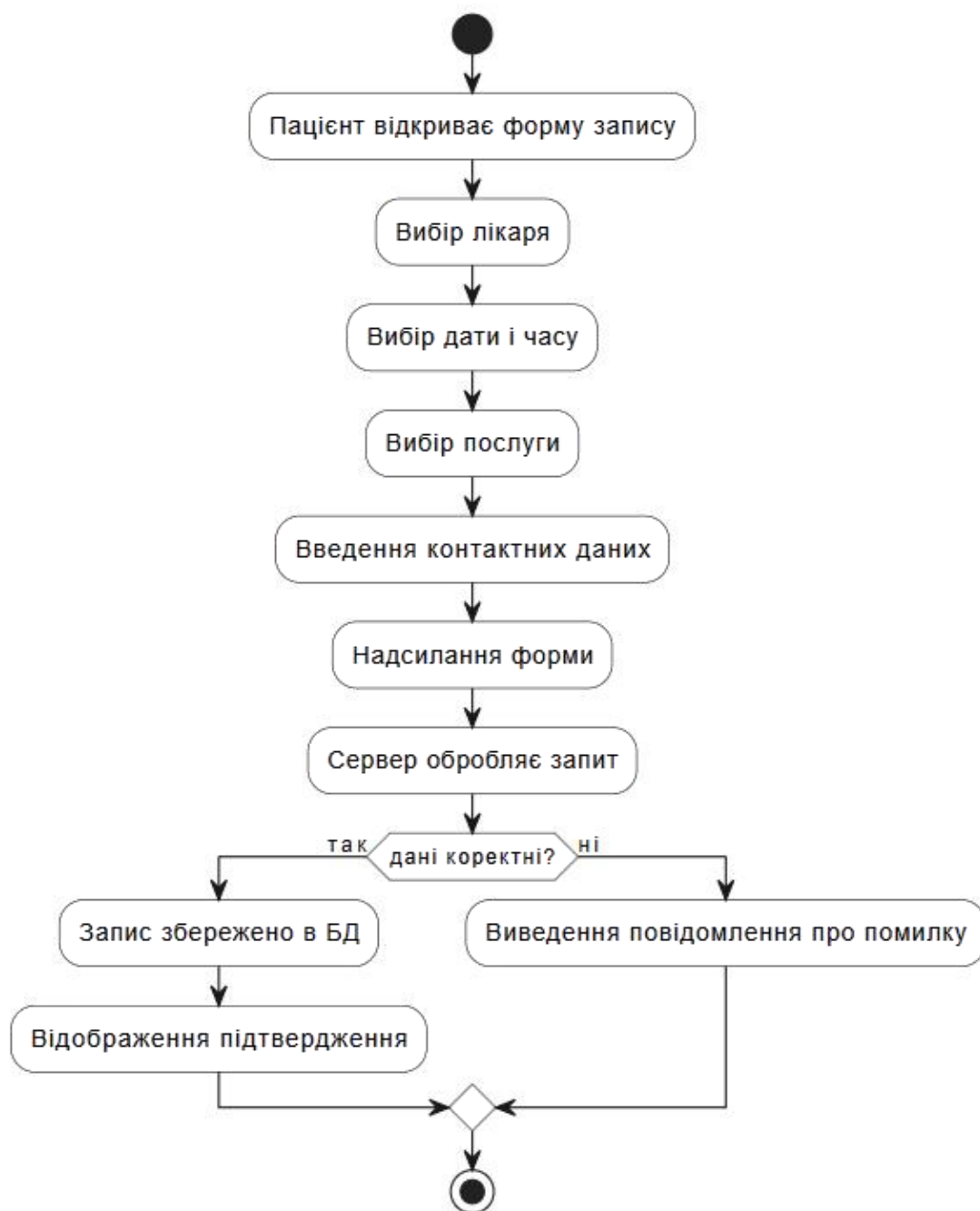


Рисунок 2.4 – Діаграма активності

Також в таблиці 2.1 описано специфікацію функціональних вимог до системи.

Таблиця 2.1 – Специфікація функціональних вимог

ID	Назва функції	Короткий опис	Пріоритет
F01	Створення заявки на прийом	Пацієнт заповнює форму запису онлайн	Високий
F02	Керування розкладом	Адміністратор встановлює графік лікарів	Високий
F03	Додавання/редагування послуг	Адміністратор оновлює дані про процедури та ціни	Високий
F04	Перегляд сторінки лікаря	Користувач переглядає профіль лікаря	Середній
F05	Модуль відгуків	Пацієнти залишають публічні коментарі	Середній
F06	Публікація статей у блозі	Адміністратор створює/редагує інформаційні матеріали	Середній

Інтерфейс створювався з урахуванням принципів мінімалізму, інтуїтивної навігації, доступності для широкої аудиторії та чіткої візуалізації структури послуг. Для прототипування використовувались інструменти Figma та Bootstrap-шаблони для ASP.NET MVC.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Розробка вебдодатку для стоматологічного кабінету потребує ретельного вибору інструментів програмування, середовищ розробки, засобів моделювання та баз даних. Важливим етапом є також формування методів обробки запитів, забезпечення безпеки даних, структуризації архітектури застосунку та реалізації інтерактивного інтерфейсу користувача.

На UML-діаграмі на рисунку 2.5 зображено ключові частини вебдодатку «LK Dental» відповідно до трирівневої архітектури. Користувач взаємодіє з інтерфейсом, побудованим на HTML, Razor і Bootstrap, який через JavaScript та jQuery викликає контролери ASP.NET MVC. Контролери, у свою чергу, передають запити до бізнес-логіки (сервісів та репозиторіїв), які обробляють дані за допомогою ORM-технології Entity Framework та з'єднуються з базою даних Microsoft SQL Server. Додатково, контролери використовують HtmlSanitizer для безпечної обробки вводу, ImageMagick – для обробки зображень, а Telegram API – для надсилання повідомлень адміністрації. Така

модульна побудова забезпечує чіткий розподіл обов'язків, легкість масштабування та супроводу системи.

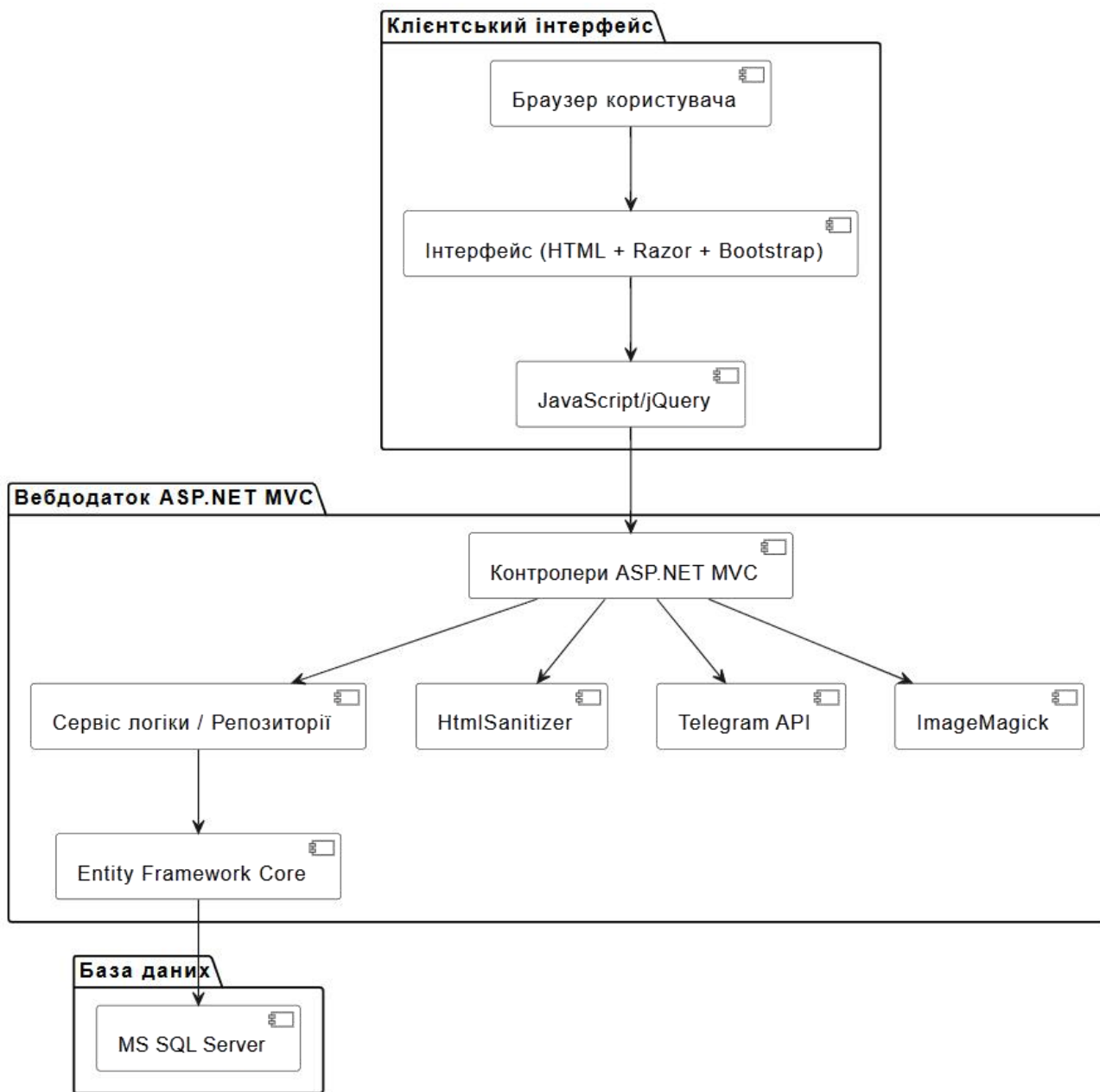


Рисунок 2.5 – Архітектура вебдодатку

Серед найпоширеніших технологічних стеків, які використовуються для розробки сучасних вебдодатків, можна виокремити кілька альтернативних рішень. Наприклад, комбінація PHP з фреймворком Laravel та СКБД MySQL забезпечує просту реалізацію CRUD-операцій, однак має обмеження при створенні складних модулів без додаткових бібліотек і компонентів. Python у поєднанні з Django та PostgreSQL характеризується високою швидкістю

розробки і зручністю при створенні REST-сервісів, але менш придатний для розгортання у середовищах з Windows-серверами, що часто використовуються в медичних установах. Java разом із Spring Boot та Oracle DB забезпечує потужну й надійну платформу для великих корпоративних систем, однак є надто ресурсомісткою й складною для реалізації у невеликих приватних медичних кабінетах. У свою чергу, стек на базі Node.js із фреймворком Express і нереляційною базою MongoDB дозволяє створювати масштабовані та продуктивні вебдодатки, але має менш чітку типізацію коду, що ускладнює підтримку великих проєктів з великою кількістю залежностей.

Обґрунтування вибору технологій для розробки вебдодатку «LK Dental» базується на поєднанні продуктивності, безпеки, зручності реалізації архітектурних рішень і гнучкості у подальшій підтримці та розширенні функціоналу. Основним інструментом реалізації клієнт-серверної взаємодії обрано платформу ASP.NET MVC із застосуванням мови програмування C# [11]. Архітектура Model-View-Controller (MVC) дозволяє ефективно розділити логіку програми на три взаємодоповнювальні компоненти: модель, яка відповідає за бізнес-дані, контролер, що керує логікою обробки запитів, і подання (View), яке формує інтерфейс користувача. У проєкті ця архітектура реалізована через відповідні структурні папки Controllers, Models, Views, що сприяє кращій організації коду, його масштабуванню та підтримці. Додаткові папки Core, Scripts та Template відображають багаторівневий підхід до структурування бізнес-логіки, зовнішнього вигляду та допоміжних інструментів.

Для роботи з базою даних було обрано технологію Entity Framework, яка виконує роль об'єктно-реляційного відображення (ORM). Вона дозволяє розробнику працювати з даними на рівні об'єктів без необхідності створення SQL-запитів вручну [12]. У цьому проєкті використано підхід Code First, що передбачає створення моделей у вигляді C# класів, з яких автоматично генерується база даних. Це забезпечує гнучкість у зміні структури даних,

спрощує міграції та зменшує ризик помилок. Крім того, ORM підвищує безпеку обробки запитів, запобігаючи SQL-ін'єкціям шляхом параметризації.

Основною системою управління базами даних виступає Microsoft SQL Server [13]. Її вибір обґрунтовано високою сумісністю з .NET-технологіями, зокрема через повну інтеграцію з Entity Framework. SQL Server забезпечує надійне зберігання, обробку та захист даних, включаючи підтримку транзакційності, цілісності даних, масштабованість на майбутнє. Його зручна інструментальна база, зокрема SQL Server Management Studio (SSMS), дозволяє легко адмініструвати таблиці, створювати запити, індекси, процедури, контролювати доступ до даних і здійснювати резервне копіювання.

Для створення сучасного та адаптивного інтерфейсу користувача використано фреймворк Bootstrap, який дозволяє швидко створювати кросбраузерні й мобільно-орієнтовані макети [14]. Верстка адаптована до різних розмірів екранів, що особливо важливо в умовах, коли пацієнти можуть взаємодіяти із сайтом не лише з ПК, а й зі смартфонів або планшетів. Візуальну частину подань реалізовано за допомогою Razor-синтаксису – вбудованої системи шаблонів у ASP.NET, яка дозволяє вставляти C#-код у HTML-розмітку. Це забезпечує зручність формування динамічного контенту, збереження логіки подання та прозорість зв'язку між поданням і контролером.

Для реалізації динамічної взаємодії з користувачем, валідації форм та відправки даних без перезавантаження сторінки використовуються JavaScript і бібліотека jQuery [15]. Розміщені у відповідній директорії Scripts, ці скрипти забезпечують такі функції як перевірка правильності заповнення полів, динамічне оновлення елементів сторінки, реалізація асинхронних запитів до сервера через AJAX. Застосування jQuery значно пришвидшує розробку завдяки великій кількості готових методів для роботи з DOM, подіями та анімацією.

В цілому, вибір технологій для реалізації вебдодатку ґрунтується на їхній стабільності, масштабованості, безпеці, активній підтримці розробницькою

спільнотою, а також здатності ефективно реалізувати як бізнес-логіку, так і сучасний інтерфейс для користувача.

У вебдодатку реалізовано кілька важливих алгоритмів:

1) алгоритм створення запису на прийом – включає перевірку доступності лікаря у вибраний день, валідацію введених користувачем даних та збереження об'єкта в БД;

2) алгоритм обробки відгуків – включає створення, збереження та модерацію відгуків адміністрацією;

3) фільтрація та сортування послуг/лікарів – реалізовано через LINQ-запити в контролерах із динамічною побудовою виразів;

4) алгоритм захисту адміністративної панелі – реалізовано автентифікацію користувача (Forms Authentication / ASP.NET Identity) з розмежуванням доступу до певних контролерів за ролями.

У процесі розробки було також створено кілька допоміжних діаграм. Діаграма компонентів (рис. 2.6) – демонструє поділ системи на логічні частини: інтерфейс, логіка, база даних.

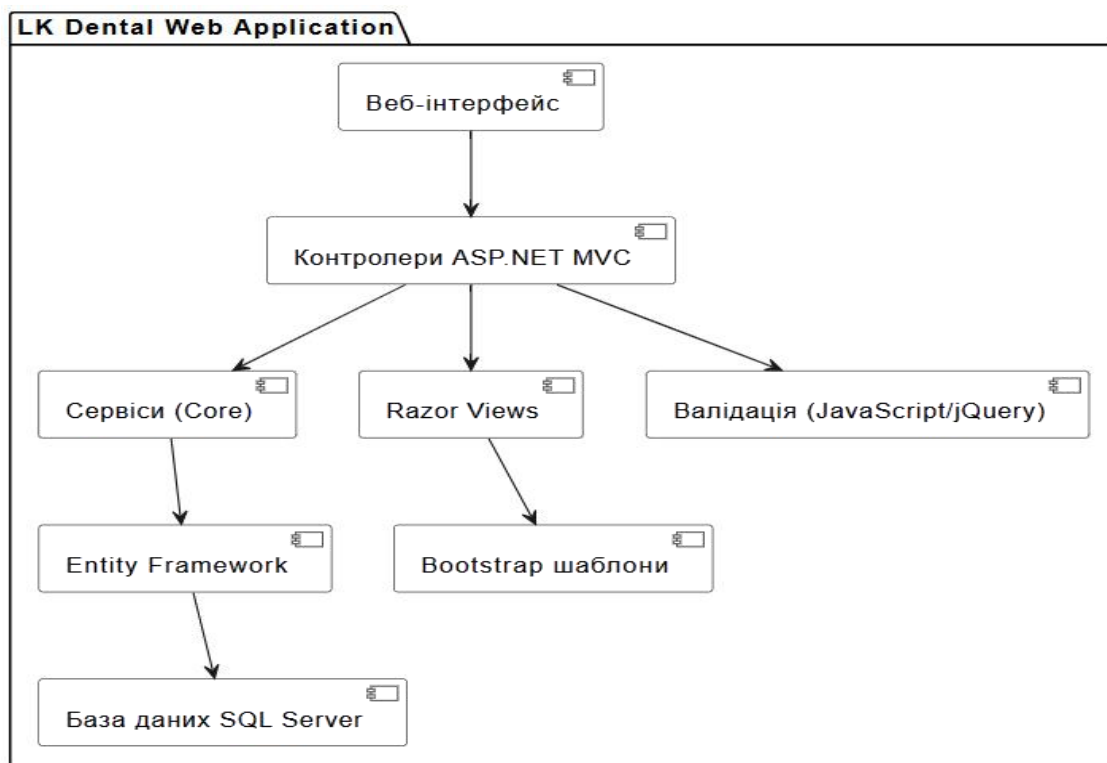


Рисунок 2.6 – Діаграма компонентів

Діаграма розміщення (рис. 2.7) – описує розгортання системи на IIS-сервері з підключенням до SQL Server.

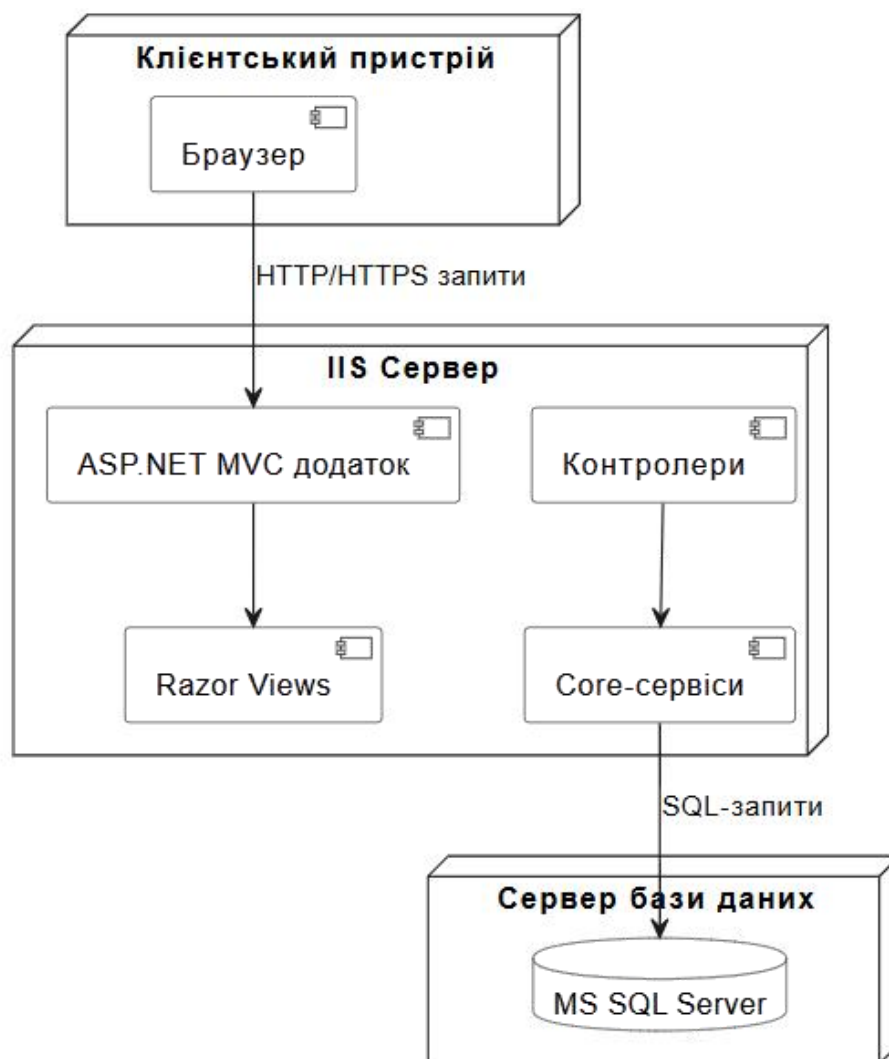


Рисунок 2.7 – Діаграма розміщення

Обрані засоби ASP.NET MVC, EF Core та MS SQL Server у поєднанні з сучасними інструментами фронтенду дозволили реалізувати ефективний, безпечний і масштабований вебдодаток, орієнтований на потреби стоматологічної практики. Усі компоненти системи логічно структуровані, легко підтримуються й відповідають принципам модульності та повторного використання коду.

Висновки до розділу 2

У другому розділі було сформовано та обґрунтовано вимоги до функціональності й структури вебдодатку для стоматологічного кабінету «LK Dental». Проведено аналіз функціональних і нефункціональних вимог, які відповідають потребам як пацієнтів, так і адміністративного персоналу. Зокрема, визначено основні функції системи: онлайн-запис на прийом, управління даними пацієнтів і лікарів, публікація послуг, ведення блогу, обробка відгуків.

Для побудови архітектури програмного забезпечення обрано модель MVC, яка забезпечує чітке розділення між логікою, даними та представленням. Це сприяє структурованості, гнучкості й масштабованості проєкту. Визначено доцільність використання ORM-технології Entity Framework у поєднанні з базою даних MS SQL Server для ефективного зберігання та обробки медичних даних.

У процесі аналізу альтернативних інструментів розробки було обґрунтовано вибір технологічного стеку ASP.NET MVC, що повністю відповідає цілям дипломної роботи. Також сформовано UML-діаграми – варіантів використання, класів, компонентів, розміщення – які відображають логічну структуру проєкту, архітектурні рішення та спосіб взаємодії між компонентами.

Загалом, на основі проведеного аналізу сформовано концептуальну модель вебдодатку та визначено засоби, необхідні для його реалізації, що створює основу для подальшого практичного впровадження системи.

РОЗДІЛ 3

РОЗРОБКА ВЕБДОДАТКУ СТОМАТОЛОГІЧНОГО КАБІНЕТУ

3.1 Практична реалізація об'єкта проєктування

У процесі розробки вебдодатку було використано мову програмування C# у середовищі Microsoft Visual Studio 2022 з шаблоном ASP.NET MVC 5. Додаток побудований за принципами Model–View–Controller (MVC), що дозволяє розділити логіку програми, користувацький інтерфейс та доступ до даних на окремі компоненти.

Для зберігання та обробки даних застосовано технологію LINQ to SQL з використанням *.dbml моделі Dentist.dbml, яка містить сутності, що відповідають таблицям бази даних: Doctor, Response, Blog, Categories тощо. Усе взаємодіє з базою даних через репозиторійний шаблон проєктування, прикладом чого є клас DoctorRepository (ліст. 3.1).

Лістинг 3.1 – Метод отримання списку всіх лікарів із бази даних

```
public IEnumerable<Doctor> GetAllDoctors()  
{  
    return _datacontext.Doctor.OrderBy(t => t.name);  
}
```

кінець лістингу 3.1

Цей метод дозволяє отримати список усіх зареєстрованих лікарів у базі даних, сортує їх за іменем та передає далі у контролер. Сторінка лікарів зображена на рисунку 3.1.

На рівні контролера, наприклад у HomeController, виконується логіка запиту та формування представлення (View), яке відображається користувачу (ліст. 3.2).

Лістинг 3.2 – Контролер виводу лікарів на сторінці залежно від параметра

```
public ActionResult Doctor(string id = "")  
{  
    if (id == "")
```

```

{
    ViewData["Doctor"] = _dataManager.DoctorR.GetAllDoctors();
    ViewBag.H1 = "Наші лікарі";
}
else
{
    DoctorType type = _dataManager.DoctorR.GetSingleType(id);
    ViewData["Doctor"] = _dataManager.DoctorR.GetDoctorFromType(id);
    ViewBag.H1 = type.value;
}
return View();
}

```

кінець лістингу 3.2

Цей метод обробляє запит сторінки лікарів і в залежності від наявності параметра id формує вивід або всіх лікарів, або лише тих, які належать до певного типу.

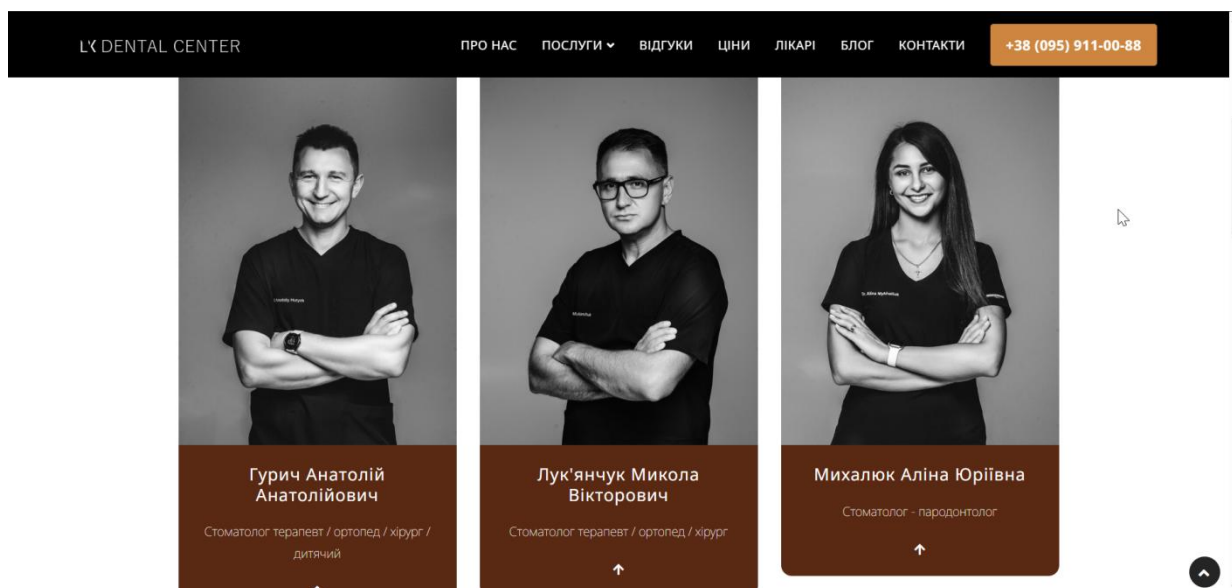


Рисунок 3.1 – Сторінка лікарів

Функціональність додавання відгуків реалізовано з урахуванням безпеки – зокрема, використано HtmlSanitizer для фільтрації шкідливих HTML-вставок (ліст. 3.3).

Лістинг 3.3 – Очищення тексту відгуку за допомогою HtmlSanitizer

```

var sanitizer = new HtmlSanitizer();

```

```
model.text = sanitizer.Sanitize(model.text); // очищення тексту відгуку
```

кінець лістингу 3.3

Інтерфейс сторінки відгуків зображено на рисунку 3.2.

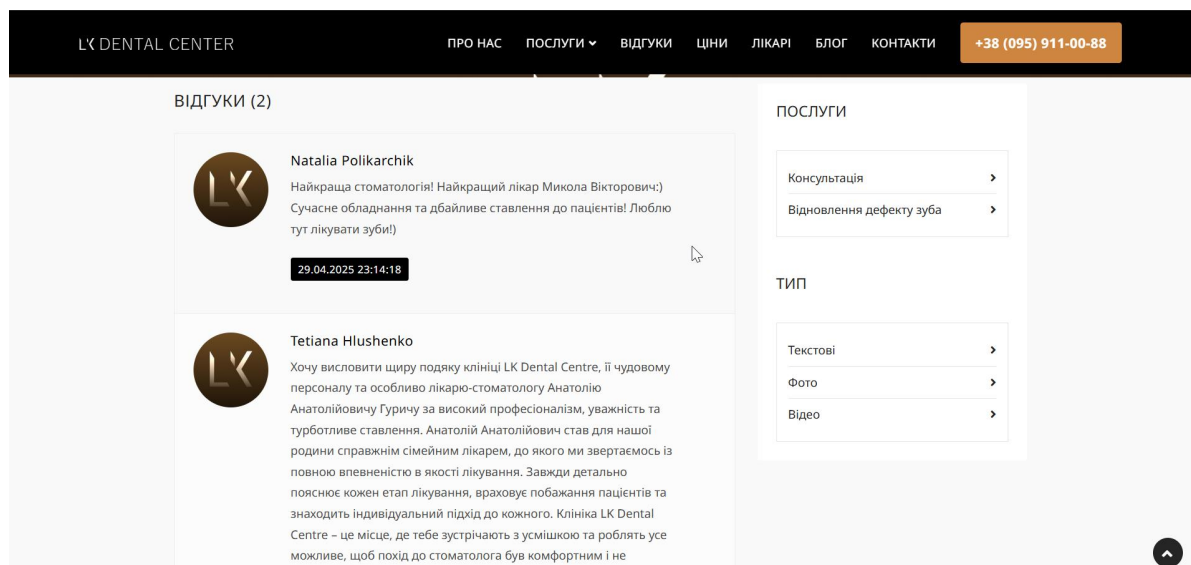


Рисунок 3.2 – Сторінка відгуків

Для завантаження зображень, наприклад, до блогу або відгуку, застосовується бібліотека ImageMagick (Magick.NET), яка дозволяє перетворити будь-яке зображення у формат JPG перед збереженням (ліст. 3.4).

Лістинг 3.4 – Обробка та збереження зображення у форматі JPG з використанням Magick.NET

```
using (var image = new MagickImage(uploadedTitle.InputStream))
{
    image.Format = MagickFormat.Jpg;
    image.Write(filePath);
}
```

кінець лістингу 3.4

Це дозволяє забезпечити єдиний формат зображень і оптимізувати їх для вебперегляду.

Також було реалізовано надсилання повідомлень через Telegram API при подачі заявки на послугу зі сторінки «Контакти». Для цього створено окремий

асинхронний метод `SendMessageToTelegram`, який викликається з `SendOrder` (ліст. 3.5).

Лістинг 3.5 – Надсилання повідомлення до Telegram через API

```
private static async Task SendMessageToTelegram(string message)
{
    using (var httpClient = new HttpClient())
    {
        var url =
$"https://api.telegram.org/bot{telegramApiToken}/sendMessage";
        var data = new FormUrlEncodedContent(new[]
        {
            new KeyValuePair<string, string>("chat_id", telegramChatId),
            new KeyValuePair<string, string>("text", message)
        });

        await httpClient.PostAsync(url, data);
    }
}
```

кінець лістингу 3.5

Відповідно, адміністратор отримує оперативне повідомлення про заявку без необхідності перевіряти панель керування.

Також на рисунку 3.3 зображено головну сторінку вебсайту LK Dental Center.

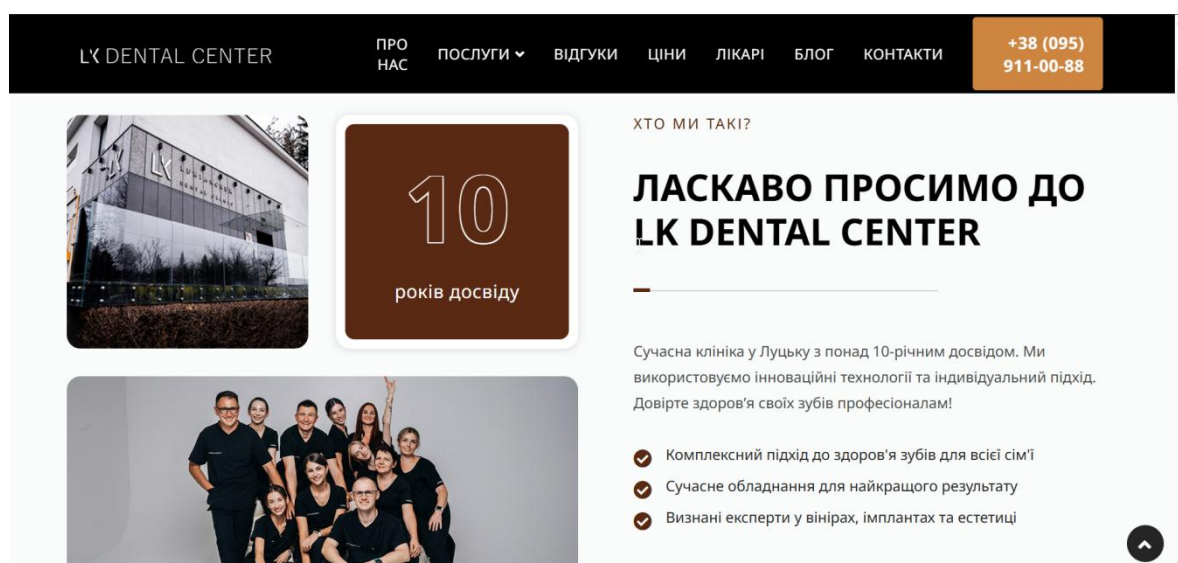


Рисунок 3.3 – Головна сторінка сайту

Інтерфейс виконано у сучасному стилі з акцентом на контрастну візуальну структуру: вгорі – навігаційне меню, у центрі – вітальний блок із заголовком, підзаголовком і фотографіями клініки та колективу. Дизайн підкреслює досвід клініки («10 років досвіду») та основні переваги – індивідуальний підхід, новітнє обладнання і професіоналізм команди.

Загалом, практична реалізація проєкту демонструє здатність здобувача самостійно будувати сучасний багатофункціональний вебдодаток з використанням інструментів мови C#, фреймворку ASP.NET MVC, шаблонів програмування (репозиторій, MVC), а також засобів забезпечення безпеки та інтеграції зі сторонніми сервісами.

3.2 Тестування та налагодження системи

У процесі розробки вебдодатку для стоматологічного кабінету «LK Dental» було проведено тестування і налагодження всіх основних модулів системи з метою перевірки їх працездатності, стабільності та відповідності функціональним вимогам. Тестування здійснювалось у декілька етапів: ручне (модульне, інтеграційне) та функціональне, що охоплювало всі основні сценарії взаємодії з системою.

На етапі модульного тестування перевірялися окремі частини системи, зокрема контролери та сервіси. Було створено тестові виклики методів контролерів для перевірки коректної обробки запитів. Наприклад, перевірялися дії `AppointmentController.Create`, `DoctorController.Index`, `ServiceController.Edit`, щоб впевнитися в правильному збереженні об'єктів у базі даних.

Було також проведено інтеграційне тестування, тобто тестувалися взаємозв'язки між модулями – наприклад, правильність отримання списку лікарів у формі запису на прийом, робота LINQ-запитів з кількома включеними таблицями (`Include()`), перевірка виведення пов'язаних об'єктів на Razor-поданнях. Також перевірялися маршрутизація та доступ до сторінок за допомогою атрибутів `[Authorize]` та `[HttpPost]`.

Під час функціонального тестування було змодельовано типові дії користувача: заповнення форми запису до лікаря, перегляд списку послуг, надсилання відгуку тощо.

Кожна дія перевірялась на відповідність очікуваному результату. Для перевірки валідації було протестовано поведінку системи при введенні некоректних даних (наприклад, неправильний формат email, порожні поля).

Було також протестовано працездатність додатку на хостингу з сервером IIS Express, а також при розгортанні на віртуальному сервері з Windows Server 2019. Перевірялась коректність підключення до бази даних SQL Server, обробка помилок, стійкість до навантаження (одночасне надсилання кількох форм запису).

У процесі тестування було виявлено кілька помилок:

1) некоректна обробка дати та часу у формі запису – виправлено шляхом зміни формату валідації;

2) помилки в прив'язці моделей до Razor-форм – усунуто завдяки уточненню полів моделі Appointment;

3) недостатня фільтрація відгуків, що могли містити заборонені слова – реалізовано фільтр стоп-слів у сервісі відгуків;

4) проблеми з відображенням інтерфейсу на мобільних пристроях – адаптовано шаблон Bootstrap для mobile-first.

Для успішної роботи вебдодатку «LK Dental» необхідні наступні параметри. На стороні сервера:

1) операційна система: Windows 10/11 або Windows Server 2016+;

2) вебсервер: IIS 10.0 або вище;

3) .NET SDK: версія 6.0 або 7.0;

4) MS SQL Server 2019 або вище;

5) ОЗП: від 4 ГБ;

6) місце на диску: від 2 ГБ.

На стороні клієнта:

1) сучасний веббраузер (Chrome, Edge, Firefox);

- 2) доступ до інтернету (для віддаленої роботи з сервером);
- 3) роздільна здатність екрана від 1024×768.

У результаті багаторівневого тестування було забезпечено стабільну роботу вебдодатку згідно з вимогами, визначеними на етапі проєктування. Виявлені помилки були оперативно усунені. Отримані результати підтверджують надійність і функціональну повноцінність системи при роботі в різних умовах і з різними категоріями користувачів.

3.3 Розробка бази даних

У рамках розробки вебдодатку для стоматологічного кабінету було створено базу даних, яка забезпечує зберігання, доступ і обробку основної інформації: про лікарів, послуги, відгуки, категорії, користувачів, блоги тощо. Розробка бази даних здійснювалася за допомогою візуального конструктора Dentist.dbml у середовищі Visual Studio, що реалізує технологію LINQ to SQL, яка автоматично генерує SQL-команди для роботи з базою даних Microsoft SQL Server.

Модель .dbml охоплює такі таблиці:

- 1) Doctor – таблиця лікарів, що зберігає їхні ПІБ, контактні дані, типи спеціалізації, текстові описи, посилання на соціальні мережі;
- 2) DoctorType – типи лікарів (наприклад, терапевт, хірург);
- 3) Response – зберігає відгуки пацієнтів, їхній текст, ім'я автора, дату, статус публікації;
- 4) Blog – публікації блогу з текстовим контентом, заголовком, датою;
- 5) Categories – категорії послуг;
- 6) Price – таблиця з переліком послуг і відповідних цін;
- 7) User – таблиця авторизованих користувачів системи (використовується в Identity Framework).

Кожна таблиця має первинний ключ id і зв'язки між собою (наприклад, Doctor має зовнішній ключ typeid, який посиляється на DoctorType).

Для виконання операцій вибірки, вставки, оновлення або видалення даних використані наступні запити:

- 1) отримання лікарів певного типу (ліст. 3.6);

Лістинг 3.6 – Отримання лікарів за типом

```
SELECT * FROM Doctor
INNER JOIN DoctorType ON Doctor.typeid = DoctorType.id
WHERE DoctorType.value = 'Ортодонт';
```

кінець лістингу 3.6

- 2) додавання нового відгуку (ліст. 3.7);

Лістинг 3.7 – Додавання відгуку

```
INSERT INTO Response (name, text, categoryId, status, date)
VALUES ('Іван', 'Дуже задоволений сервісом!', 3, 1, GETDATE());
```

кінець лістингу 3.7

- 2) оновлення ціни на послугу (ліст. 3.8);

Лістинг 3.8 – Оновлення ціни

```
UPDATE Price
SET value = 1200
WHERE id = 5;
```

кінець лістингу 3.8

- 3) видалення старого блогу (ліст. 3.9).

Лістинг 3.9 – Видалення блогу

```
DELETE FROM Blog
WHERE id = 10;
```

кінець лістингу 3.9

База даних розгорталася з використанням:

- 1) MS SQL Server – серверна СКБД;

2) Visual Studio + LINQ to SQL – для проєктування моделі даних та генерації коду доступу;

3) SQL Server Object Explorer – для перегляду вмісту та виконання ручних SQL-запитів;

4) Dbml-файл (Dentist.dbml) – візуальний редактор сутностей.

Цей підхід поєднує зручність ORM (Object-Relational Mapping) для C# з можливістю виконання складних SQL-запитів вручну, коли це необхідно. Він дозволяє легко масштабувати структуру бази, забезпечує швидке прототипування та інтеграцію із середовищем ASP.NET MVC.

3.4 Захист інформаційно-комп'ютерної системи

У розробленому вебдодатку стоматологічного центру реалізовано базові, але ефективні механізми захисту інформаційно-комп'ютерної системи. Система побудована на ASP.NET MVC з використанням стандартних засобів авторизації, автентифікації та захисту від поширених вебзагроз.

Для керування обліковими записами користувачів застосовується ASP.NET Identity. Всі дії, пов'язані з персональними даними або адміністративними функціями, захищено за допомогою атрибута [Authorize], що дозволяє обмежити доступ лише авторизованим користувачам.

Процес входу в систему реалізовано через SignInManager.PasswordSignInAsync, який виконує перевірку облікових даних користувача. Крім того, підтримується двофакторна автентифікація (2FA), що суттєво підвищує рівень безпеки доступу.

Паролі користувачів не зберігаються у відкритому вигляді. ASP.NET Identity автоматично хешує паролі за допомогою алгоритму PBKDF2 з сіллю. Це забезпечує захист від атак типу «перебору» та «rainbow tables». Формат хешу не розкривається, але реалізація відповідає сучасним рекомендаціям OWASP.

Для запобігання атакам типу XSS (міжсайтове скриптування) використовується бібліотека `HtmlSanitizer`, яка очищує HTML-текст у полях введення користувача, зокрема у відгуках і коментарях. Це дозволяє безпечно відображати введені дані у вигляді HTML без ризику виконання шкідливого коду.

Усі запити, пов'язані з автентифікацією, використовують токени CSRF-захисту, які впроваджено через атрибути `[ValidateAntiForgeryToken]`. Вони запобігають підробленим запитам, які могли б бути виконані від імені користувача без його згоди.

Передача повідомлень на сервер Telegram при замовленні послуги також відбувається через HTTPS-з'єднання, що гарантує шифрування даних у каналі передачі.

Користувачі неавторизовані не можуть отримати доступ до захищених ресурсів, зокрема панелі адміністрування CMS. Також реалізовано маршрутизацію, що чітко обмежує доступ лише до відповідних контролерів згідно з роллю користувача.

Висновки до розділу 3

У межах третього розділу було реалізовано практичну частину розробки вебдодатку для стоматологічного кабінету на основі архітектури ASP.NET MVC. Відповідно до поставлених завдань, здійснено розробку структури проєкту, побудовано компоненти, що відповідають за логіку відображення, обробки запитів і зберігання даних. Було створено функціональні модулі для відображення переліку лікарів, блогів, послуг, коментарів і відгуків із можливістю інтерактивної взаємодії.

На основі моделі LINQ to SQL (файл `Dentist.dbml`) реалізовано роботу з базою даних через ORM-рівень, що дозволило значно скоротити обсяг SQL-запитів і централізувати доступ до таблиць через репозиторії. Розроблено

основні класи-репозиторії для роботи з сутностями: Doctor, Blog, Category, Response, що забезпечує зручність супроводу та модифікації коду.

Здійснено повноцінне тестування системи, під час якого були перевірені основні функціональні сценарії (перегляд сторінок, надсилання заявки, додавання коментаря), налагоджено обробку можливих помилок і реалізовано захист на рівні валідаторів, HTML-фільтрації та перевірки CSRF-токенів.

У рамках розробки системи захисту реалізовано механізми авторизації та автентифікації користувачів, захист паролів із використанням ASP.NET Identity, а також інтеграцію з Telegram API для оперативного отримання повідомлень про заявки. Усі вхідні дані очищуються за допомогою HtmlSanitizer, що захищає вебдодаток від атак типу XSS.

Загалом, на основі сучасного стеку технологій було успішно реалізовано повноцінний вебдодаток, що відповідає потребам стоматологічної клініки та забезпечує безпечну, зручну і функціональну взаємодію з користувачем.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на тему «Розробка вебдодатку для стоматологічного кабінету з використанням ASP.NET MVC та базою даних MS SQL Server» було повністю реалізовано поставлену мету та досягнуто всі передбачені завдання. Розроблено повнофункціональну інформаційно-комп'ютерну систему, що дозволяє автоматизувати облік лікарів, пацієнтів, послуг, відгуків, категорій, а також забезпечує ефективну взаємодію між користувачами і адміністратором.

З урахуванням сучасних світових тенденцій у галузі цифрової трансформації медичних послуг, запропонований вебдодаток відповідає актуальним вимогам до зручності, безпеки та масштабованості. Система реалізована з використанням технологій ASP.NET MVC, LINQ to SQL, Razor, HTML5, CSS3, Bootstrap, JavaScript, jQuery, Magick.NET та Telegram API, що дозволяє забезпечити як ефективне керування контентом, так і інтерактивність користувацького інтерфейсу.

У ході аналізу сучасного стану проблеми було виявлено, що більшість наявних інформаційних систем для стоматологічної сфери орієнтовані на загальні медичні процеси, не враховуючи специфіку роботи приватного кабінету. Існуючі аналоги часто не мають зручного вебінтерфейсу або не підтримують гнучку взаємодію з пацієнтами. Загалом, обґрунтовано необхідність створення спеціалізованого вебрішення з оптимізованим функціоналом.

Під час дослідження вимог до інформаційної системи було визначено основні функції, якими повинен володіти вебдодаток, включаючи управління записами пацієнтів, публікаціями, зображеннями, зворотним зв'язком. Спроектовано загальну архітектуру, що базується на парадигмі MVC, та структуру бази даних, яка забезпечує підтримку основних бізнес-процесів у діяльності стоматологічного закладу.

Було проаналізовано можливі інструменти для розробки та обґрунтовано вибір технологічного стека ASP.NET MVC, MS SQL Server, LINQ to SQL, Razor, Bootstrap та допоміжних бібліотек. Обрані рішення забезпечують необхідну продуктивність, безпеку, підтримку розширення та відповідність сучасним стандартам веброботи в медичній галузі.

У процесі реалізації вебдодатку було створено багаторівневу систему з чіткою логікою розділення на представлення, контролери та моделі. За допомогою C# та Razor реалізовано динамічну генерацію сторінок, формування даних через репозиторії, підтримку зворотного зв'язку, інтеграцію з Telegram та обробку зображень через ImageMagick.

Для забезпечення працездатності та стабільності системи було проведено функціональне тестування компонентів, перевірено обробку помилок, валідацію форм, відповідність очікуваній поведінці. Система протестована на типових кейсах користувача, встановлено рекомендовані параметри серверного середовища, усунено незначні помилки, що були виявлені під час налагодження.

База даних проєкту була реалізована за допомогою LINQ to SQL на основі моделі .dbml, що забезпечило чітку відповідність між об'єктами мови C# та таблицями СУБД. Завдяки використанню Code First підходу, усі необхідні сутності були сформовані безпосередньо з коду, забезпечуючи зручність та узгодженість структури з логікою вебдодатку.

У реалізованій системі було впроваджено низку заходів захисту, включаючи автентифікацію користувачів, фільтрацію вхідних даних від XSS за допомогою HtmlSanitizer, параметризовані запити через ORM, обмеження доступу до адміністрування через механізми авторизації ASP.NET Identity. Також забезпечено зберігання зображень у безпечному форматі та оперативну взаємодію з адміністратором через захищене API Telegram.

Результати роботи можуть бути впроваджені в діяльність малих і середніх стоматологічних кабінетів для підвищення якості обслуговування, зменшення навантаження на адміністративний персонал, та оперативного реагування на

звернення пацієнтів. Система підтримує можливість розширення функціоналу, включно з модулем онлайн-запису, статистичним обліком, CRM-модулем.

Розробка частково базується на наукових і практичних дослідженнях, що проводяться в межах діяльності кафедри інженерії програмного забезпечення Луцького національного технічного університету. Під час виконання роботи було отримано нові результати, які можуть бути основою для написання наукових публікацій або практичних методичних рекомендацій щодо автоматизації стоматологічних послуг.

Розроблений додаток має народногосподарське та соціальне значення, оскільки сприяє покращенню доступу до медичних послуг, зменшенню черг, підвищенню зручності комунікації пацієнта з медичним закладом. У майбутньому передбачається вдосконалення системи за рахунок впровадження хмарного зберігання даних, адаптації до мобільних платформ, інтеграції з електронною системою охорони здоров'я України (eHealth).

Отже, кваліфікаційна робота має завершений, практично орієнтований характер і підтверджує здатність здобувача вищої освіти до самостійного розв'язання прикладних задач інженерії програмного забезпечення в медичній галузі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Artificial Intelligence Techniques: Analysis, Application, and Outcome in Dentistry—A Systematic Review / N. Ahmed et al. BioMed Research International. 2021. Vol. 2021. P. 1-15.
2. Ho S. B., Chew E. Y., Tan C. H. Streamlining dental clinic management for effective digitisation productivity and usability. Journal of Informatics and Web Engineering. 2024. Vol. 3, no. 2. P. 70-85.
3. Kozulia M., Soldatko O. Проектування інформаційної CRM-системи у сфері стоматологічного адміністрування. Bulletin of National Technical University" KhPI". Series: System Analysis, Control and Information Technologies. 2024. no. 2 (12). P. 38-41.
4. Гарматюк О., Горохівський А. Використання новітніх інформаційних технологій у сфері надання стоматологічних послуг. Збірник тез доповідей. 2025. С.13-14.
5. Білий В. В., Денисенко О. І. Система обліку клієнтів стоматології. Інформаційні технології: теорія і практика. Збірник тез доповідей. 2020. С. 30-31.
6. Чередниченко А. В., Чередниченко В. А. Цифровізація бізнес-процесів в стоматології. Редакційна колегія. 2024. С. 155-158.
7. Medical Billing Software : ClaimTek MedOffice Medical Billing Software. Medical Billing – Business Opportunity : ClaimTek Systems. URL: <https://library.claimtek.com/Software/Dental/> (date of access: 05.03.2025).
8. EasyPractice – The go-to platform for therapists and clinicians! URL: <https://easyppractice.net/> (date of access: 05.03.2025).
9. WebDental, LLC. URL: <https://webdental.com/> (date of access: 05.03.2025).
10. Rustagi P., Kumar Y. Mvc architecture and its application, 2022. 48 p.

11. ASP.NET MVC. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/uk-ua/aspnet/mvc/> (date of access: 14.03.2025).

12. Entity Framework Core 7 – Про Програмування. Про Програмування – ABout IT And Programming. URL: <https://abitap.com/category/entity-framework-core-7/> (дата звернення: 15.03.2025).

13. Microsoft SQL Server. URL: <https://www.microsoft.com/en-us/sql-server> (date of access: 15.03.2025).

14. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/> (date of access: 17.03.2025).

15. jQuery. URL: <https://jquery.com/> (date of access: 17.03.2025).