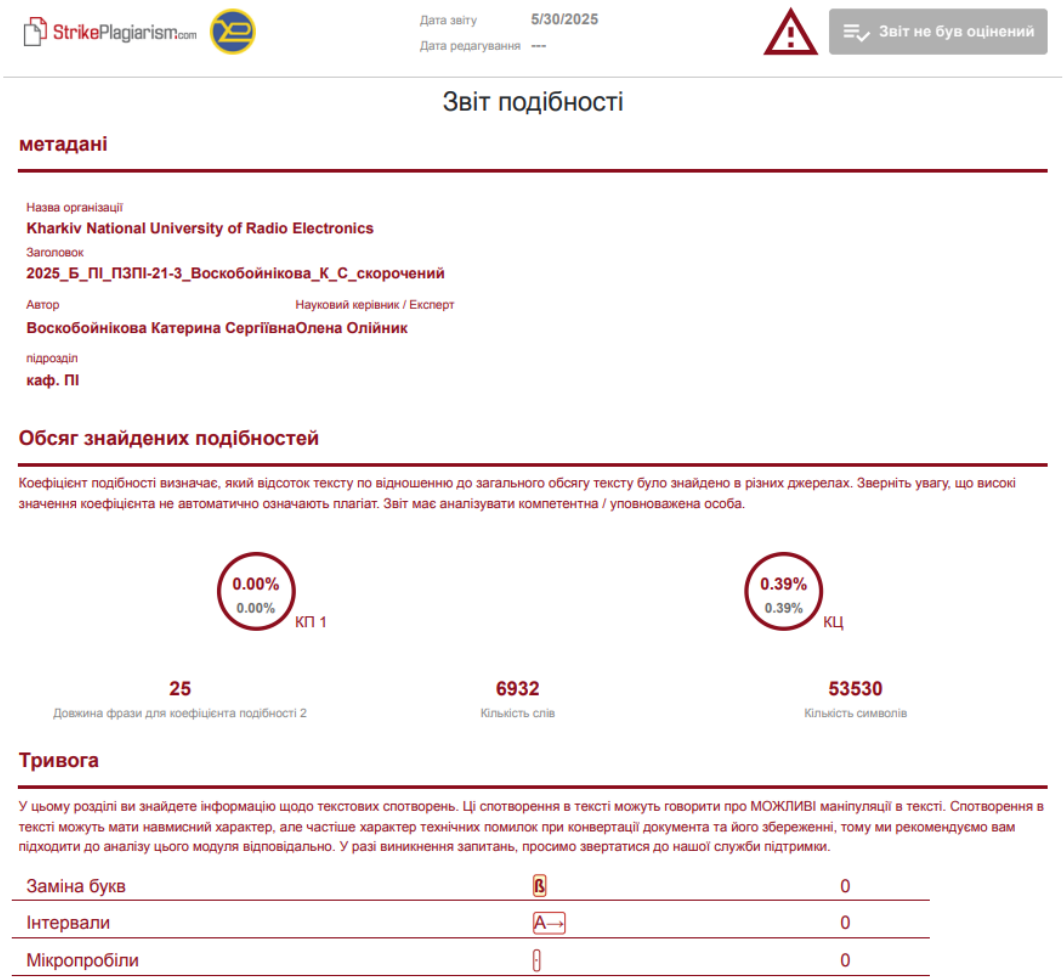
ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



StrikePlagiarism.com

Дата звіту 5/30/2025

Дата редагування ---

Звіт не був оцінений

Звіт подібності

метадані

Назва організації

Kharkiv National University of Radio Electronics

Заголовок

2025_Б_ПІ_ПЗПІ-21-3_Воскобойнікова_К_С_скорочений

Автор **Воскобойнікова Катерина Сергіївна**

Науковий керівник / Експерт **Олена Олійник**

підрозділ

каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

0.00% 0.00% КП 1

0.39% 0.39% КЦ

25 Довжина фрази для коефіцієнта подібності 2

6932 Кількість слів

53530 Кількість символів

Тривога

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

Заміна букв	R	0
Інтервали	A→	0
Мікропробіли	¶	0

Рисунок А.1 – Результат перевірки кваліфікаційної роботи бакалавра на плагіат

ДОДАТОК Б

Слайди презентації

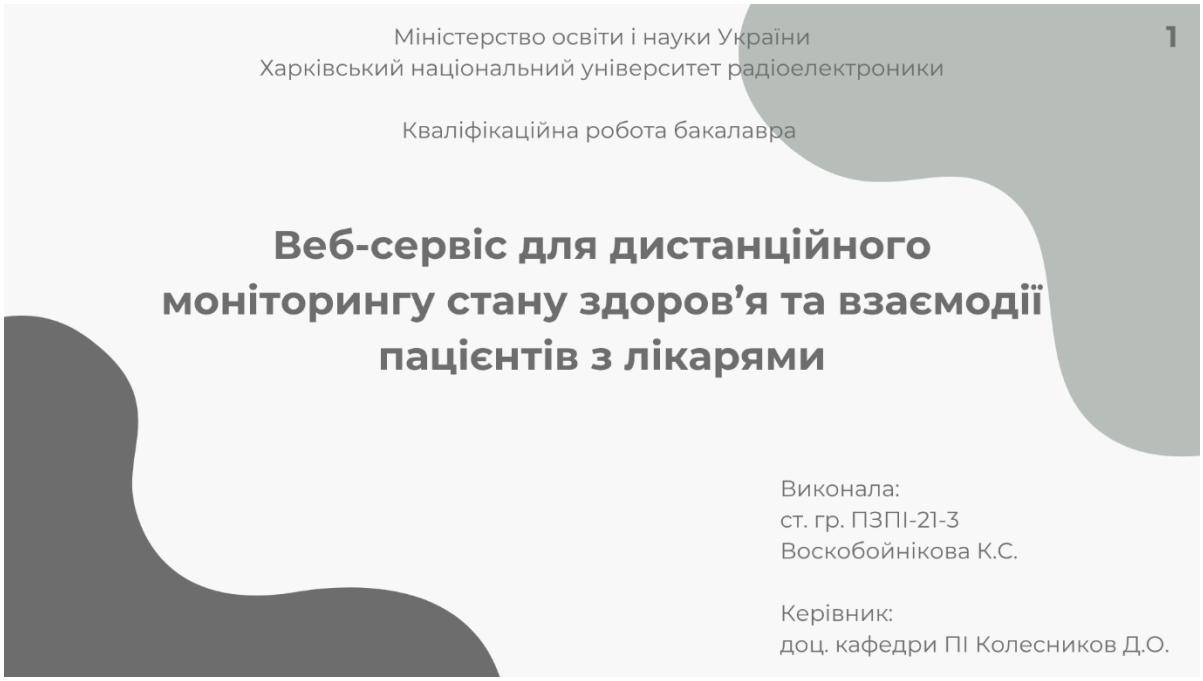


Рисунок Б.1 – Слайд 1

Рисунок Б.2 – Слайд 2

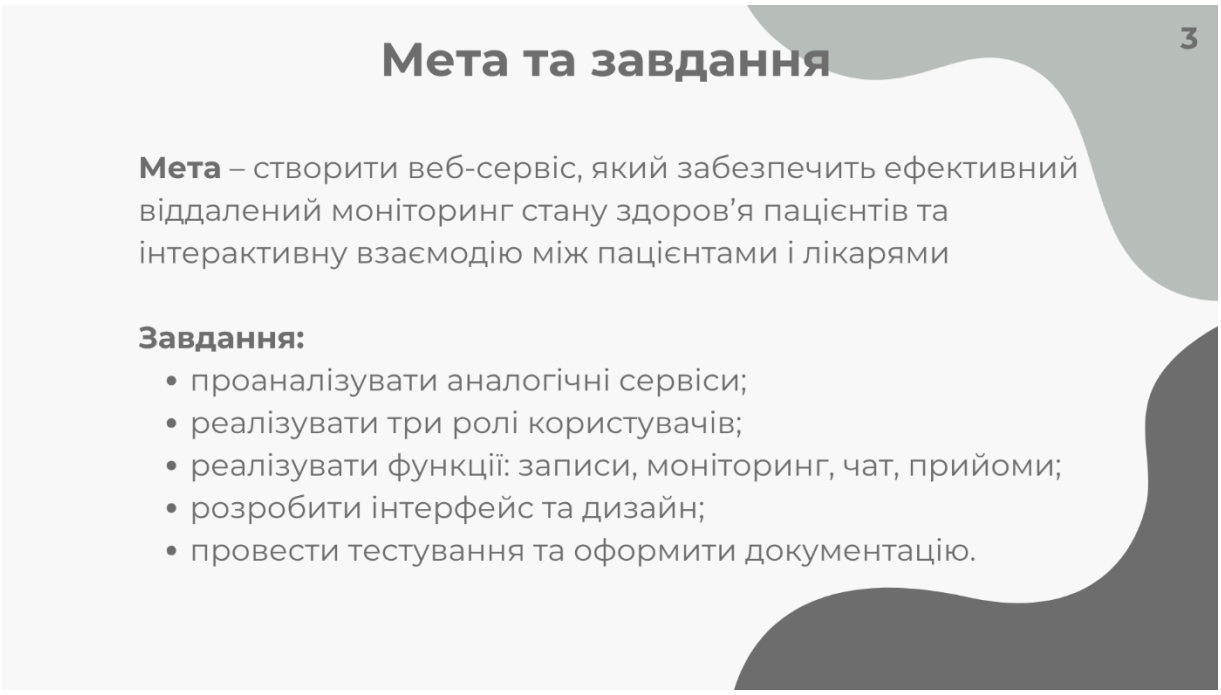


Рисунок Б.3 – Слайд 3

Рисунок Б.4 – Слайд 4

Рисунок Б.5 – Слайд 5

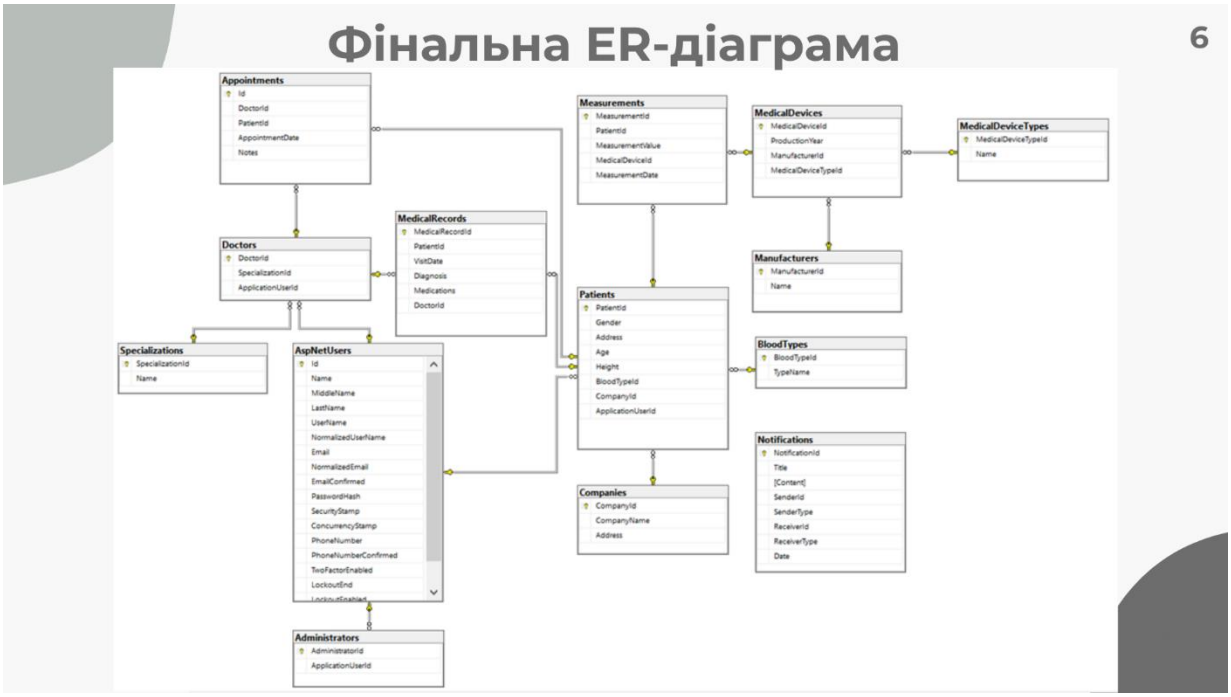


Рисунок Б.6 – Слайд 6

Рисунок Б.7 – Слайд 7

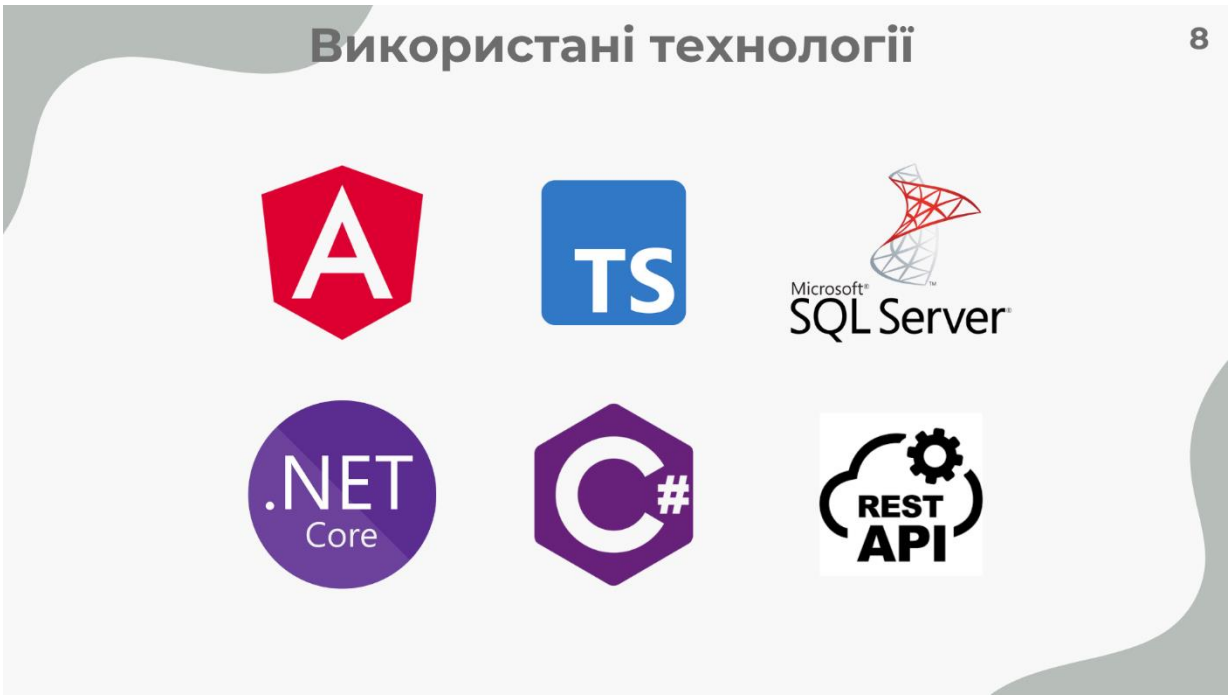


Рисунок Б.8 – Слайд 8

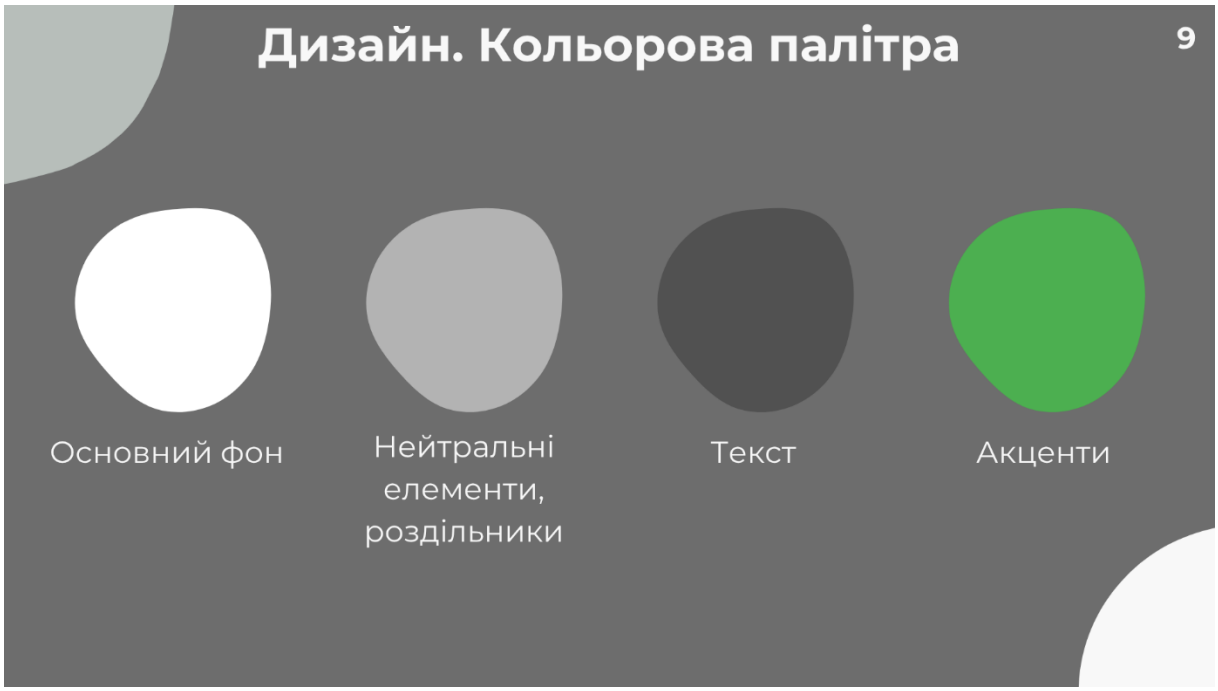


Рисунок Б.9 – Слайд 9

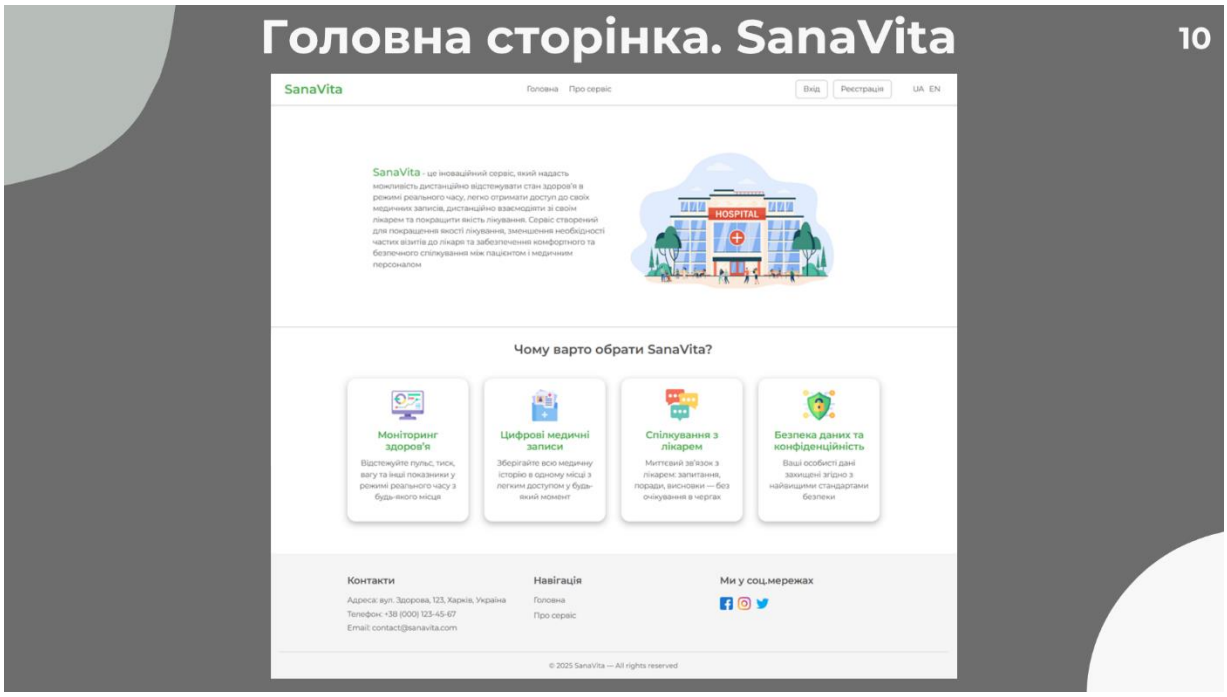


Рисунок Б.10 – Слайд 10

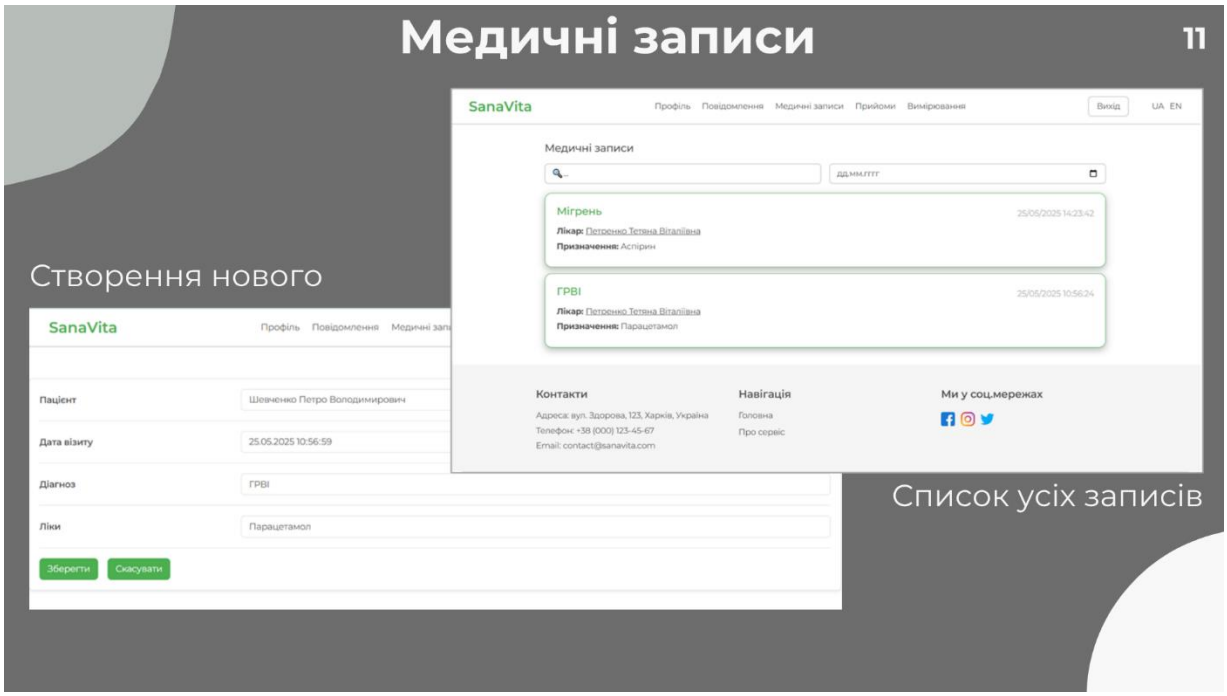


Рисунок Б.11 – Слайд 11

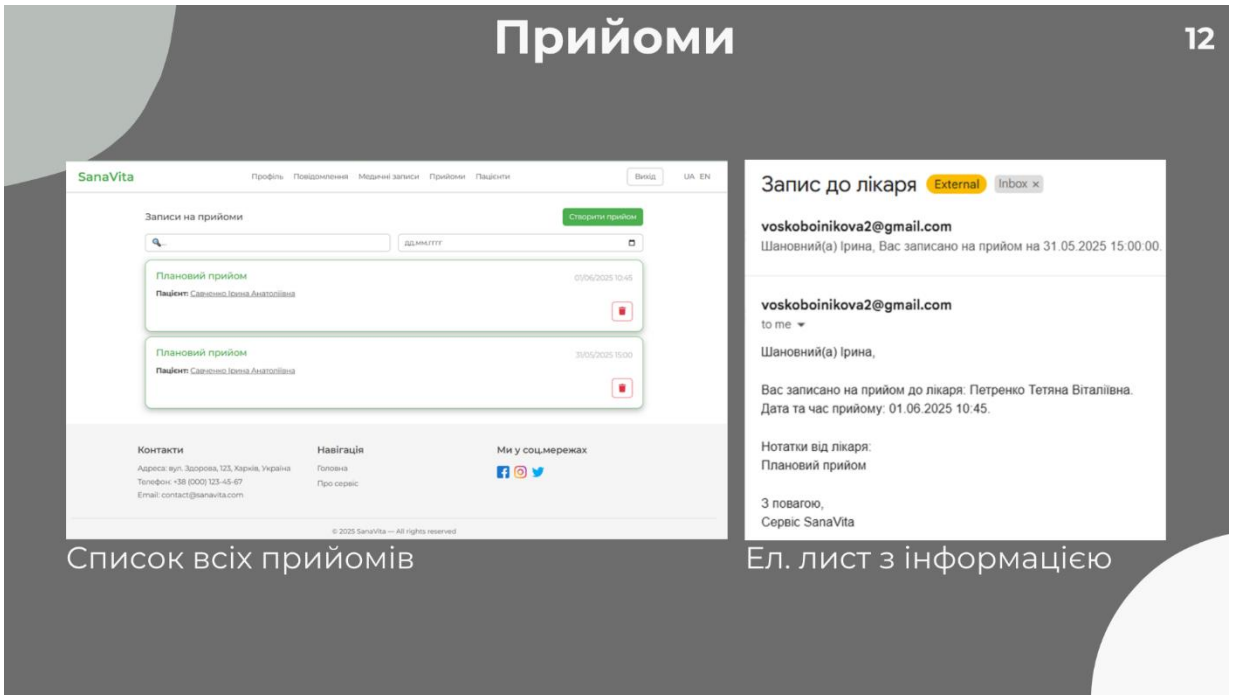


Рисунок Б.12 – Слайд 12

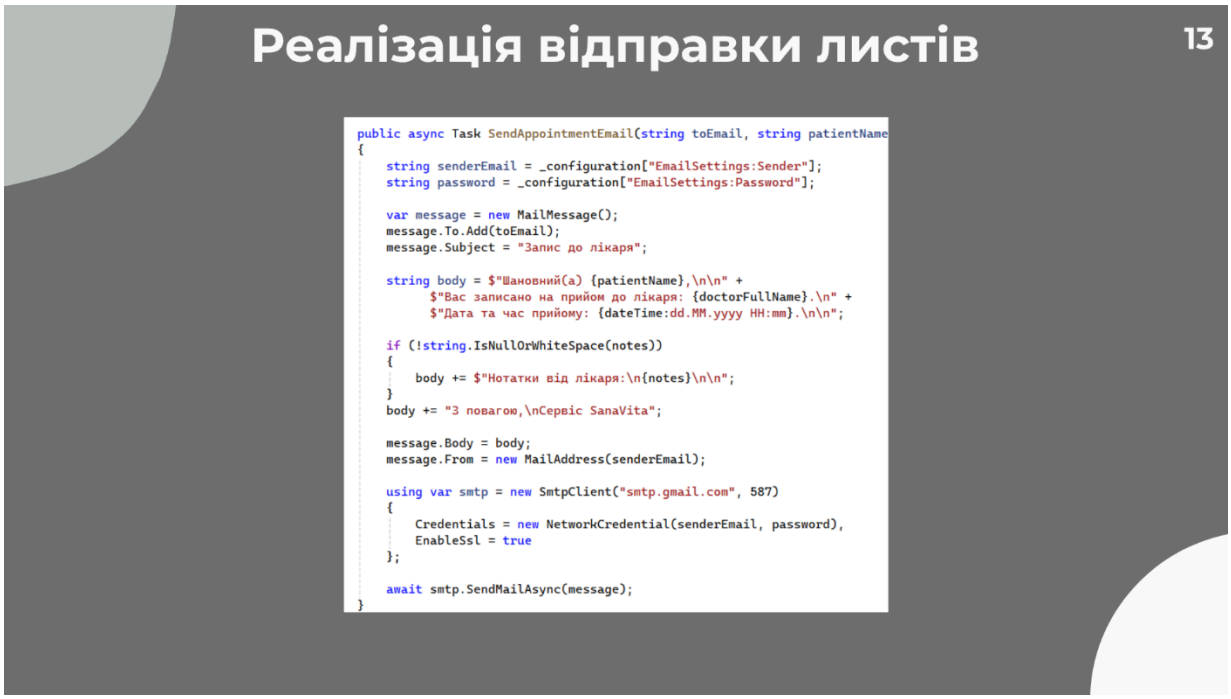


Рисунок Б.13 – Слайд 13

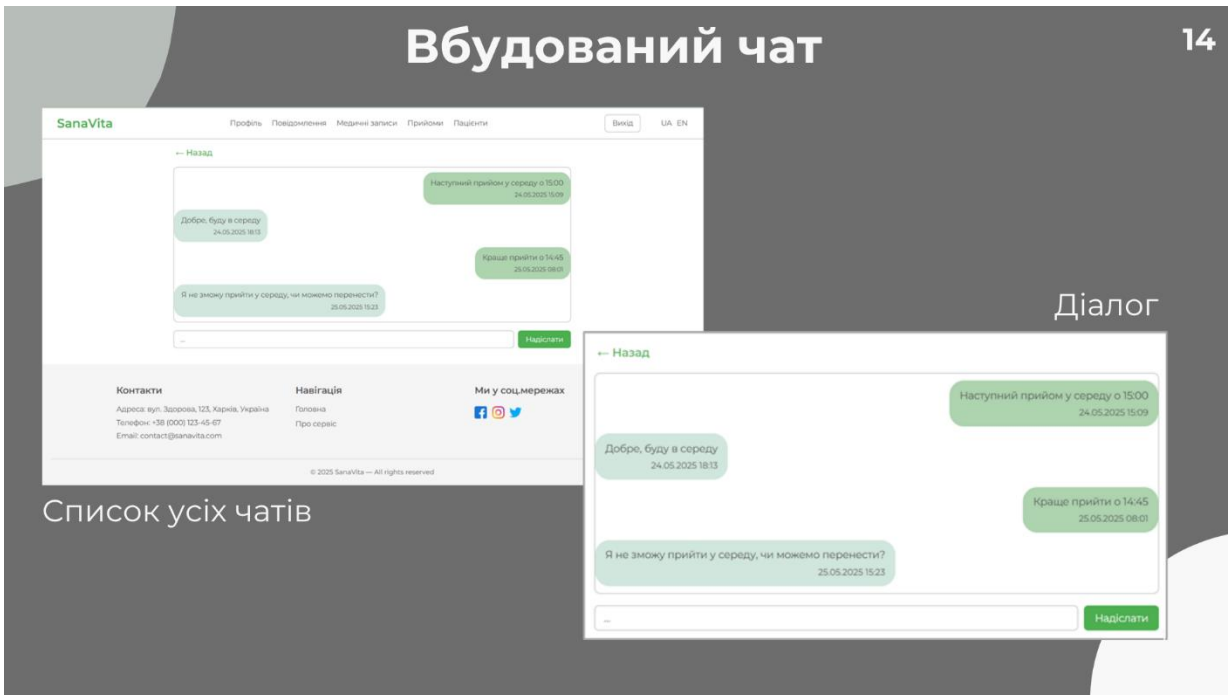


Рисунок Б.14 – Слайд 14

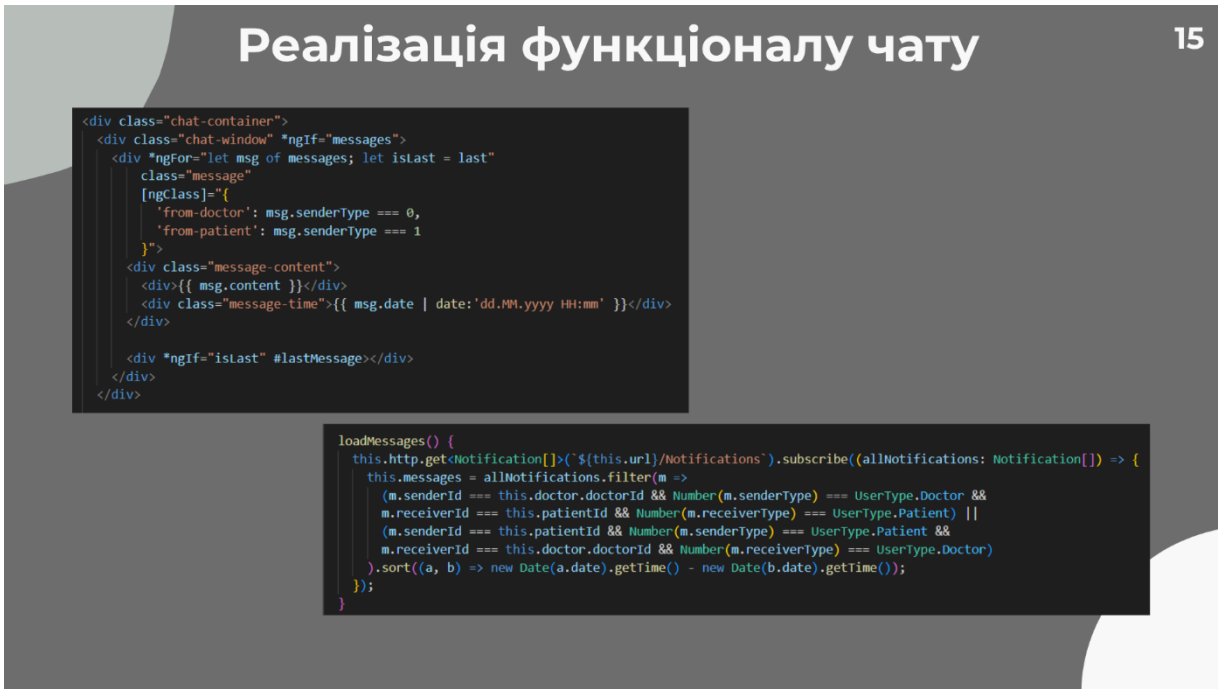


Рисунок Б.15 – Слайд 15

Рисунок Б.16 – Слайд 16

Рисунок Б.17 – Слайд 17

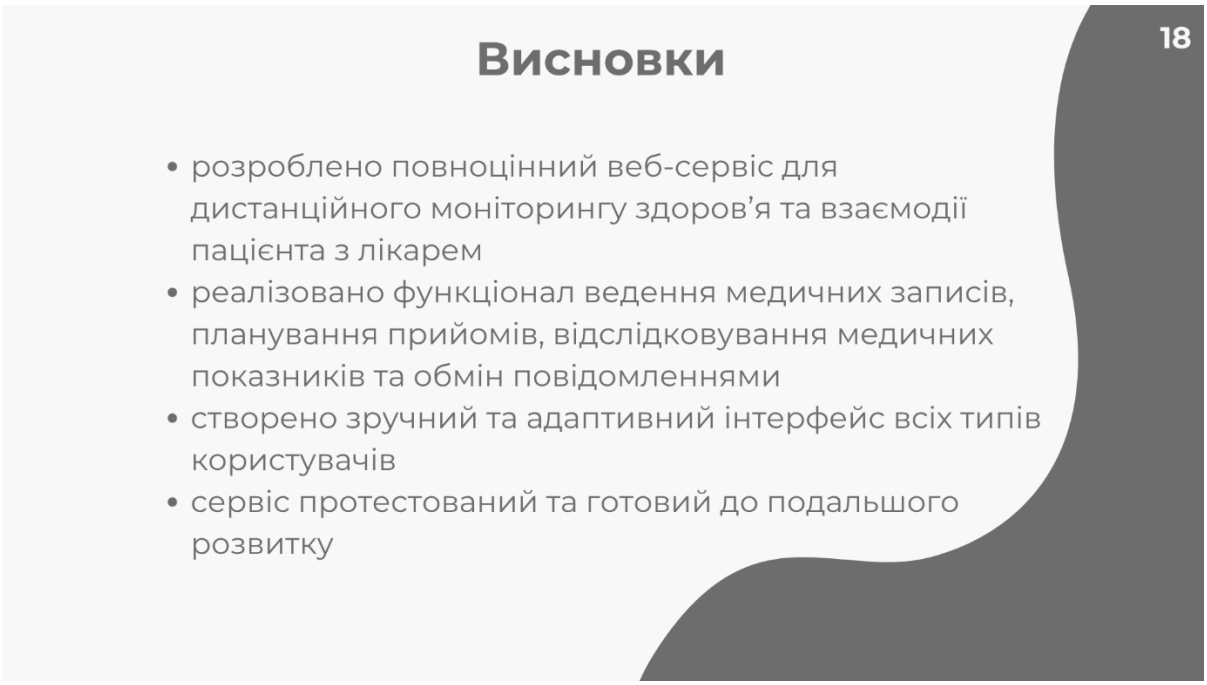


Рисунок Б.18 – Слайд 18

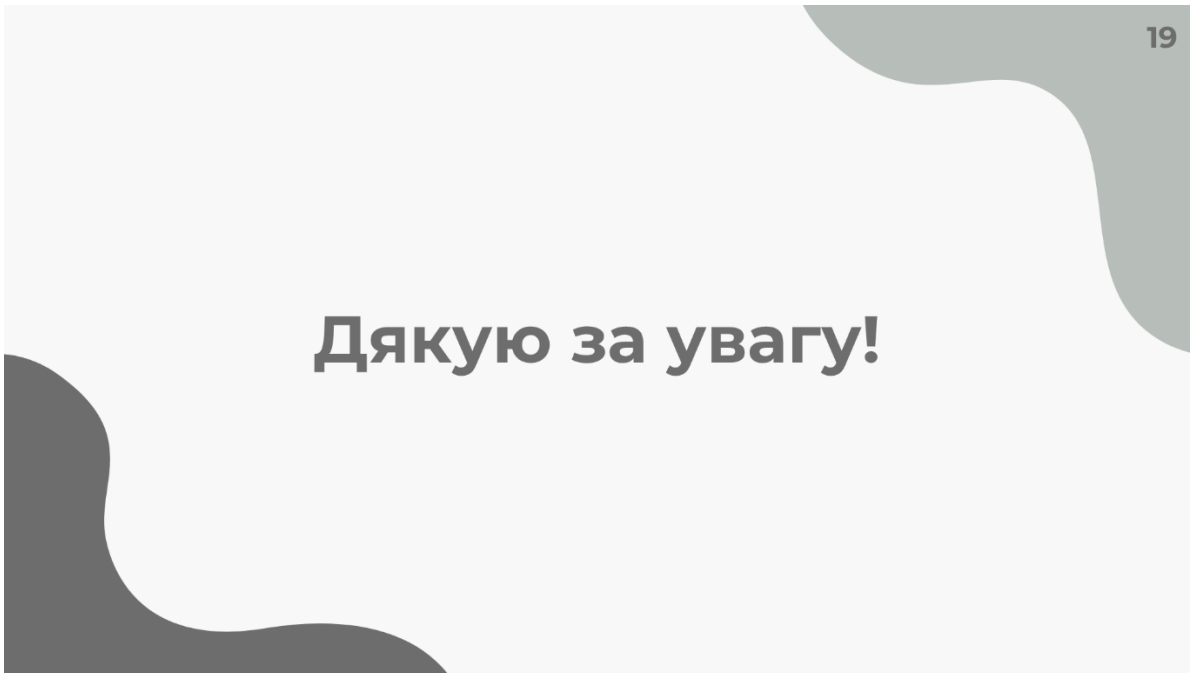


Рисунок Б.19 – Слайд 19

ДОДАТОК В

Специфікація вимог до програмного продукту

1 ВСТУП

1.1 Огляд продукту

У сучасному світі цифрових технологій, що постійно розвиваються, питання дистанційного моніторингу стану здоров’я стає особливо актуальним. Рішенням цього є створення сучасного веб-застосунку, що орієнтований на забезпечення безперервного та ефективного моніторингу життєво важливих показників здоров’я пацієнтів. Основна ідея полягає в тому, щоб надати користувачам, як пацієнтам, так і медичному персоналу, зручний і надійний інструмент для збору, зберігання та аналізу даних про стан здоров’я в режимі реального часу. Цей веб-застосунок має три рівні доступу: для адміністратора, лікаря та пацієнта. Адміністратор керує користувачами та структурою бази даних, забезпечуючи стабільну роботу системи. Лікарі можуть створювати та переглядати медичні записи, ставити діагнози, а також спілкуватися з пацієнтами через вбудовану систему повідомлень. Пацієнти, у свою чергу, мають можливість переглядати свої медичні показники в зручному вигляді, бачити медичні записи та залишати повідомлення своєму лікарю.

Архітектура побудована за принципом «клієнт-сервер»: фронтенд створено за допомогою Angular для забезпечення динамічного та адаптивного інтерфейсу, а серверна частина реалізована на ASP.NET Core з використанням C#, що забезпечує високу продуктивність і масштабованість. Суттєвою перевагою платформи є автоматичне отримання показників (таких як пульс, пульс, тощо) із використанням технологій Інтернету речей (IoT), що мінімізує потребу у ручному введенні даних.

Платформа може бути корисною як для медичних закладів (лікарень, клінік, амбулаторій), так і для приватних практикуючих лікарів, які ведуть своїх пацієнтів дистанційно.

1.2 Мета

Метою створення цієї системи є розробка інноваційного веб-застосунку, який дозволяє автоматизувати процес контролю за станом здоров’я пацієнтів у віддаленому режимі. Така система значно розширює можливості як лікарів, так і пацієнтів, оскільки усуває необхідність постійних фізичних візитів до медичних закладів, а також зменшує навантаження на персонал.

Основні цілі проєкту:

- підвищення якості медичних послуг завдяки оперативному доступу до даних про стан пацієнта;
- забезпечення безперервного моніторингу стану здоров’я навіть за межами лікарні;
- надання лікарю зручних засобів комунікації з пацієнтами та ведення їхніх медичних карток;
- спрощення взаємодії між усіма учасниками процесу лікування, включно з адміністрацією закладу;
- автоматичне отримання медичних показників, що дозволяє зменшити ризик людської помилки під час ручного введення даних.

Крім цього, система має на меті сформувати єдиний інформаційний простір між лікарем і пацієнтом. Всі дані зберігаються централізовано в захищеній базі, доступні в режимі реального часу, що підвищує загальний рівень обслуговування та прийняття рішень. Таким чином, проєкт вирішує одразу кілька важливих завдань в сфері eHealth: покращення доступності медичних послуг, підвищення ефективності лікування, збереження історії хвороб і взаємодії лікаря з пацієнтом у зручному цифровому форматі.

1.3 Межі

Незважаючи на широкі функціональні можливості системи, вона має певні обмеження, які обумовлені рамками проєкту, специфікою технологій та обраною архітектурою. Наприклад, система наразі працює з емульованими даними IoT-пристроїв, а не з реальними фізичними датчиками. Це означає, що інтеграція з реальними фітнес-браслетами, медичними моніторами або іншими смарт-пристроями потребує додаткових етапів розробки. Продукт у даний момент реалізований виключно як веб-застосунок, що означає знижену зручність використання на смартфонах у порівнянні з спеціальними мобільними застосунками. Також, хоча присутня функція повідомлень, система не реалізує відеодзвінки чи функції онлайн-прийому, вона не є діагностичною системою на основі штучного інтелекту і не замінює лікаря у прийнятті медичних рішень.

Ці межі не є недоліками, а радше орієнтирами для подальшого розвитку системи в наступних версіях.

1.4 Посилання

Проєкт тісно пов’язаний із сучасними тенденціями у сфері eHealth, цифрової трансформації медицини, а також використанням IoT у побуті. Платформа може бути зручно інтегрована в інфраструктуру приватних клінік або амбулаторій для ведення медичних записів пацієнтів та контролю їх стану. Особливо корисною вона може бути для пацієнтів з хронічними захворюваннями (наприклад, серцево-судинними хворобами або діабетом), для яких необхідне регулярне спостереження.

Система також може слугувати основою для навчання майбутніх медичних працівників у рамках симуляцій або демонстраційних моделей, де можна вивчати поведінку показників здоров’я в умовах наближених до реальних.

1.5 Означення та абревіатури

Backend – серверна логіка, база даних, API.

Frontend – клієнтська частина (веб-інтерфейс).

IoT – Інтернет речей, тобто мережа пристроїв, що можуть передавати дані через Інтернет без участі людини.

API – інтерфейс для взаємодії клієнта з сервером.

ORM – об’єктно-реляційне відображення бази даних.

HTTP – протокол для обміну даними між клієнтом і сервером.

CRUD – базові операції з даними: створення, читання, оновлення, видалення.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Ідея створення цього веб-застосунку виникла як відповідь на зростаючу потребу в ефективному контролі за станом здоров’я в умовах віддаленого медичного обслуговування. У майбутньому цей продукт може трансформуватися в повноцінну телемедичну платформу з відео консультаціями, автоматичними сповіщеннями про критичні показники, інтеграцією з державними електронними медичними реєстрами, а також аналітикою, заснованою на AI.

Розроблене програмне забезпечення має чітку архітектурну структуру, що дозволяє легко масштабувати його функціональність. Надалі можливе впровадження модулів:

– прямої інтеграції з реальними медичними пристроями;
– мобільного застосунку з push-сповіщеннями;
– онлайн-консультацій через відеозв’язок;
– аналітики стану здоров’я з використанням AI/ML.

Таким чином, веб-застосунок закладає фундамент для довготривалого розвитку, залишаючись при цьому гнучким і масштабованим.

2.2 Функції продукту

Система підтримує широкий спектр функцій, поділених відповідно до ролей користувачів. Кожна роль має свій набір прав і доступу до окремих розділів.

Для пацієнта:

– перегляд особистих медичних показників (пульс, вага, температура) у вигляді таблиць і графіків;
– отримання медичних записів і діагнозів від лікаря;
– надсилання повідомлень лікарю через вбудовану систему спілкування;
– перегляд історії повідомлень і відповідей;

– доступ до медичної картки зі всіма записами.

Для лікаря:

– перегляд списку пацієнтів, з якими він працює;
– створення та редагування медичних записів;
– аналіз показників пацієнтів із візуалізацією даних;
– надсилання повідомлень пацієнтам;
– контроль за динамікою стану здоров’я.

Для адміністратора:

– додавання, редагування та видалення користувачів (пацієнтів і лікарів);
– управління структурою бази даних;
– забезпечення безперебійної роботи сервісу.

Загалом, функціональність дозволяє створити цілісний медичний процес, від збору даних до консультацій, без фізичної присутності.

2.3 Характеристики користувачів

Система орієнтована на три основні групи користувачів, кожна з яких має свої потреби та особливості взаємодії із застосунком.

1. Пацієнти.

Зазвичай це особи, які потребують регулярного контролю за станом здоров’я. Більшість із них – це люди середнього та старшого віку. Тому інтерфейс для пацієнта створено максимально простим, зручним і інтуїтивно зрозумілим. Головна мета – дати користувачу змогу швидко знайти свої показники та зв’язатися з лікарем у кілька кліків.

2. Лікарі.

Це спеціалісти, які ведуть пацієнтів у рамках приватної або державної медичної практики. Їм важливо мати швидкий доступ до аналітики, історії змін показників та медичних записів. Інтерфейс лікаря орієнтований на швидке введення інформації, аналіз та взаємодію з пацієнтами через повідомлення.

3. Адміністратор.

Це технічний користувач, який не займається медичною діяльністю, а відповідає за підтримку системи. Для нього важливі стабільність, безпека та цілісність бази даних. В інтерфейсі адміністратора зосереджено засоби для керування користувачами та підтримки логічної структури застосунку.

2.4 Загальні обмеження

Загальні обмеження системи:

- підтримуються сучасні браузери (Chrome, Edge, Firefox);
- сервіс функціонує тільки у наявності інтернет-з’єднання;
- відсутність офлайн-режиму;
- відсутність функцій авторизації через сторонні сервіси, наприклад, реєстрація через Google або Facebook поки не підтримується;
- всі дані зберігаються на сервері; резервне копіювання має виконуватися вручну.

2.5 Припущення й залежності

Під час створення системи було зроблено декілька технічних і логічних припущень, що впливають на її роботу:

- користувачі мають стабільний доступ до Інтернету;
- сервер розміщено на підтримуваній платформі (.NET-сумісний хостинг);
- користувачі ознайомлені з основами комп’ютерної безпеки;
- показники, введені через пристрої пацієнта, вважаються коректними, клінічна валідація даних – це відповідальність лікаря.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейс користувача є критичним елементом системи, оскільки саме він забезпечує зручну та ефективну взаємодію між користувачем і програмним продуктом. Для його реалізації було використано Angular – сучасний фреймворк, що дозволяє створювати динамічні односторінкові додатки з високою продуктивністю. Залежно від ролі користувача, інтерфейс змінює свій вигляд та доступні функції, що дозволяє уникнути перевантаження інформацією та забезпечити інтуїтивну взаємодію. Наприклад, пацієнти бачать тільки інформацію про свій стан здоров’я, тоді як лікарі мають доступ до медичних карток пацієнтів, а адміністратори – до управління всією системою.

Розробка дизайну відбувалася відповідно до принципів UX/UI, із застосуванням адаптивної верстки, що забезпечує зручність користування як на десктопах, так і на мобільних пристроях. Усі форми мають валідацію, зручні підказки та повідомлення про помилки. Ієрархія інтерфейсу дотримується чітко: є панель навігації для переходу між основними розділами та основна область контенту, в якій виконується основна взаємодія.

3.1.2 Апаратний інтерфейс

У рамках цього програмного продукту немає прямої взаємодії з апаратними компонентами в класичному розумінні. Система працює із даними, які передаються через мережу, зокрема імітацію пристроїв IoT. Дані, які зазвичай надходили б із пульсометрів, ваг або оксиметрів, у цьому випадку передаються за допомогою емуляторів. Такий підхід дозволяє зосередитися на логіці обробки інформації без залежності від фізичних пристроїв. У разі потреби систему можна легко адаптувати для інтеграції з реальними сенсорами через відповідні API.

3.1.3 Програмний інтерфейс

Вся серверна логіка реалізована з використанням REST API, що надає стандартизований механізм взаємодії між клієнтською та серверною частинами. Комунікація відбувається через HTTP-запити, а дані передаються у форматі JSON. Це дозволяє легко масштабувати систему або інтегрувати її з іншими сервісами в майбутньому. Усі запити проходять перевірку на автентифікацію, яка реалізована за допомогою JWT-токенів. Клієнт, побудований на Angular із використанням TypeScript, виконує запити до API, отримує дані та відображає їх у зручному вигляді для користувача. Таким чином досягається гнучкість, безпека та підтримуваність системи.

3.1.4 Комунікаційний протокол

Передача даних між клієнтом і сервером здійснюється з використанням протоколу HTTPS, що гарантує шифрування трафіку та захист інформації від перехоплення чи стороннього втручання. У роботі з API застосовуються основні методи HTTP: GET для отримання даних, POST для створення, PUT для оновлення та DELETE для видалення. Це дозволяє реалізувати повноцінний цикл CRUD-операцій та забезпечити логічну і послідовну взаємодію з базою даних.

3.1.5 Обмеження пам’яті

Клієнтська частина не вимагає значних ресурсів для своєї роботи, що дозволяє запускати її навіть на малопотужних пристроях. Зі сторони сервера система спроектована таким чином, щоб обробляти великі об’єми даних. Зокрема, сервер повинен бути здатен підтримувати не менше 10 000 записів у базі даних без погіршення продуктивності. У разі зростання кількості користувачів можна передбачити масштабування на рівні бази даних та серверної інфраструктури.

3.1.6 Операції

Для кожного типу користувача визначений набір функцій. Пацієнти можуть переглядати власну медичну картку, а також вводити показники свого стану здоров’я. Лікарі мають змогу створювати нові записи, переглядати історію пацієнтів, а також надсилати повідомлення. Адміністратор здійснює контроль за всією системою, має можливість редагувати дані, блокувати або розблоковувати облікові записи та стежити за стабільністю роботи сервісу.

3.2 Властивості програмного продукту

Програмний продукт характеризується стабільною роботою в сучасних умовах, відповідає вимогам до продуктивності та має гнучку архітектуру. Завдяки застосуванню Angular для клієнтської частини досягається висока швидкість реакції інтерфейсу та ефективне використання ресурсів. Архітектура системи модульна, що дає змогу вносити зміни та доповнення без потреби змінювати основний код. Продукт підтримує актуальні браузери (Chrome, Firefox, Edge, Safari), а також має адаптивний дизайн, що дозволяє зручно працювати з системою на різних типах пристроїв, від комп’ютерів до смартфонів. Також система побудована з урахуванням можливості масштабування, що дозволить підтримувати більшу кількість користувачів у майбутньому.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Система має вбудовані механізми для обробки помилок як на клієнтській, так і на серверній стороні. Всі вхідні дані проходять перевірку, щоб уникнути некоректного введення. У серверній частині використано конструкції try-catch, які

дозволяють обробляти винятки та запобігати падінню застосунку при виникненні помилок.

3.3.2 Доступність

Сервіс розроблений таким чином, щоб бути доступним користувачам цілодобово. Якщо хостинг працює стабільно, то жодних обмежень у доступі не існує. Єдина можливість обмеження доступу – це втручання адміністратора, який може тимчасово деактивувати акаунт або обмежити доступ у разі порушень політики використання.

3.3.3 Безпека

Уся система побудована з урахуванням безпеки. Для захисту особистих даних користувачів реалізовано автентифікацію та авторизацію, права доступу чітко розмежовані відповідно до ролі. Паролі зберігаються в хешованому вигляді. Також передбачено захист від SQL-ін’єкцій та XSS-атак, а весь трафік зашифровано через HTTPS.

3.3.4 Супроводжуваність

Код програми організований у вигляді модулів і компонентів, що значно спрощує його підтримку та розширення. Дотримання принципів чистого коду та наявність внутрішньої документації дозволяє іншим розробникам швидко орієнтуватися у структурі проекту. За потреби можна легко додати нові функції або змінити існуючі, не порушуючи роботу всієї системи.

3.3.5 Переносимість

Проєкт легко переноситься на різні середовища. Його можна розгорнути на будь-якому сервері, який підтримує .NET та Microsoft SQL Server. Крім того, можлива контейнеризація системи з використанням Docker, що дозволяє запускати її на різних платформах, у тому числі у хмарних середовищах таких як Azure або AWS.

3.3.6 Продуктивність

Оптимізація SQL-запитів, кешування результатів, використання асинхронних запитів та ефективне управління станом клієнта забезпечують швидку реакцію системи. У середньому відповідь на запит надходить протягом 1-2 секунд навіть за умов стандартного навантаження. Це робить систему зручною для щоденного використання.

3.4 Вимоги до бази даних

База даних розроблена з використанням Microsoft SQL Server. При її проєктуванні було дотримано принципів нормалізації, зокрема 3-ї нормальної форми, що дозволяє уникнути дублювання даних і забезпечити логічну структуру зберігання. Структура бази передбачає використання зовнішніх ключів, що гарантує цілісність даних. Також реалізовано збереження історичних змін, щоб мати можливість відстежувати попередні значення у разі потреби. Усі таблиці мають індекси на ключових полях, що забезпечує швидкий доступ до даних.

3.5 Інші вимоги

Серед інших вимог, що забезпечують повноцінне функціонування програмного продукту, варто відзначити необхідність надання користувачам детальної інструкції з користування. Також передбачено проведення модульного тестування кожного з компонентів. Вся серверна частина супроводжується документацією API, що дозволяє легко інтегруватися з іншими системами або проводити аудит. Уся система відповідає сучасним стандартам UI/UX, що позитивно впливає на взаємодію користувача з продуктом. Також важливою вимогою є відповідність GDPR – загальному регламенту захисту персональних даних, що гарантує конфіденційність та безпеку для кожного користувача.