

Міністерство освіти і науки України
Дрогобицький державний педагогічний університет імені Івана Франка
кафедра фізики та інформаційних систем

«До захисту допускаю»

завідувач кафедри фізики та інформаційних систем

кандидат фізико-математичних наук, доцент

_____ Віталій ГОЛЬСЬКИЙ «____» ____ 2024 р.

Розробка веб-сервісу для збереження персональної медичної історії

Спеціальність: 122 «Комп'ютерні науки»

Магістерська робота
на здобуття кваліфікації – «Магістр з комп'ютерних наук»

Автор роботи: Любас Ольга Андріївна _____
підпис

Науковий керівник: кандидат фізико-математичних наук,
доцент кафедри фізики та інформаційних систем
Лешко Роман Ярославович _____
підпис

Дрогобич 2024

Захист кваліфікаційної (бакалаврської, магістерської) роботи

(тема роботи)

Оцінка за стобальною шкалою: _____

Оцінка за національною чотирибальною шкалою: _____

Коротка мотивація захисту:

Голова ЕК _____

Дата

підпис

прізвище, ім'я

Секретар ЕК _____

Дата

підпис

прізвище, ім'я

АНОТАЦІЯ

Любас О.А. Веб-сервіс для збереження персональної медичної історії. Магістерська робота, Дрогобицький державний педагогічний університет імені Івана Франка, Дрогобич, 2024 р. До захисту подано магістерську роботу, у якій досліджено особливості збереження медичної історії пацієнтів з акцентом на самостійне зберігання і контролювання даних. Проведено аналіз доступних схожих сервісів, переваги веб-сервісу BodyBio і його актуальність. Надано шляхи подальшого розвитку веб-сервісу.

Ключові слова: веб-сайт, веб-сервіс, медична історія пацієнта, медицина.

ANNOTATION

Olha Liubas. Web service for saving personal medical history. Master's Thesis, Drohobych Ivan Franko State Pedagogical University, Drohobych, 2024. This master's thesis presents a study on the features of storing patients' medical histories, emphasizing data saving and management by patient. The analysis includes an overview of similar existing services, the advantages of the BodyBio web service, and its relevance. The thesis also outlines potential directions for development in the future.

Keywords: website, web service, patient medical history, medicine.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1.Важливість сервісу BodyBio.....	8
1.1.Огляд функціоналу BodyBio.....	8
1.2.Порівняння з аналогами.....	10
1.3.Переваги BodyBio над аналогами.....	12
РОЗДІЛ 2.Цільова аудиторія і функціонал BodyBio.....	13
2.1.Цільова аудиторія BodyBio.....	13
2.2.Юзер сторі користувача сайту.....	16
РОЗДІЛ 3.Технологічний стек BodyBio.....	20
3.1.Опис бекенду BodyBio.....	22
3.2.Опис фронтенду BodyBio.....	27
ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	33

ВСТУП

В сучасному світі зберігання медичної інформації є доволі складним процесом. В нашій країні зазвичай вся відповідальність зазвичай покладається на лікаря. Проте часто трапляється, що лікарі заповнюють паперові історії хвороби чи навіть не заповнюють історій хвороби зовсім. Приватні медичні установи мають свої сайти, держава пропонує близько десяти різних варіантів сайтів для запису хворого, адміністрування клінік і лікарень та зберігання медичної історії пацієнтів.

Протягом всієї нашої історії українці зіштовхувались з еміграцією. Деякі з громадян України і до війни жили на дві країни, повертаючись додому для залагодження справ, але проживали і працювали в ближніх країнах Європи. Після повномасштабного вторгнення відсоток українських емігрантів у всьому світі значно збільшився. Кожна країна має свої способи збереження інформації про історію хвороби подібно до України .

В результаті пацієнт зіштовхується з ситуацією, коли його історія хвороби зберігається у вигляді окремих файлів, записів або взагалі недоступна, що ускладнює якісне медичне обслуговування. Найгірший недолік така ситуація буває критичною в разі загрози життю пацієнта. Уявіть випадок, коли потрібен негайний доступ до попередніх діагнозів, даних обстежень, результатів аналізів, рецептів чи інформації про можливі алергії на медичні препарати, а їх немає чи пацієнт не може ними поділитись, бо знаходиться в критичному стані. Проте, можливо нарахувати і безліч інших дрібніших недоліків. Таким чином, проблема зберігання та доступності медичних даних є актуальною і потребує вирішення.

Проблему реально вирішити двома способами. Перший і мало реалістичний - це введення єдиної системи даних про здоров'я пацієнта для всіх країн. Другий і більш реальний - це перекласти відповідальність за збереження своїх медичних даних на самого пацієнта. Розроблений мною сервіс BodyBio допоможе пацієнтам мати під рукою всю необхідну медичну

інформацію, яка б дозволяла їм ефективно взаємодіяти з лікарями незалежно від місця їхнього перебування.

Актуальність дослідження. Проблеми доступності медичних даних при переміщенні пацієнтів як у межах однієї країни, так і при міжнародних поїздках стають все потрібніше через збільшення динамічного переміщення населення. Веб-додаток BodyBio розроблений для того, щоб надати пацієнту інструмент для самостійного збереження та управління медичною історією. Це дозволяє кожній людині мати під рукою свою історію хвороби, що сприяє покращенню якості медичної допомоги, незалежно від країни чи медичного закладу. У зв'язку з цим, BodyBio вирішує дві основні проблеми: ускладненість доступу до медичних даних при зміні місця проживання чи подорожей, а також відсутність єдиної системи для зберігання медичної інформації.

Метою роботи є розробка веб-сервісу для зберігання і управління медичною історією пацієнтів, що дозволяє пацієнтам самостійно зберігати та використовувати свої медичні дані. Створення сервісу BodyBio надасть можливість пацієнтам мати під рукою потрібну медичну інформацію, що сприятиме продуктивній взаємодії з лікарями та поліпшенню процесу діагностики й лікування.

Для досягнення мети виконано наступні завдання:

Створено базу даних, яка підтримує збереження різних типів медичних документів (включаючи рентгенівські знімки, аналізи крові, електрокардіограми тощо) і забезпечує зручний і постійний доступ пацієнтів до своєї медичної інформації.

Розроблено інтерфейс, який дозволяє користувачам переглядати медичні записи, додавати їх чи керувати ними та надавати доступ до своїх даних медичним працівникам в разі необхідності.

Впроваджено технології для захисту конфіденційності та безпеки медичних даних, зокрема функції авторизації і верифікації користувачів.

Налаштовано веб-сервіс для забезпечення зручного доступу емігрантів та мандрівників до медичних даних з будь-якої країни та можливості додавати записи з різних закладів.

Додано функцію вибору мови та адаптивний дизайн інтерфейсу для зручного використання веб-сервісу різними групами користувачів, зокрема лікарями, пацієнтами та довіреними особами.

Об'єктом дослідження є процес зберігання та доступу до медичних даних пацієнтів у цифровому середовищі, забезпечення можливості швидкого доступу до даних.

Предметом дослідження є методи та засоби забезпечення надійного зберігання медичних даних пацієнтів у веб-додатку для забезпечення доступності та конфіденційності інформації.

Важливість розробки сервісу BodyBio полягає у тому, що він дозволяє своїм користувачам самостійно керувати своєю медичною інформацією. Пацієнт має можливість уникати повторних обстежень або зайвих аналізів, якщо попередні дані вже доступні в електронному форматі. Це також дозволяє скоротити витрати на медичні послуги, що є безсумнівним плюсом для багатьох пацієнтів.

Таким чином, BodyBio є хорошим і новим підходом до вирішення проблеми доступності та безпечного зберігання медичних даних. Сервіс орієнтований на забезпечення пацієнта потрібними інструментами для збереження, перегляду та передачі своєї медичної інформації, що є важливим для покращення медичного обслуговування і створює новий стандарт в сфері збереження медичних даних.

РОЗДІЛ 1. ВАЖЛИВІСТЬ SERVICU BODYBIO

Сучасні вимоги до медичної інформації пацієнтів є високими, тому що вчасний і повний доступ до історії хвороб, результатів обстежень та інших медичних даних сильно впливає на якість і швидкість наданої медичної допомоги. Існуючі в Україні системи дуже часто не задовольняють потреби тих, хто часто переїжджає чи подорожує. Для емігрантів, яким прийшлося регулярно змінювати місце проживання з різних причин, в тому числі, внаслідок війни, стало актуальним питання доступності й цілісності медичних даних. Вперше я стикнулася з цією проблемою самотійно, адже в дитинстві мої батьки мали досвід еміграції і життя в кількох містах, пізніше і я сама жила в еміграції кілька років. Втрата важливої інформації про стан здоров'я, наприклад, результатів попередніх обстежень або призначень лікарів, може серйозно ускладнити процес діагностики й лікування в іншій країні.

Сервіс BodyBio створено для того, щоб надати пацієнтам універсальну платформу для зберігання особистої медичної історії самотійно. Це забезпечує постійний доступ до всіх медичних записів незалежно від місця перебування пацієнта чи лікарні, яка його обслуговує. Унікальність BodyBio полягає в простоті користування, доступності та інтуїтивно зрозумілому інтерфейсі. Сервіс враховує специфічні потреби українських емігрантів та інших груп, що переїжджають: він дозволяє завжди мати під рукою критично важливі дані про здоров'я, попередні діагнози, алергії, інші параметри, що значно полегшує процес медичного обслуговування за кордоном. Також платформа розроблена так, що при наявності фінансування реально швидко додати переклади на кілька поширених мов і мов країн з розвинутим медичним туризмом.

1.1 Огляд функціоналу BodyBio

BodyBio пропонує декілька сторінок для зареєстрованих користувачів:

Головна сторінка призначена для редагування основних параметрів пацієнта, таких як зріст, вага, наявні алергії та особисті дані. Тут також можна налаштувати зв'язки з іншими користувачами, наприклад, дозволити доступ лікарю чи довірєній особі.

The screenshot shows the 'Body Bio' main page with a dark green header containing the logo and navigation links: 'Головна', 'Запис', and 'Моя медична історія'. The main content area is light blue and features a 'Редагувати' (Edit) button at the top. Below it is a form with the following fields:

- Ім'я: Input field with 'Olha' and a red question mark icon.
- Прізвище: Input field with 'Liubas' and a red question mark icon.
- Номер телефону: Input field with 'Введіть ваш Номер телефону' and a red question mark icon. Above the field is the text 'Поле обов'язкове' (Required field).
- Номер страховки: Input field with 'Введіть ваш Номер страховки' and a red question mark icon. Above the field is the text 'Поле обов'язкове' (Required field).
- Роль: Dropdown menu with 'super_admin' and a red question mark icon.
- Стать: Dropdown menu with a red question mark icon. Above the field is the text 'Поле обов'язкове' (Required field).
- Мова: Dropdown menu with 'ENG'.
- Лікарі: Input field with 'Введіть ваш Лікарі'.
- Довірені особи: Input field with 'Введіть ваш Довірені особи'.

Зображення 1.1. Головна сторінка

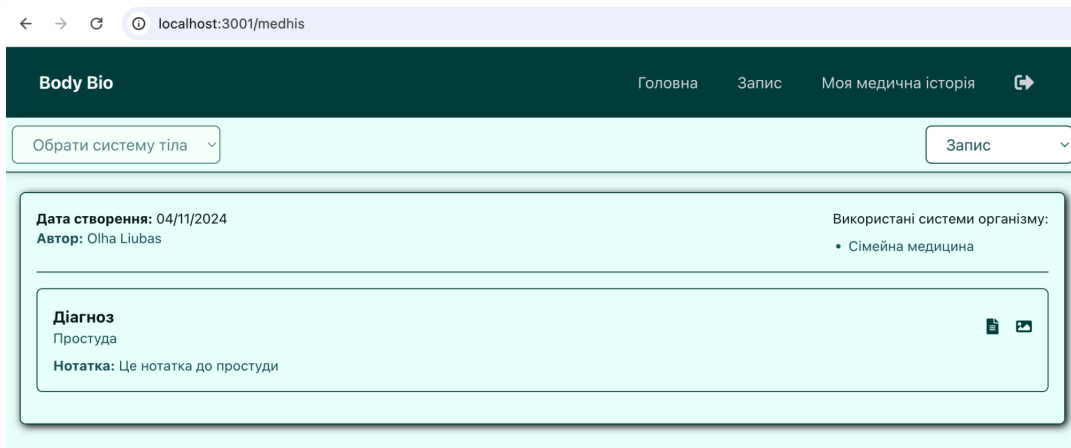
Сторінка "Запис" містить каталог хвороб, до якого користувач може додавати записи про діагнози, лабораторні дослідження та симптоми. Каталог хвороб розділений за системами тіла для зручності пошуку інформації.

The screenshot shows the 'Body Bio' 'Запис' (Record) page with a dark green header containing the logo and navigation links: 'Головна', 'Запис', and 'Моя медична історія'. The main content area is light blue and features a catalog of diseases and lab tests. The catalog is organized by body systems, with the following sections visible:

- Сімейна медицина** (Family Medicine):
 - Лікарі: Терапевт
 - Діагнози: Простуда (Cold) with a red plus icon, edit icon, and delete icon.
 - Лабораторні дослідження: Загальний аналіз крові (General blood test) with a red plus icon, edit icon, and delete icon.
- Загальна хірургія** (General Surgery): Red plus icon, edit icon, and delete icon.
- Ендокринна система** (Endocrine system): Red plus icon, edit icon, and delete icon.
- Інфекційні захворювання** (Infectious diseases): Red plus icon, edit icon, and delete icon.

Зображення 1.2. Сторінка запису

Сторінка "Моя медична історія" — призначена для зберігання повної історії хвороб і діагнозів користувача. Вона включає функцію сортування за системами організму, що значно полегшує пошук даних про конкретні захворювання або обстеження. Ця опція зручна для лікарів, які можуть швидко знайти потрібну інформацію для прийняття рішень щодо лікування.



Зображення 1.3. Сторінка медичної історії

Для незареєстрованих користувачів існують стандартні сторінки входу і реєстрації. Також 404 сторінка для неіснуючого адресу.

У BodyBio існують різні ролі користувачів: пацієнт, лікар, суперадміністратор, адміністратор та довірена особа. Це забезпечує гнучкість доступу до даних і можливість делегувати право на перегляд та управління інформацією. Також BodyBio має низку особливостей щодо змінюваності даних: пацієнти можуть змінювати особисту інформацію (як-от номер телефону), не проходячи складних перевірок. Це робить сервіс зручним для користувачів, які цінують легкість та мобільність у доступі до своїх даних.

1.2 Порівняння з аналогами

Існує декілька схожих сервісів, що також працюють у сфері зберігання медичної інформації. Жоден з них не покладає обов'язок зберігати свої медичні дані на пацієнта. Нижче наведено огляд основних сервісів, з якими я порівнювала BodyBio. Medics[1] українська платформа, орієнтована на управління медичними записами. Більшість її функцій спрямована на адміністрування медичних закладів, а не на зручність пацієнтів. Helsi[2]

інтегрована з багатьма українськими лікарнями та клініками, що дозволяє пацієнтам відстежувати записи про прийоми та деякі обстеження. Але не використовується всіма медичними закладами і пацієнт сам не може редагувати свої записи. Поліклініка без черг[3] – це система, яка виконує функцію запису до лікаря, а не повного збереження медичних даних. Її функціонал обмежується наданням пацієнту лише можливості запису на прийом. Asker.net[4] – сервіс, що забезпечує запис на прийоми та дозволяє часткове зберігання медичної історії. Однак Asker.net має певні обмеження: пацієнти не можуть самостійно змінювати номер телефону, і додати діагноз, поставлений за кордоном, неможливо. В мене особисто був випадок втрати номеру телефону, що прикріплений до медичної декларації. В висновку це призвело до втрати доступу до платформи. Змінити номер телефону в декларації я пробувала з сімейною лікаркою, але після 6 невдалих спроб і збоїв в системі ми здались. EMCImed[5] – система, яка зберігає медичні записи, але не має інтеграції з приватними клініками, що ускладнює доступ пацієнта до повного профілю його здоров'я. Сервіс не є популярним. Health24[6] – платформа для запису до лікарів, але також не для збереження медичних даних. Пацієнти можуть переглядати лише обмежений обсяг інформації про стан здоров'я. Тому не можу вважати цей сервіс є хорошим рішенням для збереження історії лікування. Microsoft HealthVault[7] – колишня міжнародна платформа для зберігання медичних записів, яка мала великий потенціал для пацієнтів, але припинила свою роботу, що створило відсутність пропозицій подібних послуг.

Як видно, з попереднього порівняння я не помітила жодної платформи, де сам пацієнт може зберігати свою медичну історію. Але при цьому я бачила потребу в такому продукті і помічала, що інші емігранти теж говорили про цю проблему в соціальних мережах.

1.3 Переваги BodyBio над аналогами

BodyBio має низку важливих переваг про які я розповім далі над існуючими аналогами. Пацієнти мають можливість самостійно додавати нові записи, зберігати результати обстежень і аналізів, встановлені діагнози, навіть якщо вони були зроблені за кордоном, в приватній клініці чи іншому місті. BodyBio дозволяє користувачам змінювати номер телефону чи інші особисті дані без ускладнень, що особливо важливо для тих, хто часто змінює місце проживання або втрачав телефон. Єдине, що є незмінним і унікальним це електронна пошта, яку легко відновити в разі втрати. BodyBio має можливість об'єднати записи з різних медичних установ, створюючи повну картину здоров'я, що особливо важливо для тих, хто відвідує лікарів у різних закладах. Має можливість сортувати діагнози, дослідження і симптоми за системами організму, що значно прискорює пошук потрібної інформації та робить роботу лікарів із даними більш зручною. Сервіс BodyBio підтримує кілька ролей, таких як лікар, пацієнт, довірена особа, що дозволяє пацієнтам делегувати доступ до своєї медичної інформації іншим людям за необхідності. BodyBio доступний для використання в будь-якій країні, що робить його корисним для багатьох людей, яким важливо мати під рукою інформацію про стан здоров'я у випадку, якщо вони звертаються за допомогою до лікарів за межами рідної країни чи міста.

Підсумовуючи, можна сказати, що BodyBio є важливим сервісом для зберігання медичної інформації, особливо для людей, які часто змінюють місце проживання та потребують доступу до своєї медичної історії в різних країнах. На відміну від існуючих аналогів, BodyBio пропонує гнучкість у зміні даних, можливість додавати записи, зроблені за кордоном і забезпечує високу доступність будь-де.

РОЗДІЛ 2. ЦІЛЬОВА АУДИТОРІЯ І ФУНКЦІОНАЛ BODYBIO

Цільова аудиторія BodyBio включає користувачів, серед яких емігранти, подорожуючі, медичні туристи, лікарі та родичі пацієнтів. Найбільшою цільовою аудиторією є для українці, які були змушені емігрувати через війну, залишивши свої медичні записи в Україні або в інших країнах. BodyBio дає цим користувачам спосіб зберігати і керувати медичною інформацією. Це дозволяє уникнути втрати важливих даних під час переїздів. Сервіс розроблений для того, щоб кожен пацієнт мав повний контроль над своєю медичною історією, міг швидко знайти потрібні записи та мати доступ до медичних даних незалежно від місця перебування.

Всі користувачі платформи можуть зберігати і впорядковувати свою медичну інформацію, додавати нові записи, переглядати історію хвороб і ділитися інформацією з лікарями. Водночас лікарі отримують зручний інструмент для ознайомлення з медичною історією пацієнтів, що значно покращує точність діагностики. Користувачі можуть переглядати діагнози, лікування, історію захворювань, а довірені особи можуть швидко знайти записи своїх близьких, щоб вчасно допомогти в екстрених випадках.

Функціонал BodyBio дозволяє пацієнтам зберігати результати обстежень, діагнози, рекомендації, рецепти, лабораторні аналізи і структуровано розміщувати їх за системами організму. Користувачі можуть використовувати систему для контролю свого здоров'я, а лікарі – для точнішого лікування. Таким чином BodyBio - це особистий архів медичних даних та інструмент для запобігання втрати інформації

2.1.Цільова аудиторія BodyBio

Основна цільова аудиторія сервісу BodyBio – це люди, які потребують постійного доступу до медичної інформації незалежно від свого місця проживання. В умовах поточної ситуації багато українців змушені емігрувати до інших країн через війну на батьківщині. Багато сімей опинилися в ситуації, коли попередня історія хвороб залишилася в Україні, або ж частково

зберігається в різних країнах, через які вони вимушено переміщалися. Втрата медичних даних ускладнює діагностику, а часто й повторну терапію у разі хронічних захворювань чи регулярних обстежень.

Крім емігрантів, BodyBio буде корисний людям, які часто подорожують, експатам, медичним туристам, а також тим, хто змінює роботу в різних країнах. Завдяки BodyBio пацієнти зможуть підтримувати повний обсяг інформації про своє здоров'я, уникати необхідності повторного проходження процедур і надавати лікарям доступ до актуальної інформації про попереднє лікування та діагнози в нових умовах. Завдяки широкому функціоналу та зручному інтерфейсу BodyBio стає універсальним інструментом для збереження медичних даних.

Уявімо кілька детальніших прикладів цільової аудиторії BodyBio в наступних абзацах.

Приклад 1

Марічка народилась в Харкові за кілька років перед війною. В лікарні її медична історія була записана в паперовому зошиті. Потім вона відвідувала огляди в сімейного лікаря з мамою. Сімейний лікар записував її вагу, дитячі хвороби, прививки на одному з українських ресурсів, доступ до якого був прикріплений до номеру телефону матері. Коли розпочалась війна, лікарня згоріла, мати і Марічка виїхали до Польщі. З часом український номер втрапився, а отже втратилась і вся медична історія. В Польщі Марічка півтори року відвідувала дитячий садок, отримала кілька щеплень, захворіла скарлатиною і була поставлена на облік логопеда. А потім через вплив інших біженців Польща була перенаселена, продукти і житло подорожчали, мама Марічки втратила роботу і була змушена емігрувати вдруге. Тепер матері прийдеться вже в третій країні відновлювати медичну історію своєї дитини, вона не зможе точно згадати на які медикаменти мала алергію її дитина і можливо щось наплутає. Втрата записів призведе до того, що лікарям буде важко спостерігати за розвитком дитини в динаміці. BodyBio

допоміг би цій сім'ї зберегти повну історію медичних даних Марійки, доступну в будь-якій країні, що б полегшило роботу лікарів і забезпечило якісне лікування для дівчинки.

Приклад 2

Олександр мандрівник, який любить досліджувати нові країни, місцеву кухню та культуру. Протягом своїх подорожей він відвідав десятки країн і часто звертався за медичною допомогою в різних місцях. Під час своєї поїздки до Індії він захворів на шлункову інфекцію, у Кенії отримав опіки після довгого перебування на сонці, а в Аргентині змушений був звернутися до лікаря через сильний кашель, який виявився наслідком алергії на місцеву рослинність. Олександрові проводили різні обстеження та аналізи, в історії його хвороб з'явилися нові записи, але збережені вони були в місцевих клініках або на електронних сервісах цих країн.

Коли Олександр повернувся додому, він мав проблеми з отриманням доступу до цих записів. Коли у нього знову почалися алергічні реакції, він не зміг швидко пояснити лікарю, які обстеження вже пройшов, і отримав ті ж аналізи вдруге. BodyBio допоміг би Олександрові зберігати всю медичну інформацію в одному місці, доступному у будь-якій країні, куди б він не поїхав. Це дозволило б уникнути повторних витрат на обстеження, полегшило б роботу лікарів та дало йому повний контроль над історією здоров'я. BodyBio створив би для Олександра безперервний ланцюжок медичних записів, незалежний від країни чи клініки, де проводилося лікування.

Приклад 3

Якщо б BodyBio став популярним і зберіг свою популярність з плином часу наступні покоління б могли отримувати дані про здоров'я попередніх поколінь. Це б дозволило ефективніше попереджувати спадкові захворювання. Уявімо ситуацію: у Марії є дочка Анна, яка нещодавно стала повнолітньою. Після проходження медичного обстеження лікарка

рекомендувала Анні звернути увагу на можливі ризики, пов'язані з генетичними захворюваннями, що можуть передаватися по жіночій лінії. Завдяки BodyBio Анна мала доступ до історії хвороб своєї мами і бабці та змогла переглянути всі записи про діагнози, лікування та обстеження, які жінки в її родині проходили раніше. Це дало Анні цінну інформацію і зберегло її здоров'я в майбутньому. Такий підхід до збереження медичних даних дозволяє попереджати можливі ризики для здоров'я і робить BodyBio необхідним інструментом для турботи про здоров'я на рівні поколінь.

2.2. Юзер сторі користувача сайту

Пацієнт

Я, як пацієнт, реєструюся на сайті BodyBio, щоб мати можливість зберігати та організовувати свою медичну історію. Після реєстрації і входу на сайт я потрапляю на головну сторінку, де одразу приступаю до заповнення своїх особистих даних. Вношу основну інформацію: свій зріст, вагу, дані про алергії, групу крові та будь-які інші важливі параметри, які можуть знадобитися лікарям у майбутньому. Після цього переходжу до вкладки “Запис”, де можна додавати інформацію з останнього прийому у лікаря.

У мене була зустріч із кардіологом, і тепер я додаю деталі в систему. Спочатку вказую свій діагноз – “артеріальна гіпертензія”, обираючи його з каталогу захворювань. Потім додаю результати обстежень: рентген знімок грудної клітки та результати лабораторних аналізів, щоб зберегти всю інформацію в одному місці. У каталозі всі хвороби, симптоми та обстеження чітко розділені на системи організму, тож я легко знаходжу потрібну інформацію у розділі “Серцево-судинна система”. Додаю й інші деталі прийому – симптоми, які мене турбували, рекомендації лікаря щодо лікування та рецепт на препарати, які мені призначили.

Після цього я переходжу у вкладку “Медична історія”, щоб перевірити, чи не був у мене раніше схожий діагноз. Вибираю сортування за системою організму і переглядаю всі попередні апойнменти та діагнози, пов'язані із

серцево-судинною системою. Виявляється, що раніше я вже проходив обстеження, але артеріальна гіпертензія в мене діагностована вперше. Це полегшує аналіз свого стану і дає можливість у будь-який момент продемонструвати інформацію лікареві. У майбутньому, лікар або адміністратор медичного закладу зможе приєднати мене до профілю лікаря, щоб йому було зручно переглядати мою історію.

Лікар

Я, як лікар, використовую BodyBio для управління інформацією про пацієнтів та для зручного доступу до їхніх даних. Входжу в систему, переглядаю запит пацієнта на отримання доступу до його історії хвороби. Я маю можливість переглянути його обстеження, симптоми та діагнози, які він додав із попередніх прийомів. Після зустрічі з пацієнтом я хочу додати новий діагноз, який нещодавно йому поставили, тому переходжу у вкладку “Запис” і додаю інформацію про виявлене захворювання та призначене лікування.

Для зручності я використовую каталог, де всі діагнози, симптоми й обстеження розділені за системами організму, і це допомагає мені швидко знайти потрібні записи у моїй спеціалізації. У майбутніх версіях сайту, я як лікар зможу самостійно додавати нові діагнози до каталогу в розділі, де працюю. Це зробить мою роботу ще простішою, дозволяючи оновлювати список актуальних хвороб, з якими я найчастіше стикаюся у своїй практиці. Крім того, я можу працювати на платформі не лише як лікар, але й як пацієнт, зберігаючи особисті медичні записи в своєму профілі.

Адміністратор

Я, як адміністратор сайту BodyBio, маю доступ до управління обліковими записами користувачів і можу допомагати пацієнтам та лікарям приєднуватися до платформи. Після входу в систему я переглядаю особисті дані нових користувачів, але не маю доступу до їхніх медичних записів, що гарантує конфіденційність даних. Якщо пацієнт хоче надати доступ до своєї

історії хвороби конкретному лікареві, я можу допомогти здійснити цей процес, приєднавши профіль лікаря до пацієнта.

Також у мої обов'язки входить приєднання довірених осіб до профілю пацієнтів, якщо вони на це дали дозвіл. Завдяки цьому довірена особа може переглядати інформацію, яку пацієнт дозволив їй бачити. У майбутніх оновленнях сайту планується розширення функціоналу ролі адміністратора, що дозволить мені більш ефективно підтримувати роботу платформи і допомагати користувачам.

Довірена персона

Як довірена персона, я маю доступ до медичних даних пацієнта, який довірив мені свої записи. Наприклад, моя дружина потрапила в аварію і зараз перебуває без свідомості, а лікарі потребують інформації про її стан здоров'я. Завдяки BodyBio я можу увійти в систему і переглянути її профіль. Оскільки вона довірила мені доступ до свого облікового запису, я маю можливість бачити необхідні медичні дані, такі як її алергії, групу крові та поточні ліки, які їй призначені. Це дає лікарям критично важливу інформацію для прийняття рішень перед операцією.

У майбутньому функціонал довіреної персони буде розширено, щоб забезпечити більше можливостей для взаємодії з обліковим записом пацієнта, а також можливість отримувати автоматичні повідомлення про зміни в стані здоров'я, якщо пацієнт вирішить їх надавати. Ця роль допомагає мені як близькій людині підтримувати медичну безпеку моєї родини в екстрених ситуаціях.

Супер адміністратор

Як супер адміністратор, я відповідаю за загальне адміністрування та технічну підтримку BodyBio. У цій ролі я маю доступ до всіх функцій платформи і можу переглядати, редагувати або видаляти будь-які облікові записи користувачів, якщо це необхідно для обслуговування сайту. Я також можу тестувати новий функціонал, оновлювати систему і забезпечувати

конфіденційність та захист даних користувачів, що є важливим для безпеки всієї платформи.

BodyBio – це інноваційний сервіс для зберігання і доступу до медичних даних, орієнтований на потреби сучасного користувача. Цільова аудиторія платформи охоплює людей, які цінують зручність у збереженні та доступі до медичних записів. Платформа є особливо корисною для тих, хто часто змінює місце проживання, відвідує лікарів у різних країнах або потребує контролю над даними своїх близьких.

Функціонал BodyBio охоплює основні потреби користувачів: від збереження історії хвороб і обстежень до можливості надання доступу лікарям та траст-персонам. Користувачі можуть легко додавати нові записи, переглядати попередні візити до лікаря та лікування, а також упорядковувати медичні дані. Для лікарів BodyBio стає джерелом інформації, яке дозволяє отримати повний огляд стану здоров'я пацієнта та забезпечити персоналізований підхід до лікування. Довірені персони отримують можливість допомагати близьким, а адміністратори забезпечують надійну роботу системи.

Завдяки BodyBio пацієнти зможуть зберігати свою медичну інформацію в одному місці, завжди мати доступ до неї та уникнути складнощів, пов'язаних з втратою або розпорошеністю даних. BodyBio створює нові стандарти у збереженні медичних даних, об'єднуючи зручність і функціональність для кожного користувача.

РОЗДІЛ 3. ТЕХНОЛОГІЧНИЙ СТЕК BODYBIO

MySQL[9]. Для збереження та організації даних у BodyBio використовується реляційна база даних MySQL[9]. Вона забезпечує структуроване зберігання та швидкий доступ до інформації і є поширеною технологією. MySQL[9] надає можливості для створення таблиць, обмежень цілісності даних, а також підтримує складні запити та агрегації. Це робить його оптимальним вибором для додатків з великими обсягами даних, таких як медичні записи.

TypeORM[10]. Для взаємодії з базою даних MySQL[9] обраний ORM інструмент TypeORM[10], який допомагає організовувати запити до бази даних за допомогою об'єктів. TypeORM[10] забезпечує легкість у роботі з MySQL[9] і дозволяє створювати запити на рівні коду без написання SQL. Цей підхід полегшує підтримку та знижує можливість помилок.

Class-validator[11]. Ця бібліотека використовується для валідації даних, що надходять з фронтенду. За допомогою class-validator[11] можна налаштувати правила перевірки полів в запитах і віддавати повідомлення про помилки.

NestJS[8] – це фреймворк для створення серверних додатків, який використовує TypeScript та організовує код за принципами модульності та використання залежностей. Це дозволяє структурувати код у вигляді модулів, що полегшує роботу над великими проектами, інтуїтивне сприйняття проекту і спрощує підтримку та тестування.

Docker[12] використовується для створення контейнерів в середовищі хостингу, спрощує налаштування середовища розробки та хостинг додатку.

Для хостингу серверної частини BodyBio використовується Railway[13], що дозволяє швидко розгортати додатки з MySQL базою даних у хмарі. Railway[13] забезпечує безперервне розгортання і надає зручний інтерфейс для управління. Серверна частина BodyBio доступна за посиланням [23], що дозволяє лікарям і пацієнтам взаємодіяти з додатком в реальному часі.

GitHub Repository. Код бекенду знаходиться у відкритому доступі на GitHub[20].

React[15] є основою для будування інтерфейсу користувача BodyBio, що забезпечує швидке оновлення контенту на сторінці без повного перезавантаження. Завдяки компонентній архітектурі React[15], розробники можуть створювати динамічні компоненти, масштабувати і перевикористовувати їх, що полегшує додавання нового функціоналу і підвищує продуктивність додатка.

Для стилізації в BodyBio використовується Tailwind CSS[18] – фреймворк, що дозволяє створювати дизайни без написання власного CSS. Tailwind[18] надає готові класи для швидкого створення стилів.

Redux Toolkit[16] використовується для управління станом додатку. Це поширений інструмент, що дозволяє зберігати та передавати дані між різними компонентами і змушує компоненти реагувати на зміни стану. Це забезпечує стабільну роботу додатку без повторного запиту інформації з сервера.

Redux Saga[17] часто використовується разом з попереднім інструментом і дозволяє організувати складну бізнес-логіку на стороні фронтенду, відділивши її від стилів і логіки компонентів. Це допомагає обробляти великі обсяги даних з бекенду та гарантує синхронізацію стану додатка з сервером.

Для підтримки мультимовності на платформі BodyBio інтегрований i18n[19]. Це робить додаток зручним для користувачів із різних країн, особливо враховуючи, що BodyBio орієнтований на емігрантів і медичного туризму. Також багатомовність покращить взаємодію пацієнта і лікаря, що говорять різними мовами.

Фронтенд BodyBio розгорнутий на платформі Netlify[21], яка забезпечує надійний хостинг для SPA-додатків. Netlify підтримує безперервне розгортання та інтеграцію з GitHub, що спрощує оновлення коду. Додаток доступний за посиланням [22], що дозволяє користувачам взаємодіяти з ним у будь-який час.

GitHub Repository. Код фронтенду зберігається на GitHub за посиланням [23].

3.1 Опис бекенду BodyBio

Для розгортання бекенду BodyBio необхідно виконати декілька кроків. Спершу запускаємо базу даних MySQL[9], використовуючи команду `brew services start mysql` (якщо базу потрібно перезапустити, можна скористатися `brew services restart mysql`). Після цього відкриваємо MySQL[9] через термінал, використовуючи `mysql -u root -p`, що дозволяє отримати доступ до бази даних під користувачем `root`.

Далі переходимо до запуску самої програми, яка побудована на фреймворку NestJS[8]. Для запуску в режимі розробки використовуємо команду `npm run start:dev`. У процесі розробки можна створювати нові модулі, контролери та сервіси для програми за допомогою команд, таких як `nest g module users`, `nest g controller users/controllers/users` і `nest g service users/services/users`.

Для розгортання проєкту з використанням Docker[12] необхідно спочатку видалити всі існуючі мережі, контейнери та образи, які можуть бути пов'язані з додатком. Це робиться за допомогою команд `docker stop`, `docker rm`, `docker rmi` і `docker network rm`. Після очищення середовища створюємо нову мережу Docker (`docker network create body-bio-network`) та ініціалізуємо контейнер MySQL з параметрами бази даних.

Наступним кроком є створення Docker-образу для додатку за допомогою команди `docker build -t body-bio-image .`, після чого запускаємо контейнер додатку, налаштовуючи змінні середовища, такі як `DATABASE_HOST`, `DATABASE_PORT`, `DATABASE_USERNAME`, `DATABASE_NAME` тощо, та запускаємо на порту 3000.

Для перевірки роботи контейнерів можна скористатися командами `docker logs mysql-container` та `docker logs body-bio`. Додаток буде доступний за адресою `http://localhost:3000/` + шлях, що дозволяє перевірити підключення до

бекенду. Якщо в додатку внесено зміни, контейнер потрібно оновити за допомогою `docker stop`, `docker rm`, `docker build`, а потім знову запустити додаток.

Докер я використовувала для правильного деплоюменту додатку і моєю рекомендацією є розгорнути бекенд частину без докера.

У бекенд-структурі `BodyBio User` є основною сутністю, яка представляє користувача додатку. Ключове поле `id` є унікальним і генерується автоматично у форматі `uuid`. Це забезпечує унікальну ідентифікацію кожного користувача. Серед основних полів є `name` та `surname`, які мають обмеження на довжину за допомогою бібліотеки `class-validator` для забезпечення коректності введених даних. Поле `email` є обов'язковим і перевіряється на правильний формат електронної пошти. Саме за допомогою емейлу, а не номеру телефону чи номеру страховки юзер може відновити свій профіль. Для додаткових даних, таких як `contactNumber` та `insuranceId`, встановлено опцію `nullable` (це робить їх необов'язковими), а поле `password` зберігається як приховане (`select: false`) для підвищення безпеки. Також є поля для зберігання ідентифікаторів, які зберігають список користувачів інших ролей, таких як `patientsIds`, `doctorsIds`, `adminsIds` та `trustPersonsIds` для організації зв'язків з іншими користувачами.

У користувачів є відношення один до багатьох з іншими сутностями (`Appointment`, `UsersLabs`, `UsersDiagnoses`, `UsersFinding`). Це означає, що користувач може мати кілька пов'язаних записів про візити, лабораторні дослідження, діагнози та висновки. Контролер `UserController` надає ендпоінти для роботи з користувачами, такі як `getUsers` для отримання списку користувачів, `getUserById` для отримання інформації про конкретного користувача, `updateUserById` для оновлення даних користувача, `deleteUserById` для видалення, а також `signUpUser` для реєстрації нового користувача та `loginUserByEmail` для авторизації.

Сервіс `UserService` забезпечує бізнес-логіку, наприклад методи для хешування паролів і перевірки паролів під час авторизації. Всі операції

взаємодіють із базою даних за допомогою `userRepository`, який надає можливості для створення, збереження, оновлення та видалення записів.

```
@Entity({name: 'users'})
export class User {

    @PrimaryGeneratedColumn("uuid")
    id: string;

    @Column({nullable: false})
    @MinLength(1, { message: 'Name is too short'})
    @MaxLength(30, { message: 'Name is too long'})
    name: string;

    @Column({nullable: false})
    @MinLength(1, { message: 'Surname is too short'})
    @MaxLength(30, { message: 'Surname is too long'})
    surname: string;

    @Column({
        type: "enum",
        enum: UserRole,
        default: UserRole.PATIENT,
    })
    role: UserRole;

    @Column({ unique: true })
    @IsEmail({}, {message: 'Use correct email'})
    email: string;

    @Column({ select: false })
    password: string;

    @Column({nullable: true})
    contactNumber: string;

    @Column({nullable: true})
    insuranceId: string;

    @CreateDateColumn()
    createdAt: Date;
}
```

Зображення 3.1. Приклад сутності користувача в коді

Сутність `BodySystem` є батьківською для медичного каталогу, до якого також належать `Lab`, `Diagnoze`, та `Finding`. Кожна з цих дочірніх сутностей має зв'язок багато до одного з `BodySystem`. У `BodySystem` визначені базові поля, такі як `id` та `nameKey`, що визначають ідентифікатор і назви систем тіла, за допомогою якого потім на фронтенді знаходиться переклад. Сутність `Lab`, яка дуже схожа до `Diagnoze` та `Finding`, зберігає дані про лабораторні дослідження, діагнози чи симптоми. Контролер `DiagnozesController` надає ендпоінти для взаємодії з сутністю `Diagnoze`, включаючи методи `findAll()` для отримання всіх діагнозів, `findDiagnozesByIds()` для пошуку діагнозів за списком ідентифікаторів і `create()` для створення нового запису.

Сервіс `DiagnozesService` обробляє бізнес-логіку для взаємодії з `Diagnoze`, забезпечуючи методи для отримання всіх записів, пошуку за ідентифікаторами і створення нових записів. При створенні нового запису

виконується перевірка існування BodySystem за bodySystemId. Також перевіряється, чи не перевищено максимальний рівень вкладеності (2 рівні: bodySystem => parent). В LabsService визначено методи для отримання всіх лабораторних тестів (findAll), пошуку лабораторних тестів за ідентифікаторами (findLabsByIds) і створення нового лабораторного запису (create). Ендпоінти та функціонал у Diagnose та Finding аналогічні до Lab, що забезпечує уніфіковану структуру для трьох типів дочірніх сутностей в BodySystem.

```
@Entity()
export class BodySystem {
  @PrimaryGeneratedColumn('uuid')
  id: string;

  @Column()
  name: string;

  @Column()
  nameKey: string;

  @Column({ type: 'int', default: 0 })
  level: number;

  @OneToMany(() => DoctorSpecialization, (specialization) => specialization.bodySystem)
  doctorsSpecializations: DoctorSpecialization[];

  @OneToMany(() => Lab, (lab) => lab.bodySystem)
  @JoinColumn({ name: 'id' })
  laboratories: Lab[];

  @OneToMany(() => Finding, (finding) => finding.bodySystem)
  @JoinColumn({ name: 'id' })
  findings: Finding[];

  @OneToMany(() => Diagnose, (diagnose) => diagnose.bodySystem)
  @JoinColumn({ name: 'id' })
  diagnoses: Diagnose[];
}
```

Зображення 3.2. Приклад сутності системи тіла в коді

Сутність Appointment в BodyBio є батьківською сутністю для записів, пов'язаних із медичними записами, і забезпечує структуру для записів лабораторних аналізів (UsersLabs), діагнозів (UsersDiagnoses) та симптомів (UsersFinding). Основні поля Appointment включають id, user (користувач, якого стосується зустріч), author (користувач, який створив запис), дату створення, та масив usedBodySystems, який зберігає ідентифікатори систем тіла, задіяних під час зустрічі. Зв'язок багато до одного з User реалізовано як

для поля `user`, так і для `author`, що дозволяє кожному запису належати певному користувачеві та мати інформацію про того, хто створив запис. Зустрічі також мають зв'язки один до багатьох із дочірніми сутностями `UsersLabs`, `UsersDiagnozes`, і `UsersFinding`. Дочірня сутність `UsersLabs` зберігає інформацію про лабораторні аналізи, прив'язані до певного прийому. Поля `bodySystemId`, `parentLabId` описують зв'язок між аналізами з каталогу, що дає можливість структурувати дані відповідно до специфіки медичних систем. `UsersDiagnozes` та `UsersFinding` мають аналогічну структуру, що дозволяє однаково обробляти діагнози та медичні знахідки в межах каталогу. Усі дочірні сутності (`UsersLabs`, `UsersDiagnozes`, `UsersFinding`) також містять масиви `linkedImages` та `linkedFiles`, що дозволяє зберігати пов'язані з аналізами або діагнозами файли, такі як результати обстежень або знімки, прив'язані до кожного конкретного запису. Контролер `UsersAppointmentsController` містить ендпоінти для взаємодії з `Appointment`, включаючи методи `create` для створення запису, `findAllByUserIdAndBodySystem` для фільтрації записів за ідентифікаторами користувача та системи тіла, `findOne` для отримання конкретного запису за його ідентифікатором, `edit` для оновлення запису та `delete` для видалення запису разом з усіма пов'язаними дочірніми елементами.

```

@Entity({ name: 'users_appointment' })
export class Appointment {
  @PrimaryGeneratedColumn('uuid')
  id: string;

  @ManyToOne(() => User, ({user}) => user.appointments, { nullable: false })
  @JoinColumn({ name: 'userId' })
  user: User;

  @ManyToOne(() => User, { nullable: false })
  @JoinColumn({ name: 'authorId' })
  author: User;

  @CreateDateColumn()
  creationDate: Date;

  @Column("simple-array", { nullable: true })
  usedBodySystems: string[];

  @OneToMany(() => UsersFinding, ({usersFinding}) => usersFinding.appointment)
  usersFindings: UsersFinding[];

  @OneToMany(() => UsersDiagnozes, ({usersDiagnozes}) => usersDiagnozes.appointment)
  usersDiagnozes: UsersDiagnozes[];

  @OneToMany(() => UsersLabs, ({usersLabs}) => usersLabs.appointment)
  usersLabs: UsersLabs[];
}

```

Зображення 3.3. Приклад сутності запису користувача в коді

Це дозволяє користувачам зберігати та організовувати всі записи про прийоми у зручній та структурованій формі. Сервіс UsersAppointmentsService обробляє бізнес-логіку, пов'язану з сутностями Appointment, UsersLabs, UsersDiagnozes, і UsersFinding. Він забезпечує методи для створення, редагування, отримання і видалення зустрічей разом із пов'язаними записами.

3.2. Опис фронтенду BodyBio

Розгортається фронтенд як стандартний React[15] застосунок.

Загальні компоненти в BodyBio створені, щоб забезпечити гнучкість та зручність використання у різних частинах застосунку. Зазвичай розроблені компоненти є загальними та універсальними.

Для BodyBio я також розробила кілька загальних компонентів, які використовувала в різних частинах аплікації. Нижче я наведу кілька прикладів.

Button – базовий компонент кнопки, що приймає параметри children (вміст), onClick (подія), type (тип кнопки), disabled (активність) та className (додаткові стилі). Цей компонент автоматично змінює свій стиль залежно від активності, роблячи його універсальним для різних сценаріїв, як-от підтвердження форм чи навігаційні дії.

CheckboxWithLabel – компонент, який об'єднує чекбокс із міткою для забезпечення зручності користування.

InputWithValidation – поле введення з валідацією та додатковими можливостями. Компонент включає валідацію в реальному часі з візуальними індикаторами для повідомлення про помилки, а також має функціональність для показу додаткових підказок.

SelectWithValidation – компонент вибору, який включає функцію валідації та візуальні індикатори помилок. Він дозволяє користувачам вибирати значення зі списку опцій та отримувати додаткові підказки за допомогою tooltip.

Ці компоненти забезпечують базову структуру та полегшують розробку інтерфейсу, роблячи його більш зручним для користувача.

У BodyBio використовується i18n для забезпечення мультимовності та адаптації інтерфейсу до англійської та української мов. Файли перекладів зберігаються у форматі JSON і містять ключі, які відповідають певним

текстовим елементам сервісу. Основна структура перекладів знаходиться в `translation.json` для кожної мови, що дозволяє легко і швидко додавати або оновлювати тексти в різних мовних версіях.

При ініціалізації `i18n`[19] налаштовуються мовні ресурси, включаючи англійську (`enTranslation`) та українську (`uaTranslation`). Якщо мова користувача не визначена або відсутній переклад для певного тексту, використовується значення з `fallbackLng`, що вказує на англійську мову за замовчуванням.

Для відображення текстів в інтерфейсі, як-от мітки полів або підказки, застосовується функція `t` із бібліотеки `useTranslation`. Наприклад, у компоненті `SignUp` `t('COMMON.name')` завантажує переклад для ключа `name` з розділу `COMMON`, що дозволяє відображати текст на мові, вибраній користувачем.

Скрипт `copyTranslationKeys` автоматизує процес оновлення перекладів. Він копіює ключі з базового файлу англійської мови до інших мовних файлів, зберігаючи структуру перекладів. Якщо певний ключ відсутній у файлі перекладу іншої мови, він автоматично додається як порожній, що спрощує підтримку перекладів.

Завдяки `i18n` і структурі перекладів, `BodyBio` може бути легко адаптований до інших мов у майбутньому.

У `BodyBio` використовується `Tailwind CSS`[18] для швидкого і зручного стилізування компонентів, що спрощує розробку інтерфейсу. `Tailwind CSS` дозволяє застосовувати класи для створення адаптивного дизайну без необхідності писати стилі самотужки з нуля.

```

const copyTranslationKeys = () => {
  const sourceFile = path.join(
    __dirname,
    'src',
    'locales',
    'en',
    'translation.json',
  );
  const sourceData = JSON.parse(fs.readFileSync(sourceFile, 'utf-8'));

  const localesDir = path.join(__dirname, 'src', 'locales');
  const languages = fs
    .readdirSync(localesDir)
    .filter(
      (dir) =>
        dir !== 'en' && fs.lstatSync(path.join(localesDir, dir)).isDirectory(),
    );

  languages.forEach((lang) => {
    const targetFile = path.join(localesDir, lang, 'translation.json');

    const targetData = fs.existsSync(targetFile)
      ? JSON.parse(fs.readFileSync(targetFile, 'utf-8'))
      : {};

    deepCopyKeys(sourceData, targetData);

    fs.writeFileSync(targetFile, JSON.stringify(targetData, null, 2), 'utf-8');
  });

  console.log('Keys successfully copied!');
};

```

Зображення 3.3. Функція copyTranslationKeys

У файлі конфігурації Tailwind визначено індивідуальні кольори для теми додатку, як-от darkSeaWave для кнопок та меж, lightBlue для фону сторінок, та brightLightBlue для акцентів. Це робить інтерфейс більш узгодженим, оскільки всі компоненти використовують однакову палітру кольорів.

Tailwind значно спрощує роботу зі станами компонентів. Наприклад, умова для межі поля вибору залежно від помилки чи фокусу (border-lightTheme-error або border-lightTheme-darkSeaWave) задається через використання змінних класів, що скорочує час розробки і зменшує кількість

додаткових CSS-стилів. Це оптимізує загальний розмір CSS-файлів, оскільки Tailwind видаляє невикористані класи під час білду.

У BodyBio використовується також Redux Toolkit[16] у поєднанні з Redux Saga[17] для керування станом, виконання асинхронних запитів та оптимізації взаємодії з API. Redux Toolkit[16] допомагає створювати та керувати станом додатку. Наприклад, за допомогою createSlice компоненти, як-от authSlice, мають зрозумілі ред'юсери для аутентифікації, зокрема для обробки подій входу, реєстрації та виходу.

Redux Saga[17] керує асинхронними процесами через саги. У прикладі signUpUserSaga та loginUserSaga обробляють процеси реєстрації та входу користувачів, виконуючи відповідні запити до API і, у разі успіху, оновлюючи стан через signUpUserSuccess чи loginUserSuccess. Якщо виникає помилка, саги спрямовують її до signUpUserFailure або loginUserFailure, що робить код легшим для відлагодження та підтримки. Мені подобаються ці інструменти також через наявність зручного розширення для браузеру Google Chrome, де я можу відстежувати зміни стану при розробці.

Оптимізація API здійснюється за допомогою централізованого request методу, який використовує бібліотеку axios і підтримує різні методи HTTP. Кожен запит до API здійснюється через універсальну функцію request, що знижує дублювання коду та дозволяє зберігати стабільну структуру запитів.

Також для оптимізації зберігання даних використовується localStorage, зокрема для зберігання інформації про користувача після входу в систему.

У компоненті Redux використовується для ініціалізації авторизаційного стану за допомогою useDispatch та useSelector, а також для виклику відповідного ред'юсера.

ВИСНОВКИ

В рамках дослідження я розробила структура для зберігання медичних даних, яка надає можливість ефективно організувати медичну інформацію різного типу. Це значно полегшує її використання як для пацієнтів, так і для медичних працівників. Створена система дозволяє зберігати посилання на такі види документів, як рентгенівські знімки, результати аналізів, електрокардіограми тощо. Застосована структура організації даних сприяє зручному збереженню медичної інформації і забезпечує легкий доступ до кожного запису за потреби. Такий підхід надає пацієнтам можливість мати єдину, завжди доступну електронну базу медичних записів, яка в подальшому може допомогти уникнути необхідності повторних обстежень або додаткових витрат на отримання дублікатів документів.

Важливою складовою цієї роботи стало створення функціоналу, який дає змогу пацієнтам легко завантажувати медичні записи, керувати ними, а також контролювати доступ до медичної інформації себе чи родичів. Функціонал дозволяє зберігати всю медичну історію в одному місці, незалежно від географічного положення або медичного закладу, де були створені записи. Цей фактор є важливим для пацієнтів, які часто змінюють місце проживання або подорожують, адже тепер вони мають доступ до своїх медичних записів у будь-який час. Розроблений веб-сервіс BodyBio особливо важливо для українських емігрантів та людей, що постійно переїжджають, адже в них часто втрачається доступ до всіх попередніх медичних записів. Такий сервіс дозволяє економити час та ресурси для збереження свого здоров'я. Пацієнти мають можливість уникнути повторного проведення обстежень, якщо їх результати вже збережені в електронному вигляді. Крім того, доступність інформації про алергії, попередні діагнози та проведені процедури підвищує якість медичного обслуговування та дозволяє лікарям приймати рішення про лікування точніше.

Веб-сервіс BodyBio демонструє багато переваг порівняно з існуючими аналогами завдяки своїй функціональності та орієнтованості на потреби користувачів. Завдяки BodyBio користувачі отримують повний контроль над своєю медичною інформацією, що дозволяє їм ефективніше управляти своїм здоров'ям та уникати труднощів при переході між медичними закладами в різних країнах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Medics [Електронний ресурс] — Режим доступу: URL: <https://medics.com.ua/>
2. Helsi [Електронний ресурс] — Режим доступу: URL: <https://helsi.me/>
3. Поліклініка без черг [Електронний ресурс] — Режим доступу: URL: <https://pb4.com.ua/>
4. Asker.net [Електронний ресурс] — Режим доступу: URL: <https://asker.net/>
5. EMCImed [Електронний ресурс] — Режим доступу: URL: <https://emcimed.com/>
6. Health24 [Електронний ресурс] — Режим доступу: URL: <https://health24.ua/>
7. Microsoft HealthVault [Електронний ресурс] — Режим доступу: URL: <https://healthvault.com/> (Примітка: Microsoft HealthVault припинив роботу, але історичне посилання зберігається)
8. NestJS: A progressive Node.js framework [Електронний ресурс] — Режим доступу: URL: <https://nestjs.com/>
9. MySQL Documentation [Електронний ресурс] — Режим доступу: URL: <https://dev.mysql.com/doc/>
10. TypeORM: TypeScript ORM for Node.js [Електронний ресурс] — Режим доступу: URL: <https://typeorm.io/>
11. Class-validator [Електронний ресурс] — Режим доступу: URL: <https://github.com/typestack/class-validator>
12. Docker Documentation [Електронний ресурс] — Режим доступу: URL: <https://docs.docker.com/>
13. Railway.app Documentation [Електронний ресурс] — Режим доступу: URL: <https://docs.railway.app/>

14. React Documentation [Електронний ресурс] — Режим доступу: URL: <https://react.dev/>
15. Redux Toolkit Documentation [Електронний ресурс] — Режим доступу: URL: <https://redux-toolkit.js.org/>
16. Redux Saga Documentation [Електронний ресурс] — Режим доступу: URL: <https://redux-saga.js.org/>
17. Tailwind CSS Documentation [Електронний ресурс] — Режим доступу: URL: <https://tailwindcss.com/docs/>
18. i18next: Internationalization for JavaScript [Електронний ресурс] — Режим доступу: URL: <https://www.i18next.com/>
19. netlify Documentation [Електронний ресурс] — Режим доступу: URL: <https://docs.netlify.com/>
20. Бекенд репозиторій [Електронний ресурс] — Режим доступу: URL: <https://github.com/0-L-G-A/bodyBio/tree/prod>
21. Задеплоїний сервіс [Електронний ресурс] — Режим доступу: URL: <https://bodybio.netlify.app/>
22. Фронтенд репозиторій [Електронний ресурс] — Режим доступу: URL: <https://github.com/0-L-G-A/bodyBioFront/tree/prod>
23. Запити до бекенду [Електронний ресурс] — Режим доступу: URL: <https://mbbioprod.up.railway.app/> + шлях