Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Інститут комп’ютерних систем

Кафедра інформаційних систем

Щербата Ірина Олегівна,

студент групи АІ-212

ДИСЦИПЛІНА

Об’єктно-орієнтоване програмування

КУРСОВА РОБОТА

Розробка мобільного додатку для ведення заміток

Спеціальність:

122 Комп’ютерні науки

Освітня програма:

Комп’ютерні науки

Керівник:

Годовиченко Микола Анатолійович,

кандидат технічних наук, доцент

Одеса – 2023

ЗМІСТ

Анотація 3

Вступ 4

1.Огляд систем-аналогів та технологій їх розробки………………………………….6

1.1 Огляд існуючих систем для запису на прийом до лікаря………………………..6

1.2 Використання мови програмування Java для розробки веб-додатків………….6

1.3 Використання HTML та CSS для створення інтерфейсу веб-додатків…………7

2.Проєктування веб-додатку для запису на прийом до лікаря……………………....8

2.1 Мета та завдання веб-додатку……………………………………………………..8

2.2 Визначення функціональних вимог до веб-додатку……………………………..8

2.3 Розробка бази даних для зберігання інформації про лікарів та пацієнтів……...9

2.4 Проектування користувацького інтерфейсу веб-додатку………………………11

2.5 Реалізація логіки запису на прийом та управління даними……………………14

3.Тестування веб-додатку…………………………………………………………….15

Висновки………………………………………………………………………………22

Перелік використаних джерел………………………………………………………23

АНОТАЦІЯ

Курсова робота присвячена розробці веб-додатку для для запису на прийом до лікаря. Додаток призначний для зручного запису на прийом до лікаря. Основною метою роботи є створення простого, ефективного та інтуїтивно зрозумілого інструменту, який дозволяє користувачам легко забронювати зручний для них час прийому, не стоячи довгий час у черзі. Також у разі виникнення певних особистих обставин користувач може легко відміити бронювання раніше обраного часу. В ході розробки додатку будуть використані практики розробки веб-додатків, зокрема мова програмування Java,а для створення HTML- сторінок мови: HTML, CSS та Bootstrap.

ABSTRACT

The course work is devoted to the development of a web application for making an appointment with a doctor. The application is intended for convenient appointment with a doctor. The main goal of the work is to create a simple, effective and intuitive tool that allows users to easily book an appointment time convenient for them, without standing in line for a long time. Also, in case of certain personal circumstances, the user can easily cancel the reservation before the chosen time. In the course of developing the application, web application development practices will be used, in particular the Java programming language, and HTML, CSS and Bootstrap languages will be used to create HTML pages.

ВСТУП

У сучасному світі, коли швидкість життя постійно зростає, люди все частіше стикаються з проблемою ефективного управління своїм часом і організації різних аспектів свого життя. Одним з найбільш важливих аспектів, які потребують належної організації, є медична допомога та запис на прийом до лікаря.

У рамках даної курсової роботи ми пропонуємо розробку веб-додатку, який спростить процес запису на прийом до лікаря та зробить його більш зручним та ефективним для пацієнтів. Основною метою даного додатку є полегшення процесу отримання медичної допомоги, зменшення часу очікування на прийом та покращення загальної організації роботи медичних закладів.

В даному вступі ми оглянемо актуальність даної проблеми, визначимо мету та завдання курсової роботи, а також надамо короткий огляд запропонованих рішень та технологій, які будуть використовуватись під час розробки веб-додатку.

Актуальність теми полягає в необхідності оптимізації процесу запису на прийом до лікаря та покращення доступу до медичної допомоги. Довгі черги, незручні графіки роботи, складні процедури запису - все це ускладнює отримання необхідного лікування для багатьох людей. Наш веб-додаток пропонує вирішити цю проблему, надаючи зручний і швидкий доступ до інформації про лікарів, їхній розклад роботи та можливість запису на прийом у будь-який зручний для користувача час.

Метою курсової роботи є розробка функціонального веб-додатку, який надасть пацієнтам можливість легко та зручно записуватись на прийом до лікаря. Додаток буде базуватись на сучасних веб-технологіях та принципах розробки, забезпечуючи високу ефективність та зручність використання.

Завдання курсової роботи включають:

* Огляд систем-аналогів та технологій, що використовуються для запису на прийом до лікаря;
* Проєктування інтерфейсу та функціональних можливостей веб-додатку;
* Реалізація додатку з використанням Java, HTML, CSS;
* Тестування та валідація роботи веб-додатку;
* Формулювання висновків та рекомендацій щодо подальшого розвитку та вдосконалення додатку.

В наступних розділах розглянемо детальніше огляд систем-аналогів та технологій, проєктування та реалізацію веб-додатку, а також проведемо тестування та надамо висновки щодо результатів роботи.

1.ОГЛЯД СИСТЕМ-АНАЛОГІВ ТА ТЕХНОЛОГІЙ ЇХ РОЗРОБКИ

1.1 Огляд існуючих систем для запису на прийом до лікаря

На сучасному ринку існує безліч систем для запису на прийом до лікаря, які надають різні функціональні можливості та забезпечують зручний доступ до медичних послуг. Найпоширеніші системи для запису на прийом включають:

Електронний реєстр пацієнтів: Ця система дозволяє пацієнтам створювати облікові записи та записуватись на прийом до лікаря в онлайн-режимі. Вона забезпечує зручний і швидкий доступ до інформації про лікарів, їхній розклад роботи та доступність вільних часових слотів для запису.

Мобільні додатки для запису на прийом: Ці додатки дозволяють пацієнтам записуватись на прийом до лікаря безпосередньо зі своїх смартфонів. Вони надають зручну інтерфейсну навігацію, нагадування про прийоми та можливість перегляду медичної історії.

Веб-платформи для запису на прийом: Ці системи дозволяють пацієнтам реєструватись та записуватись на прийом до лікаря через веб-сайт. Вони надають широкі можливості для пошуку лікарів за різними параметрами (спеціалізація, місцезнаходження тощо) та забезпечують зручну систему календаря для вибору вільного часу.

1.2 Використання мови програмування Java для розробки веб-додатків

Мова програмування Java є однією з найпопулярніших мов для розробки веб-додатків. Вона відома своєю надійністю, переносимістю і широким спектром доступних фреймворків та бібліотек. За допомогою Java можна створити потужний веб-додаток з багатофункціональним інтерфейсом, який забезпечує безпеку та ефективність роботи.

Java пропонує різноманітні фреймворки, такі як Spring, які спрощують розробку веб-додатків шляхом надання готових компонентів та вбудованих функцій, що полегшують роботу з базами даних, взаємодію з користувачем та керуванням сесіями.

1.3 Використання HTML та CSS для створення інтерфейсу веб-додатків

Мови розмітки HTML (HyperText Markup Language) і CSS (Cascading Style Sheets) використовуються для створення інтерфейсу веб-додатків. HTML відповідає за структуру сторінки, визначаючи різні елементи, такі як заголовки, параграфи, форми та кнопки. CSS відповідає за стилізацію і зовнішній вигляд сторінки, включаючи кольори, шрифти, розташування елементів і т.д.

За допомогою HTML та CSS можна створити привабливий та інтуїтивно зрозумілий інтерфейс для веб-додатку для запису на прийом до лікаря. Завдяки їхній гнучкості і широкому спектру можливостей, можна легко керувати виглядом та розміщенням елементів на сторінці, створюючи приємний користувацький досвід.

2.ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ ЗАПИСУ НА ПРИЙОМ ДО ЛІКАРЯ

2.1 Мета та завдання веб-додатку

Мета веб-додатку для запису на прийом до лікаря: надання зручного та ефективного інструменту для пацієнтів, що дозволить їм легко та швидко записатись на прийом до лікаря. Головним завданням такого додатку є полегшення процесу запису на прийом, зменшення навантаження на адміністративний персонал медичного закладу та забезпечення зручності для пацієнтів.

Додаток повинен забезпечувати наступні завдання:

* Реєстрація та авторизація пацієнтів: Додаток має надати можливість користувачам створювати облікові записи та увійти в систему за допомогою своїх облікових даних.
* Перегляд розкладу лікарів: Пацієнти повинні мати змогу переглядати розклад роботи лікарів, доступні часові слоти, щоб обрати зручний для себе варіант.
* Запис на прийом: Користувачі мають мати можливість обрати доступний часовий слот, інформацію про лікаря.
* Скасування запису: Пацієнти повинні мати можливість скасувати свій запис на прийом у разі потреби.

2.2 Визначення функціональних вимог до веб-додатку

User Story

1. Як користувач, я хочу мати можливість обрати лікаря за його спеціальністю, щоб знайти лікаря з потрібними знаннями.
2. Як користувач, я хочу мати можливість бачити основну інформацію про лікаря, щоб щоб вирішити, хто мені найбільше підходить.
3. Як користувач, я хочу мати можливість переглянути розклад роботи лікаря, щоб знайти зручний час для запису на прийом.
4. Як користувач, я хочу мати можливість бачити заброньований час у розкладі лікаря, щоб знати який час можу забронювати.
5. Як користувач, я хочу мати можливість скасувати запис на прийом, якщо з'являться непередбачені обставини.
6. Як користувач, я хочу мати можливість зайняти обране місце, щоб забронювати місце у розкладі лікаря.

2.3 Розробка бази даних для зберігання інформації про лікарів та пацієнтів

*1)Виділені суттєві сутності, що характеризують предметну область, та їх характеристика.*

|  |  |
| --- | --- |
| Сутність | Характеристика |
| Лікар | Містить інформацію про: П.І.Б та спеціалність за якою працює лікар. |
| Клієнт | Містить інформацію про: прізвище, ім'я, по батькові, email, домашня адреса, пароль до додатку. |
| Сеанс | Містить інформацію про дані про сеанси в розкладі всіх лікарів. |

2) Опис формалізації зв’язків у відповідності з правилами формалізації.

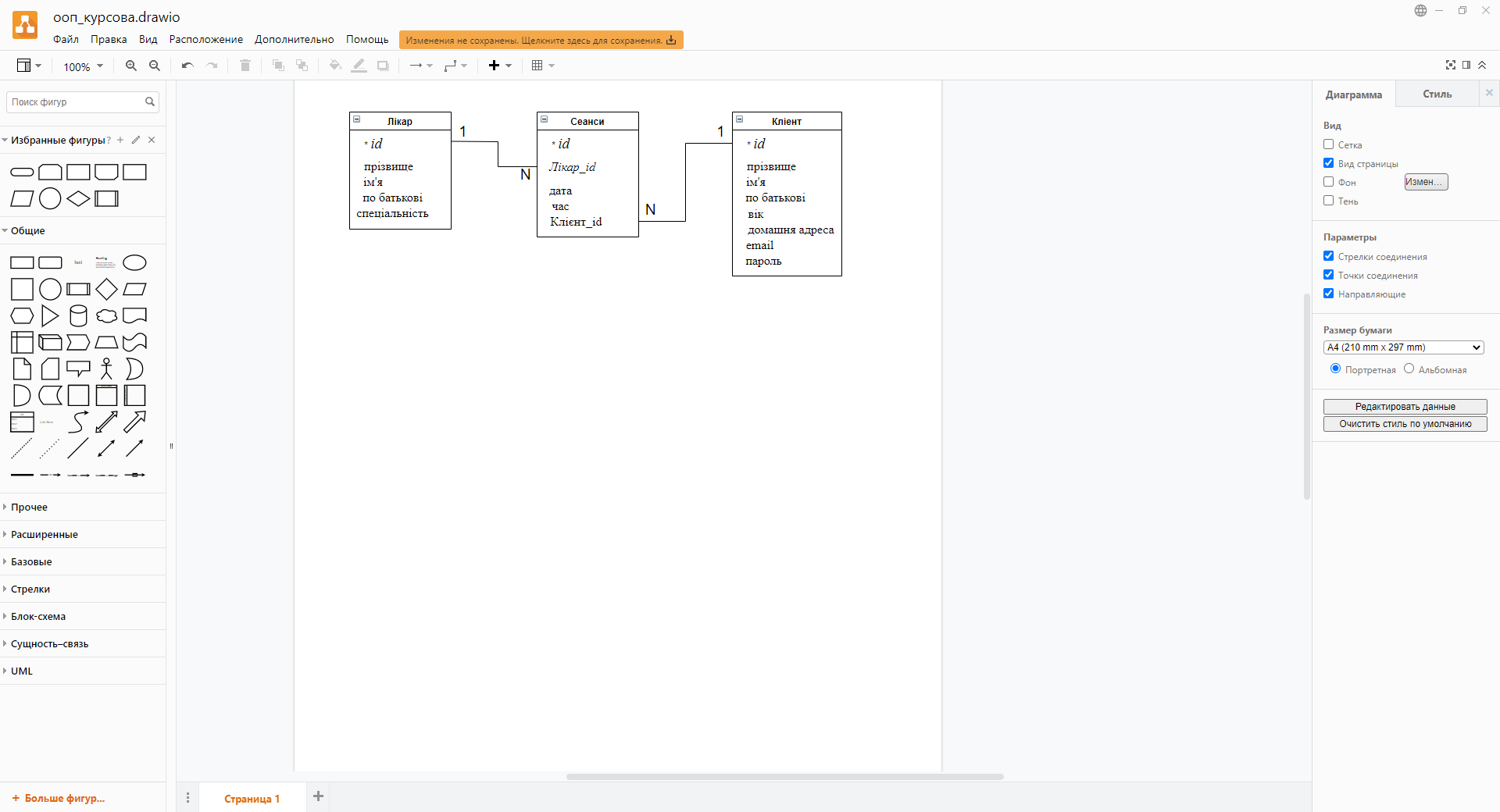
Для створення бази даних необхідно визначити зв’язки між отриманими відношеннями. Наприклад, кожний лікар може мати багато кліентів так і один кліент може мати багато лікарів, але у певний сеанс лікар може мати лише одного кліента, а також кліент у один сеанс може потрапити до одного лікаря.Отже, утворюється зв’язок багато до багатьох між сутностями Лікар та Кліент, де проміжною сутністю є Сеанс. Таким чином утворюється зв’язок один до багатьох між сутностями Кліент та Сеанс й аналогічно Лікар та Сеанс.

*Формалізуємо дані зв’язки*. У даному ввипадку зв’язки між таблиці можна побудувати наступним чином: первинні ключі Кліент та Сеанс додати до відношення Сеанси як зовнішні ключі.

2) Виділені властивості об’єктів (атрибути) з зазначенням зовнішніх ключів

|  |  |  |  |
| --- | --- | --- | --- |
| Сутність | Властивість | Тип даних | Ключ |
| Лікар | *id* | *цілочисельний* | *первинний* |
|  | прізвище | Символьний |  |
|  | ім'я | Символьний |  |
|  | по батькові | Символьний |  |
|  | спеціальність | Символьний |  |
| Кліент | *id* | *цілочисельний* | *первинний* |
|  | прізвище | Символьний |  |
|  | ім'я | Символьний |  |
|  | по батькові | Символьний |  |
|  | вік | цілочисельний |  |
|  | домашня адреса | Символьний |  |
|  | email | Символьний |  |
|  | пароль | Символьний |  |
| Сеанси | *id* | *цілочисельний* | *первинний* |
|  | *Лікар\_id* | *цілочисельний* | зовнішній |
|  | дата | DATE |  |
|  | час | TIME |  |
|  | Клієнт\_id | цілочисельний | зовнішній |

4) Представлення схеми даних з зазначенням нотації.



мал.1- схема даних з визначеними зв’язками.

Схема зроблена в draw.io.

Для роботи з базою даних використовувалося середовище PostgreSQL PgAdmin4. Далі наведено SQL-запити створення відповідних таблиць:

SQL-запити:

--таблиця лікарів

create table doctor(

id INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

last\_name VARCHAR(50) NOT NULL,

first\_name VARCHAR(50) NOT NULL,

middle\_name VARCHAR(50) NOT NULL,

specialty VARCHAR(30)

);

--таблиця кліентів

create table client(

id INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

last\_name VARCHAR(50) NOT NULL,

first\_name VARCHAR(50) NOT NULL,

middle\_name VARCHAR(50) NOT NULL,

age INTEGER,

address VARCHAR(100),

email VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL

);

--таблиця сеансів

create table sessions(

id INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

doctor\_id INTEGER REFERENCES doctor(id),

date DATE,

time TIME,

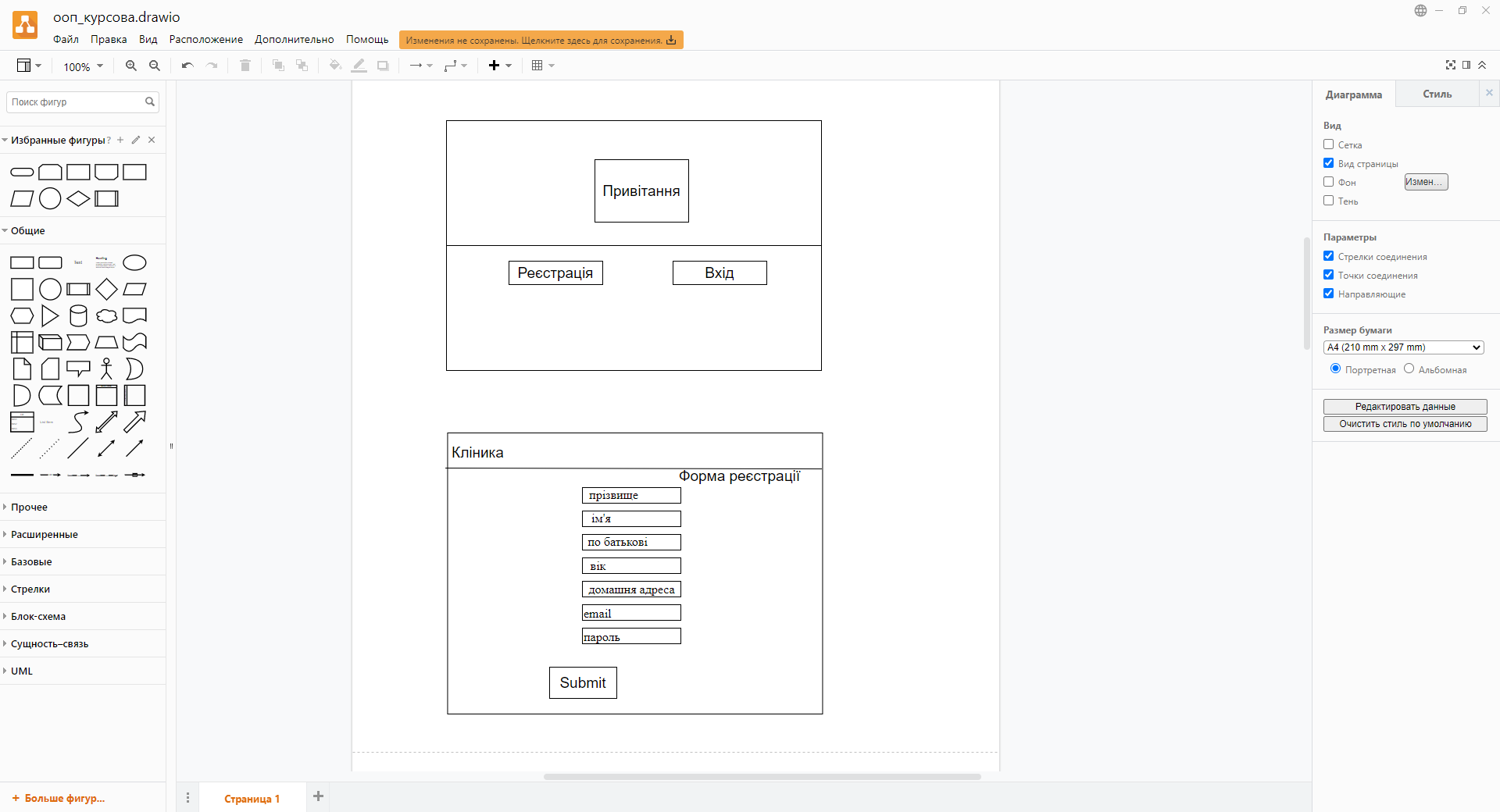
client\_id INTEGER REFERENCES client(id)

);

2.4 Проектування користувацького інтерфейсу веб-додатку

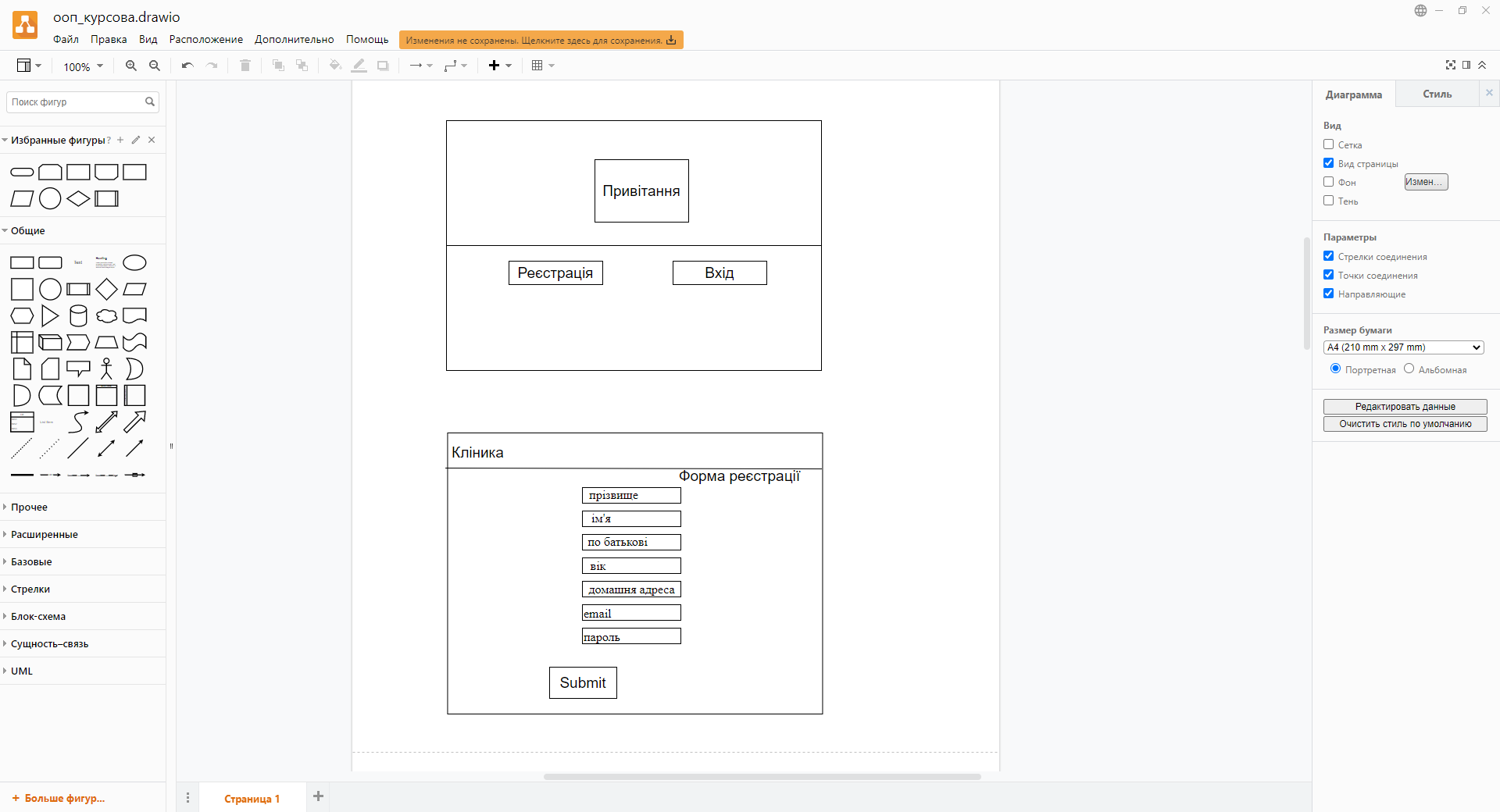
Далі буде наведено макети сторінок, які буде бачити користувач:

При початку роботи, буде видно головну сторінку:



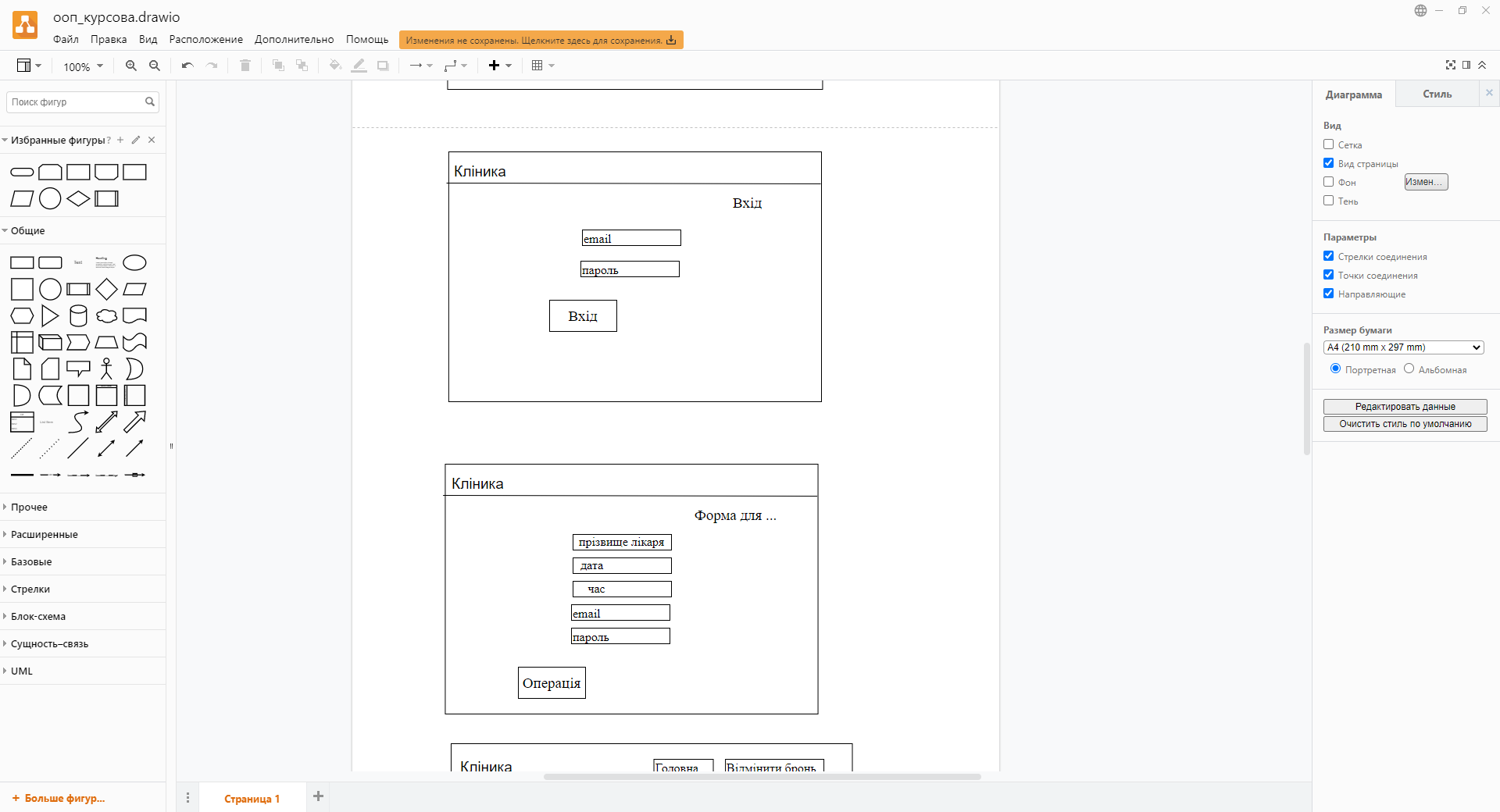
мал.2- макет головної сторінки.

Якщо обрати кнопку реєстрації, з’явиться форма типу:



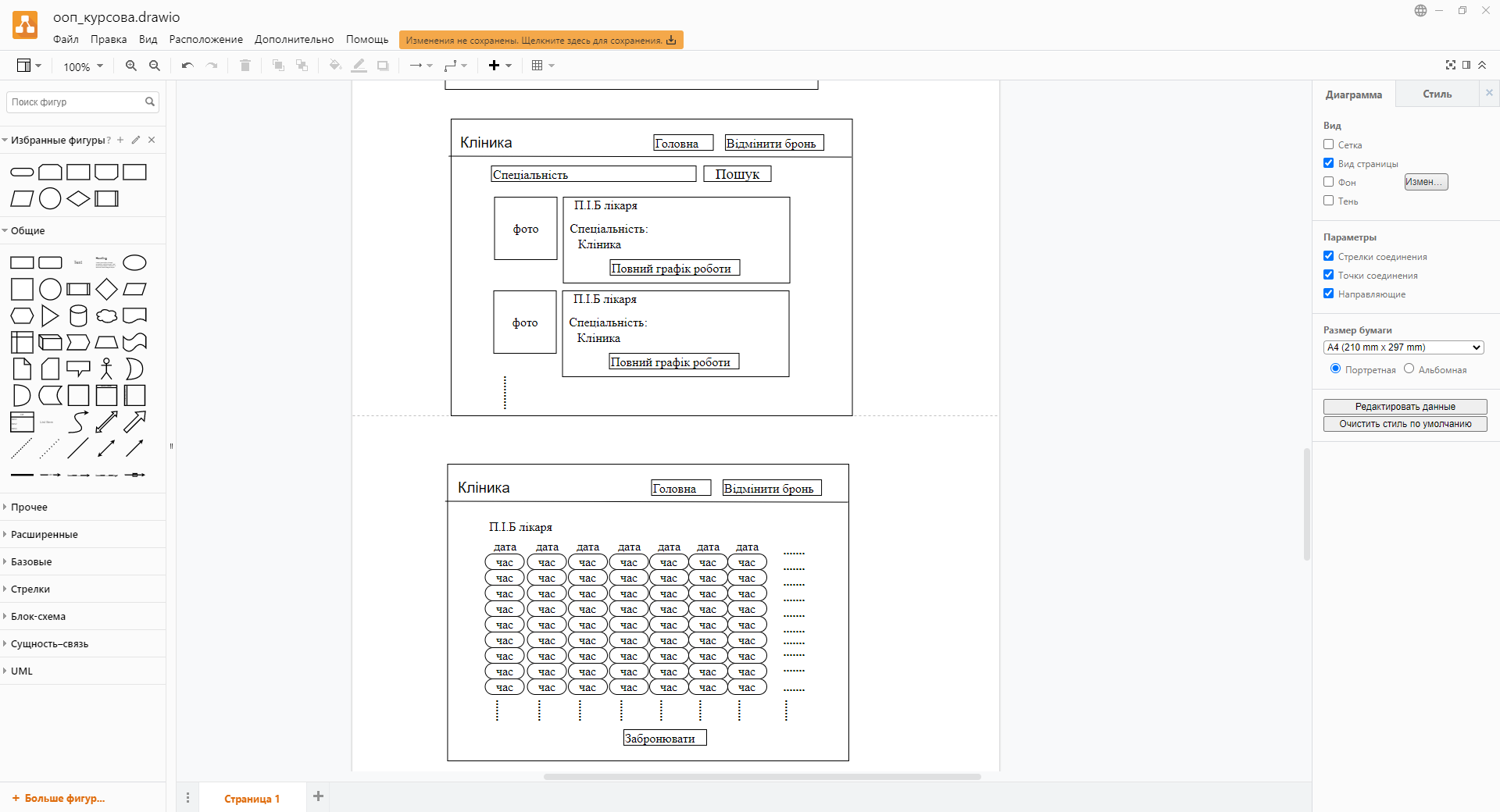
мал.3- макет сторінки-форми для реєстрації.

Якщо обрати кнопку входу, з’явиться форма типу:



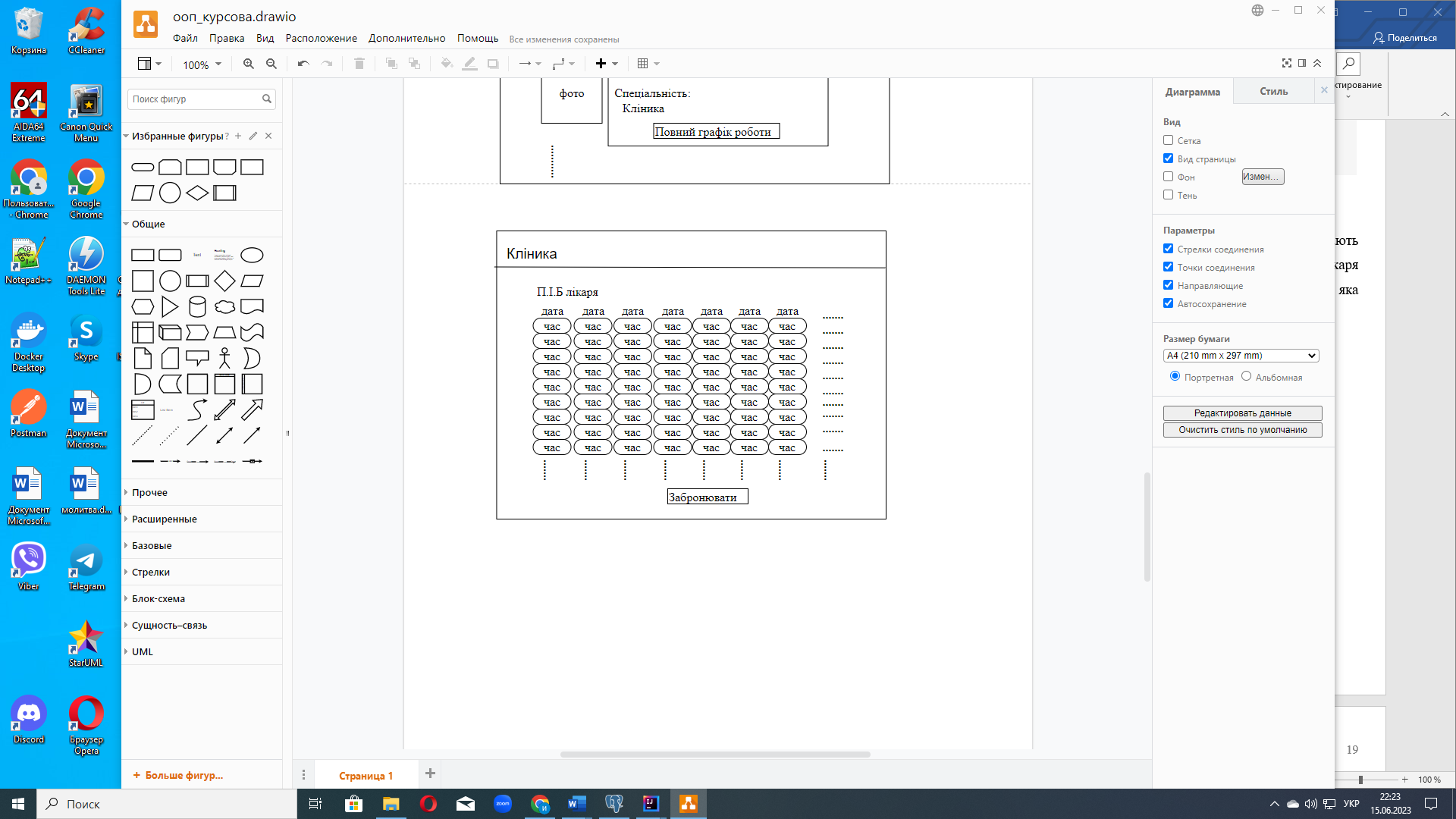
мал.4- макет сторінки-форми для входу у додаток.

Після входу перейдемо до сторінки меню.



мал.5- макет сторінки меню.

Якщо натиснути кнопку ’’Повний графік роботи’’, перейдемо на сторінку, де відображен графік роботи відповідного лікаря та показано, які з сеансів вже заброоньовані.

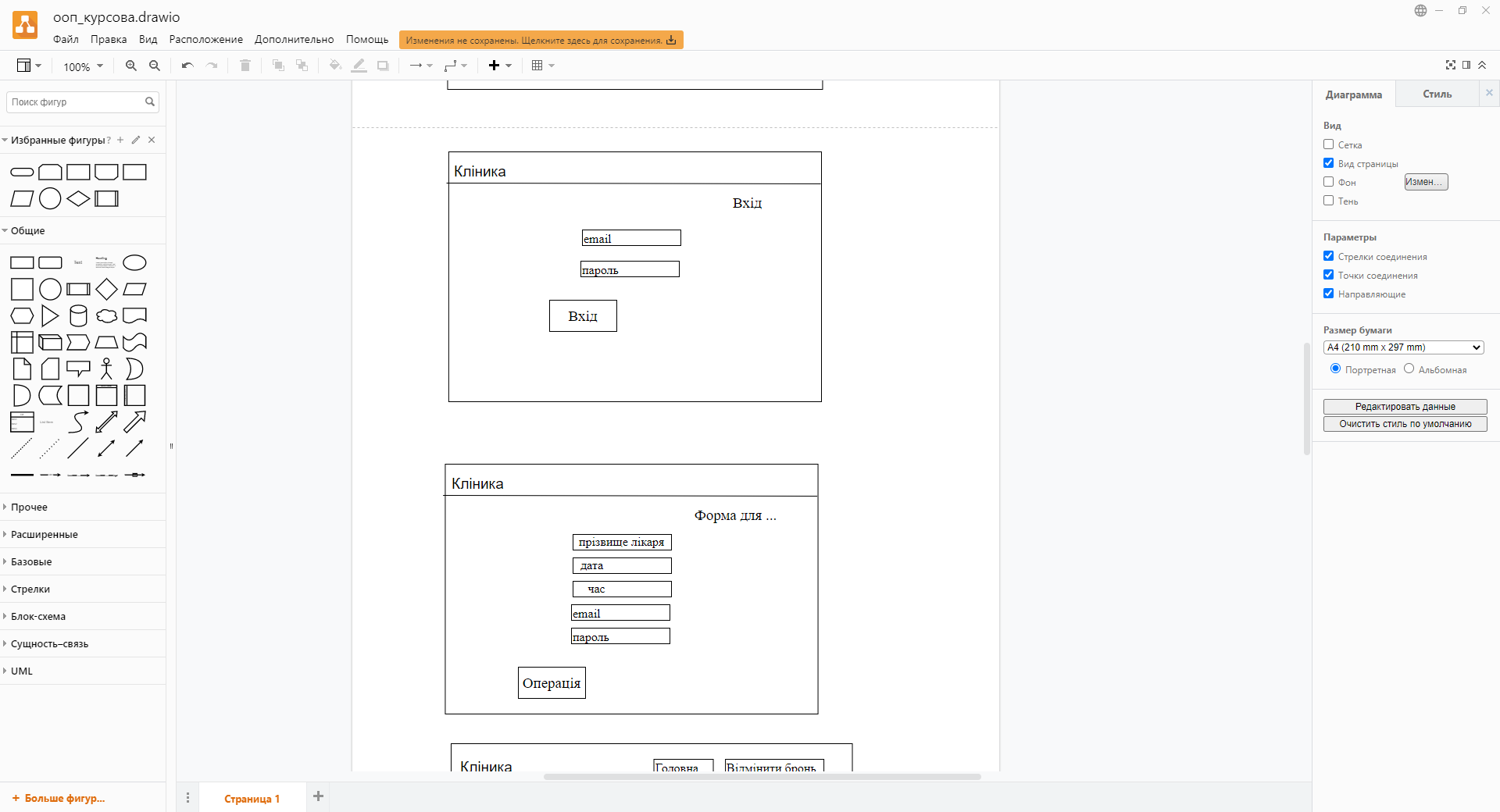


мал.6- макет сторінки з графіком роботи відповідного лікаря.

Якщо користувач хоче забронювати сеанс, який не заброньовано у графіку, йьому слід натиснути кнопку ’’Забронювати’’ (мал.6).

Якщо користувач хоче відмінити бронь на сеан, він може натиснута на кнопку ’’Відмінити бронь’’ та перейти до відповідної форми.

Форми для бронювання та його відміни мають однакову структу, яка представлена далі:



мал.7- макет сторінки-форми для юронювання/відміни броні.

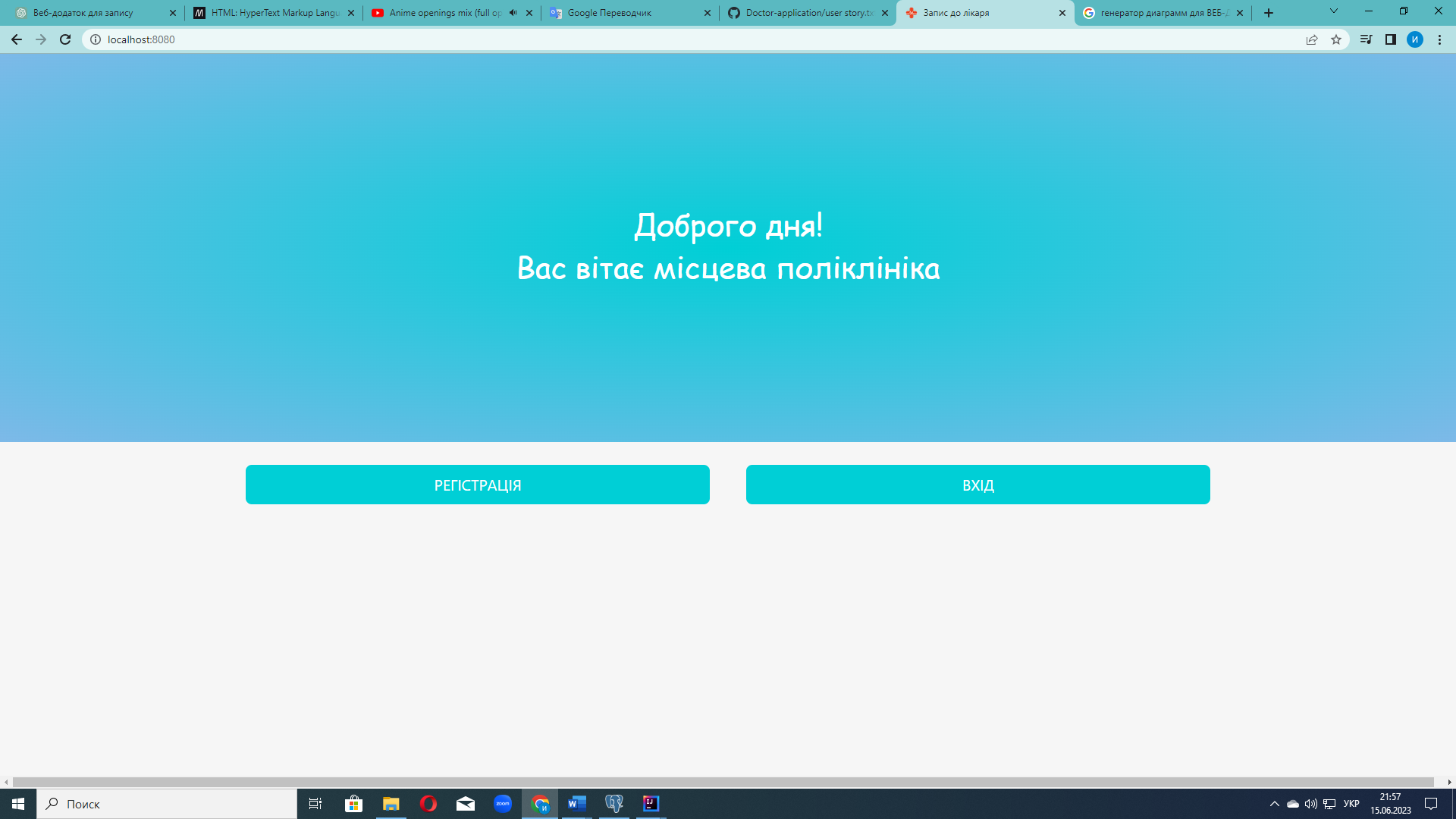
2.5 Реалізація логіки запису на прийом та управління даними

За створеною логікою можна визначити за які процеси відповідає кожна з сутностей у додатку. Отже, Doctor відповідає за зберігання та вивод інфорамції про лікарів, а також в ній реалізовано сортівання лікарів за спеціальністю для легшого пошуку лікаря потрібного напрямку. Client відповідає за реєстрацію (створення нового кліенту у базі даних) та вході, тобто перевірці чи існує такий користувач у базі даних. Sessions відповідає за бронювання сенасну та відміни броні, У базі даних таблиця зберігає сеанси, які заброньовано, посилаючись на таблиці лікаря та кліента. Коли бронь відміняють відповідна строка у таблиці сеансів видаляється.

3.ТЕСТУВАННЯ ВЕБ-ДОДАТКУ

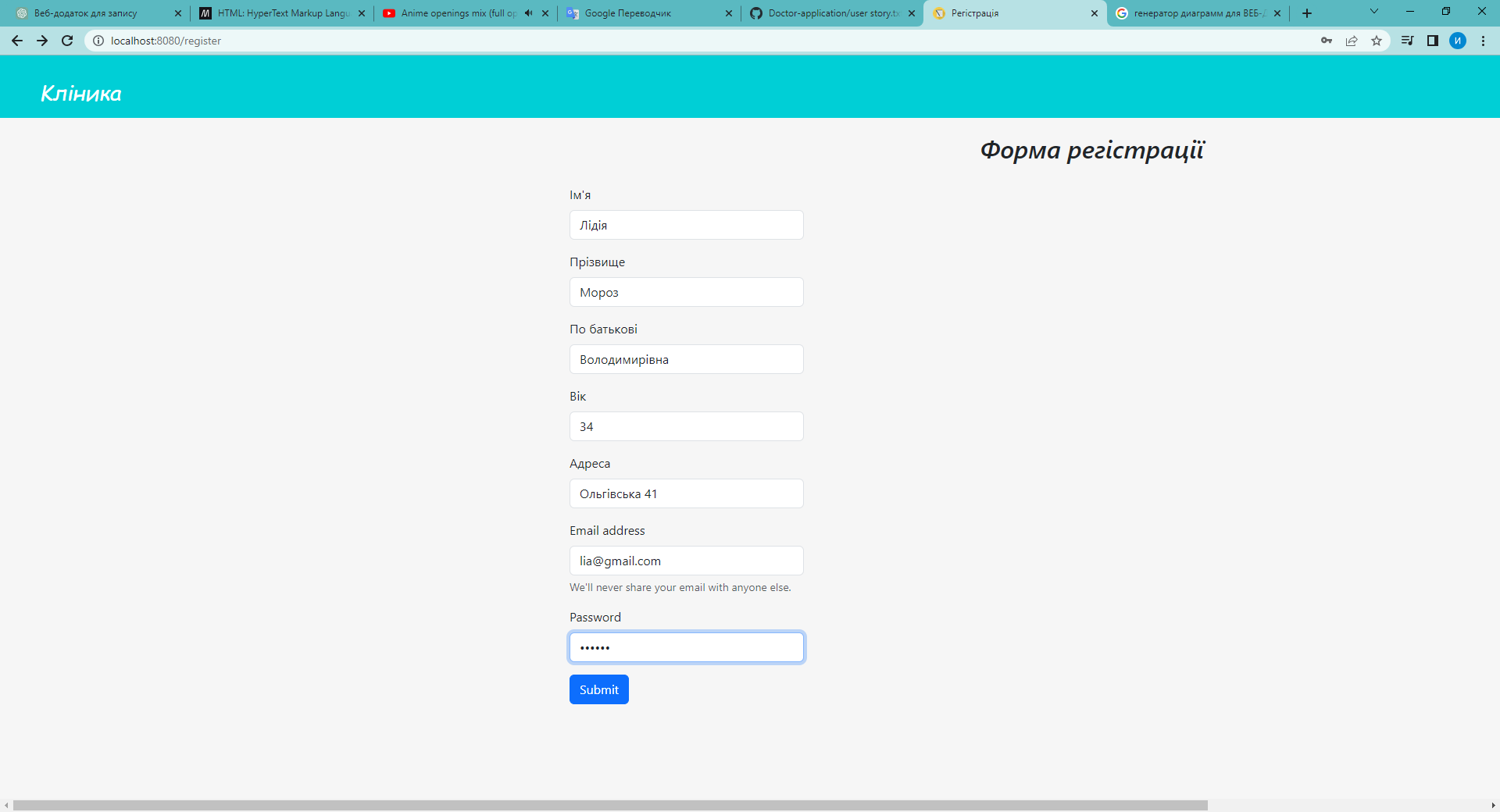
Тепер наведемо приклад роботи додатку.

Початок роботи:



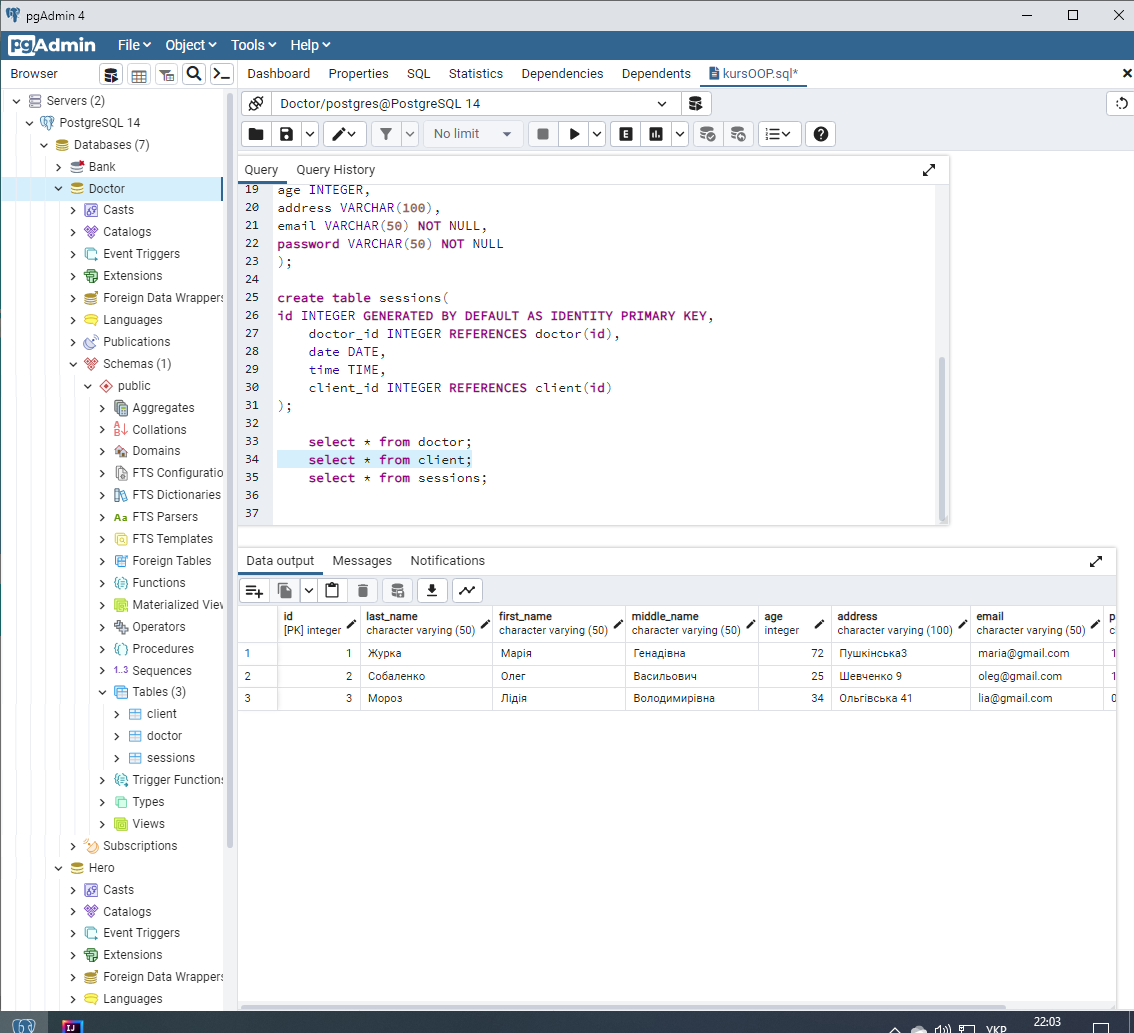
мал.8- вигляд головної сторінки.

Створимо нового користувача, для цього натиснемо ’’Регістрація’’:



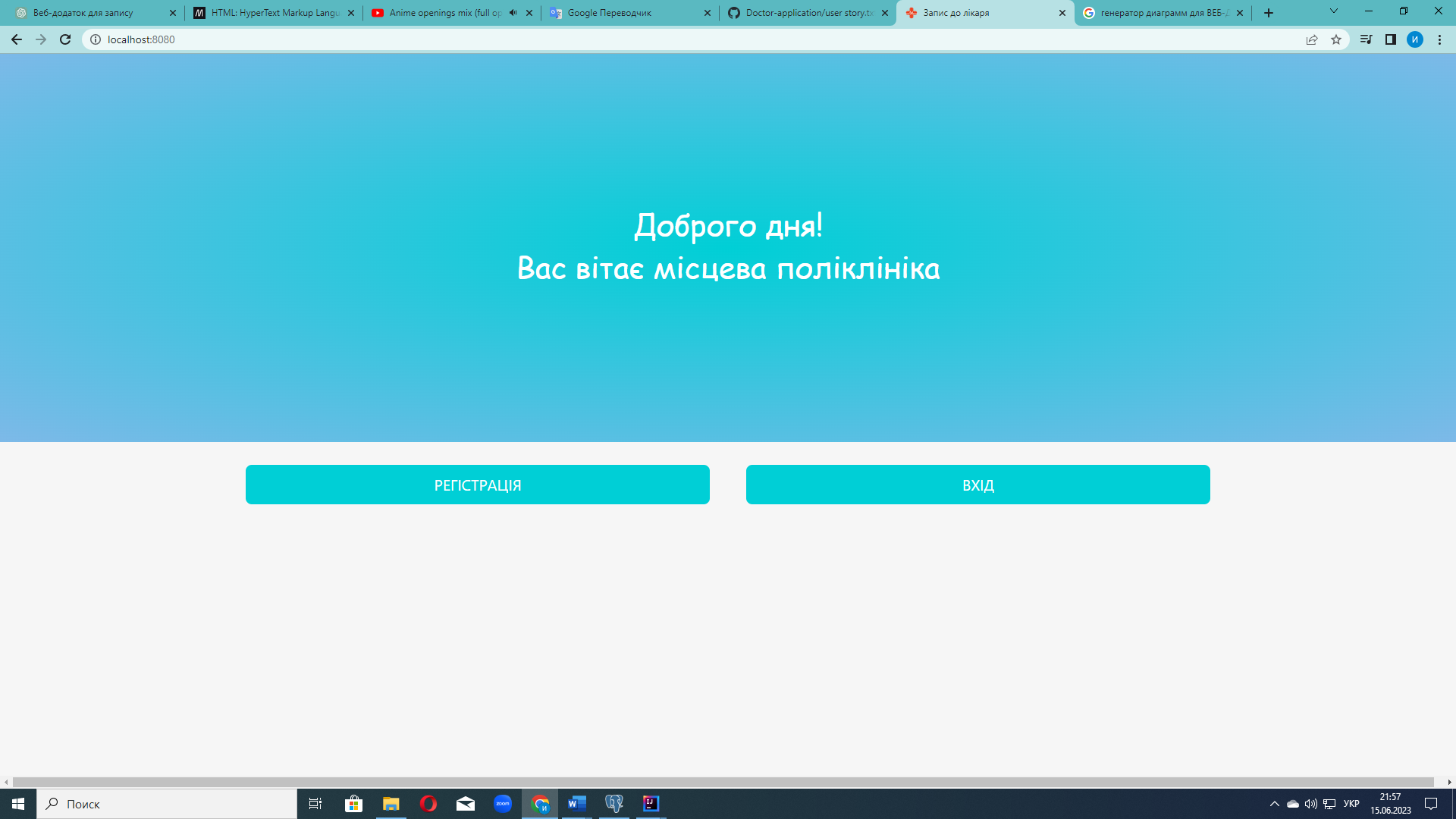
мал.9- вигляд заповненої форми для регістрації.

При відправлення форми (мал.9) у базі даних в таблиці кліентів стрворився запис про нового користувача.



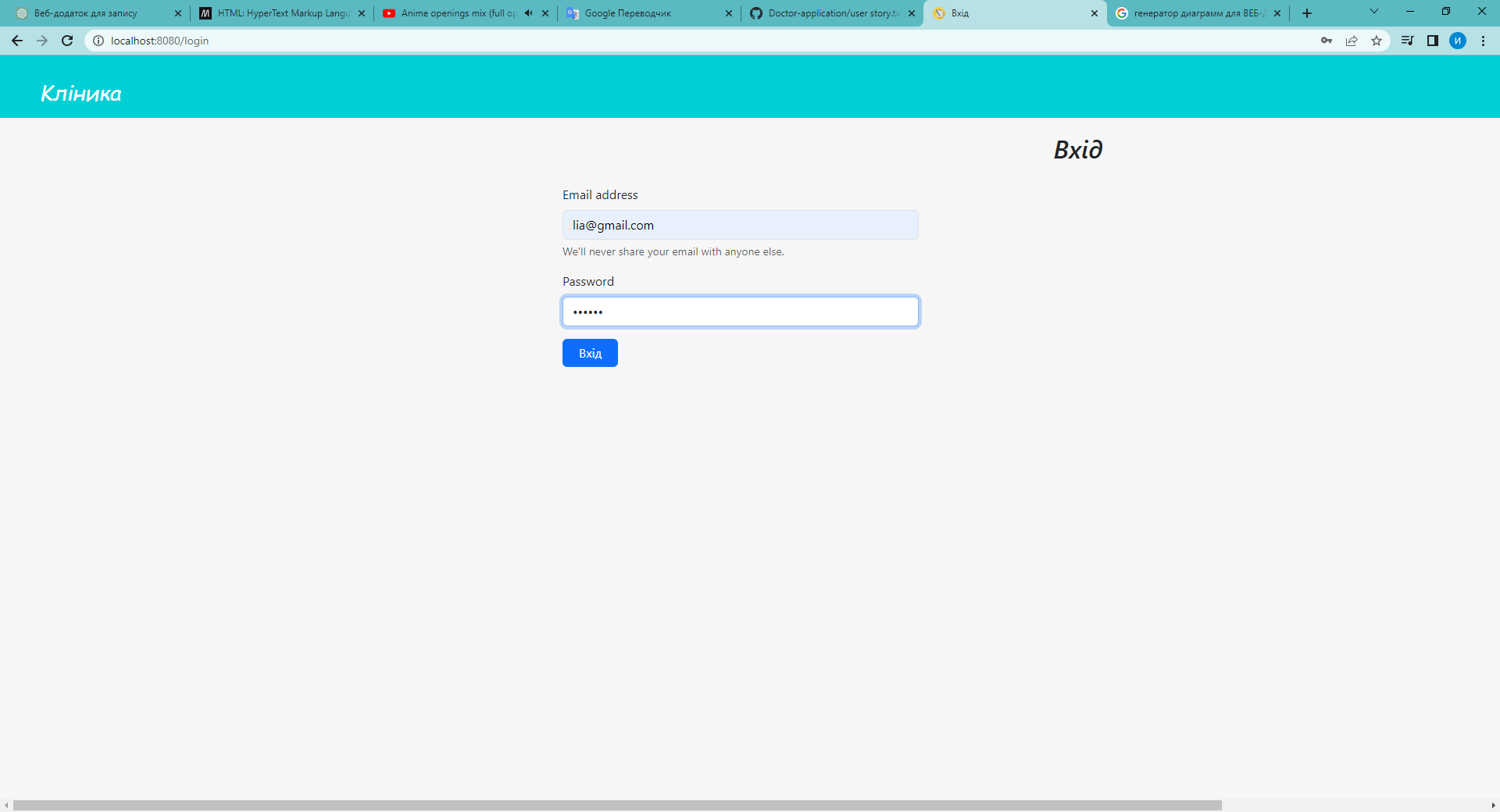
мал.10- таблиця кліентів у базі даних з новим користувачем.

Після відправленя форми ми повернулися на головну сторінку:



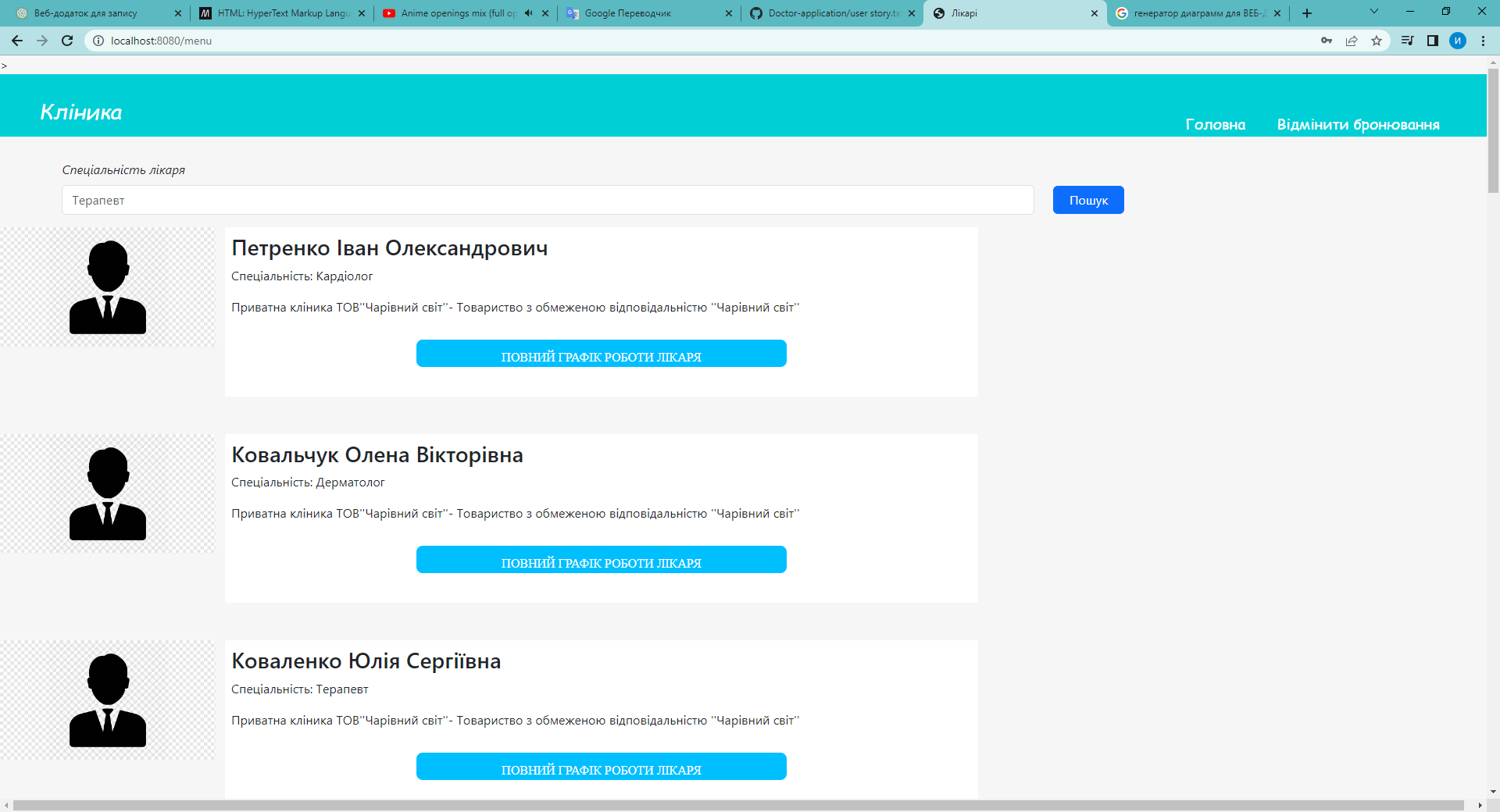
мал.10- повернення на головну сторінку.

Тепер потрібно виконати вхід:



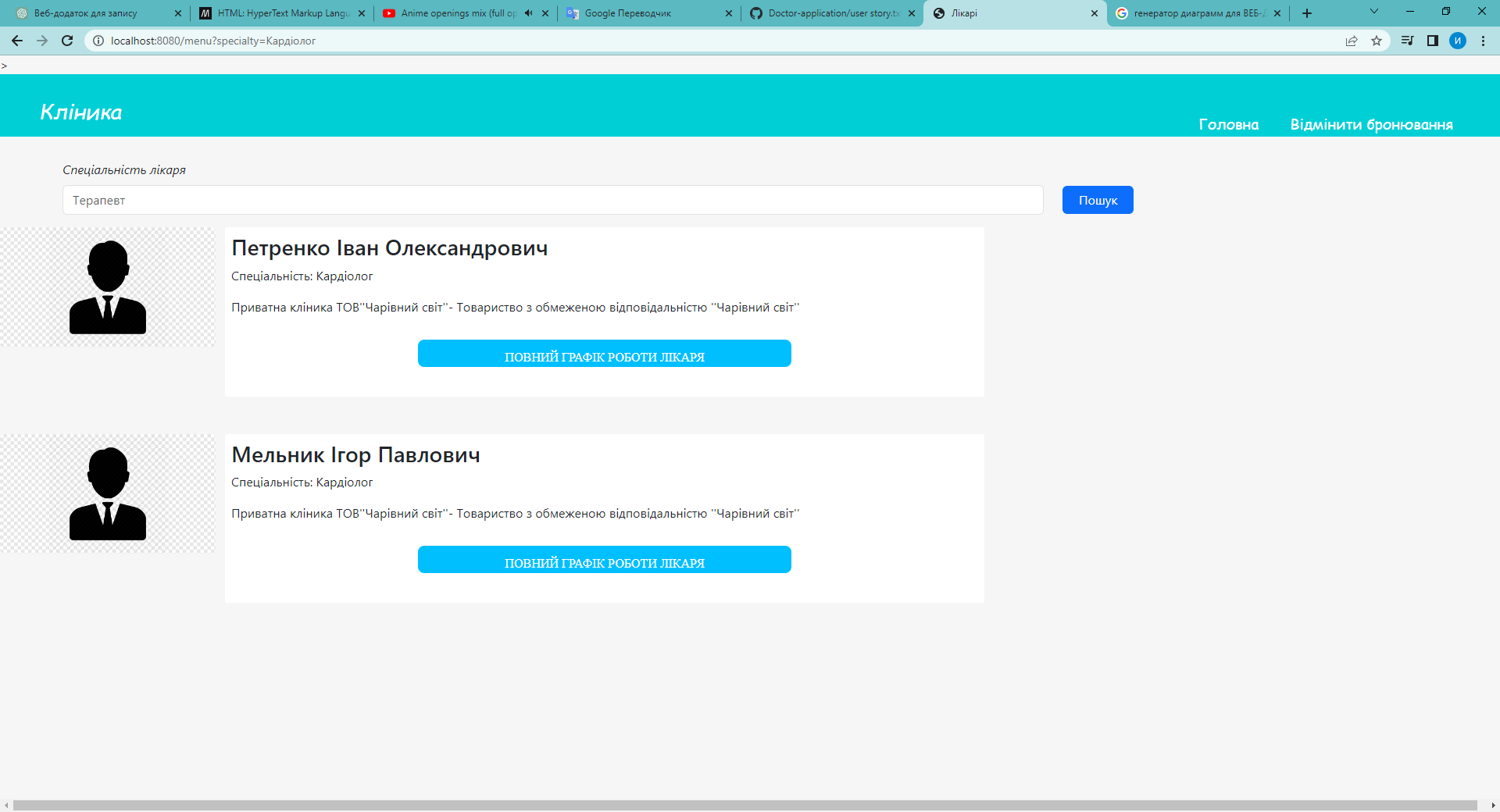
мал.11- Вигляд заповненої форми входу у додаток.

Після відправлення даної форми(мал.11), переходимо на сторінку меню:



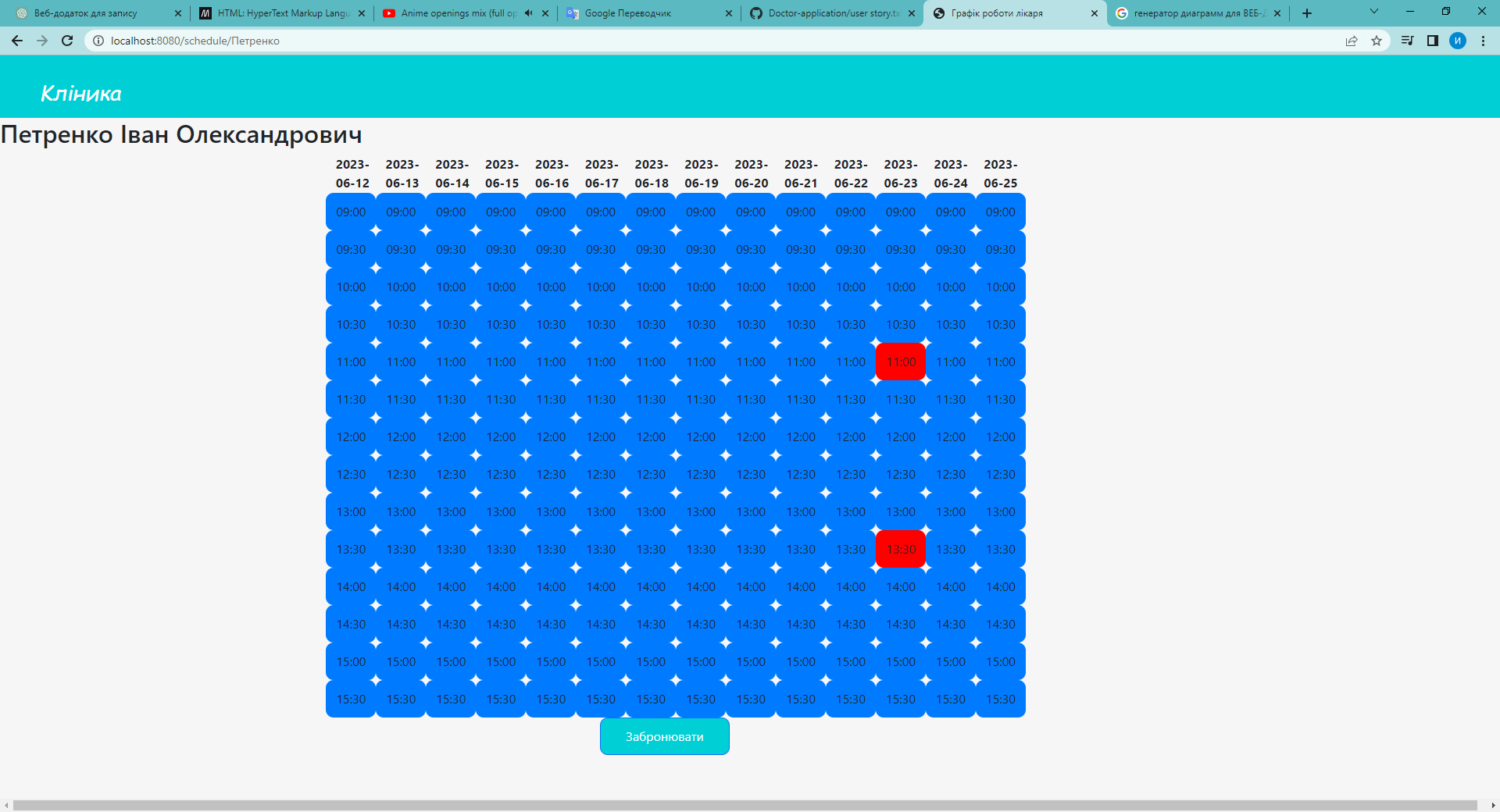
мал.12- Вигляд сторінки меню.

Бачимо, що на даний моменти виведені усі лікарі кліники. Якщо ми хочемо бачити лікарів лише потрібної спеціальності, можемо скоритатися пошуковою строкою вгорі сторінки. Наприклад, нам потрібен кардіолог, тоді вводимо Кардіолог у пошукову строку та отримаємо:



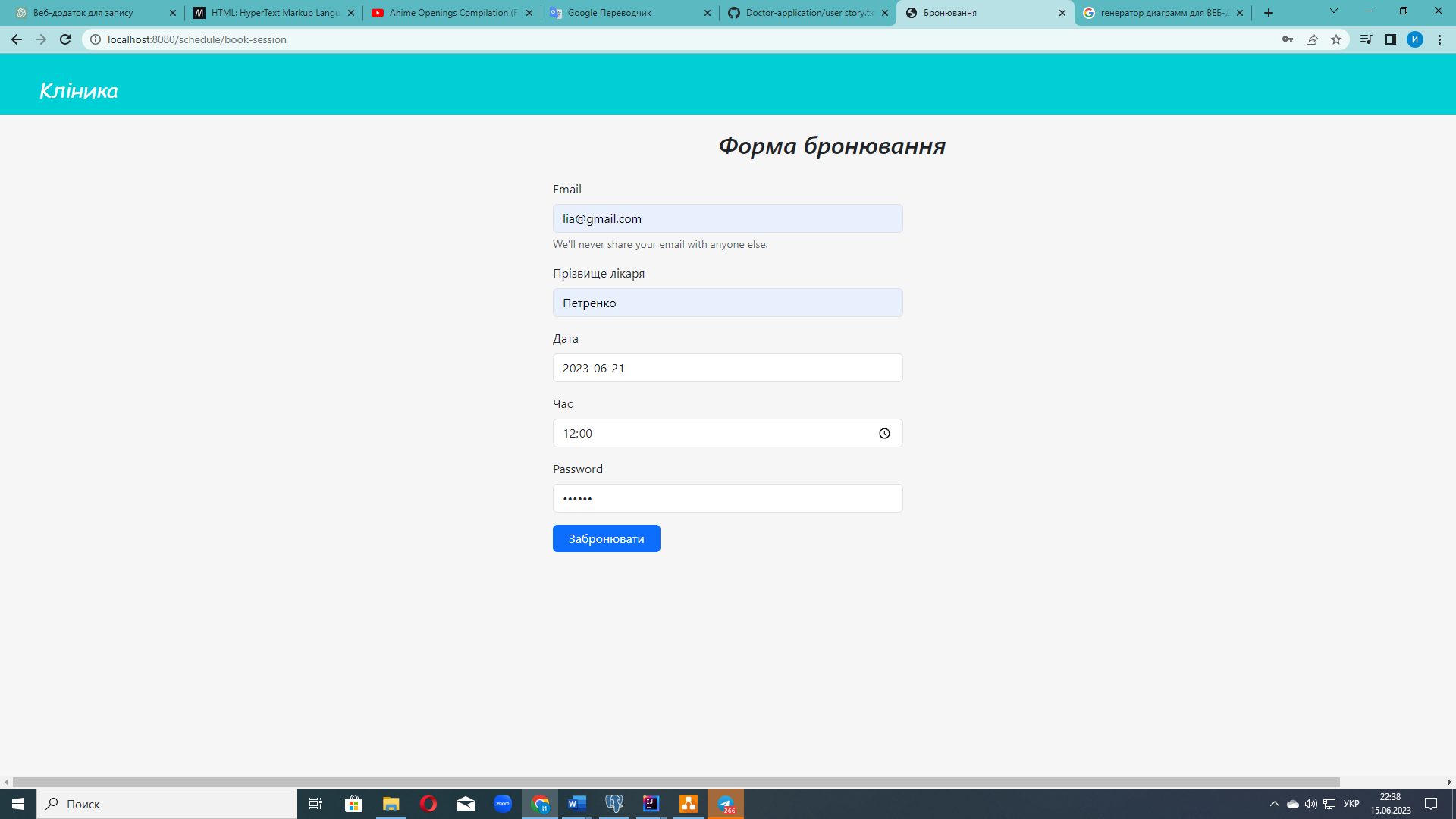
мал.12- Відфільтрований список лікарів.

Як бачимо, після фільтрації залишолося лише два лікаря та вони мають спеціальність Кардіолог. Тепер ,наприклад, хочемо забронювати сеанс до лікаря кардіолога Петренко, тоді натискаємо на кнопку ’’Повний графік роботи’’, яка знаходиться у його секції та переходимо на відповідну сторінку.



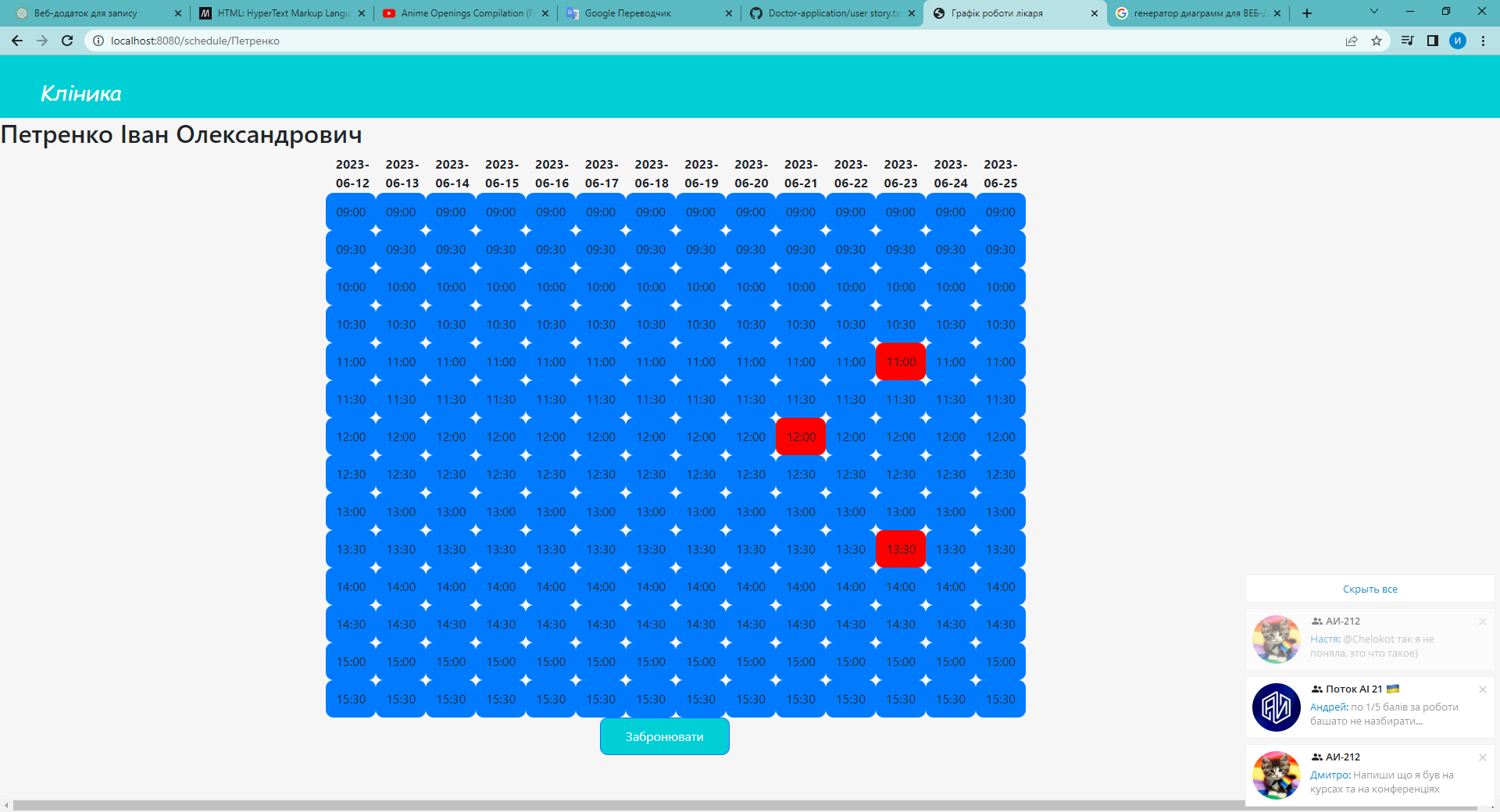
мал.13- Графік роботи лікаря Петренка.

Як видно на малюнку13, у обраного лікаря два сеанси на 23 червня, а саме на 11:00 та 13:30 відображаються іншим кольором. Це означає, що дані сеанси вже хтось забронював. Ми можемо записатись на будь-який сеанс, який на задньому фоні мають блакитний відтінок. Наприклад, забронюємо сеанс на 21 червня 12:00.



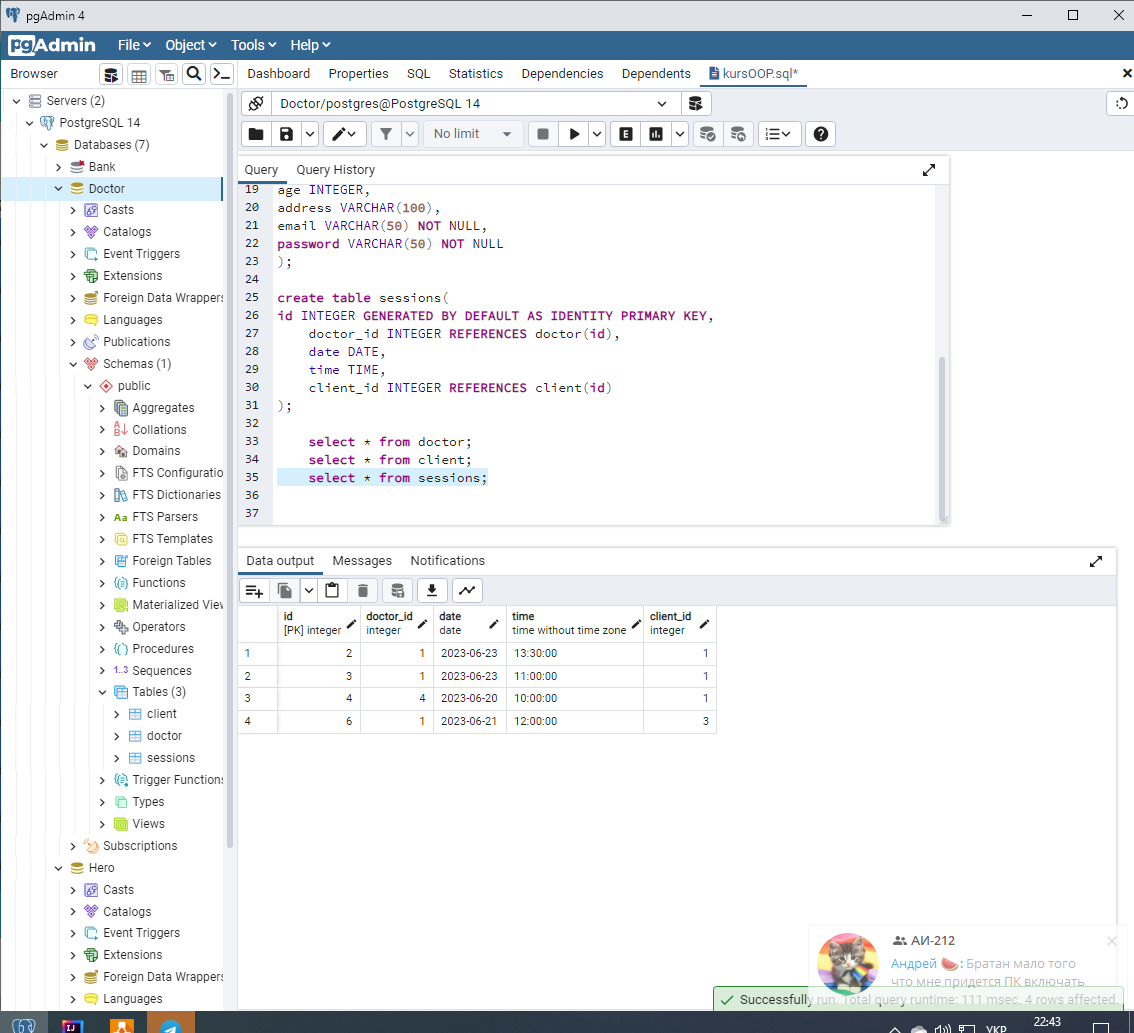
мал.14- Заповнена форма для бронювання.

Після відправлення форми, можемо побачити, що сеанс тепер заброньовано.



мал.15- Оновлений графік роботи Петренка після бронювання.

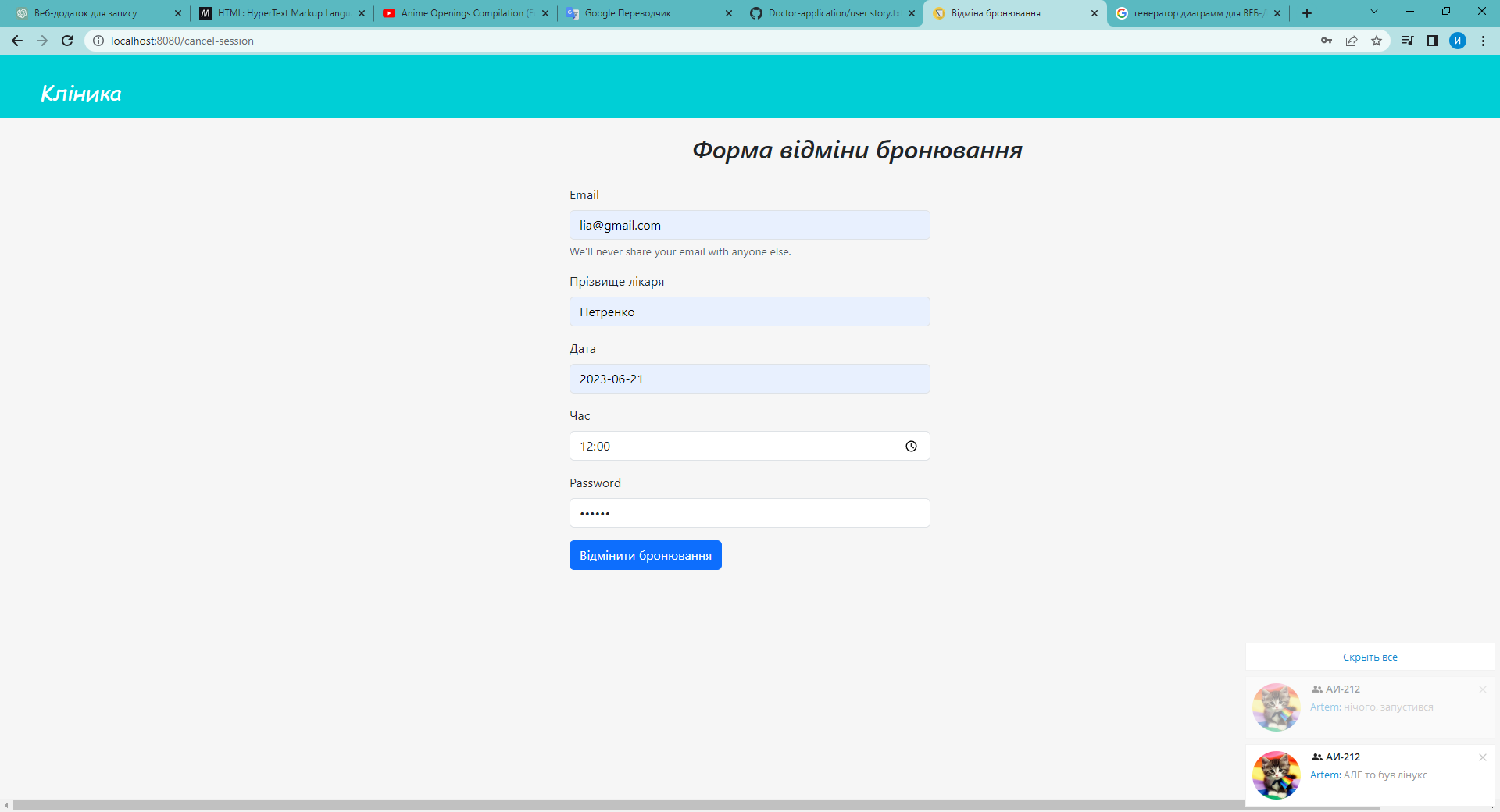
Також у базі даних до таблиці сеансів додався запис:



мал.16- Оновлена таблиця сеансів після бронювання.

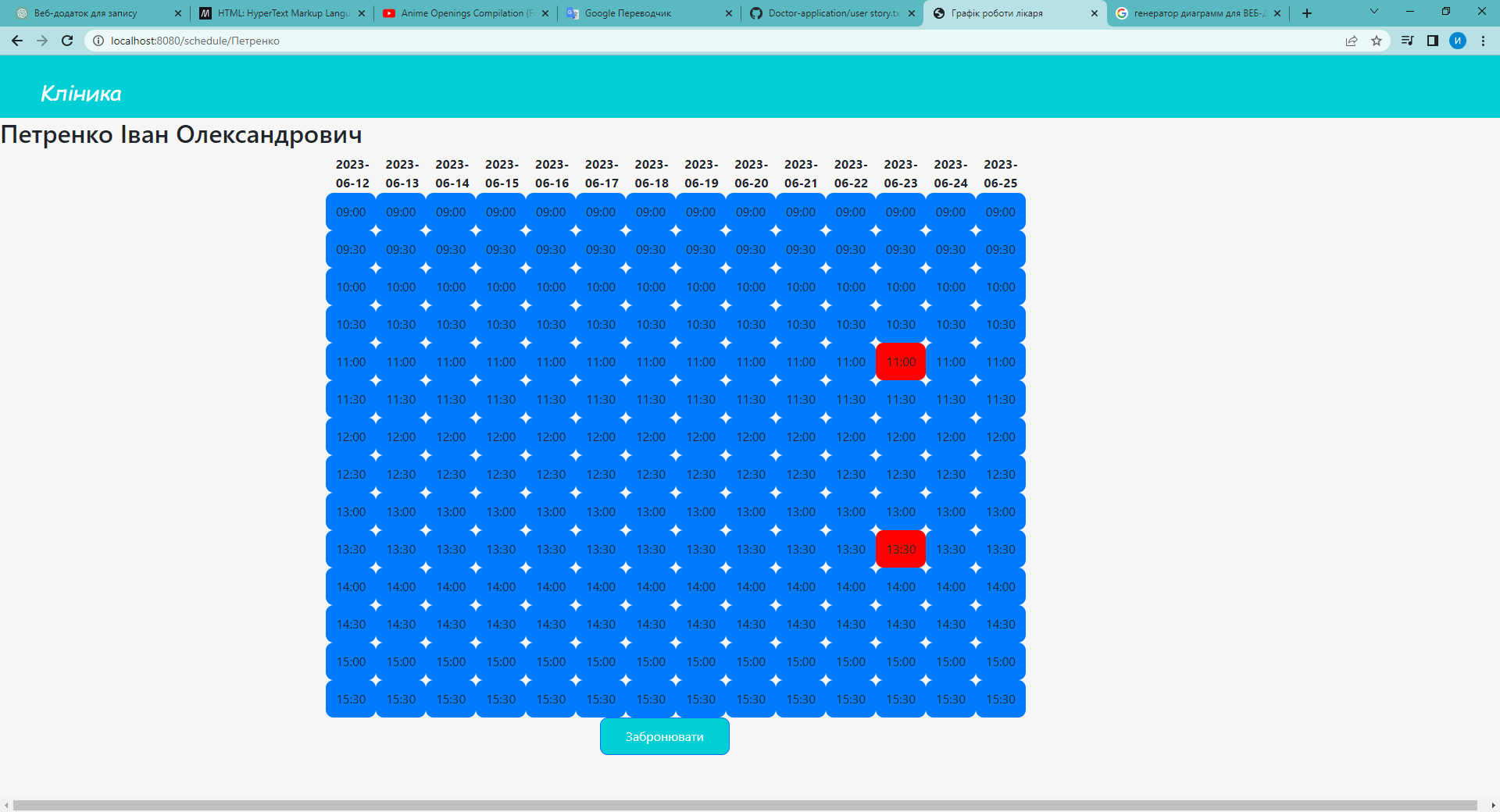
Як видно на малюнку 16, з’явився запис на 21 червня 12:00 до лікаря Петренка, який має id=1, клієнтом з id=3, тобто Лідія Мороз.

У випадку коли потрібно видалити бронь, потрібно повернутися на меню та натиснути кнопку ’’Відмінити бронь’’.

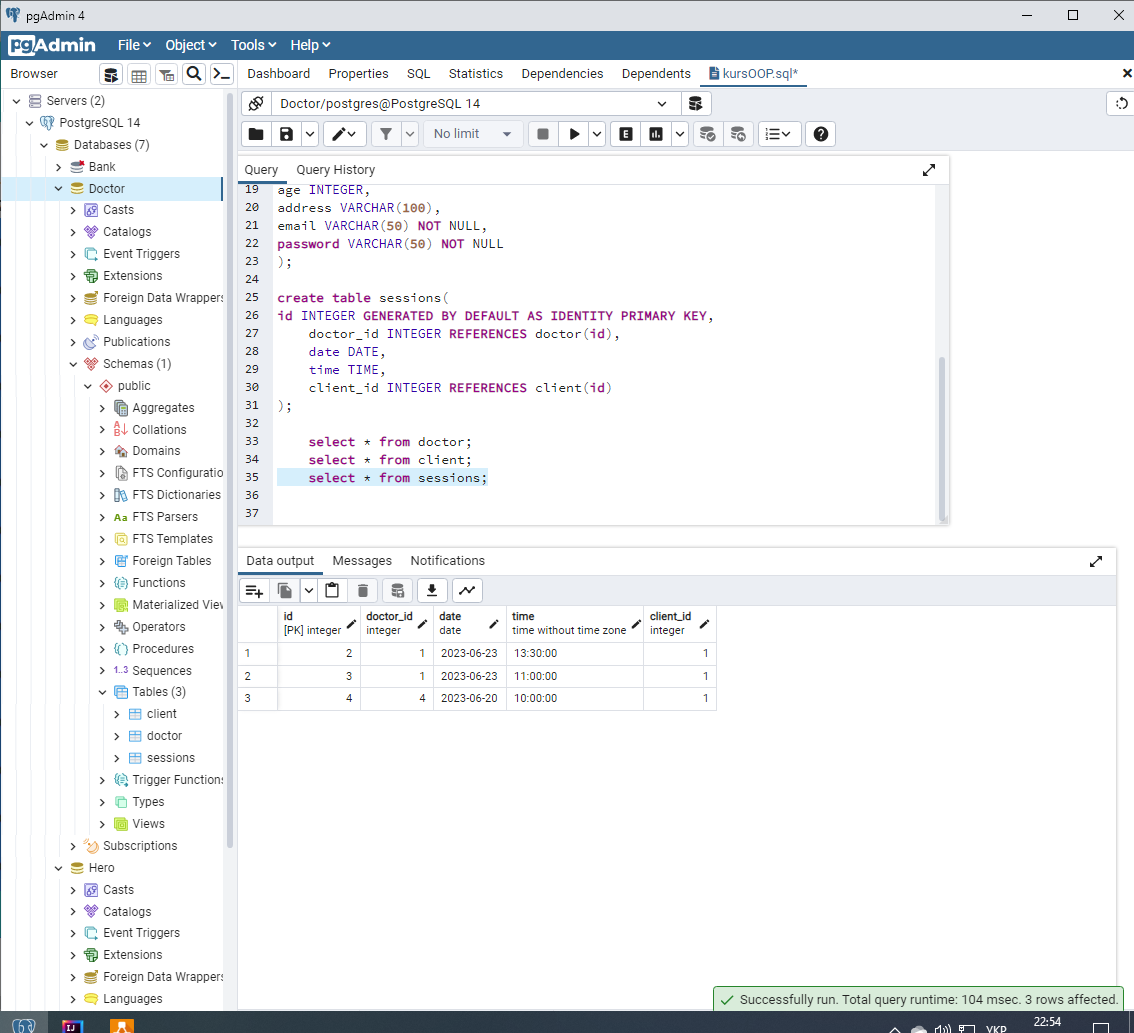


мал.17- Заповнена форма для відміни броні.

Після відправлення броні, як видно з малюнків 18 та 19 нижче, заброньований раніше сеанс видалено з бази даних та він більше не відображається у розкладі лікаря Петренка.



мал.18- Графік роботи Петренка після відміни броні.



мал.19- Оновлена таблиця сеансів після відміни броні.

Отже, я видно з навеленого прикладу, додаток працює та добре виконує поставлені завдання.

ВИСНОВКИ

В даній курсовій роботі був розроблений веб-додаток для запису на прийом до лікаря. Метою проекту було створення зручного та ефективного інструменту, що дозволить пацієнтам легко та швидко записатися на прийом до лікаря.

В ході роботи були вирішені наступні завдання:

* Був проведений огляд існуючих систем для запису на прийом до лікаря, досліджено їх функціонал та особливості. Це дало змогу зрозуміти ключові вимоги та потреби користувачів у такому додатку.
* Використовуючи мову програмування Java, була розроблена серверна частина веб-додатку. Це дало змогу забезпечити обробку та збереження даних, необхідних для запису на прийом, а також реалізувати логіку взаємодії з базою даних.
* Застосовуючи HTML та CSS, був розроблений користувацький інтерфейс веб-додатку. Це дало змогу створити зручний та привабливий інтерфейс, який дозволяє користувачам з легкістю записатися на прийом та керувати своїми даними.

Також було наведено приклад функцірнування додатку, який складався з зображень та відповідних пояснень до них. Після наведеня прикладу роботи додатку можна сказати, що додаток виконує всі поставленя цілі та завдання, тобто

має зв’язок з базою даних та завдання описані в user story виконуються, а саме:

може виконувати регістрацію користувача та здійснювати вхід до додатку, якщо користувач вже зарегістрован, виводить усіх лікарів та вправно виконує фільтріцію лікарів по їх спеціальності, показує заброньовані сеанси у графіку роботи обраного лікаря, а також виконує як бронювання сеансу до обраного лікаря так і відміну вказаної броні.

Отже можна зробити висновок, що веб-додаток для запису на прийом до лікаря був успішно розроблений. Він дозволяє пацієнтам зручно та ефективно записуватися на прийом до лікаря, спрощуючи процес і запобігаючи зайнятість лікарні. Даний додаток може бути використаний в медичних установах для поліпшення обслуговування пацієнтів та оптимізації робочих процесів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
2. <https://github.com/jooble/entry-doctor-web-application>
3. <https://www.bibliofond.ru/view.aspx?id=588156>
4. <https://itnan.ru/post.php?c=1&p=495010>
5. <https://formdesigner.ru/templates/medicine/20.html>
6. <https://coderlessons.com/articles/java/veb-prilozhenie-spring-boot-s-gradle>
7. <https://programmer.help/blogs/spring-spring-mvc-mybatis-postgresql-bootstrap-initial-integration.html>
8. <https://habr.com/ru/articles/565242/>
9. <https://www.codeflow.site/ru/article/spring-mvc__spring-mvc-form-handling-example>