МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**Звіт**

**з технологічної практики**

студента ІІ курсу групи ІПЗ-20-4

спеціальності 121 «Інженерія програмного забезпечення»

\_\_ Васьківського Віталія Юрійовича\_\_\_\_\_

(прізвище, ім’я та по-батькові)

Керівник: Кузьменко О.В.

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 2022 р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Житомир – 2022

Зміст

[ВСТУП 3](#_Toc94609040)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ 4](#_Toc94609041)

[1.1 Аналіз задачі, засобів та методів її вирішення 4](#_Toc94609042)

[1.2 Технічне завдання на практику 4](#_Toc94609043)

[Висновки до першого розділу : 6](#_Toc94609044)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 7](#_Toc94609045)

[2.1. Проектування загального алгоритму роботи веб-додатку 7](#_Toc94609046)

[2.2. Розробка програмного забезпечення 7](#_Toc94609047)

[Висновки до другого розділу : 8](#_Toc94609048)

[РОЗДІЛ 3 ОПИС РОБОТИ З ДОДАТКОМ 9](#_Toc94609049)

[3.1 Опис роботи з додатком (Опис інтерфейсу) 9](#_Toc94609050)

[3.2. Тестування веб-додатку 13](#_Toc94609051)

[Висновки до третього розділу 15](#_Toc94609052)

[ВИСНОВКИ 16](#_Toc94609053)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 17](#_Toc94609054)

[ДОДАТКИ 18](#_Toc94609055)

[Додаток А 19](#_Toc94609056)

[Додаток Б 20](#_Toc94609057)

# ВСТУП

Інтернет сьогодні виступає в якості основної рушійної сили інформаційного суспільства. У сучасному суспільстві практично немає області людської діяльності, яка не була б представлена в мережі Інтернет. На сьогодні це не тільки джерело інформації, але і зручний, а найголовніше - швидкий засіб комунікації, сприятливий простір для бізнесу, джерело розваг.

Практично кожна людина з різною частотою вдається до його послуг, так як всесвітня мережа володіє такими незаперечними перевагами як доступність, оперативність і інформативність. Тому в даний час актуально створювати різні сайти.

У даному проекті буде наведено процес розробки месенджеру. Адже останнім часом месенджери стають більш популярними, тому що дозволяють здійснювати зручні переписки будь-де та у будь-який час. У цьому й полягає актуальність даної теми.

**Метою** практики є дослідження особливостей проектування та реалізації месенджеру.

Завданням на практику є:

* аналіз теоретичних засад проектування та реалізації месенджерів;
* визначення інформаційних потреб предметної області дослідження;
* розробка адаптивного інтерфейсу веб-сайту за допомогою HTML5, CSS3, JavaScript, React, TypeScript;
* розробка математичної та алгоритмічної моделі функціонування системи на основі БД;
* проектування бази даних за визначеною предметною областю;
* написання серверної частини за допомогою ASP.NET, SignalR, Entity Framework;

**Об’єктом дослідження** є методи та засоби розробки месенджеру.

**Предмет дослідження** є використання веб-технологій для забезпечення людей по всьому світу зручним засобом для переписок.

# РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## Аналіз задачі, засобів та методів її вирішення

Для написання фронтенду було обрано бібліотеки React – для реалізація SPA, Redux – для стейт менеджменту. Для написання бекенду було обрано фреймворки ASP.NET – як основа, з доповненням SignalR для відправки даних у реальному часі.

У якості БД було обрано MySQL.

## Технічне завдання на практику

**Постановка задачі**

Необхідно розробити, здійснити тестування і розгортання веб-орієнтованого багатокористувацького месенджера.

**Функціональні можливості системи**

Загальні функції системи:

* Наявність клієнтської (frontend) та адміністративної (backend) частини;
* Каталог діалогів;
* Реєстрація користувачів;
* Відео дзвінки.

Функції адміністративної частини:

* Робота з діалогами: CRUD для діалогів, реалізація множинного зв'язку, фільтрації та сортування даних.
* Робота з повідомленнями: CRUD для повідомлень, реалізація множинного зв'язку, фільтрації та сортування даних. Розсилка по web socket протоколу
* Автентифікація.

Функції клієнтської частини:

* Стандартні функції користувачів: реєстрація, вхід
* Кабінет клієнта: редагування профілю, перегляд власних діалогів
* Перегляд користувачів: сортування, пошук, фільтрація

**Функціональні вимоги щодо системи**

Загальні вимоги:

* Кожна дія користувача, пов'язана з успішним чи неуспішним додаванням/оновленням даних, повинна супроводжуватись повідомленням.

**Функціональні вимоги клієнтської частини:**

Сторінки каталогу діалогів клієнтської частини:

* Список діалогів
* Пошук діалогів по імені

Вимоги по роботі з користувачами:

* Реєстрація користувачів здійснюється за email та паролем. В продакшн-версії необхідно здійснювати верифікацію;
* Реалізувати відновлення паролю

Профіль клієнта:

* Зареєстрований користувач має змогу створювати/редагувати власний профіль клієнта - дані про користувача.

Вимоги по відправці повідомлень:

* Сторінка з формою для відправки повідомлень.
* Містить поля: текст повідомлення, прикріплення файлів та відправка emoji.

**Функціональні вимоги адміністративної частини:**

Сторінки для роботи з сутностями:

* CRUD по діалогам
* CRUD по повідомленням
* Зв'язок між діалогам і повідомленням
* Створення/обробка повідомлень

## Висновки до першого розділу :

У ході виконання першого розділу було визначено мови і фреймворки для розробки програмного забезпечення та актуальний напрямок розробки для продукту. Також поставлено чітке технічне завдання, по якому потрібно слідувати під час розробки.

# РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2.1. Проектування загального алгоритму роботи веб-додатку

На даному етапі розробки розглянемо загальний алгоритм роботи месенджеру. Загальна схема роботи наведена на Додатку А.

Для початку розробки бекенду було створено моделі, потім для них репозиторії для роботи з базою даних, котролери для обробки роутів та signalr хаб для відправки даних у реальному часі.

Тепер спроектуємо візуальну частину. Для цього використаємо бібліотеку React для написання SPA, та також за допомогою бібліотеки React Router Dom створити роути для кожного з модулів веб додатку. Для запитів на сервер використовується бібліотека Axios, за допомогою якої можна зручно робити запити на сервер.

При запуску месенджеру користувач бачить головну сторінку, де може переглядати свої діалоги та повідомлення до цих діалогів. В лівому меню можна перейти на сторінку користувачів та на сторінку налаштувань.

## 2.2. Розробка програмного забезпечення

Більша частина програмного забезпечення розробляється відповідно до алгоритму описаного вище, тому розглянемо цей принцип на прикладі реалізованих методів повідомлень.

Лістинг відправлення повідомлення:

public async Task SendMessage(int dialogId, string messageText, FileModel[] files)

{

int messageId = \_dialogsRep.AddMessageToDialog(Context.User.Identity.Name, dialogId, messageText);

MessageModel message = \_dialogsRep.AddFilesPinnedToMessage(messageId, files);

var usersLogins = message.Dialog.Users.Select(u => u.Login).Distinct().ToList();

message.Dialog.Users = null;

await Clients.Users(usersLogins).ReceiveMessage(dialogId, message);

usersLogins.Remove(Context.User.Identity.Name);

await Clients.Users(usersLogins).ReceiveNotification(message);

}

Спочатку створюється повідомлення та прикріпляються відправлені файли. Потім

отримуються логіни користувачів, котрі знаходяться в даному діалозі, та розсилаються ним це повідомлення.

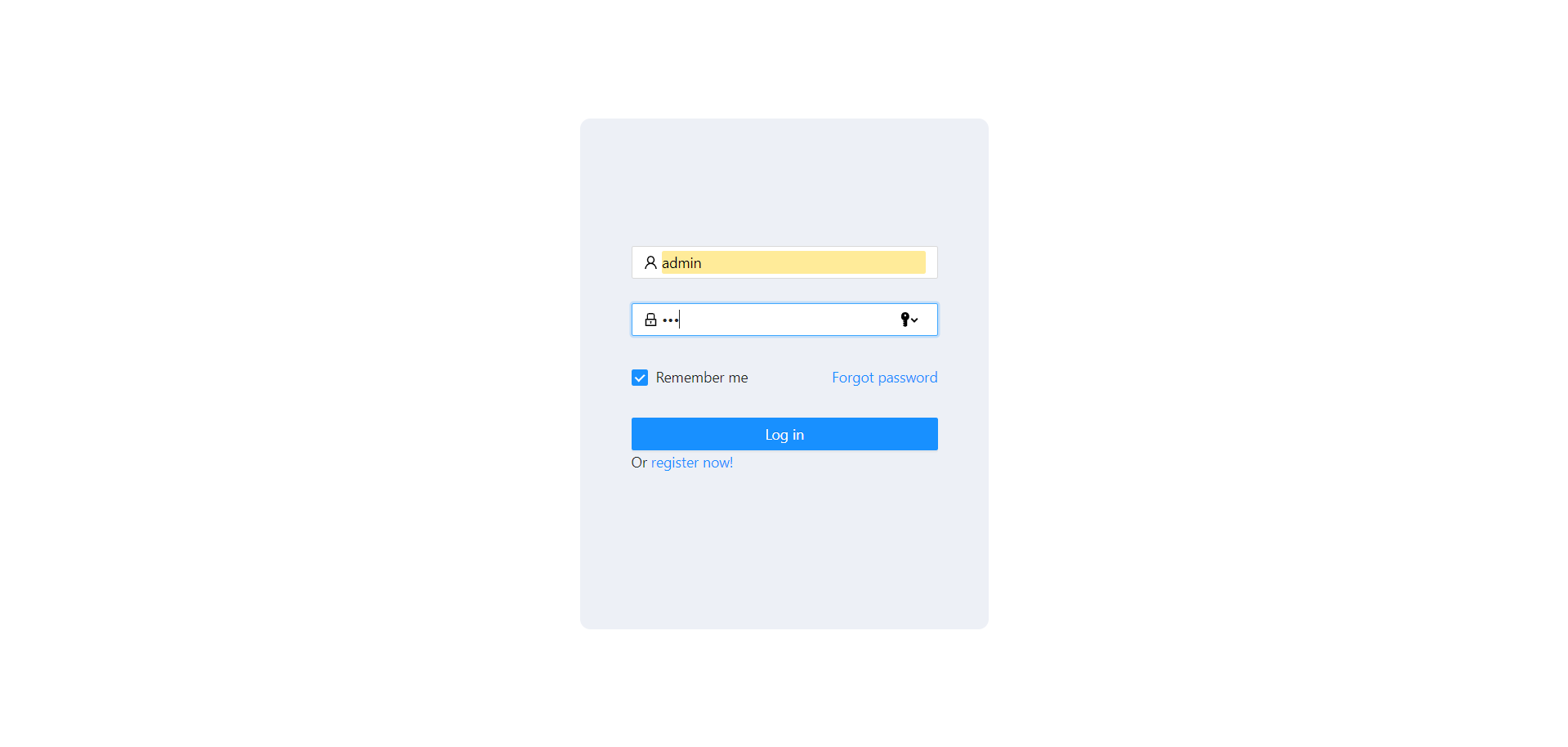
## Висновки до другого розділу :

Спроектовано загальну схему програми. Визначено та детально описано основні можливості програми. Пояснено алгоритми найважливіших методів месенджеру та приведено код їх основних частин. Приведено приклади коду деяких методів. Роз’яснено деякі нюанси роботи програми.

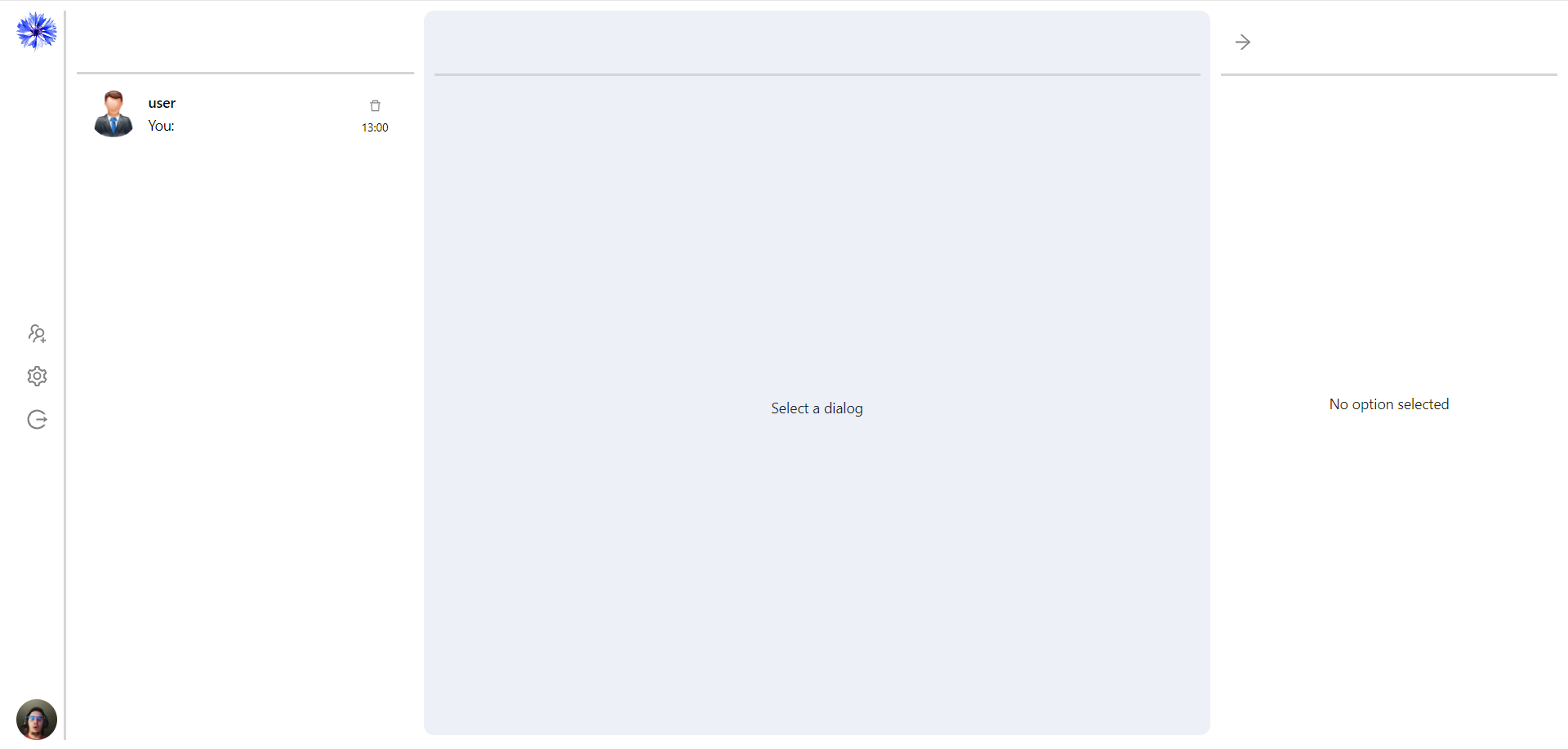
# РОЗДІЛ 3 ОПИС РОБОТИ З ДОДАТКОМ

## 3.1 Опис роботи з додатком (Опис інтерфейсу)

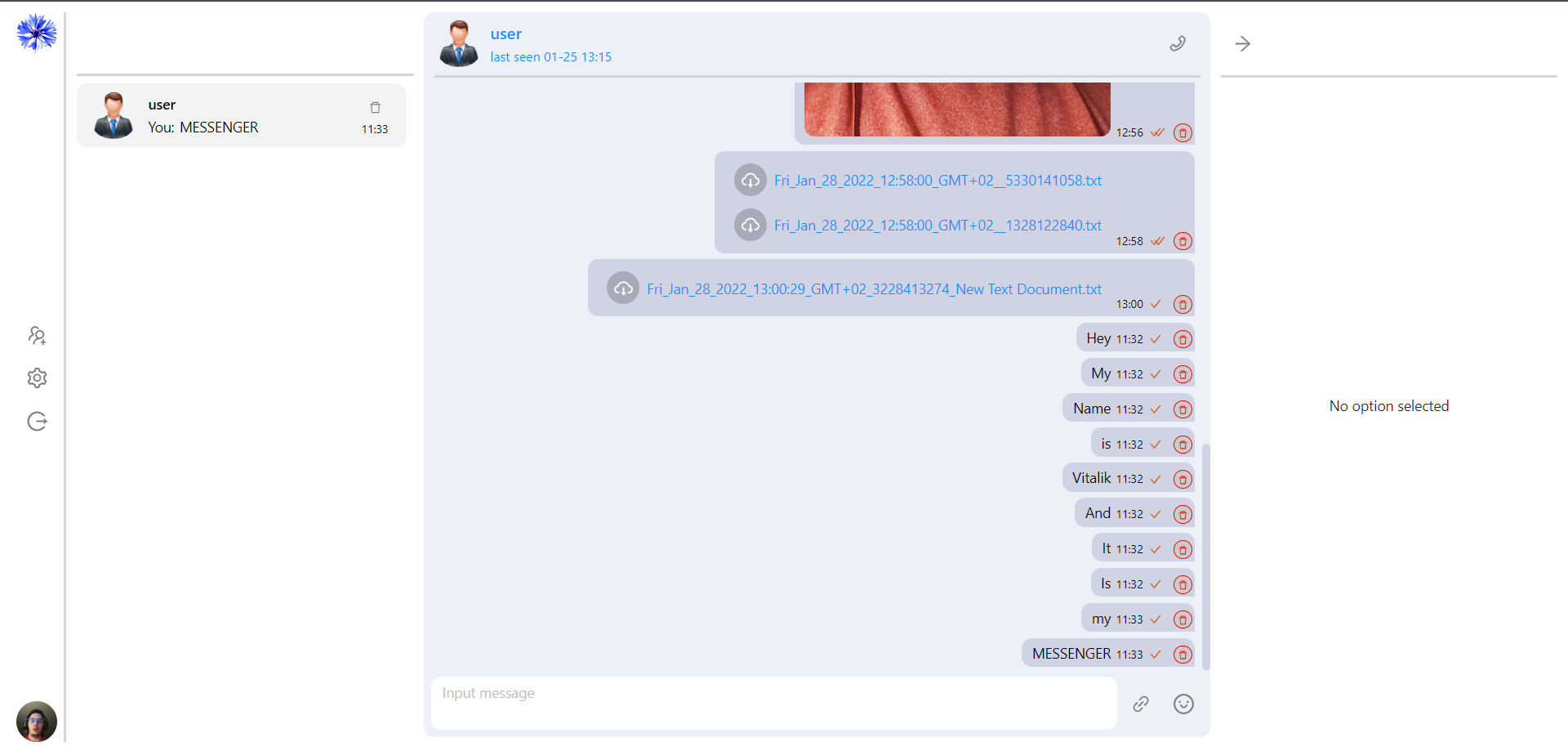
Після запуску веб-додатку з’являється сторінка входу, де користувач може ввести свої автентифікаційні дані та увійти до свого аккаунту. Або перейти на сторінку реєстрації та створити новий.

Рисунок 3.1. Сторінка «Вхід»

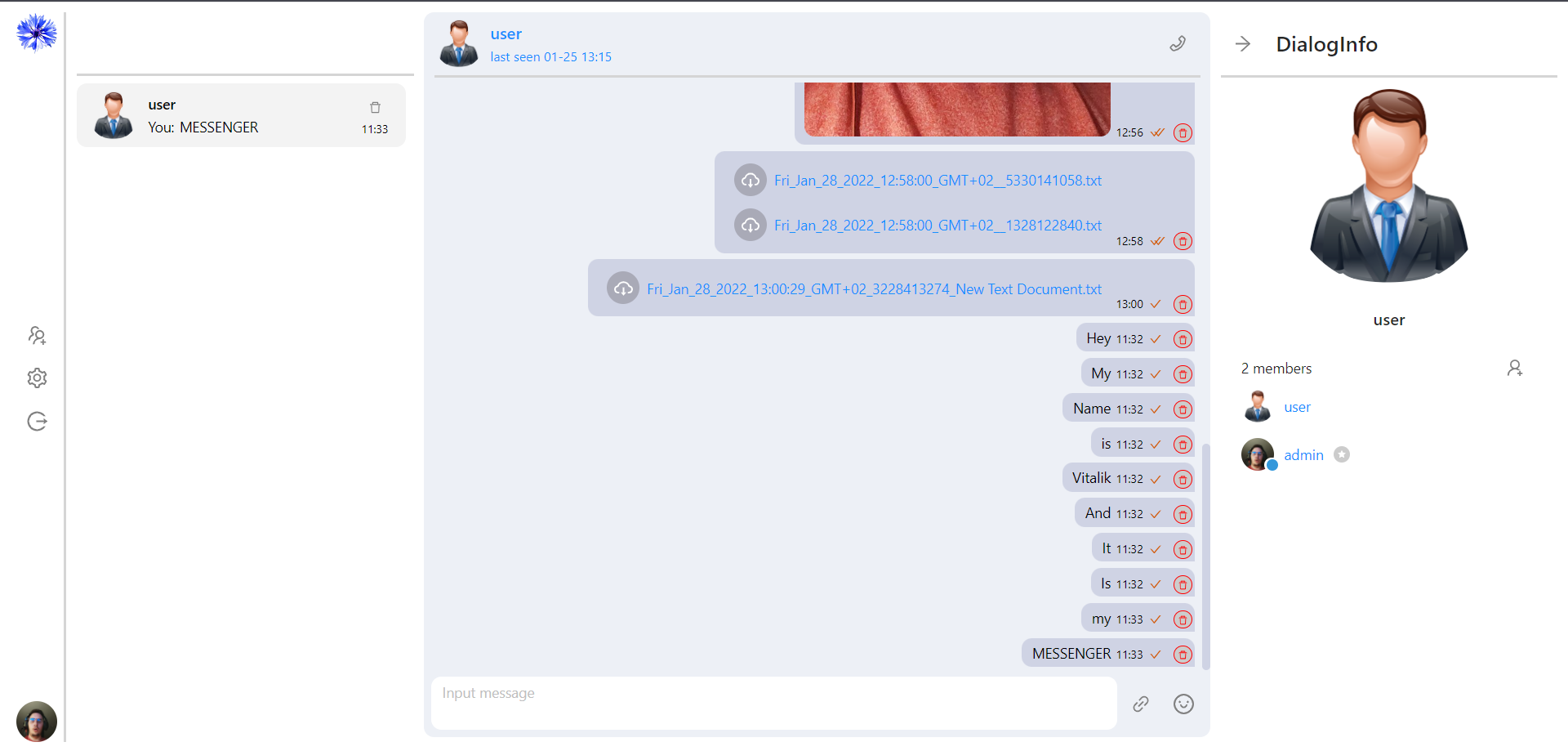
Після автентифікації користувач потрапляє на головну сторінку месенджеру. Тут може побачити список своїх діалогів з іншими користувачами.

Рисунок 3.2. Сторінка «Головна»

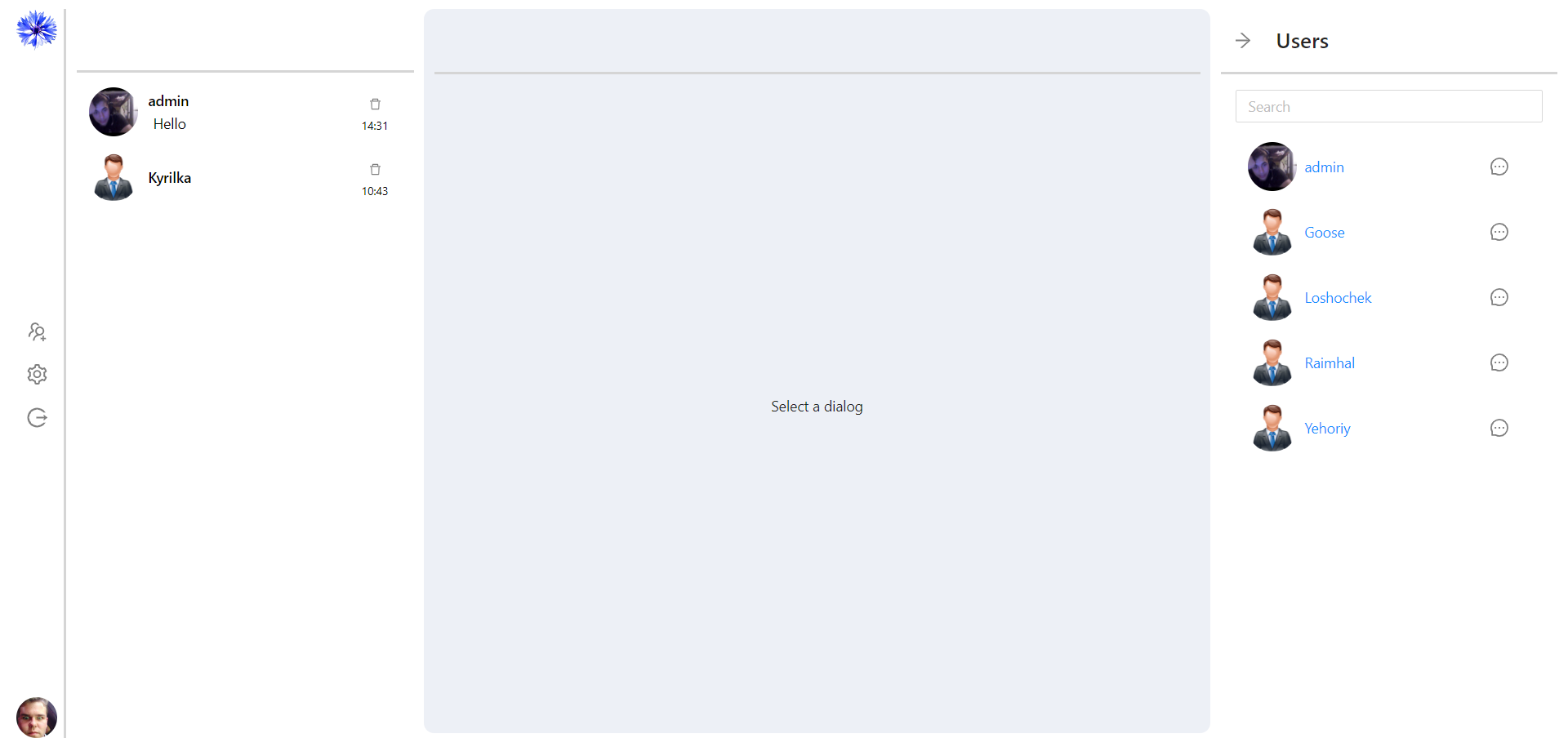
Можна вибрати якийсь конкретний діалог. Тут користувач може переглядати, видаляти та створювати нові повідомлення. Також для повідомлення він може прикріпити файл або емодзі.

Рисунок 3.3. Сторінка «Повідомлення діалогу»

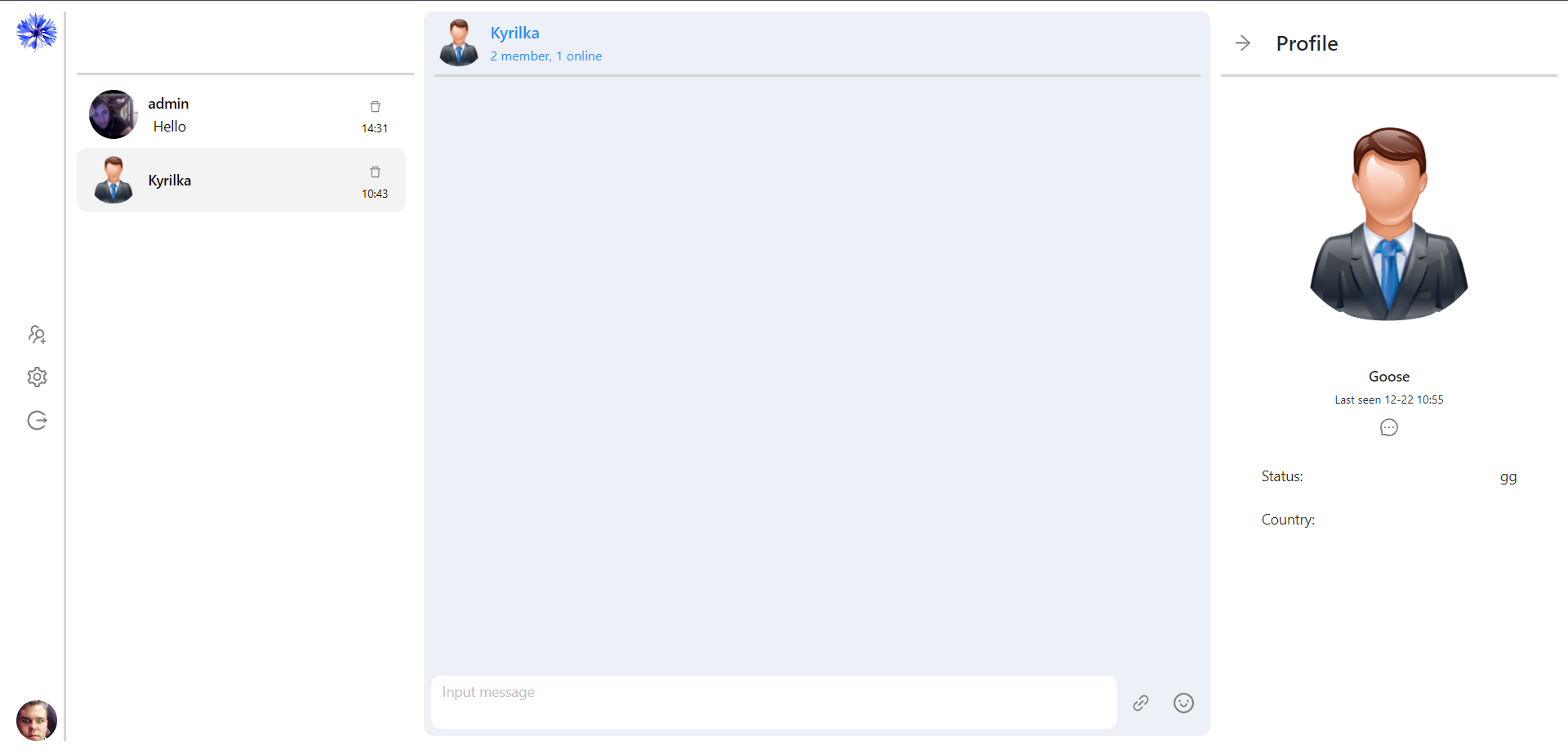
Також користувач може перейти до інформація про діалог де побачить побачить назву, хто до нього належить, хто створив даний діалог та додати нового користувача до даного діалогу.

 Рисунок 3.4. Сторінка «Інформація про діалог»

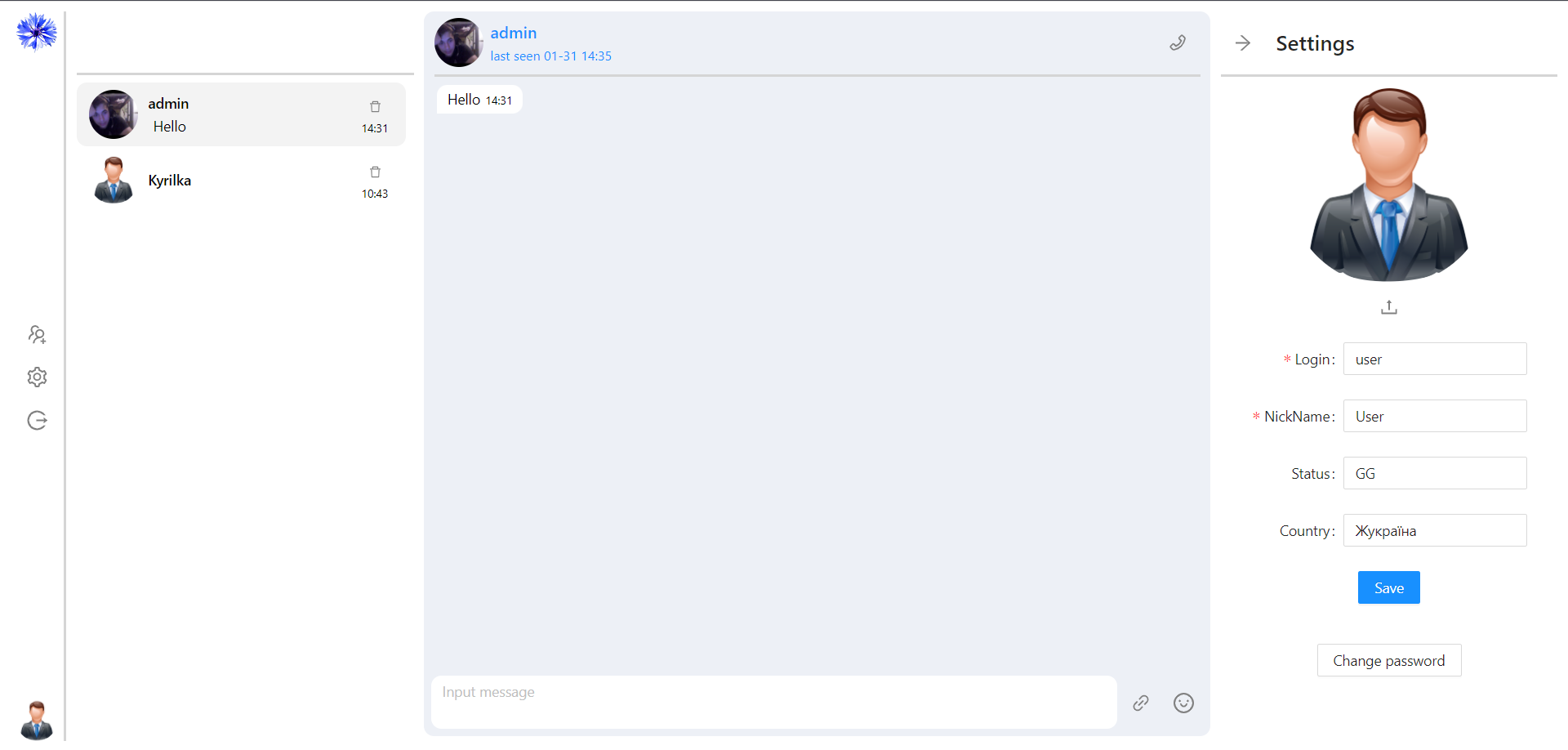
Також можна перейти на сторінку користувачів та переглянути їх

 Рисунок 3.5. Сторінка «Користувачі»

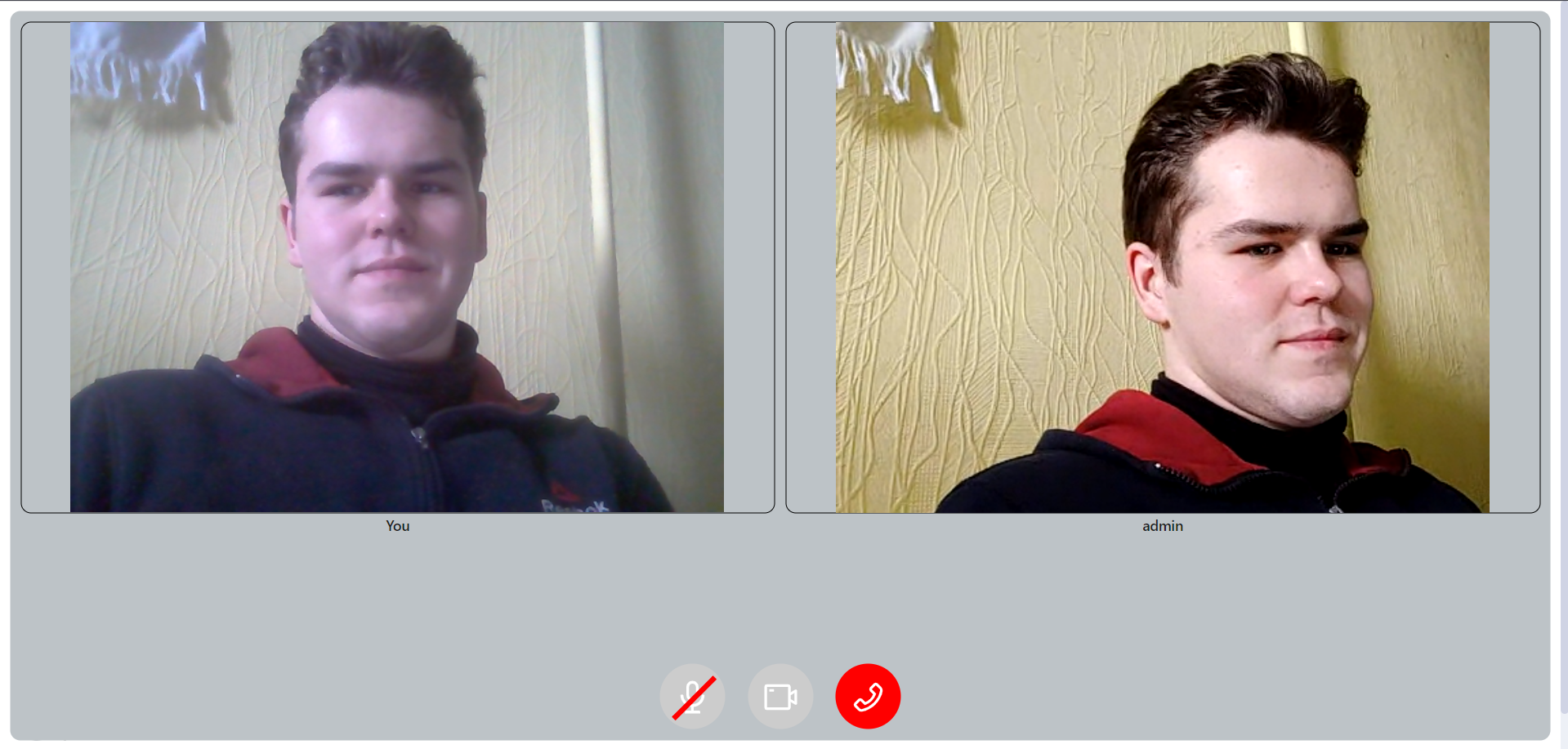
Або перейти до конкретного користувача та переглянути його профіль.

 Рисунок 3.6. Сторінка «Профіль користувача»

Також можемо перейти на сторінку налаштувань та відредагувати персональні дані та змінити пароль.

 Рисунок 3.7. Сторінка «Вхід»

Також є можливість подзвонити до будь-якого користувача. Поговорити з ним в реальному часі та побачити через веб камеру.

 Рисунок 3.8. Сторінка «Дзвінок»

## 3.2. Тестування веб-додатку

Веб-додаток відображає попередження про помилку при введенні не коректних даних. Наприклад, при спробі створення користувача без введення даних відображаються попередження про обов’язковість полів.

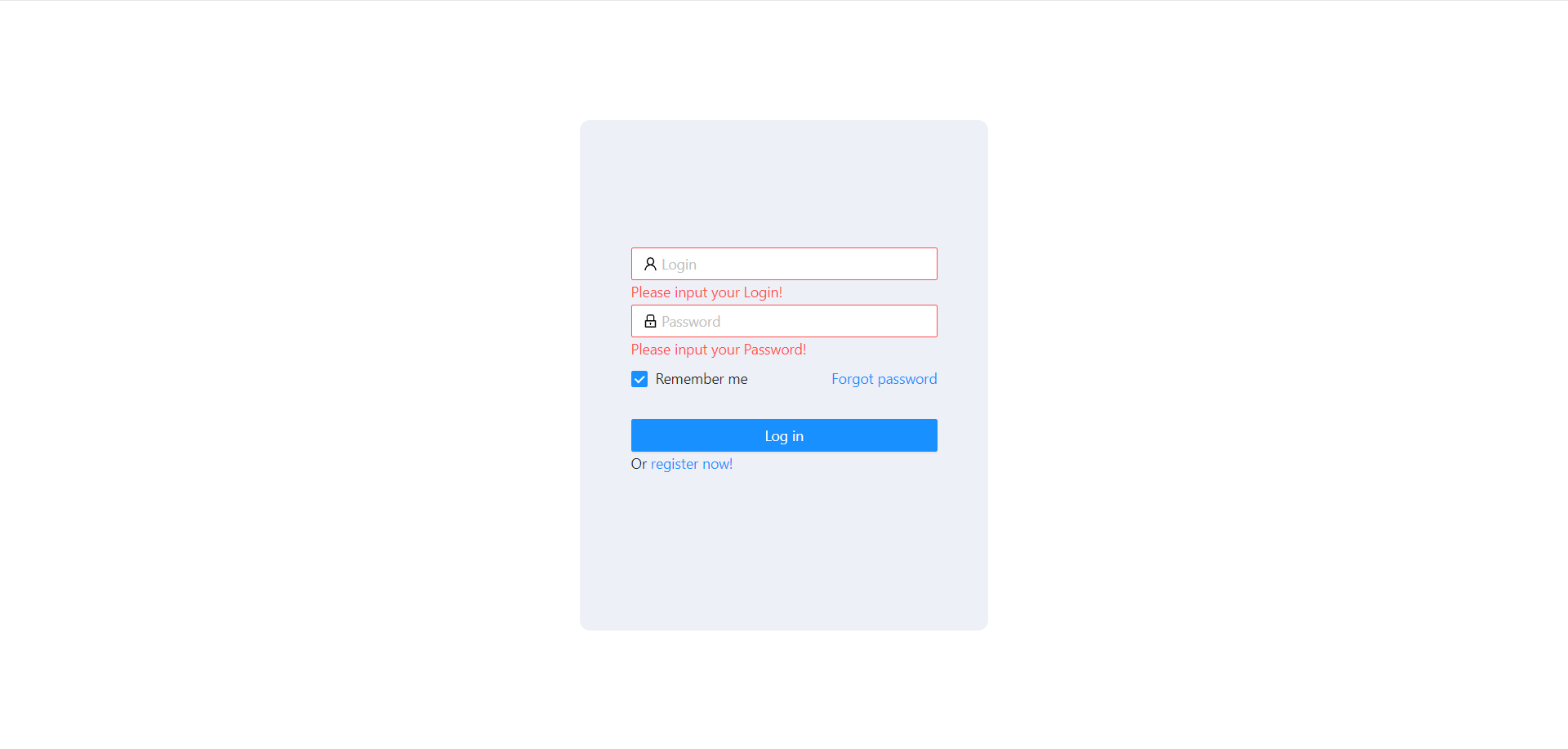


Рис 3.9. Повідомлення про необхідність заповнення

При вводі логіну, який вже існує, з’являється попередження про існування.

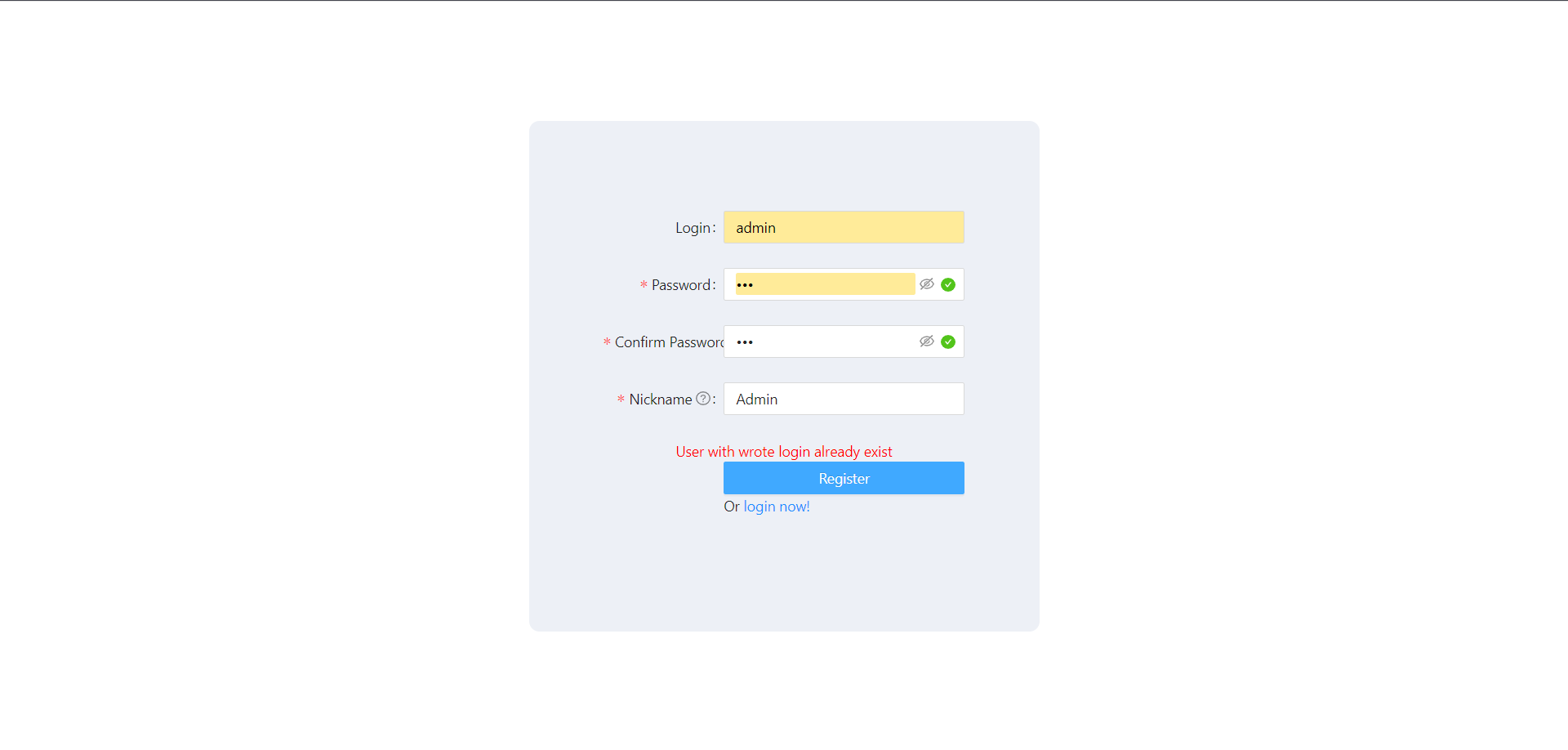


Рис 3.16. Повідомлення про існування користувача

При відправці форми з некоректними даними аутентифікації, з’являється попередження про некоректність:

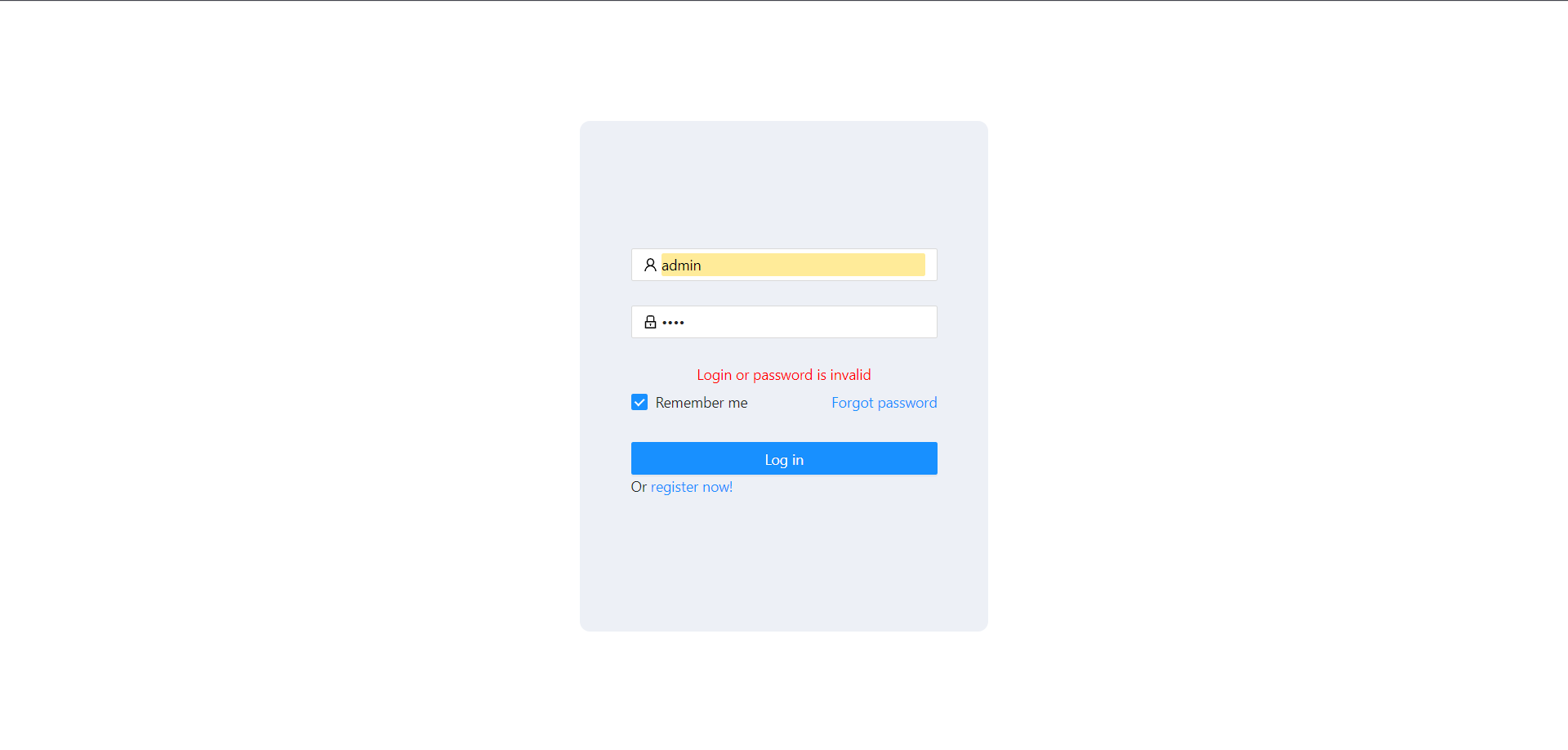


Рис 3.17. Повідомлення про некоректність логін або пароля

## Висновки до третього розділу

Було детально описано роботу програмного додатку, користувацький інтерфейс, очікувані дії користувача та їх результати. Також було розписано моменти уникнення помилок завдяки допоміжним перевіркам при введенні даних. Завдяки цим перевіркам вдалося уникнути багатьох помилок.

# ВИСНОВКИ

В результаті виконання технологічної практики було розроблено месенджер, який повністю відповідає запланованому функціоналу з технічного завдання, умовам та вимогам, поставленим на початку проектування. Інтернет-магазин є зрозумілим та зручним в користуванні, адаптивним для різних пристроїв та швидкодіючим. Також у ході виконання було закріплено знання з мов програмування C# та TypeScript, фреймворків ASP.NET, SiganlR , React та Redux.

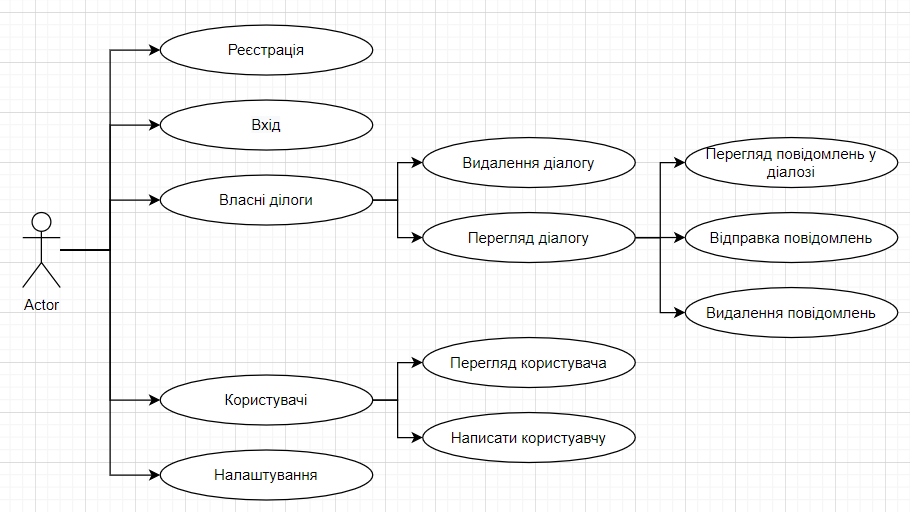
# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація React [електронний ресурс]. Режим доступу: <https://reactjs.org/> .
2. Документація Redux [електорнний ресурс]. Режим доступу: <https://redux.js.org/> .
3. Документація Axios [електорнний ресурс]. Режим доступу: <https://axios-http.com/docs/intro>.
4. Документація Ant Design [електорнний ресурс]. Режим доступу: <https://ant.design/components/overview/> .
5. Список відтворення «Курс «React JS путь самурая 1.0, уроки практика» » [електорнний ресурс]. Режим доступу: <https://www.youtube.com/playlist?list=PLcvhF2Wqh7DNVy1OCUpG3i5lyxyBWhGZ8> .
6. Документація MySQL [електронний ресурс]. Режим доступу: <https://dev.mysql.com/doc/> .
7. ASP.NET Core и C# | Полное руководство [електорнний ресурс]. Режим доступу: <https://metanit.com/sharp/aspnet6/>.
8. ASP.NET Core | SignalR Core. Первое приложение [електорнний ресурс]. Режим доступу: <https://metanit.com/sharp/aspnet5/30.1.php>.
9. Руководство по Entity Framework Core 6 и C# [електорнний ресурс]. Режим доступу: <https://metanit.com/sharp/efcore/>.

# 

# ДОДАТКИ

# Додаток А



# Додаток Б

