МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп’ютерних наук та технологій

Кафедра Комп’ютерних Інформаційних Технологій

Проєкт на тему

«Месенджер»

З дисципліни

«Крос-платформне програмування»

Виконав:

Студенти групи:

ТП-215Б

Григор А.Ю. та

Яцевілов В.О.

Перевірила:

Охремчук О.С.

2023

Назва: “BanderaChat”

Ідея: Додаток являє собою месенджер, який буде комунікуватися з сервером по апі, написаний на фреймворкі FastApi. На сервері буде своя база даних – MySQL. Буде використовуватися технологія websocket для обміну повідомленнями.

Месенджер буде доступним у вебі та в десктопному додатку, написаному на Java.

Додаток матиме вкладки:

* Чати – розпочаті переписки з іншими користувачами
* Друзі – перелік користувачі, яких Ви додали до друзів
* Обрані – перелік повідомлень, які Ви помітили в чати
* Профіль – заповнення профілю(нікнейм, фото, місто, телефон, пару слів про себе)
* Налаштування – різні налаштування сповіщень та безпеки

Користувачі матимуть змогу реєструватися, авторизуватися власноруч або через гугл та фейсбук.

Григор Андрій відповідає за розробку backend і frontend частини. Яцевілов Владислав буде розробляти desktop додаток.

План

1. Налаштувати бекенд проєкт
2. Написати моделі для бази даних
3. Створити аутентифікацію по апі
4. Створити апі по взаємодії з профілем(фото, емейл, пароль)
5. Стоврити апі для обміну повідомлень
6. Написати верстку месенджера на VueJs
7. Взаємозв’язати frontend частину месендежера і його backend api частину.

# Налаштування проєкту

## Бекенд

1. Налаштовано проект на FastAPI. [Основний файл](https://github.com/BanderosikUa/BanderaChat/blob/main/backend/src/main.py)
2. Підключена база даних MySQL на сервері, що зберігає дані месенджера. [Файл бази даних](https://github.com/BanderosikUa/BanderaChat/blob/main/backend/src/database.py)
3. Налаштовані [докерфайли](https://github.com/BanderosikUa/BanderaChat/blob/main/backend/Dockerfile) для проєкту, щоб додаток можна було запускати в Docker контейнерах.
4. Додано налаштування для логів додатку, щоб детально відслідковувати дії користувачів та помилки у додатку.
5. Написані тести для перевірки роботи додатку та визначення проблем.

[Conftest файл](https://github.com/BanderosikUa/BanderaChat/blob/main/backend/tests/conftest.py)

1. Створені міграції для збереження та оновлення схеми бази даних за допомогою alembic. [Конфігурацій файл](https://github.com/BanderosikUa/BanderaChat/blob/main/backend/alembic/env.py)
2. Налаштовано Kubernetes файли та розгорнуто проєкт в сервісі Google Cloud. [Всі yaml файли](https://github.com/BanderosikUa/BanderaChat/tree/main/k8s)

Посилання на документацію: <http://34.120.190.133/docs>

## Фронтенд

На даному етапі було налаштовано модуль фронтенду на Vue.js. Для розробки було використано Vue CLI, що дозволяє швидко створити новий проєкт та налаштувати його структуру.

Для підключення до серверу було встановлено бібліотеку Axios, яка дозволяє здійснювати HTTP запити з фронтенду до бекенду. Для зручності використовується [конфігураційний файл](https://github.com/BanderosikUa/BanderaChat/blob/main/frontend/src/main.js), в якому зберігаються URL-адреси для запитів до серверу.

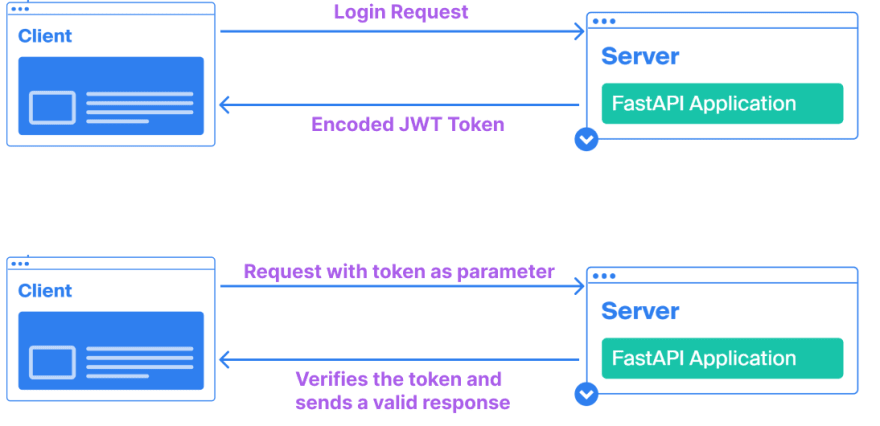
Також було налаштовано [роутер](https://github.com/BanderosikUa/BanderaChat/blob/main/frontend/src/router/index.js), який дозволяє переходити між різними сторінками додатку без перезавантаження сторінки.

Налаштовано Kubernetes файли та розгорнуто фронтенд частину в сервісі Google Cloud

Посилання на сайт: <http://35.184.58.77/home>

# Аутентифікація

Зроблено [модуль](https://github.com/BanderosikUa/BanderaChat/tree/main/backend/src/auth) аутентифікації користувачів за допомогою JWT (JSON Web Token). Цей модуль дозволяє користувачам реєструватися в системі, увійти в свій обліковий запис та отримувати JWT токени, які можуть бути використані для авторизації під час доступу до різних ресурсів в системі. Модуль включає в себе [перевірку](https://github.com/BanderosikUa/BanderaChat/blob/main/backend/src/auth/oa2auth.py) токенів при кожному запиті до серверу та відправку помилки, якщо токен не валідний. Створено [модель](https://github.com/BanderosikUa/BanderaChat/blob/main/backend/src/auth/models.py) для користувачів, де всі дані зберігаються в базі даних MySQL, яка була підключена до серверної частини проєкту.

До фронтенд частини було додано кілька допоміжних компонентів, таких як компонент входу в систему, компонент реєстрації користувача. Налаштовано звертання до апі та запис токену в локальне сховище Вашого браузера, щоб при перезагрузці сторінки не зникала авторизація.

Приклад реєстрації юзера по апі: [http://34.120.190.133/docs#/Auth/register\_user\_auth\_register\_post](http://34.120.190.133/docs%23/Auth/register_user_auth_register_post)

Форма рееєстрації юзера на сайті: <http://35.184.58.77/login>

# DESKTOP ДОДАТОК

## План

1. Налаштування проекту.  
2. Розробка дизайну за допомогою бібліотеки JavaFx та додатку SceneBuilder.  
3. Налаштування API запитів для аунтефікації за допомогою бібліотеки OkHTTP.  
4. Налаштування API запитів для взаємодії з профілем.  
5. Налаштування API запитів для обміну повідомлень.  
6. Імпорт дизайну та пов’язання його з класами з API запитами.

API Запити

В тестовому варіанті представлені API запити на гітхабі   
[API TEST](https://github.com/Seinia/Bandera-Chat-Java-version) згідно з http://34.120.190.133/docs

Дизайн

1. Розроблено дизайн авторизації та викладено на github у вигляді файлу [hello-view.fxml](https://github.com/Seinia/Bandera-Chat-Java-version/blob/main/hello-view.fxml)   
Переглянути можна за посиланням  
[Дизайн логіну](https://github.com/Seinia/Bandera-Chat-Java-version/blob/main/Authexample.PNG)