МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет безопасности информационных технологий**

**Направление подготовки: 10.03.01 Информационная безопасность**

**Образовательная программа: "Информационная безопасность / Information security"**

**Дисциплина:**

**«Web программирование»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

**«Разработка Веб сервиса»**

**Выполнил студент(ы):**

группа/поток \_\_\_N3345\_\_\_\_

\_Молитвин Илья Алексеевич\_\_/\_\_\_\_**Изображение выглядит как зарисовка

Контент, сгенерированный ИИ, может содержать ошибки.**\_\_\_\_\_\_

*ФИО Подпись*

**Проверил:**

\_\_Менщиков Александр Алексеевич\_\_/\_\_\_\_\_\_\_\_\_\_

*ФИО Подпись*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Отметка о выполнении (один из вариантов:*

*отлично, хорошо, удовлетворительно, зачтено)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Дата*

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc181108318)

[ВВЕДЕНИЕ 3](#_Toc181108319)

[1. Описание архитектуры ПО 4](#_Toc181108320)

[2. Описание структуры базы данных 5](#_Toc181108321)

[3. Описание протокола и форматов передачи данных 6](#_Toc181108322)

[4. Описание API 7](#_Toc181108323)

[5. Тестирование функционала ПО 8](#_Toc181108324)

[ЗАКЛЮЧЕНИЕ 12](#_Toc181108325)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc181108326)

[ПРИЛОЖЕНИЕ 14](#_Toc181108327)

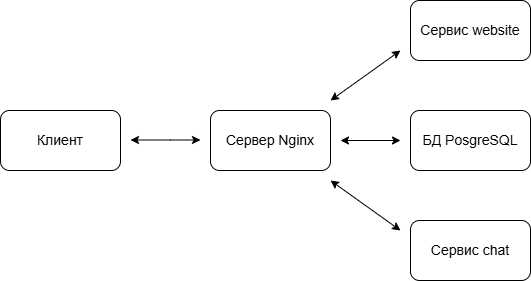
ВВЕДЕНИЕ

Цель работы – разработать онлайн чат для пользователей, состоящий из двух микросервисов, написанных на языке Python с использованием фреймворка FastAPI:

* Первый должен отвечать за регистрацию, аутентификацию, создание и поиск чат-комнат.
* Второй - за функционал Websocket чата.

# Описание архитектуры ПО

В веб сервисе реализована структура, представленная на рисунке 1.



1. Архитектура ПО

Клиент взаимодействует с системой через веб-интерфейс. Все запросы от клиента проходят через сервер nginx, который перенаправляет их к соответствующим микросервисам.

**Сервис website**:

* Отвечает за веб-приложение, которое обрабатывает запросы клиента, связанные с регистрацией, логином и работой с веб-интерфейсом.
* Выдает JWT-токены для аутентификации в чат.

**Сервис chat**:

* Поддерживает Websocket протокол для работы чата.
* Принимает сообщения и поддерживать рассылку сообщения по всем пользователям, подключенным к данной чат комнате.
* Аутентифицирует пользователей через проверку JWT-токенов

# Описание структуры базы данных

В БД содержатся 2 таблицы:

* Пользователи;
* Чат-комнаты;

На рисунке 2 приведена ER-диаграмма БД.



1. ER-диаграмма БД

# Описание протокола и форматов передачи данных

На рисунке 3 показан процесс взаимодействия между клиентом и сервером с использованием WebSocket:

1. Клиент отправляет запрос на установку WebSocket-соединения.
2. Сервер подтверждает запрос и открывает соединение.
3. Происходит двусторонний обмен сообщениями.
4. Когда обмен завершается, одна из сторон закрывает соединение.

Изображение выглядит как текст, снимок экрана, линия, Параллельный

Автоматически созданное описание

Рисунок 3 - Схема передачи данных по Websocket

В данной работе используется тактовый формат при передаче данных в виде: «почта пользователя: сообщение пользователя»

# Описание API

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 - API веб сервиса

# Тестирование функционала ПО

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5 - Запущенный docker-контейнер

Изображение выглядит как снимок экрана, программное обеспечение, Мультимедийное программное обеспечение, текст

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 6 - Запущенные Докер-контейнеры

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 7 - страница регистрации

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 8 -Ошибка при попытке зарегистрироваться под уже существующем email-омИзображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9 – Вход пользователя

Изображение выглядит как текст, программное обеспечение, Шрифт, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 10 - Ошибка идентификации или аутентификации

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 11 - Дэшбоард

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 12 – Поиск чатов

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 13 – Комната чата

Изображение выглядит как текст, число, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 14 – общение в чате

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы был разработан веб-сервис онлайн-чата, состоящий из двух основных микросервисов – website и chat. Микросервисы реализованы на языке Python с использованием фреймворка FastAPI. Аутентификация и регистрация пользователей, а также функционал создания, удаления и поиска чат-комнат обеспечиваются благодаря использованию базы данных PostgreSQL и ORM-библиотеки. Фронтенд-составляющая представлена шаблонами Jinja2, которые обеспечивают удобное взаимодействие пользователей с сервисом. Общение между пользователями реализовано через WebSocket-протокол, что позволяет поддерживать обмен сообщениями в реальном времени.

Сервис также включает дэшбоард, где авторизованный пользователь может управлять своими чатами – создавать новые комнаты, просматривать список созданных чатов, удалять их и осуществлять поиск по названию. Значимые изменения в визуальном оформлении выполнены с использованием Bootstrap, что позволяет обеспечить современный и удобный интерфейс для пользователей.

Вся система развёрнута с использованием Docker, а многосервисная архитектура описана в файле docker-compose.yml, включающем микросервисы website, chat, сервер Nginx для балансировки запросов и базу данных PostgreSQL. Nginx настроен для проксирования HTTP-запросов к сервису website и WebSocket-соединений к сервису chat.

Таким образом, была успешно реализована полноценная система онлайн-чата с поддержкой аутентификации пользователей, управлением чат-комнатами и обменом сообщениями в реальном времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. FastAPI Documentation. [Электронный ресурс] // URL: https://fastapi.tiangolo.com.
2. PostgreSQL Documentation. [Электронный ресурс] // URL: <https://www.postgresql.org/docs/13/index.html>.
3. Docker Documentation. [Электронный ресурс] // URL: https://docs.docker.com/get-started/overview/.
4. JWT.IO. JSON Web Tokens (JWT): What is JSON Web Token?. [Электронный ресурс] // URL: <https://jwt.io>
5. Nginx Documentation. [Электронный ресурс] // URL: https://nginx.org/en/docs/.

ПРИЛОЖЕНИЕ

<https://github.com/dimdid2004/Web_1lab> - ссылка на исходный код