1. **Структура контейнеризации и настройка окружения**

Приложение развернуто с использованием Docker Compose и состоит из двух основных сервисов:

|  |  |  |
| --- | --- | --- |
| **Сервис** | **Образ** | **Назначение** |
| db | postgres:latest | Хранение данных пользователей, чатов, сообщений |
| web | Кастомный (FROM python:3.10-slim) | Flask-приложение (веб-интерфейс) |

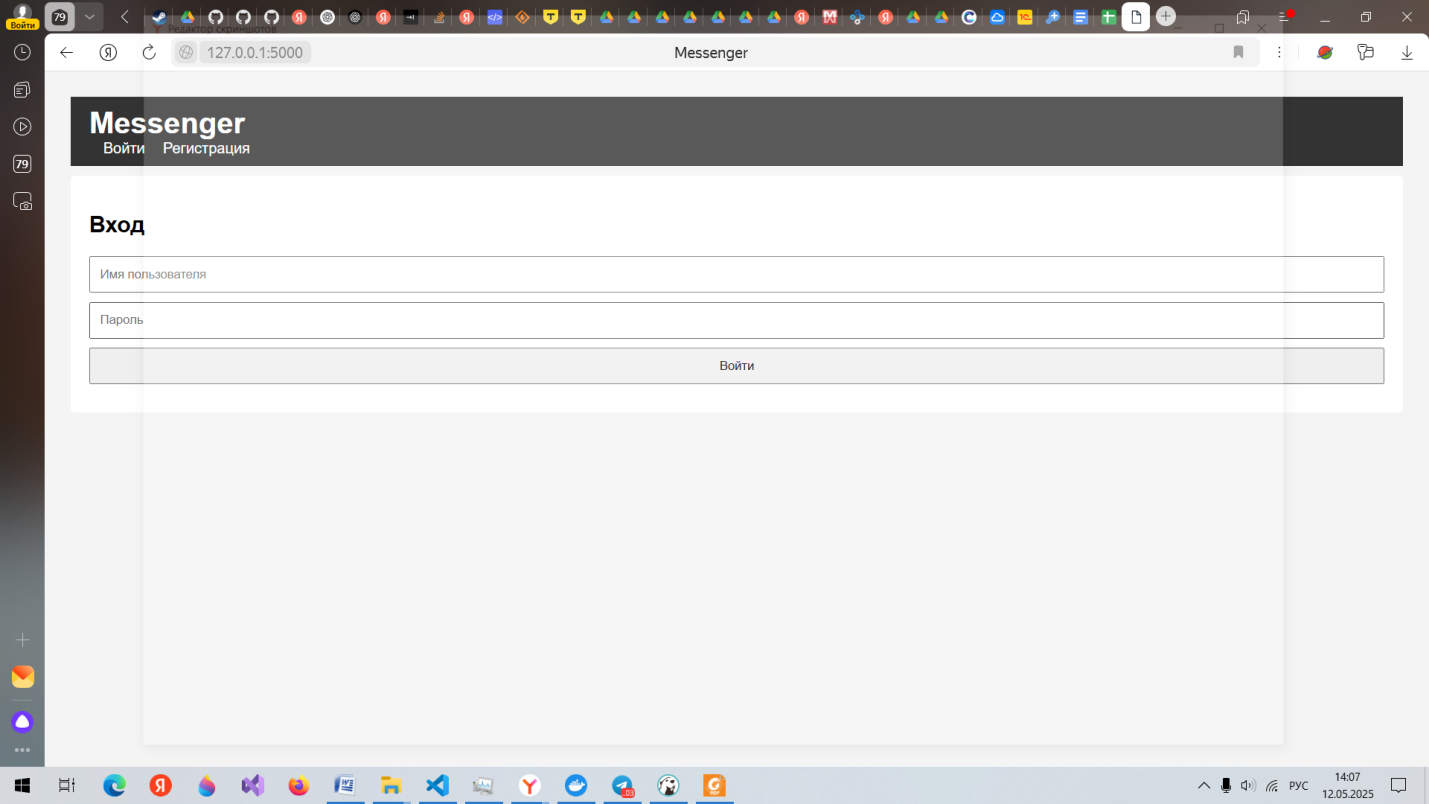
**Файлы:**

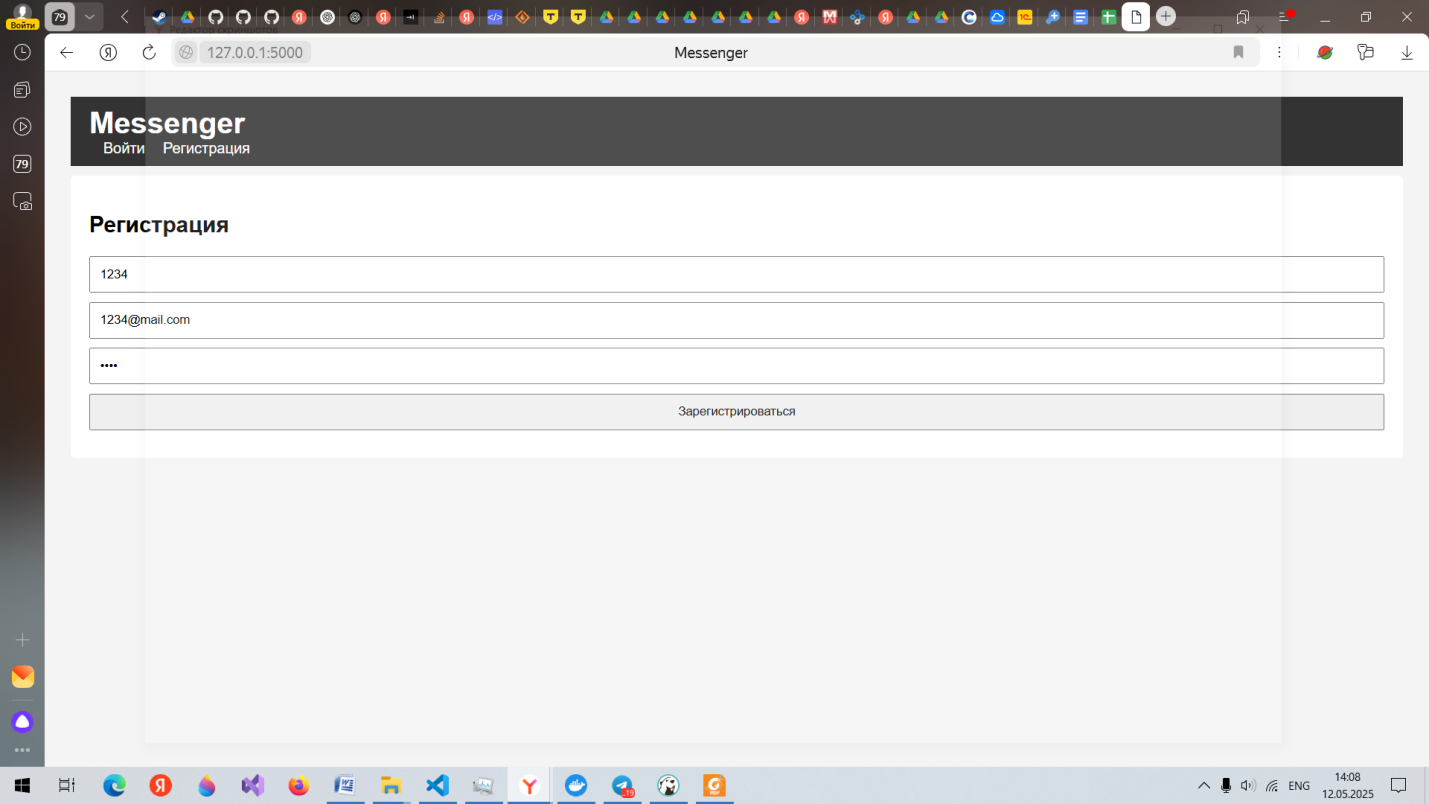
* docker-compose.yml: описывает оба сервиса, монтирует том для PostgreSQL.
* Dockerfile: создает образ Flask-приложения.
* requirements.txt: содержит зависимости (Flask, SQLAlchemy, psycopg2 и др.).

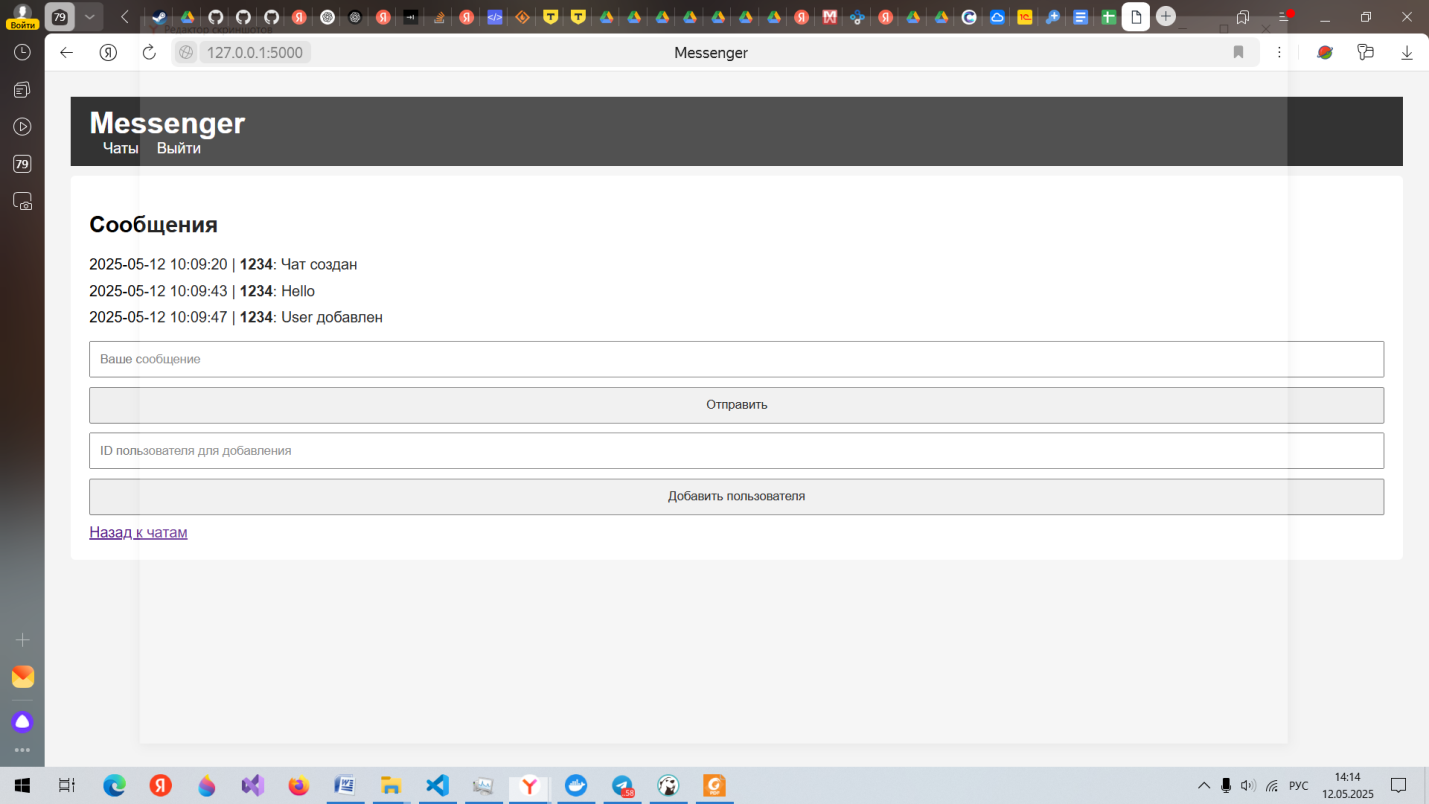
**Пример запуска:**

docker-compose up --build

1. **Скриншоты работающего приложения в контейнерах**







1. **Описание API (в формате таблицы)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Маршрут** | **Метод** | **Назначение** | **Аутентификация** |
| / | GET | Редирект на /login-page | нет |
| /register-page | GET/POST | Регистрация нового пользователя | нет |
| /login-page | GET/POST | Вход в систему | нет |
| /logout-page | GET | Выход из системы | нет |
| /chats-page | GET/POST | Просмотр чатов / создание нового | да |
| /chats/<chat\_id>/leave | POST | Покинуть чат | да |
| /chats/<chat\_id>/messages-page | GET/POST | Просмотр сообщений и взаимодействие | да |

### Архитектура разработанного приложения

**Основные компоненты:**

* Flask (Backend) — веб-приложение, обрабатывающее маршруты, формы и шаблоны.
* Jinja2 (Frontend) — серверная генерация HTML.
* PostgreSQL — СУБД для хранения данных.
* SQLAlchemy — ORM для работы с БД.

**Основные сущности:**

* User: хранит имя пользователя, email и хеш пароля.
* Chat: групповые чаты.
* Message: сообщения, привязанные к чатам.
* ChatMember: связующая таблица many-to-many между пользователями и чатами.

**Поток взаимодействия:**

1. Регистрация / логин.
2. Создание чатов, автоматическое добавление автора.
3. Отправка сообщений.
4. Добавление участников по ID.
5. Удаление чата при выходе последнего участника.