PR7-ZENTRATECH-DEV-TEST

Web Application for User Interests and Messaging

Table of contents

Chapter 1

Project Overview

1.1 Introduction

This document provides an overview of a web application designed to facilitate user interactions through interest messages and real-time chat. The application will be built using Django for the backend and Angular or React for the frontend.

1.2 Scope

The application allows users to:

- Register and log in.
- Browse a list of other users and send interest requests.
- View, accept, or reject received interest requests.
- Engage in real-time chat if an interest request is accepted.

Chapter 2

Core Features

2.1 User Authentication

- Implement a user authentication system to support registration and login.
- Ensure secure password storage and handling.

2.2 Sending Interests

- Allow logged-in users to browse other users.
- Provide a mechanism to send interest messages to other users.

2.3 Accepting/Rejecting Interests

- Enable users to view received interest messages.
- Allow users to accept or reject these messages.

2.4 Chat System

- Implement a chat interface for users whose interest has been accepted.
- Ensure real-time messaging functionality.

Chapter 3

Design Choices

3.1 Backend

- **Framework**: Django
- **Database**: SQLite
- **Authentication**: JWT Authentication
- **Real-Time Communication**: Socket.io server

3.2 Frontend

- **Framework**: React
- **App Routing & Navigation**: React Router Dom
- **Build Tool**: Vite
- **UI Library**: Material UI & Tailwind CSS
- **Real-Time Communication**: Socket.io client

Chapter 4

Setup and Running Instructions

4.1 Backend Setup

1. Clone the repository: `git clone https://github.com/anirudhasj441/pr7-zentra-dev-test.git`
2. Navigate to the backend directory: `cd ./pr7-zentra-dev-test/backend`
3. Create a Virtual Environment (if not already created): `python -m venv venv`
4. Activate the virtual environment:
    - Windows: `venv\Scripts\activate`
    - Unix/MacOS: `source ./venv/bin/activate`
5. Install dependencies: `pip install -r requirements.txt`
6. Navigate to Source Directory: `cd ./src`
7. Apply migrations: `python manage.py migrate`
8. Create a Superuser (if needed): `python manage.py createsuperuser`
9. Run the development server: `python server.py`
10. Access the Application:
    - Open your web browser and go to http://127.0.0.1:8000/ to view the application.
    - The Django admin interface is available at http://127.0.0.1:8000/admin/.
11. Run test cases: python manage.py test

Note: Ensure that you have the frontend project running as well.

Additional Note: Redis server should be up and running at port 6379. You can start Redis using the following command if it's not already running: `redis-server`

4.2 Frontend Setup

1. Clone the repository: `git clone https://github.com/anirudhasj441/pr7-zentra-dev-test.git`
2. Navigate to the frontend directory: `cd pr7-zentra-dev-test/frontend`
3. Install dependencies: `npm install`
4. Start the development server: `npm run dev`
5. Access the Application:
    - Open your web browser and go to http://127.0.0.1:3000/ to view the application.

Note: Please ensure that the backend server is running before starting the frontend application.

Chapter 5

Incomplete Aspects and Next Steps

- Integrate advanced features such as notifications for new messages or interest requests.
- Implement additional security measures such as email verification or two-factor authentication.
- Implement end-to-end encryption for secure communication.
- Enhance user interface and experience based on user feedback.

Chapter 6

Conclusion

This documentation outlines the core features, design choices, setup instructions, and next steps for the project. By adhering to the outlined expectations and focusing on critical functionality, the project aims to deliver a functional and user-friendly application within the specified time constraints.