Sabhyasha Retail Tech Pvt Ltd

Position: Django Developer
Task: Chatting Platform Development
Objective: Build a chatting platform using the Django framework that allows two users to communicate in real-time. This feature will eventually be integrated into our existing platform for seller-to-customer service interactions.

Task Overview

Develop a simple chat application where two users can engage in a conversation. This feature will act as a foundation for integrating chat functionality into our platform. The chat system should be designed with future scalability in mind, ensuring it can accommodate a larger user base and more advanced features.

Task Requirements

1. **User Authentication**
    - Implement a simple authentication system where two users can log in and access the chat interface.
    - Ensure basic user management functionalities: login, logout, and registration.
2. **Chat Interface**
    - Design a minimal, functional user interface where two authenticated users can send and receive real-time messages.
    - Display messages in chronological order, showing the sender's name and timestamp.
3. **Real-Time Communication**
    - Use Django Channels to implement real-time communication.

    - Messages should appear instantly on both users' screens without requiring a page reload.
4. **Database**
    - Store chat history in a database using Django models.
    - Ensure that the chat history can be accessed after logging in and out.
5. **Security Considerations**
    - Basic security implementation, such as CSRF protection, HTTPS, and user input validation.
6. **Documentation & Code Quality**
    - The code should be well-structured, with clear comments.
    - Provide basic documentation outlining how the system works, how to set it up, and how to use it.

Additional Features (Optional but Encouraged)

- Implement a typing indicator.
- Allow users to send media (images, files).

Evaluation Criteria

- **Functionality**: Does the chat system work as expected with two users chatting in real time?
- **Code Quality**: Is the code well-organized, modular, and easy to understand?
- **UI/UX**: Is the user interface intuitive and easy to use?
- **Documentation**: Is there sufficient documentation for understanding and running the system?

Submission Guidelines

1. Submit your task via a Git repository (GitHub).
2. Include a README.md with setup instructions.
3. Ensure the project runs locally without issues and include steps for running it.

Deadline: 21-10-2024

Contact for Queries: *8249351488 | pratik@sabhyasha.com*