ChatterBox

Real-Time WebSocket Chat Application

Siddhi Kumari
Infosys Springboard Internship – Batch 11
Mentor: Megha M
19 February 2026

Project Overview

- ChatterBox is a full-stack real-time chat application developed using FastAPI and WebSockets.
- The system provides:
- • Secure user registration and login using JWT
  • Real-time messaging using WebSocket protocol
  • Admin dashboard for system monitoring
  • ML-based bad word detection
  • Automated warning and blocking system
  • CSV data export functionality

Problem Statement

- Modern chat applications often face:
- • Lack of real-time efficiency
  • Weak authentication mechanisms
  • No proper moderation system
  • Limited admin monitoring capabilities
- This project solves these issues using a secure and scalable backend architecture.

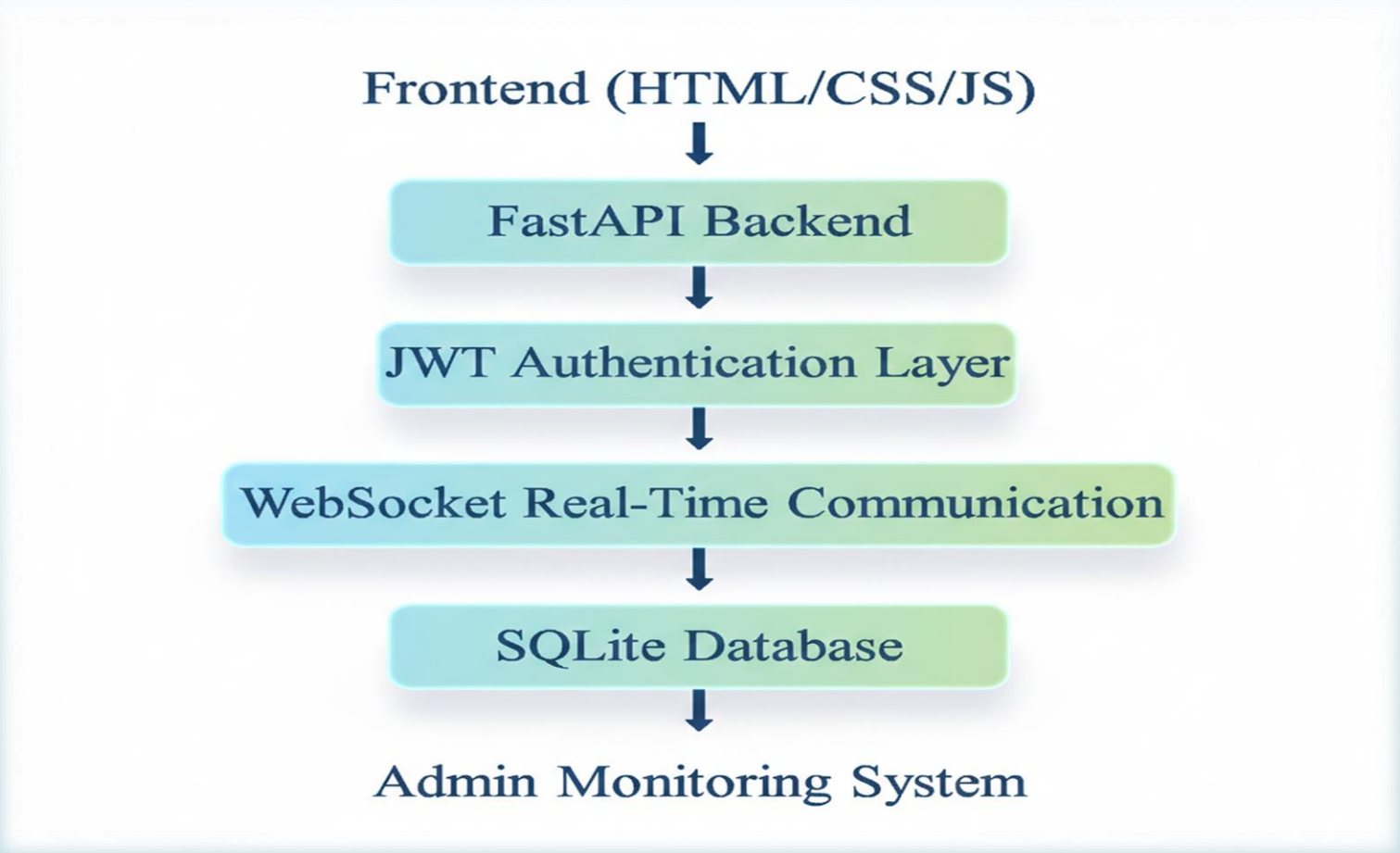
Project Objectives

- Build a real-time chat system using WebSockets
- Implement JWT-based secure authentication
- Develop individual and group messaging
- Add automatic bad-word detection
- Implement warning-based blocking system
- Create separate Admin interface
- Enable CSV export of users and messages

Technology Stack

- Backend:
  - Python
  - FastAPI
  - WebSockets
  - SQLAlchemy
  - SQLite
- Authentication:
  - JWT (python-jose)
  - Passlib + Bcrypt
- Frontend:
  - HTML
  - CSS
  - JavaScript
- Machine Learning:
  - Custom bad-word detection logic

Project Architecture

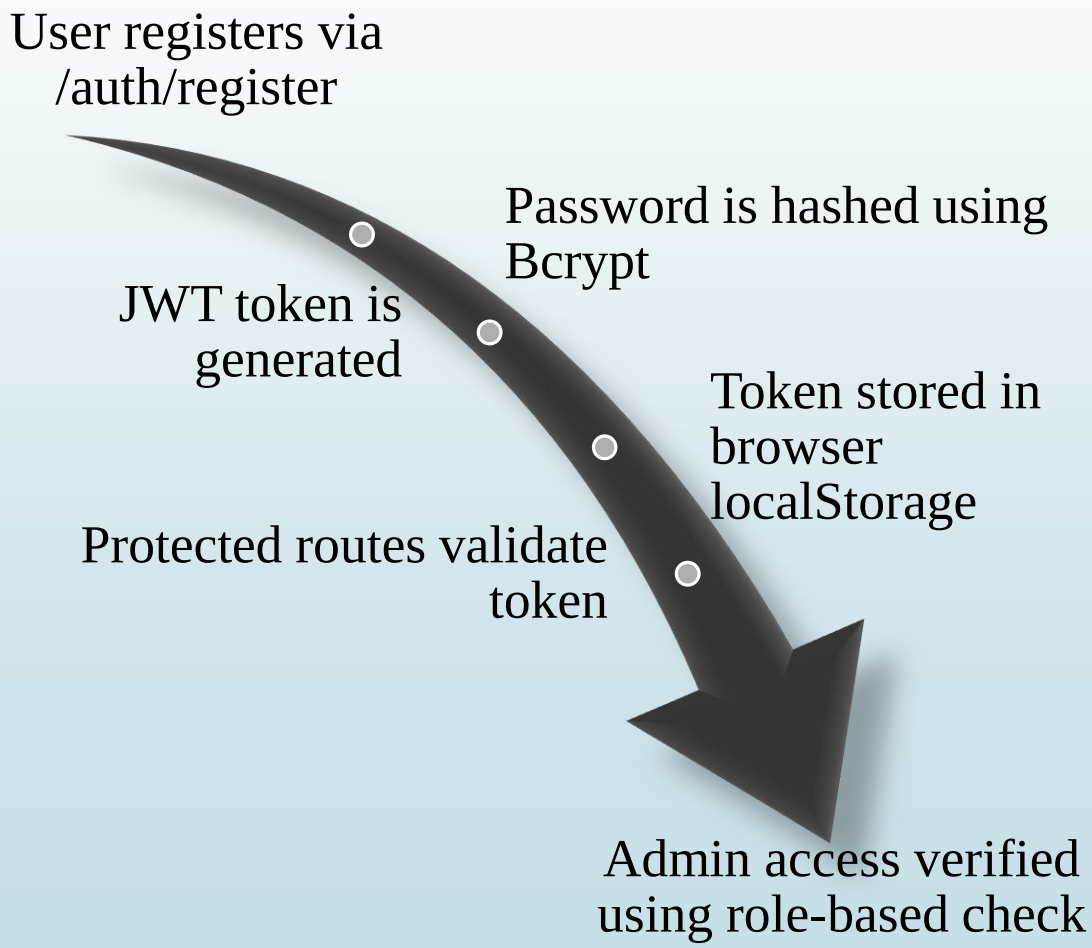


Project Structure

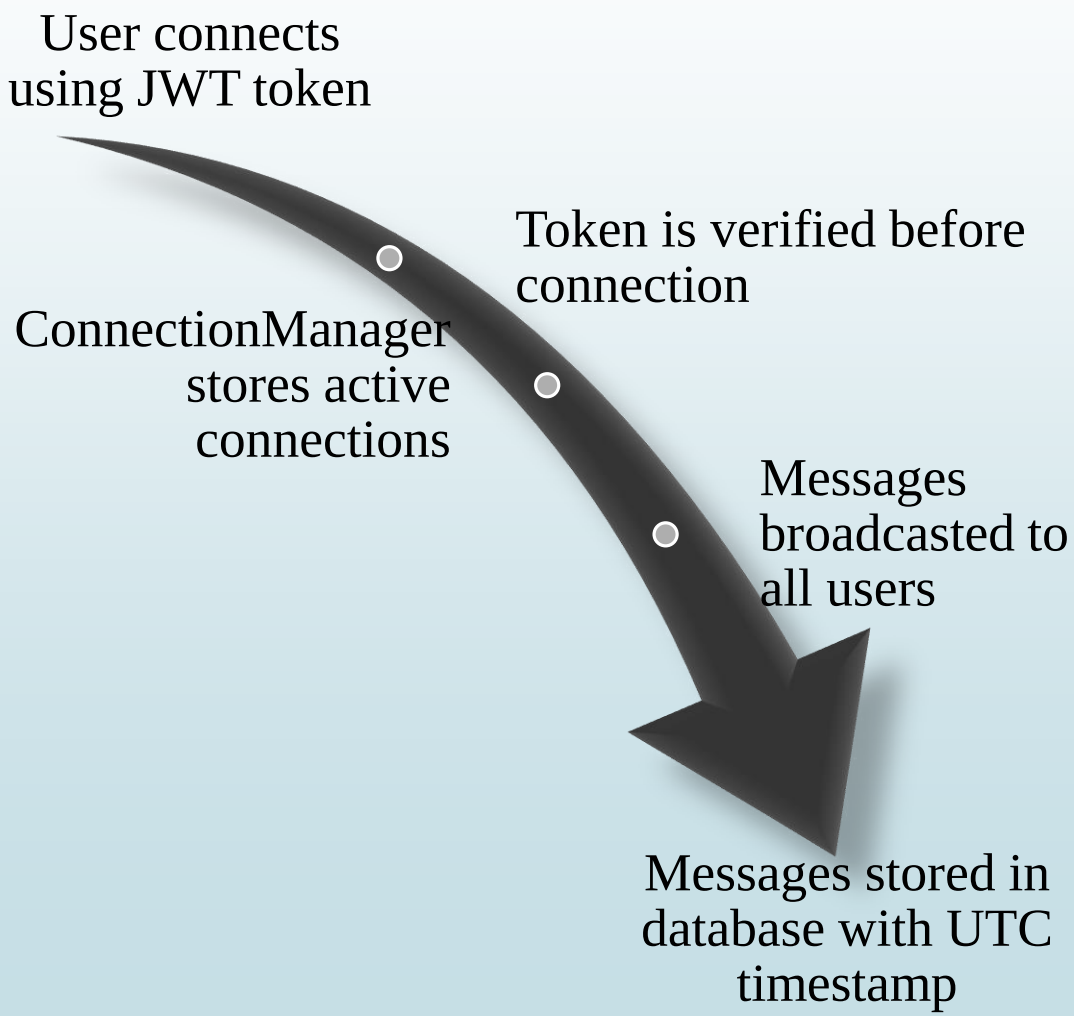
- Backend:
  - Auth Module
  - Admin Module
  - Chat Module
  - ML Moderation Module
  - Utils & Security Module
  - Database & Models

- Frontend:
  - User Login & Registration
  - Real-Time Chat Interface
  - Admin Login & Dashboard

Authentication Flow



Real-Time WebSocket Flow



Machine Learning Moderation System

- • Custom bad-word detection module
  • Tracks warning_count per user
  • 1st violation → Warning
  • 2nd violation → Warning
  • 3rd violation → Auto Block
- Blocked users cannot:
  • Login
  • Send messages

Admin Features

- Secure Admin Login
- Dashboard Summary
- Total Users Count
- Total Messages Count
- Blocked Users Count
- Recent Messages Monitoring
- Download Users CSV
- Download Messages CSV

Database Design

- Users Table:
  - id
  - username
  - email
  - password_hash
  - is_admin
  - is_blocked
  - warning_count
  - created_at

- Messages Table:
  - id
  - sender_id
  - receiver_id
  - room
  - content
  - timestamp

Key Features

- Real-time WebSocket communication
- JWT-based authentication
- Role-based admin control
- ML-based moderation
- Warning & Auto-block system
- CSV Export functionality
- Clean and responsive UI
- Date separator logic (Today / Yesterday)
- UTC timestamp handling

Challenges Faced

- Implementing secure JWT authentication
- Managing WebSocket token verification
- Handling Passlib & Bcrypt compatibility
- Designing auto-block logic
- Implementing CSV streaming response
- Timezone mismatch issues

Learning Outcomes

- Deep understanding of FastAPI architecture
- Real-time communication using WebSockets
- Secure authentication using JWT
- Database modeling using SQLAlchemy
- Role-based access control
- Backend + Frontend integration
- Admin monitoring systems

Conclusion

- The ChatterBox project successfully demonstrates:
- • A scalable real-time communication system
  • Secure authentication and authorization
  • Intelligent moderation mechanism
  • Complete admin control system
  • Professional full-stack development skills

Future Improvements

- Deploy on cloud server
- Add private chat rooms
- Implement proper ML model training
- Add file/image sharing
- Improve UI animations

Thank You
Questions?