

Multi-Chatbot Creation Platform Specification

Objective

Build a **functional-first platform** to create, manage, and interact with AI chatbots, focusing on the **core backend functionality**. The UI is secondary, and candidates are encouraged to use AI-generated UI tools for faster development. The evaluation will prioritize **code structure**, logic, and implementation of AI-driven features over UI aesthetics.

Core Features

1. User Management

- **Registration:**
 - Unique username, email, and strong password.
 - Basic validation (no email verification).
- **Login:**
 - Username/email + password with session timeout after 24 hours.
- **Dashboard:**
 - Summary of created chatbots with stats (total bots, creation date).

2. Chatbot Creation Workflow

- **Step-by-Step Wizard:**
 1. **Basic Information:** Define chatbot name and description.
 2. **Personality Setup:** Configure tone and behavior via prompts.
 3. **Knowledge Base:** Upload documents (PDF/TXT, max 5MB).
- **Document Processing:**
 - Chunk large files for embedding generation.
 - Use any open-source embedding model to store vectors in a database (e.g., Chroma).
- **Testing Interface:**
 - Real-time testing of chatbot during setup to validate responses.

Core Functionalities

AI-Powered Responses

- Use **open-source APIs** to handle chatbot responses (e.g., GPT-like models).im not providing my api for task cause it could be misused

- Ensure the model can leverage the vectorized knowledge base for contextual replies.

Knowledge Base Enhancements

- Preprocess uploaded documents (split, clean, and vectorize).
- Implement **semantic search** to fetch precise knowledge snippets.

Optional Add-Ons (For Bonus Points)

1. **Dynamic Personality Configuration:**
 - Allow users to tweak tone via sliders (friendly, formal, witty).
2. **Advanced Document Management:**
 - Add **version control** for uploaded documents.

Recommendations for UI

- **AI-Generated UI:**

Use AI-based tools (e.g., Figma AI, ChatGPT plugins for front-end generation) to quickly scaffold UI designs.

- **Focus on Functionality:**

The UI should only serve as a bridge to showcase backend functionality.

Technology Stack

Backend

- Python frameworks like Django/Flask/FastAPI for core logic.
- Open-source embedding models (e.g., sentence-transformers, HuggingFace).
- Vector database (e.g., Chroma) for document embeddings.

Deployment

- anywhere just so that we can test the bot

Evaluation Criteria

1. **Functionality (60%):**
 - End-to-end chatbot creation and real-time testing.

- Reliable document processing and embedding generation.
- 2. **Code Quality (30%):**
 - Clean, modular, and well-commented code.
 - Logical structure with a clear separation of concerns.
- 3. **Optional Features (10%):**
 - Implement advanced features like dynamic personalities or API integrations.

Submission Requirements

1. **Source Code:**
 - GitHub repository with README and requirements file.
2. **Live Demo:**
 - Deployed version with test credentials for evaluation.