

Backend Assessment: AI Agent Platform

Overview

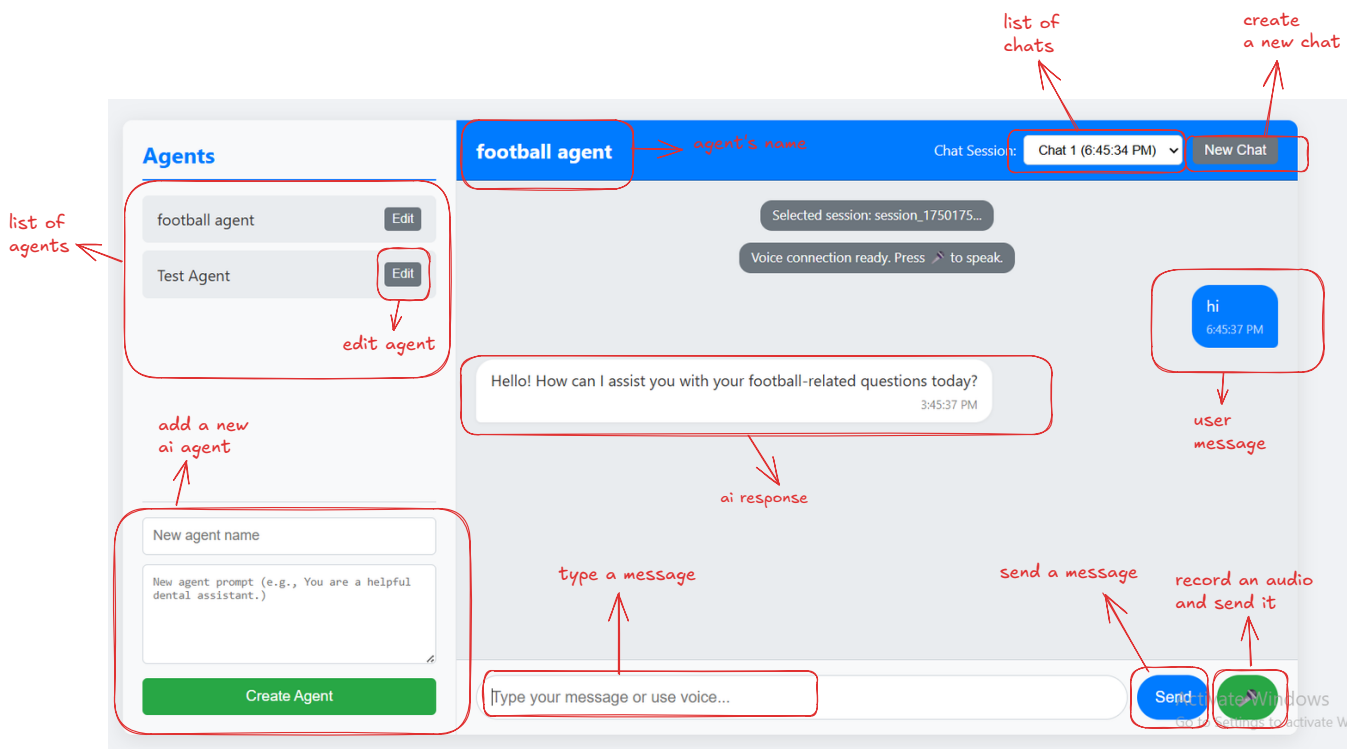
This assessment evaluates your skills as a backend software engineer. You will design and implement a **backend service** for an AI Agent Platform, enabling users to create, manage, and interact with AI agents via text and voice. The platform will integrate with OpenAI APIs for chat, speech-to-text, and text-to-speech functionalities.

You are expected to demonstrate proficiency in Python, RESTful API design, database modeling, third-party API integration, and best practices in code quality, testing, and documentation. You are encouraged to use AI tools (e.g., GitHub Copilot, ChatGPT) to assist your workflow.

Frontend Mockup

building frontend is optional.

The backend service you build should support the functionalities depicted in the following user interface mockup. It illustrates the core features: agent creation and management, session handling, and interaction via text and voice.



Functional Requirements

1. Agent Management

- Users can create, update, and list AI agents.
- Each agent has a name and a prompt (system message).

2. Chat Sessions

- Users can start and manage multiple chat sessions per agent.
- Each session stores a chronological history of messages between the user and the agent.

3. Messaging

- Users can send text messages to an agent within a chat session.
- The backend should use OpenAI's API to generate agent responses.
- All messages (user and agent) are stored in the database with timestamps.

4. Voice Interaction (Asynchronous)

- Users can record and send a complete voice message (e.g., via a push-to-talk button that sends the file upon release).
- The backend receives an audio file, converts it to text (Speech-to-Text), generates a chat response, and converts that response back into an audio file (Text-to-Speech).
- The final audio response is sent back to the client to be played. This is an asynchronous request/response model, not a real-time stream.

5. API Documentation

- Provide interactive API documentation (e.g., Swagger/OpenAPI) for all endpoints.

6. Testing

- Implement tests to cover core backend functionality.

7. Postman Collection

- Supply a Postman collection for testing all API endpoints.

8. Deployment

- Provide a Dockerfile to containerize the backend service.
- Include a `requirements.txt` listing all dependencies.

9. Documentation

- Write a clear `README.md` with setup, configuration, and usage instructions.

10. (Optional) Simple Web Interface

- (Bonus) Provide a minimal frontend to interact with your backend.

Technical Requirements

- **Language:** Python 3.10+
- **Framework:** FastAPI (preferred) or similar
- **Database:** SQLite (or any SQL database)
- **ORM:** SQLAlchemy (async recommended)
- **OpenAI Integration:** Use provided API key for chat, speech-to-text, and text-to-speech.

- **API Design:** Use standard REST principles for interactions, including file uploads for voice messages.
 - **Testing:** pytest or unittest
 - **Documentation:** Swagger/OpenAPI (auto-generated by FastAPI or similar)
 - **Containerization:** Docker
-

Evaluation Criteria

- **Functionality:** Completeness and correctness of the backend features
 - **Code Quality:** Structure, readability, maintainability, and use of best practices
 - **Testing:** Coverage and reliability of tests
 - **Documentation:** Clarity and completeness of API docs and README
 - **Deployment:** Ease of setup and containerization
-

Deliverables

- Source code for the backend service
 - Database models and migration scripts (if any)
 - API documentation (Swagger/OpenAPI)
 - Postman collection for API testing
 - Test suite with instructions to run tests
 - Dockerfile and requirements.txt
 - README.md with setup and usage instructions
 - **a short video showing the functionality of the backend service.**
 - (Optional) Simple frontend for demonstration.
-

Notes

- You will be provided with an OpenAI API key (with limited access) for development and testing.
 - Do **not** commit sensitive information (e.g., API keys) to version control.
 - You are encouraged to use AI tools to assist your development process.
-

Good luck!