

Project Summary Report: AI-INTERVIEWER

1. Project Overview

AI-INTERVIEWER is an AI-powered interview simulation tool that enables users to practice for interviews by interacting with an intelligent agent. It uses **Python**, **Streamlit**, and **OpenAI's GPT models** to generate real-time, dynamic interview questions and feedback based on a user's uploaded resume or profile.

- **GitHub Repository:** [jiakash424/AI-INTERVIEWER](#)
- **Primary Technology Stack:** Python, Streamlit, OpenAI API

2. Key Features

-  **Resume/Profile Upload:** Users can upload their resume for personalized questions.
-  **Real-Time Interactive Q&A:** Simulates dynamic interview conversations.
-  **Streamlit Interface:** Provides an intuitive, easy-to-use web application.
-  **API Integration:** Connects with OpenAI's models for generating interview questions and responses.
-  **Testing Framework:** Basic testing structure included for development validation.

3. Repository Structure

File/Folder	Purpose
Ai_interviewer.py	Main application file to run the Streamlit app.
app_test.py	Contains tests to validate basic app functionalities.
requirements.txt	Production dependencies required to run the app.
requirements-dev.txt	Additional development dependencies (e.g., for testing, linting).
.ruff.toml	Configuration file for the Ruff Python linter.

4. Setup Instructions

- **Clone the Repository:**

bash

CopyEdit

```
git clone https://github.com/jiakash424/AI-INTERVIEWER.git
```

```
cd AI-INTERVIEWER
```

- **Install Requirements:**

bash

CopyEdit

```
pip install -r requirements.txt
```

- **Set OpenAI API Key:**

bash

CopyEdit

```
export OPENAI_API_KEY='your-api-key'
```

- **Run the Application:**

bash

CopyEdit

```
streamlit run Ai_interviewer.py
```

5. Current Development Status

- The project appears to be in an **early or prototype stage** with limited commits and features.
- There is **potential for further expansion**, such as:
 - Adding mock interview scoring.
 - More resume analysis for deeper personalized interviews.

- Integration of behavioral and technical question sets.
- Summarized feedback after each mock interview.

6. Potential Improvements

- **Add interview feedback summaries** (strengths, weaknesses).
- **Enhance resume parsing** for better contextual questioning.
- **Build scoring/rating mechanisms** for interview answers.
- **Expand tests** to ensure app reliability.
- **UI/UX enhancements** with more polished Streamlit designs.

7. References

- [AI-INTERVIEWER GitHub Repository](#)
- Streamlit Official Documentation
- [OpenAI API Documentation](#)