

User Manual

Title of Your Project: AI Interviewer APP

**User Manual**

**Title of Your Project: AI Interviewer App**

**1. About the App**

**Title of App: AI Interviewer App**

**Student Developer:**

DeerajKumar S M(R22EF053)

Riya K Janali (R22EN065)  
Pruthvi Anil Gola (R22EN060)

**Description:**

The **AI Interviewer App** is an advanced Python-based application that simulates real-time job interviews using artificial intelligence. It supports both **text and voice-based interaction**, allowing candidates to communicate in the mode they’re most comfortable with. The app conducts interviews by asking **technical and coding-related questions**, adapting dynamically based on the user's responses.

One of its standout features is the built-in **code execution environment**, where users can write, run, and submit their code within the app itself. After the interview session, the system analyzes the responses—both verbal and written—and provides **detailed feedback** highlighting strengths, areas for improvement, and performance scores.

This application is ideal for students and professionals preparing for technical interviews, offering a comprehensive and interactive mock interview experience.

**2. System Requirements**

**Hardware Requirements::**

* **Minimum Requirements:**
  + Processor: Dual-Core CPU (Intel i3 or equivalent)
  + RAM: 4 GB
  + Storage: 1 GB free disk space
  + Microphone and speakers/headphones for voice input/output
* **Recommended Requirements:**
  + Processor: Quad-Core CPU (Intel i5/i7 or equivalent)
  + RAM: 8 GB or higher (for smoother multitasking and faster processing)
  + GPU: Optional (recommended if you use local model inference)
  + High-speed internet for API communication and real-time response

**Software Requirements:**

* **Operating System Compatibility:**
* Windows 10 or later
* macOS Catalina or later
* Ubuntu/Linux (latest LTS version recommended)
* **Programming Language:**
* Python 3.10 or higher
* **Libraries and Dependencies:**

The following Python libraries and tools are required for the functionality of the app:

* **User Interface & Web Framework**
  + streamlit – for creating the web interface
* **Audio Input/Output**
  + sounddevice – for recording user audio
  + scipy – for saving recorded audio
  + pyttsx3 / gTTS – for converting text responses to speech
  + pygame – for playing audio output
* **Speech-to-Text**
  + faster\_whisper – for transcribing audio using Whisper ASR models
* **API Integration**
  + groq – to connect and communicate with Grok API
* **Utilities & System**
  + uuid, datetime, time, os, subprocess, tempfile, threading – standard libraries for system utilities and process handling
* **Data Processing & Visualization**
  + numpy – for numerical computations
  + matplotlib – for generating performance charts and plots
* **Text & Image Handling**
  + PIL (Pillow) – for creating feedback cards with custom text/images
  + textwrap – for formatting text output
* **Others**
  + base64, io – for handling encoded media and file streams

**Optional Tools (Recommended for Development)**

* Visual Studio Code / PyCharm – IDE for development
* Git – for version control
* Postman – for API testing

**3. Installation Guide:**

**Step 1: Clone the Project Repository**

* Download or clone the project from GitHub:

<https://github.com/Deeraj1509/AI-Interviewer-App>

**Step 2: Navigate to the Project Directory**

Use the terminal or command prompt to move into the project folder:

* **cd path\to\ai-interviewer-app**

**Step 3: Install Required Libraries**

Make sure Python 3.10 or later is installed. Then install all required Python libraries using:

* **pip install -r requirements.txt**

If a requirements.txt file is not provided, install the libraries manually:

* **pip install streamlit sounddevice scipy faster-whisper pyttsx3 groq matplotlib numpy pillow gtts pygame**

**Step 4: Get Your Groq API Key**

* Sign up at https://console.groq.com
* Generate your API key.
* Create a .env file in the root of your project and add the following line:
* GROQ\_API\_KEY=your\_api\_key\_here

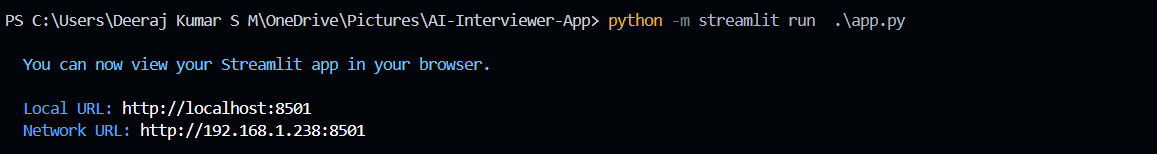
**Step 5: Run the App**

Start the Streamlit app using the following command:

* **python -m streamlit run .\app.py**

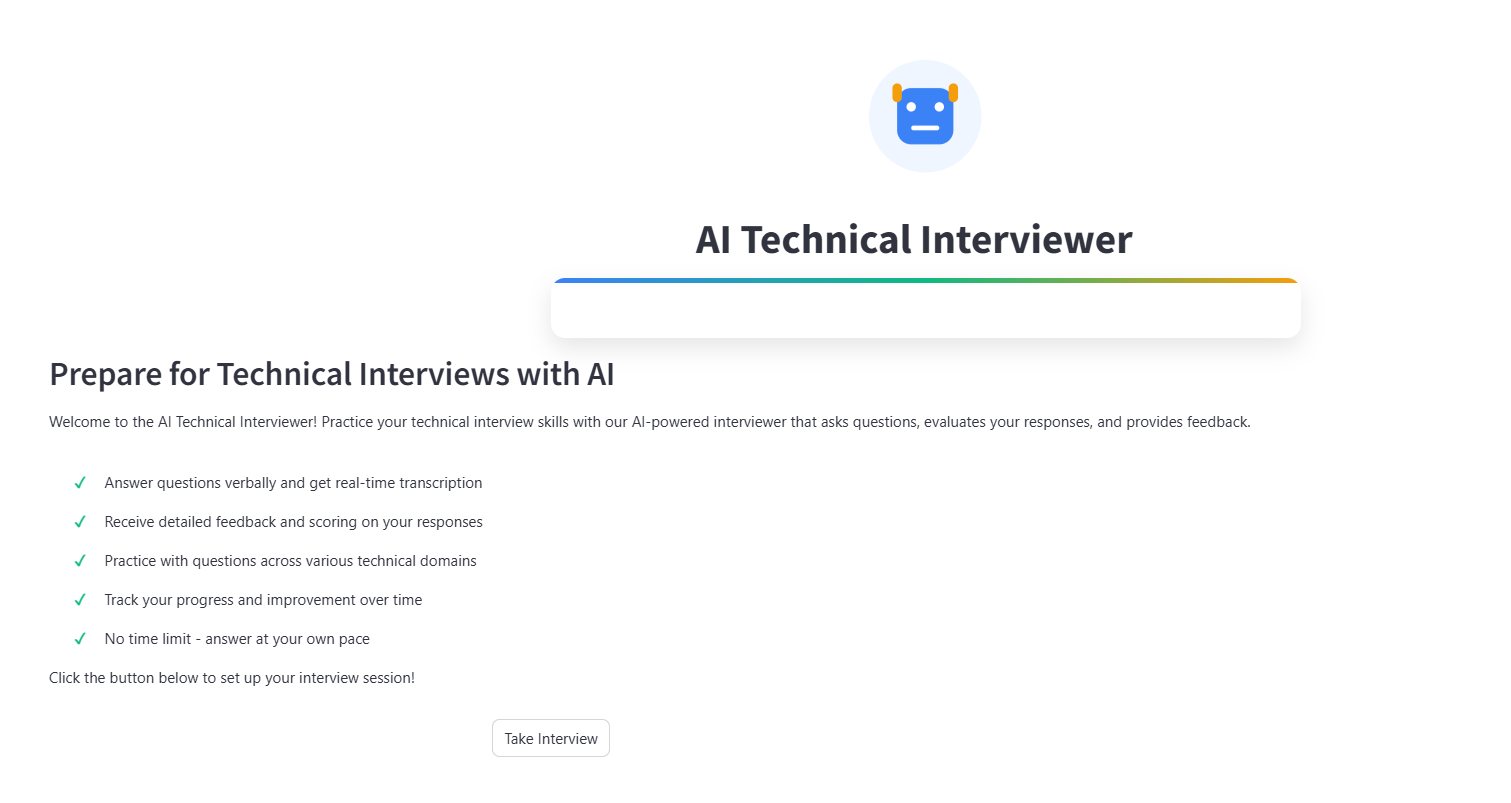
**4. Using the App**

After running the application, open your browser and navigate to:



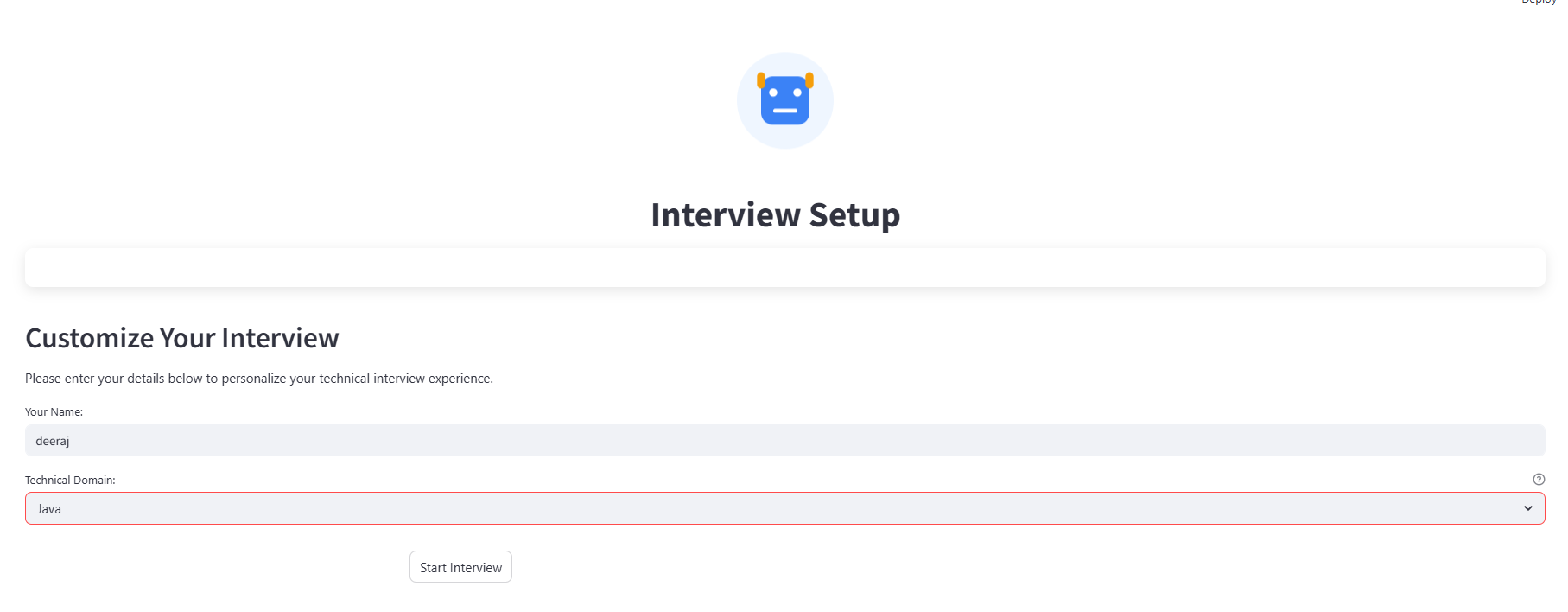
**Step 1: Start the Interview Process**

* Click on the **"Take Interview"** button on the homepage to begin.



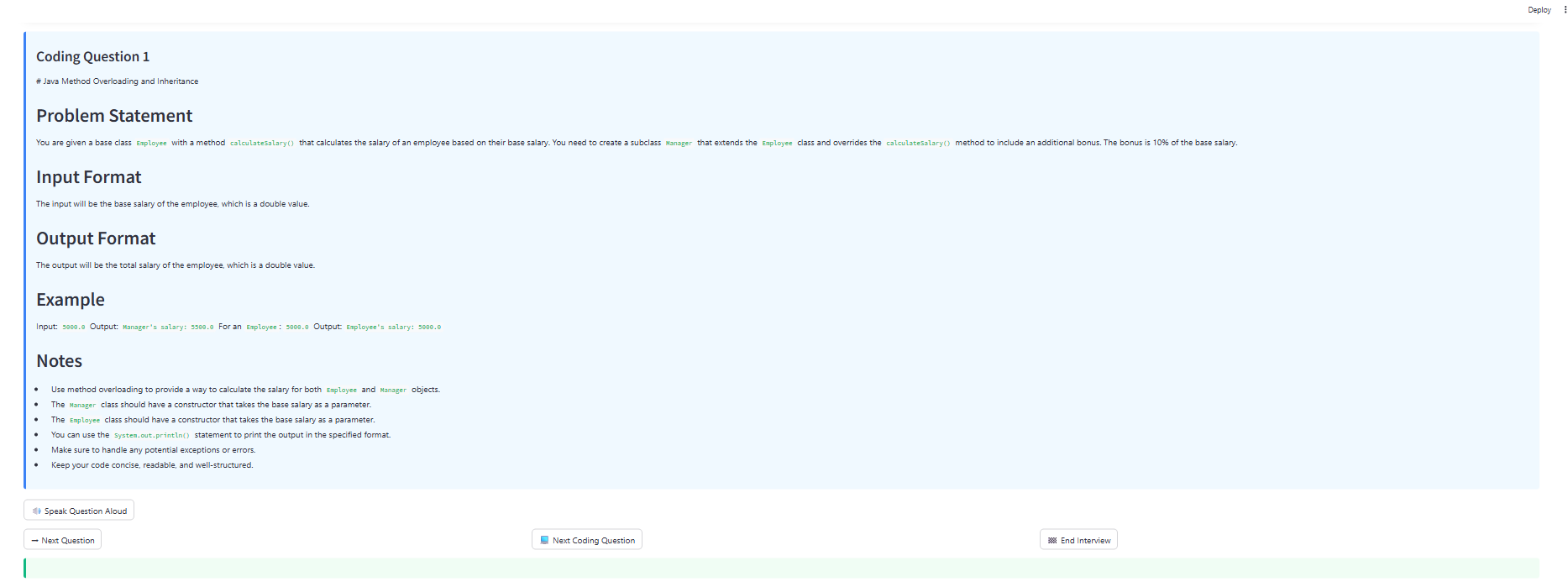
**Step 2: Enter Candidate Details**

* Enter your **name**.
* Choose your **domain/field** (e.g., Python, Web Development, Data Science).
* Click **"Start Interview"** to begin the session.



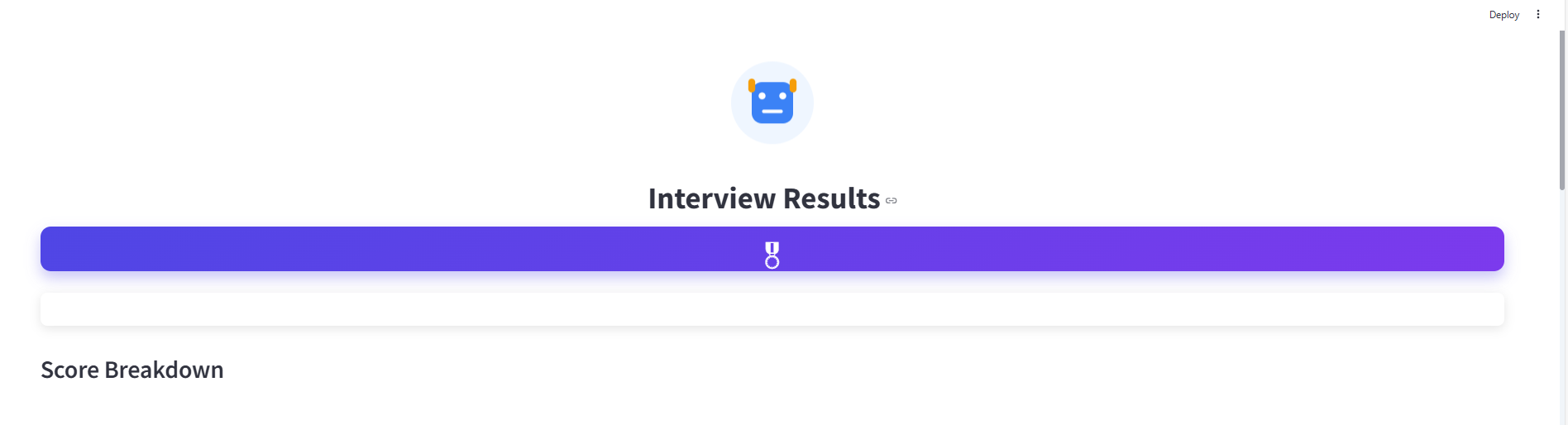
**Step 3: Interview Flow**

* The app will begin asking **coding or soft skill questions** one by one.
* You can **answer using text or voice** input.
* After answering each question, click **"Submit"**.
* The next question will appear automatically—this continues for both coding and behavioral (soft skill) sections.
* After all questions, click **"End Interview"** to finish.



**Step 4: View Results and Download Certificate**

* Once you end the interview, a **scorecard** will be displayed showing:
  + Performance summary
  + Scores in technical and soft skills
  + Feedback and suggestions
* A **certificate of participation** will be generated, and you will have the option to **download it as a PDF**.





**5. Features Explained**

**1. Text and Voice Input Support**

* Users can choose to answer interview questions using either typed text or spoken voice input.
* Voice input is transcribed using the Whisper model (faster-whisper), ensuring accurate and fast speech-to-text conversion.

**2. Grok API Integration**

* The app integrates with the Grok API to fetch intelligent, domain-specific interview questions.
* It dynamically selects coding and behavioral questions tailored to the user's chosen field and expertise level.

**3. Coding Question Execution**

* For technical questions, the app provides a built-in code editor where candidates can write and test their code.
* Code is executed securely using Python’s subprocess library, enabling real-time output and validation of coding answers**.**

**4. Soft Skill Assessment**

* The app includes HR and soft skill questions to evaluate behavioral attributes such as communication, decision-making, and leadership.
* Answers are evaluated and logged for post-interview analysis and scoring.

**5. Interactive Interview Flow**

* The interview proceeds in a structured manner: one question at a time, auto-advancing after each submission.
* It mimics a real-time interview session for both technical and soft skills.

**6. Scorecard Generation**

* After completing the interview, a detailed scorecard is presented.
* It highlights performance in coding and communication, along with personalized feedback.

**7. Certificate of Completion**

* A visually designed certificate is generated automatically at the end, which users can download as a PDF for their records.

**6. Troubleshooting**

1. **App not starting**
   * **Cause:** Missing dependencies or incorrect environment setup.
   * **Solution:** Make sure all required libraries are installed using pip install -r requirements.txt. Also, ensure that Python 3.10 or later is correctly installed.
2. **Audio not recording or microphone not detected**
   * **Cause:** Microphone not connected or access not granted.
   * **Solution:** Check if the microphone is properly connected and accessible by the system. On Windows, verify microphone permissions in privacy settings.
3. **Voice input not converting to text**
   * **Cause:** Whisper model not loaded correctly or incompatible audio format.
   * **Solution:** Ensure the faster-whisper library is installed and functional. Check for supported audio formats (typically .wav) and confirm that the model directory is properly configured.
4. **Grok API not responding**
   * **Cause:** Invalid or missing API key.
   * **Solution:** Make sure your Grok API key is correctly placed in the .env file with the variable GROQ\_API\_KEY. Also, ensure you have internet connectivity.
5. **Code not executing or crashing**
   * **Cause:** Syntax error in user-submitted code or subprocess failure.
   * **Solution:** Validate the syntax before executing. Make sure the code editor handles Python code securely and uses subprocess with proper error handling.
6. **Streamlit app not loading in browser**
   * **Cause:** Port conflict or firewall blocking.
   * **Solution:** Make sure no other app is using port 8501. You can change the port by running:

****

1. **Certificate not downloading**
   * **Cause:** File write permissions or browser issues.
   * **Solution:** Check browser download permissions and ensure the directory where the certificate is saved is writable.

**7. Contact / Support**

For any queries, technical support, or feature requests, feel free to reach out to the developers:

1. **Deeraj Kumar S M**  
   Email: smdeeraj.2002@gmail.com  
   LinkedIn: [www.linkedin.com/in/deerajkumar-s-m-7211a3261](https://www.linkedin.com/in/deerajkumar-s-m-7211a3261)
2. **Riya K Janali**  
    Email: riyakj93@gmail.com  
    LinkedIn: [linkedin.com/in/riya-kj-8936a1294](https://www.linkedin.com/in/riya-kj-8936a1294)
3. **Prithvi Anil Gola**  
   Email: pruthvigola2020@gmail.com  
   LinkedIn: <https://www.linkedin.com/in/pruthvi-anil-gola-bb217135b>