INFERNO

**-AI-Powered Interview Assistant**

📌 Project Overview

The AI-Powered Interview Assistant is an advanced interview simulation tool that leverages AI to evaluate candidates performance based on real-time video, voice analysis, and technical skills assessment. It provides dynamic questioning, facial recognition, speech analysis, and coding evaluation to enhance the hiring process and also helps the candidate to prepare for the interview by providing mock interviews.

**🚀 Features**

* **Interview Simulation**: Predefined questions with AI-based evaluation and feedback.
* **Dynamic Questioning**: Adapts based on candidate responses.
* **Facial Analysis**: Detects confidence, hesitation in face , and engagement using DeepFace, OpenCV, and DeepGaze.
* Voice Analysis: evaluates the candidates voice tone, confidence and keyword detection using whisper.
* **Gaze Tracking**: Analyzes eye contact and screen focus.
* **Technical Skills Assessment**: Supports coding challenges with Judge0 API.
* **Real-Time Feedback**: Uses WebRTC for live interview evaluation.

**🛠️ Tech Stack**

* **Frontend**: HTML,CSS,Javascript
* **Backend**: FastAPI (Python)
* **Machine Learning**: DeepFace (Facial Recognition), DeepGaze (Eye Tracking), OpenCV (Video Analysis)
* **Speech Analysis**: Whisper
* **Real-Time Processing**: WebRTC (Live Video & Audio)
* **Coding Evaluation**: Judge0 API (Multi-language Support)
* **Cloud Deployment**: AWS/GCP

**📂 Folder Structure**

├── frontend/ # WordPress-based UI components  
├── backend/ # FastAPI backend logic  
│ ├── models/ # AI models for face, gaze, and speech analysis  
│ ├── routes/ # API endpoints  
│ ├── services/ # Processing and evaluation logic  
├── data/ # Sample datasets & test cases  
├── requirements.txt # Python dependencies  
├── README.md # Project Documentation  
└── run.sh # Script to launch the application

**🔧 Installation & Setup**

**Prerequisites**

* Python 3.10+
* Node.js & npm (if modifying frontend)
* Virtual environment (recommended)
* WinRAR/ZIP tool to extract files

**Backend Setup**

cd backend  
python -m venv venv  
source venv/bin/activate # On Windows: venv\Scripts\activate  
pip install -r requirements.txt  
uvicorn main:app --reload

**Frontend Setup (if needed)**

cd frontend  
npm install  
npm start

**🚀 Running the Project**

1. Start the backend: uvicorn main:app --reload
2. Start the frontend: npm start
3. Access the application via <http://localhost:3000>

**📌 Contributors**

* Mr. Tejdeep Reddy K– Backend Development
* Mr. Ramcharan P – AI & Machine Learning
* Mr. Sai Vishwak K – Frontend & UI/UX
* Mr.Ramakrishna -Worked on API keys

**📜 License**

This project is open-source and available under the MIT License.

**📞 Contact**

For queries, reach out at [Inferno@tech.in](mailto:Inferno@tech.in) or visit our GitHub repository.