Interview Bot

**1.AIM :**

The primary aim of the project is to develop an AI-powered interview bot that:

1. **Automates the Interview Process**: The bot guides candidates through various rounds, including Multiple Choice Questions (MCQs) and technical Q&A, based on the role they have applied for.
2. **Provides Dynamic Content**: Customizes questions according to the selected job role to make the interview relevant and focused.
3. **Enhances Interview Integrity**: Detects potential cheating attempts by identifying multiple faces and suspicious objects in the camera feed, ensuring a fair assessment of candidates.
4. **Ensures Data Logging**: Records and stores candidates' responses and relevant metadata in structured files for easy review and analysis

**2.INTRODUCTION :**

In recent years, the demand for remote interview solutions has increased significantly, driven by the need for virtual hiring processes and the growing trend of digital transformation. This project, the "Interview Bot Application," aims to streamline the interview process by automating candidate interactions, recording responses, and dynamically selecting questions based on the candidate's role. To enhance the integrity of the interview, we have integrated advanced cheating detection features, including multi-person detection and object tracking, which ensure a fair interview environment.

**3. OBJECTIVE :**

The project objectives include:

* **Dynamic Question Selection**: To dynamically load MCQs and technical questions tailored to specific job roles (e.g., Java Developer, MERN Stack Developer, Full Stack Developer, AI/ML Developer).
* **Automated Response Logging**: To capture and log candidates’ responses in real-time, ensuring accurate data collection for review.
* **Real-Time Cheating Detection**: To monitor the video feed and detect potential cheating by identifying additional faces, digital devices, and unauthorized materials during the interview process.
* **User-Friendly Interface**: To develop a simple, intuitive interface that guides candidates smoothly through the interview process.

**4 . PROJECT TECNOLOGIES :**

This project utilizes a combination of web development, machine learning, and computer vision technologies:

* Flask: Used as the primary framework for developing the backend of the application. Flask enables the server to render HTML templates and handle form submissions.
* HTML, CSS, JavaScript: Used for designing the front-end interface, providing an engaging and interactive user experience.
* OpenCV: A powerful library for real-time computer vision, employed for detecting faces and objects within the camera feed to monitor potential cheating behaviors.
* Machine Learning Models: Pre-trained models are integrated for object detection to detect electronic devices and other suspicious objects during the interview.
* File Management (CSV): For logging and storing candidate responses along with interview metadata, ensuring data is easily accessible for review.
* Datetime Module: For timestamping candidate responses, adding another layer of data integrity.

**5. SYSTEM DESIGN AND ARCHITECTURE :**

**5.1 Application Structure**

The application consists of several core components:

1. Start Page: Gathers initial candidate information, including name, email, college, and job role applied for. The selected role determines the questions asked in the subsequent rounds.
2. MCQ Round: Presents multiple-choice questions relevant to the chosen job role. The bot logs each answer in real-time for future review.
3. Q&A Round: Asks technical questions related to the candidate’s selected role. Answers are captured and stored for assessment.
4. Cheating Detection Module: Continuously monitors the video feed, detecting additional faces, objects, and gaze direction to flag suspicious activity.
5. Interview Completion Page: Indicates the end of the interview process, thanking candidates and providing feedback where necessary.

**5.2 Data Flow and Logging**

Data captured during the interview (e.g., responses, timestamps) is logged in a CSV format. This structured format allows easy data retrieval, analysis, and reporting, ensuring transparency in the interview evaluation process.

**6. Implementation Details**

6.1 Dynamic Question Selection

The application retrieves questions dynamically based on the role the candidate applies for, ensuring a customized interview experience. For example:

* Java Developer: Questions on Java syntax, object-oriented programming, and Java-specific tools.
* MERN Stack Developer: Questions on MongoDB, Express, React, and Node.js.
* Full Stack Developer: A combination of frontend and backend questions.
* AI/ML Developer: Questions on machine learning, data science, and Python.

**6.2 Cheating Detection Using OpenCV**

OpenCV is used to enable real-time face and object detection:

* **Face Detection**: The application continuously scans the video feed for multiple faces. If additional faces are detected, the bot logs a cheating alert.
* **Object Detection**: Object detection models recognize specific items such as smartphones, tablets, or books, which may indicate cheating. If detected, alerts are flagged.
* **Gaze Detection**: Gaze tracking monitors the candidate’s line of sight, ensuring they remain focused on the screen. Prolonged gaze deviations may also trigger alerts.

**6.3 Logging System**

The application logs candidate responses, timestamps, and interview metadata in a CSV file. This file allows:

* Easy Review: Logged data can be reviewed by recruiters to make data-driven decisions.
* Timestamp Verification: Ensures transparency by logging the time each question was answered.

**8. CONCLUSION :**

The Interview Bot Application provides an innovative and efficient solution for conducting remote interviews. By integrating dynamic questioning, automated logging, and real-time cheating detection, it simplifies the process for both candidates and interviewers, ensuring integrity and reliability. This project demonstrates the potential of AI and machine learning to enhance traditional processes, laying the groundwork for more secure and efficient remote hiring practices.