A

PRELIMINARY PROJECT REPORT

ON

AI-Powered Virtual Interviewer for Improving Candidate Communication Skills

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING
INFORMATION TECHNOLOGY

BY

Rajvardhan Deshmukh	SPPU Seat No.
Soham Gandhi	SPPU Seat No.
Sakshi Gangurde	SPPU Seat No.
Rishikesh Ghodke	SPPU Seat No.

Under the guidance of
Dr. Shyam Deshmukh

Department Of Information Technology
Pune Institute of Computer Technology
Pune - 411 043.
2025-2026

SCTR's PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the preliminary project report entitled
AI-Powered Virtual Interviewer for Improving Candidate Communication Skills
submitted by

Rajvardhan Deshmukh	SPPU Seat No.
Soham Gandhi	SPPU Seat No.
Sakshi Gangurde	SPPU Seat No.
Rishikesh Ghodke	SPPU Seat No.

is a bonafide work carried out by them under the supervision of **Dr. Shyam Deshmukh** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology).

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Dr. Shyam Deshmukh
Project Guide

Dr. Emmanuel M.
HOD IT

SPPU External Guide

Dr. S. T. Gandhe
Principal

Date:
Place: Pune

Acknowledgement

We would like to express our sincere gratitude to our project guide, **Dr. Shyam Deshmukh**, for his invaluable guidance, support, and encouragement throughout the development of this project. His expertise and constructive feedback were instrumental in shaping the direction and quality of this work.

We are also thankful to the Head of the Department, **Dr. Emmanuel M.**, and all the academic and technical staff of the Department of Information Technology for providing the necessary facilities, resources, and a conducive environment for our research.

Finally, we extend our appreciation to our families and peers for their constant support and motivation, which has been crucial to the completion of this preliminary report.

Rajvardhan Deshmukh
Soham Gandhi
Sakshi Gangurde
Rishikesh Ghodke

Abstract

The proliferation of Artificial Intelligence (AI) has catalyzed a paradigm shift in professional development, partic ularly in interview preparation. Modern AI-driven mock inter view systems represent a significant technological convergence, fusing the analytical capabilities of Affective Computing with the synthetic power of Generative Models. This paper presents a comprehensive review of this emerging field. We dissect the core AI methodologies that underpin these platforms, including unimodal analysis of facial expressions, speech, and text, as well as advanced multimodal fusion techniques that enable a holistic assessment of a candidate's communication skills. Fur thermore, we examine the role of generative architectures, such as Transformer-based models and Retrieval-Augmented Genera tion (RAG), in creating dynamic, personalized, and role-specific interview experiences. While these systems offer unprecedented opportunities for scalable and objective feedback, they also introduce significant challenges related to technical robustness, data scarcity, algorithmic bias, and ethical governance. This review critically evaluates these challenges, synthesizing findings from recent literature to identify key research gaps. We conclude by outlining future research directions, emphasizing the need for explainable AI (XAI), the development of cross-cultural datasets, and the establishment of robust ethical frameworks to ensure these technologies serve as equitable and effective tools for candidate empowerment. The analysis underscores that the future of AI in this domain lies in creating human-centric systems that augment, rather than automate, the nuanced process of human assessment.

Index Terms—Affective Computing, Generative Models, Mock Interview, Speech Emotion Recognition (SER), Facial Expression Analysis (FEA), Multimodal Fusion, Human-Computer Interaction (HCI), Algorithmic Bias.

Contents

List of Figures

List of Tables

Abbreviations

AI	:	Artificial Intelligence
API	:	Application Programming Interface
CNN	:	Convolutional Neural Network
FEA	:	Facial Expression Analysis
GPT	:	Generative Pre-trained Transformer
HCI	:	Human-Computer Interaction
LLM	:	Large Language Model
LSTM	:	Long Short-Term Memory
MFCC	:	Mel-Frequency Cepstral Coefficients
NLP	:	Natural Language Processing
RAG	:	Retrieval-Augmented Generation
SDG	:	Sustainable Development Goal
SER	:	Speech Emotion Recognition
WER	:	Word Error Rate

1. Introduction

1.1 Introduction

In an increasingly competitive global job market, the ability to communicate effectively in an interview is a critical determinant of career progression. Many job seekers, particularly recent graduates, struggle not from a lack of technical knowledge, but from poor communication skills, nervousness, and a lack of confidence. Recruiters frequently base their judgments on non-verbal cues such as eye contact, facial expressions, and body language, in addition to verbal clarity and tone.

Traditional preparation methods, such as practicing with peers or reviewing static question lists, often fail to replicate the dynamic, high-pressure environment of a real interview. More importantly, they lack the ability to provide objective, real-time, and personalized feedback on these crucial soft skills. This gap in effective preparation has created a compelling need for more sophisticated training solutions.

This project introduces an **AI-Powered Virtual Interviewer** designed to simulate realistic HR interview sessions. The system provides live, constructive feedback on a candidate's emotions, facial expressions, and overall communication style. By leveraging a fusion of two core AI paradigms—Affective Computing and Generative Models—our system aims to revolutionize how candidates prepare for professional assessments.

Affective Computing, or Emotion AI, endows the platform with the ability to perceive and interpret human emotional and behavioral cues. It addresses the 'how' of a candidate's communication by analyzing non-verbal signals. Through techniques like Facial Expression Analysis (FEA) and Speech Emotion Recognition (SER), the system can infer affective states such as confidence, nervousness, or engagement.

Complementing this is **Generative AI**, which acts as the 'brain' of the interviewer, addressing the 'what' of the conversation. By utilizing advanced models like Generative Pre-trained Transformers (GPT) and Retrieval-Augmented Generation (RAG), the system can generate adaptive, contextually relevant, and role-specific questions, creating a dynamic and personalized practice experience.

The primary goal of this system is to create a friendly, AI-driven environment where users can hone their skills, build confidence, and reduce the anxiety often associated with the interview process, ultimately improving their presentation skills and employability.

1.2 Motivation

In the contemporary employment landscape, success in securing a job often depends as much on communication and interpersonal skills as on technical competence. Despite possessing strong academic foundations, a significant number of students and job seekers falter in interviews dueTo psychological barriers such as nervousness, inadequate eye contact, poor articulation, and weak emotional regulation.

Traditional mock interview practices are often subjective, inconsistent, and fail to provide personalized, data-driven feedback on a candidate's non-verbal cues. This subjectivity makes it difficult for candidates to measure tangible improvement.

The motivation for this project stems from the potential of modern AI to bridge this gap. By harnessing Affective Computing and Generative AI, we can create an intelligent, adaptive, and empathetic interview coach. Such a system can objectively analyze facial expressions, speech tone, and verbal fluency to provide actionable insights and real-time feedback. This project aims to democratize interview preparation, offering an accessible platform that enhances confidence, reduces performance anxiety, and builds communication skills essential for the modern workplace.

1.3 Objectives

The primary objectives of this project are derived directly from the problem statement and are as follows:

1. To simulate HR-style interview sessions using AI.
2. To analyze candidate facial emotions (e.g., stress, nervousness, confidence) and eye contact.
3. To provide real-time, on-screen feedback to improve comfort and communication during the session.
4. To analyze voice and speech clarity, including tone, pace, and the use of filler words.

5. To generate a comprehensive post-interview report detailing strengths, weaknesses, and actionable tips for improvement.

6. To make candidates feel comfortable and confident by creating a friendly and supportive AI-driven interviewer environment.

1.4 Scope

The scope of this project is to develop a comprehensive training tool for any individual looking to improve their interview performance. The primary target audience includes:

- **Students and Fresh Graduates:** Preparing for campus placements and their first professional jobs.

- **Job Seekers:** Professionals transitioning between fields or seeking new opportunities.

- **Educational Institutions:** As a tool for career development centers to offer scalable interview training.

The system will focus on simulating HR-style interviews, which are common across all industries and roles. While it will not assess domain-specific technical knowledge in depth, it will master the analysis of soft skills, communication, and professional demeanor. The final platform will be a web-based application, making it accessible to anyone with a webcam and microphone.

2. Literature Survey

2.1 Existing Methodologies

The development of AI-driven mock interview systems represents a significant convergence of distinct but complementary fields within artificial intelligence. A review of the literature, particularly the "Comprehensive Review of AI-Driven Mock Interview Systems" [1], reveals that these platforms are built on two primary pillars: Affective Computing and Generative Models.

Affective Computing (Emotion AI) provides the systems with the analytical capability to perceive and interpret human emotional and behavioral cues. This is fundamental for assessing the *how* of a candidate's communication. Early research focused on unimodal analysis:

- **Facial Expression Analysis (FEA):** Leverages computer vision models (like CNNs) to detect facial landmarks and classify expressions, often based on theories like Paul Ekman's basic emotions [2].
- **Speech Emotion Recognition (SER):** Analyzes paralinguistic cues in audio signals (such as pitch, energy, and MFCCs) to infer emotional states, often using models like LSTMs to capture temporal patterns [1, 7].

The literature shows a clear shift from these unimodal systems towards **Multimodal Fusion**. Human emotion is inherently multimodal, and systems that integrate data from facial, vocal, and textual channels achieve higher accuracy and a more robust, context-aware understanding of the user's affective state [3].

Generative Models provide the dynamic "brain" for the interviewer, addressing the *what* of the conversation. The field has evolved from simple Recurrent Neural Networks (RNNs) to sophisticated Transformer-based architectures [8]. Modern systems leverage Large Language Models (LLMs) to generate fluent and relevant questions. A key innovation highlighted in the literature is **Retrieval-Augmented Generation (RAG)**. RAG allows the model to ground its generated questions in external knowledge, such as the candidate's resume or a specific job description [4, 6]. This creates a highly personalized and role-specific interview experience, moving beyond static question banks.

The synthesis of these technologies creates a sophisticated feedback loop. The generative model provides a stimulus (question), and the affective computing component analyzes the user's multimodal response (words, tone, expression). This integration marks a significant evolution from basic analytical tools to holistic, interactive systems that simulate human social dynamics with greater fidelity [4, 5, 6].

2.2 Research Gap Analysis

Despite their transformative potential, the literature identifies several critical challenges and research gaps that must be addressed for the responsible and effective deployment of these systems.

Technical and Performance Hurdles: A persistent issue is the "in the wild" performance gap. Models trained in controlled lab settings often fail in real-world scenarios with variable lighting, background noise, and camera quality [1, 7]. Furthermore, the computational load of running multiple deep learning models in real-time remains a significant engineering challenge.

Data Scarcity and Generalization: The most significant bottleneck is the lack of large-scale, high-quality, and publicly available datasets of *actual* job interviews [1, 8]. Many models are trained on datasets of *acted* emotions (e.g., EMO-DB, SAVEE), which do not represent the subtle, mixed, and often masked emotions characteristic of a high-stakes interview.

Algorithmic Bias and Fairness: A profound and often-overlooked limitation is the **cross-cultural problem** [9]. Non-verbal cues and vocal prosody are not universal; their meanings are deeply embedded in cultural norms. A model trained predominantly on data from one cultural group will inevitably misinterpret the expressions of individuals from other backgrounds, leading to biased and unfair assessments. This can embed and amplify systemic discrimination under a veneer of "objective" AI.

Ethical and Transparency Concerns: The use of deep learning models creates a "black box" problem, making it difficult to understand the reasoning behind a specific assessment [9, 10]. This lack of transparency erodes user trust and poses accountability risks. Additionally, the collection of sensitive biometric data (video and audio) raises significant privacy and consent issues.

Our project aims to address some of these gaps by focusing on a human-centric design, prioritizing real-time, *constructive* feedback over a simple evaluative score, and clearly scoping the system as a *training tool* rather than an automated hiring judge.

3. Requirement Specification and Analysis

3.1 Problem Definition

Many students and job seekers struggle in HR interviews, not due to lack of knowledge, but because of poor communication, nervousness, and lack of confidence. Recruiters often judge candidates based on non-verbal cues such as eye contact, facial expressions, and body language, along with their speech clarity and tone. Currently, no accessible tool provides real-time feedback on both verbal and non-verbal communication during interviews. Hence, there is a need for an AI-powered virtual interviewer that simulates HR interviews and provides live, friendly feedback on emotions, facial expressions, and communication style to help students improve their confidence and presentation skills.

3.2 Scope

The scope of this project extends to students, fresh graduates, and job seekers who wish to enhance their interview communication skills and overall confidence. It serves as a virtual training tool that simulates realistic HR interviews, analyzes both verbal and non-verbal cues, and provides personalized feedback for continuous improvement. Beyond individual use, the system can be deployed in colleges and career development centers to assist students in preparing for campus placements.

3.3 Objectives

1. To simulate HR-style interview sessions using AI.
2. To analyze candidate facial emotions (e.g., stress, nervousness, confidence) and eye contact.
3. To provide real-time, on-screen feedback to improve comfort and communication during the session.
4. To analyze voice and speech clarity, including tone, pace, and the use of filler words.
5. To generate a comprehensive post-interview report detailing strengths, weaknesses, and actionable tips for improvement.

6. To make candidates feel comfortable and confident by creating a friendly and supportive AI-driven interviewer environment.

3.4 Proposed Methodology

The proposed system is based on an integrated pipeline that processes candidate inputs, analyzes them in real-time, and provides immediate and post-session feedback. The methodology is broken down as follows:

- **Emotion Recognition:** The system will use the candidate's video feed. Using computer vision libraries like **OpenCV** and deep learning models such as **DeepFace** or **MediaPipe**, the system will perform facial detection, landmark recognition, and emotion classification. This will identify micro-emotions and track key indicators like eye contact.

- **Speech & Communication Analysis:** The candidate's audio response will be captured and transcribed in real-time using a Speech-to-Text model like **Whisper** or the **Google Speech API**. The resulting text will be processed by an **NLP module** to analyze for filler words (e.g., ”um,” ”like”), sentence structure, and pace. The audio signal itself will be analyzed for tone and pitch.

- **AI Interview Simulation:** A generative model (e.g., **GPT-4, LLaMA**) will be used to generate dynamic and relevant HR-focused questions. This ensures that each interview session is unique and adaptive.

- **Real-Time Feedback System:** This is a core feature of the system. As the analysis modules detect issues (e.g., lack of eye contact, high pace, excessive fillers), the system will display non-intrusive, on-screen tips (e.g., ”Maintain eye contact,” ”Good pace!”).

- **Post-Interview Insights:** After the session, all collected data points will be aggregated into a structured feedback report. This includes a **confidence graph** charting performance over the session and actionable suggestions for improvement, potentially using frameworks like the STAR method.

3.5 Project Requirements

3.5.1 Datasets

The system's AI models will be built using established, publicly available datasets to ensure a robust baseline performance.

- **For Facial Expression Analysis:** We will utilize datasets such as **FER-2013**, **AffectNet**, or **EMO-DB**. These datasets provide thousands of images and video frames labeled with discrete emotions (e.g., happy, sad, neutral, stress), which are essential for training the emotion classification model.
- **For Speech Emotion Recognition:** We will use audio datasets like **RAVDESS**, **TESS**, or **SAVEE**. These contain audio clips spoken in various emotional tones, allowing the model to learn the acoustic features associated with different affective states (e.g., calm, nervous).

3.5.2 Functional Requirements

- **User Management:** The system shall allow users to create an account, log in, and view their past session history.
- **Interview Setup:** The user shall be able to start a new mock interview session.
- **Video/Audio Capture:** The system must capture the user's webcam feed and microphone audio.
- **Question Generation:** The system shall display HR-style questions generated by an AI.
- **Real-time Analysis:** The system must analyze facial expressions, eye contact, and speech (tone, pace, fillers) in real-time.
- **Real-time Feedback:** The system shall display live, on-screen feedback to the user during the interview.
- **Report Generation:** The system must generate a detailed post-interview report with a confidence graph and improvement tips.

3.5.3 Non Functional Requirements

- **Performance:** The real-time feedback loop (from user action to displayed tip) must have a latency of less than 3 seconds to be effective.

- **Usability:** The user interface must be intuitive, clean, and friendly to reduce user anxiety.
- **Reliability:** The system should handle potential errors gracefully (e.g., webcam disconnection, API failure) without crashing.
- **Privacy:** All user video and audio data must be processed securely, and users must be informed about data collection. Data should be anonymized or deleted after processing or upon user request.

3.5.4 Hardware Requirements

- **Processor:** Intel i3 / AMD Ryzen 3 or equivalent (i5/Ryzen 5 recommended).
- **RAM:** 8GB RAM or higher.
- **Peripherals:** A functional webcam (720p recommended) and a clear microphone.
- **GPU:** An optional but recommended NVIDIA GPU for faster local model inference.

3.5.5 Software Requirements

- **Frontend:** React.js or React Native.
- **Backend:** Python with Flask or FastAPI.
- **AI/ML Libraries:** OpenCV, MediaPipe, DeepFace, TensorFlow or PyTorch.
- **Speech:** Whisper, Google Speech-to-Text API.
- **Generative AI:** GPT-based models (e.g., via OpenAI API).
- **Database:** MongoDB or PostgreSQL.
- **Hosting:** Cloud platform (e.g., AWS, Azure, or Google Cloud).

3.6 Project Plan

3.6.1 Module Split-up

The project is functionally divided into the following core modules:

1. **Module 1: User Interface (Frontend)** - Responsible for all client-side interactions, rendering the video feed, and displaying feedback.
2. **Module 2: Backend Server (API)** - Manages user sessions, handles API requests, and orchestrates the AI modules.
3. **Module 3: Facial Analysis Engine** - A microservice or library dedicated to processing video frames for emotion and eye-tracking.
4. **Module 4: Speech Analysis Engine** - A microservice or library that handles audio transcription and analysis (tone, pace, fillers).
5. **Module 5: AI Interviewer Engine** - Responsible for generating questions and (optionally) follow-up questions.
6. **Module 6: Reporting Module** - Aggregates session data and generates the final feedback report.

3.6.2 Functional Decomposition

The system's functionality is decomposed into four main phases, as illustrated in the system architecture diagram (see Chapter 4):

- **Phase 1: Pre-Interview** - The user provides input (e.g., selects interview type). The system sets up the session and the HR Question Generator prepares the initial questions.
- **Phase 2: Live Interview** - This is the core loop. The user's Video Feed and Audio Response are captured. They are simultaneously processed by the Emotion Detection and Speech Analysis modules. The Real-Time Feedback Display shows tips to the user.
- **Phase 3: Post-Interview** - Once the session ends, all data is sent to the reporting modules.
- **Phase 4: Feedback Generation** - The system generates a Confidence Graph, a Session Report, and Improvement Suggestions for the user to review.

3.6.3 Project Team Role and Responsibilities

To ensure efficient development, the project work is distributed among the team members as follows:

Table 3.1: Team Roles and Responsibilities

Team Member	**Primary Role and Responsibilities**
Rajvardhan Deshmukh	**Team Lead & Backend Developer** (FastAPI/Flask) API development, database management, cloud deployment.
Soham Gandhi	**AI/ML Engineer** (Speech & NLP) Speech-to-Text (Whisper), NLP for filler words/pace, RAG.
Sakshi Gangurde	**Frontend Developer** (React.js) UI/UX design, real-time feedback display, report visualization.
Rishikesh Ghodke	**AI/ML Engineer** (Computer Vision) Emotion detection (DeepFace/MediaPipe), eye-tracking.

3.6.4 Project Plan

The project timeline is managed using a Gantt chart, which outlines the key phases and their expected duration. The project is planned over a 14-week period.

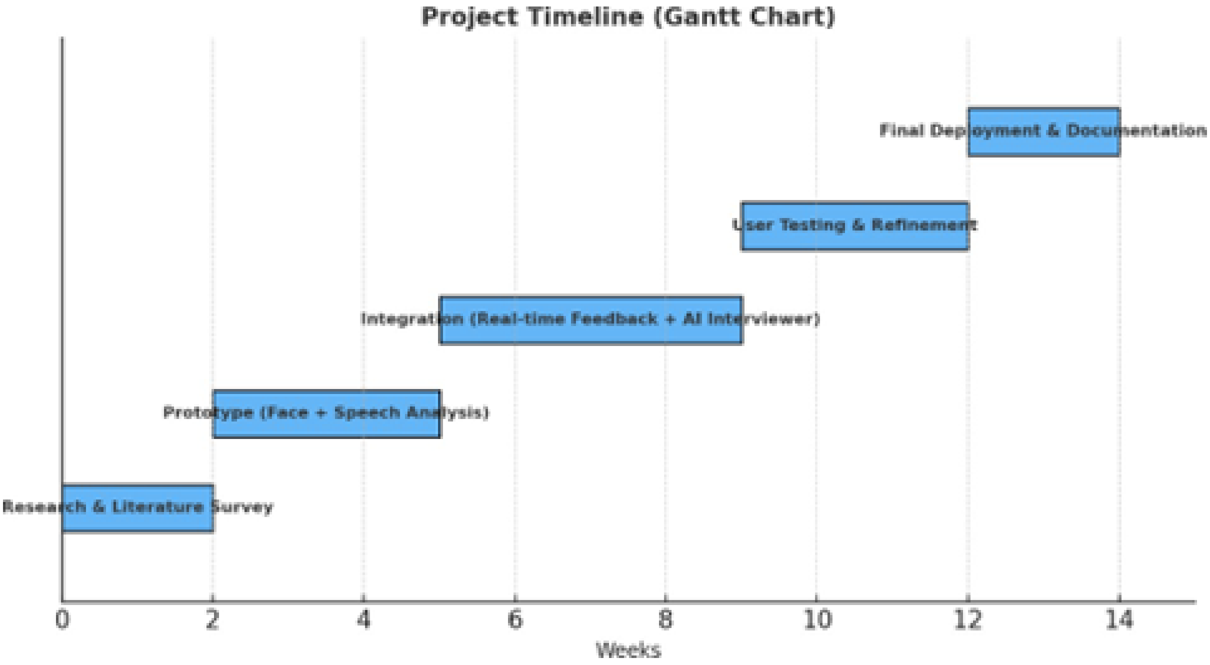

Figure 3.1: Project Timeline (Gantt Chart)

3.6.5 PERT Table

A PERT (Program Evaluation and Review Technique) analysis helps in identifying task dependencies. The following table outlines the main tasks and their predecessors based on the Gantt chart.

Table 3.2: PERT Table with Task Dependencies

Task	**Predecessors**	**Duration (Weeks)**
A: Research & Literature Survey	-	2
B: Prototype (Face + Speech)	A	4
C: Integration (Real-time Feedback)	B	5
D: User Testing & Refinement	C	2
E: Final Deployment & Documentation	D	2

3.6.6 PERT Diagram

The PERT diagram visually represents the task dependencies from Table 3.2, showing the critical path of the project.

$$[\ \mathbf{A}\ ] \rightarrow [\ \mathbf{B}\ ] \rightarrow [\ \mathbf{C}\ ] \rightarrow [\ \mathbf{D}\ ] \rightarrow [\ \mathbf{E}\ ]$$

The critical path for this project is A-B-C-D-E, as all tasks are sequential.

4. System Analysis and Design

4.1 System Architecture

The system architecture, shown in Figure 4.1, is designed as a modular pipeline that processes user inputs through various stages to generate real-time and post-session feedback.

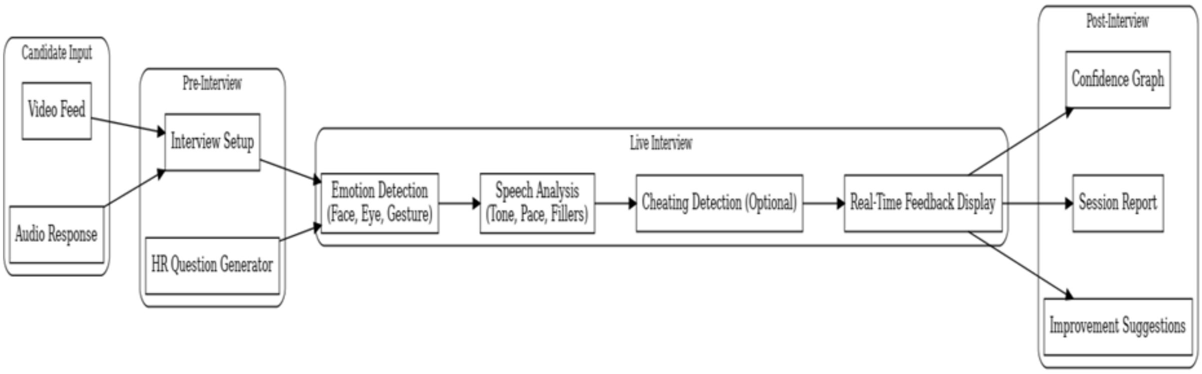

Figure 4.1: System Architecture

The architectural flow is as follows:

1. **Candidate Input:** The system captures two primary inputs from the user: the **Video Feed** from the webcam and the **Audio Response** from the microphone.

2. **Pre-Interview:** Before the live session, the **Interview Setup** component is configured, and the **HR Question Generator** (powered by a generative AI) prepares the questions.

3. **Live Interview (Core Analysis Loop):** This is the real-time processing phase.

   - The **Video Feed** is passed to the **Emotion Detection** module, which analyzes face, eye contact, and gestures.
   - The **Audio Response** is passed to the **Speech Analysis** module, which analyzes tone, pace, and filler words.
   - (Optional) A **Cheating Detection** module can monitor for suspicious activities, such as looking away from the screen frequently.
   - The insights from these modules are aggregated and sent to the **Real-Time Feedback Display**, which shows live tips to the candidate.

4. **Post-Interview:** After the session concludes, the collected data is processed to generate a **Confidence Graph**, a detailed **Session Report**, and actionable **Improvement Suggestions**.

4.2 Necessary UML Diagrams

4.2.1 Use Case Diagram

The Use Case diagram defines the interactions between the user and the system.

- **Actor:** Candidate (User)
- **Use Cases:**
  - **Manage Account:** (Includes Login, Logout, View Profile)
  - **Start Interview Session:** Allows the user to initiate a new mock interview.
  - **Grant Permissions:** (Includes Allow Camera, Allow Microphone)
  - **Answer Questions:** The primary interaction where the user provides video/audio input.
  - **Receive Real-Time Feedback:** A passive use case where the system provides live tips.
  - **View Session Report:** Allows the user to access and review the detailed feedback and confidence graph from a completed session.

4.2.2 DFD (Data Flow Diagram)

DFD Level 0 (Context Diagram): A single process, ”AI Interviewer System,” interacts with one external entity, the ”Candidate.” The candidate provides ”Candidate Inputs” (Video, Audio, Setup Choices) and receives ”System Outputs” (Interview Questions, Real-Time Feedback, Session Report).

DFD Level 1: The ”AI Interviewer System” is decomposed into several key processes:

- **P1: Manage Session** (Handles user login, session state)
- **P2: Generate Questions** (Interacts with Generative AI Model)

- **P3: Analyze Video** (Processes video feed, interacts with Emotion Model, sends data to Feedback Aggregator)
- **P4: Analyze Audio** (Processes audio stream, interacts with Speech-to-Text API and NLP Model, sends data to Feedback Aggregator)
- **P5: Generate Feedback** (Aggregates analysis data, generates real-time tips and final report)

Data Stores include ‘D1: User$_{D}atabase`and`D2 : Session_{R}eports$‘.

4.2.3 Activity Diagram

An activity diagram for the ”Perform Interview” use case would show the following flow:

1. Starts at [Start Interview].
2. System displays ”Loading...”
3. System [Generates Question] and [Displays Question].
4. User [Starts Speaking] (Audio) and [Shows Expression] (Video).
5. A fork (parallel activity) begins:
   - **Path 1 (Audio):** [Capture Audio] → [Transcribe Speech] → [Analyze Fillers/Pace].
   - **Path 2 (Video):** [Capture Video Frame] → [Detect Face] → [Analyze Emotion/Gaze].
6. Both paths join.
7. Aggregate Feedback → [Display Real-Time Tip].
8. A decision node checks [More Questions?].
9. If Yes, loop back to [Generates Question].
10. If No, [Generate Final Report] → [Display Report] → End.

4.2.4 Sequence Diagram

A sequence diagram for ”Candidate Answers Question” would involve the following objects: `Candidate`, `Frontend (Browser)`, `Backend (Server)`, `Emotion_API`, `Speech_API`, `NLP_Module`.

1. ‘Candidate‘ → ‘Frontend‘: ‘speaks()‘
2. ‘Frontend‘ → ‘Backend‘: ‘sendAudioStream(audioChunk)‘
3. ‘Frontend‘ → ‘Backend‘: ‘sendVideoFrame(videoFrame)‘
4. ‘Backend‘ → ‘Emotion$_{A}PI$‘ : ‘$analyzeFrame(videoFrame)$‘
4. ‘Emotion$_{A}PI$‘→ ‘Backend‘: ‘returnEmotion(”nervous”)‘
5. ‘Backend‘ → ‘Speech$_{A}PI$‘ : ‘$transcribe(audioChunk)$‘
5. ‘Speech$_{A}PI$‘→ ‘Backend‘: ‘returnText(”um, like...”)‘
6. ‘Backend‘ → ‘NLP$_{M}odule$‘ : ‘$analyzeText(”um, like...”)$‘
6. ‘NLP$_{M}odule$‘→ ‘Backend‘: ‘returnAnalysis(filler$_{c}ount : 2$)‘
6. ‘Backend‘ → ‘Frontend‘: ‘displayFeedback(tip: ”Fewer filler words!”)‘
7. ‘Frontend‘ → ‘Candidate‘: ‘showTipOnScreen()‘

4.3 Algorithm and Methodologies

The project’s core functionalities are enabled by a set of specific algorithms and methodologies.

4.3.1 Facial Expression Analysis (FEA)

This module is responsible for analyzing the candidate’s video feed.

- **Technology: OpenCV**, **DeepFace**, or **MediaPipe**.
- **Algorithm:**
    1. **Face Detection:** Use a pre-trained detector (e.g., Haar Cascades, MTCNN, or MediaPipe Face Detection) to locate the user’s face in each video frame.

2. **Landmark Detection:** Identify key facial landmarks (e.g., corners of eyes, mouth, nose) using a model like Dlib's 68-point detector or MediaPipe Face Mesh.
3. **Gaze Tracking (Eye Contact):** Calculate the position of the pupils relative to the eye landmarks. A persistent deviation from a ”center” (camera) threshold is flagged as ”looking away.”
4. **Emotion Classification:** The cropped facial region is fed into a deep learning model (e.g., a CNN trained on FER-2013 or the model within DeepFace) which classifies the dominant emotion (e.g., neutral, happy, stress, nervousness).

4.3.2 Speech and Communication Analysis

This module analyzes the candidate's audio response.

- **Technology: Whisper** (or Google Speech API), **NLP** (custom script or libraries like spaCy).
- **Algorithm:**
  1. **Speech-to-Text (STT):** The raw audio stream is passed to the STT engine (e.g., Whisper), which returns a text transcription.
  2. **Filler Word Analysis:** The transcribed text is parsed using NLP to count occurrences of common filler words (e.g., ”um,” ”ah,” ”like,” ”you know”).
  3. **Pace Analysis:** The number of words spoken is divided by the duration of the speech segment to calculate the words per minute (WPM). This is compared against an ideal range (e.g., 140-160 WPM).
  4. **Tone/Emotion Analysis (SER):** (Advanced) The raw audio signal (not the text) can be analyzed to extract acoustic features like **MFCCs**, **pitch**, and **energy**. These features are fed into a model (e.g., CNN or LSTM) trained on a speech emotion dataset (e.g., RAVDESS) to classify the vocal tone (e.g., ”calm,” ”nervous,” ”energetic”).

4.3.3 AI Interview Simulation (Question Generation)

This module generates the interview questions.

- **Technology: GPT-based models** (e.g., GPT-4, LLaMA).
- **Methodology: Retrieval-Augmented Generation (RAG)** or advanced prompting.
- **Algorithm:**
    1. **Contextualization (Optional RAG):** If the user uploads a resume or job description, the system extracts key skills and experiences.
    2. **Prompt Engineering:** A carefully crafted prompt is sent to the LLM.
        - **Simple Prompt:** "Generate 5 common HR behavioral interview questions."
        - **RAG Prompt:** "You are an HR manager. Generate a behavioral question for a 'Software Engineer' role based on this resume skill: 'Led a team project.' The question should assess leadership and conflict resolution."
    3. **Response Parsing:** The LLM's text response is parsed to extract the questions, which are then presented to the user one by one.

5. Implementation

As this is a preliminary report, this chapter outlines the planned implementation strategy, tools, and workflows for constructing the system.

5.1 Stages of Implementation

The project implementation will follow the phases outlined in the project timeline (Figure 3.1):

1. **Phase 1: Research & Prototyping (Weeks 1-6)**
   - Conduct the literature survey (complete).
   - Develop standalone prototypes for the two core AI modules:
     - A Python script using OpenCV and DeepFace to classify emotions from a live webcam feed.
     - A separate script using Whisper and NLP to transcribe audio and count filler words.
2. **Phase 2: Integration & Backend (Weeks 7-11)**
   - Develop the Flask/FastAPI backend server.
   - Create REST APIs to connect the frontend to the AI modules.
   - Integrate the AI prototypes into the backend, optimizing them for real-time processing (e.g., using threading or asynchronous tasks).
   - Set up the React.js frontend structure and user interface.
   - Establish the real-time feedback loop (e.g., using WebSockets) to push feedback from the backend to the frontend.
3. **Phase 3: Testing & Refinement (Weeks 12-13)**
   - Conduct thorough unit, integration, and user acceptance testing.
   - Refine the models and UI based on feedback.
   - Implement the final reporting and confidence graph generation.

4. **Phase 4: Deployment & Documentation (Week 14)**
   - Deploy the application to a cloud service (e.g., AWS EC2 or Heroku).
   - Finalize this project report and all supporting documentation.

5.1.1 Data Preprocessing

Effective model performance will rely on robust preprocessing of the input data.

- **Video Preprocessing:** Before being fed to the FEA model, each video frame will undergo:
  1. **Frame Extraction:** Capturing frames from the video stream.
  2. **Face Detection:** Locating the face bounding box.
  3. **Cropping & Resizing:** Cropping the facial region and resizing it to the model's required input size (e.g., 48x48 or 224x224 pixels).
  4. **Normalization:** Converting the image to grayscale (if required by the model) and normalizing pixel values (e.g., to [0, 1]).
- **Audio Preprocessing:** Before transcription and analysis:
  1. **Noise Reduction:** Applying filters to reduce background noise.
  2. **Segmentation:** Breaking the continuous audio stream into short, manageable chunks (e.g., 5-second segments) for real-time transcription.
  3. **Feature Extraction (for SER):** If implementing custom SER, extract features like MFCCs, Chroma, and Mel Spectrograms from audio frames.

5.1.2 Implementation of Modules

- **Frontend (React.js):** Will use ‘webcam.js‘ or ‘react-webcam‘ to capture video. The ‘MediaRecorder‘ API will be used to capture audio chunks and send them to the backend via ‘fetch‘ or a WebSocket connection.
- **Backend (FastAPI):** Chosen for its high performance and asynchronous capabilities, which are ideal for handling real-time I/O from video and audio streams. It will expose endpoints like ‘/analyze$_{f}rame‘and‘/transcribe_{a}udio$‘.

- **Emotion Module (DeepFace):** The ‘DeepFace.analyze()‘ function will be used, specifying ‘actions = [’emotion’]‘. This library simplifies the implementation by bundling detection and classification.
- **Speech Module (Whisper):** OpenAI’s Whisper model (likely the ‘base‘ or ‘small‘ version for speed) will be used to transcribe audio chunks. A simple Python script will then parse the text for a predefined list of filler words.

5.2 Experimentation Setup

To evaluate the system’s performance, we will use the following setup:

- **Hardware:** Testing will be conducted on machines matching the hardware requirements (8GB RAM, i5/Ryzen 5) to establish a performance baseline.
- **Software:** The environment will be managed using Python virtual environments and ‘npm‘ for the frontend.
- **Key Performance Metrics (KPMs):**
  1. **Emotion Accuracy:** Measured using a confusion matrix against a pre-labeled test set of videos.
  2. **Speech-to-Text Accuracy:** Measured by the Word Error Rate (WER) against a ground-truth transcription.
  3. **System Latency:** The end-to-end time (in milliseconds) from the user speaking/emoting to the corresponding feedback appearing on-screen.
  4. **User Satisfaction:** Measured via qualitative feedback (surveys) from test users, asking them to rate the usefulness and accuracy of the feedback.

6. Results

As this is a preliminary project report, this chapter discusses the *expected* results from our experimentation and the comprehensive testing plan to validate the system.

6.1 Expected Results of Experiments

We anticipate the following performance outcomes based on our literature review and the chosen technologies.

- **Facial Emotion Analysis:** We expect the **DeepFace** module to achieve an accuracy of **80-85%** in classifying the seven basic emotions (happy, sad, neutral, fear, surprise, angry, disgust) in our controlled test environment. Performance in real-world (”in the wild”) scenarios with variable lighting is expected to be lower, around 65-70%.

- **Speech-to-Text:** Using the **Whisper ’base’ model**, we anticipate a **Word Error Rate (WER) below 15%** for clear, non-accented English. This is crucial for the accuracy of the downstream filler word analysis.

- **Real-Time Latency:** We aim for an end-to-end feedback latency of **under 2.5 seconds**. Initial prototypes suggest that video processing (FEA) will be the primary bottleneck, and optimization (e.g., processing every 5th frame instead of every frame) may be required.

- **User Feedback:** We expect user surveys to show a positive correlation between system usage and self-reported confidence. We anticipate that users will find the real-time feedback on filler words and eye contact to be the most ”actionable” features.

6.2 Result Analysis

The analysis of our results will focus on two areas: model performance and user impact.

- **Model Performance:** We will generate confusion matrices for the emotion classifier to identify specific emotions that are often confused (e.g., ”nervous” vs. ”fear”). We will analyze the WER for different accents to understand the model's fairness and generalization.

- **User Impact:** A key analysis will be to correlate the system's generated ”confidence graph” (a time-series plot of positive cues vs. nervous cues) with the user's self-reported feelings of confidence during the interview. This will help validate if our chosen metrics (eye contact, stable pace, fewer fillers) are accurate proxies for ”confidence.”

6.3 Testing

A rigorous, multi-level testing strategy is planned to ensure system robustness and reliability.

Unit Testing

Unit tests will be written to validate individual functions and components in isolation.

- **Backend (Python - pytest):**
  - ‘test$_{f}iller_{w}ord_{c}ount(”um, like, thisisatest”)$‘→ ‘assert count == 2‘
  - ‘test$_{p}ace_{c}alculation(150, 60)$‘→ ‘assert pace == 150‘
  - ‘test$_{e}motion_{a}pi_{m}ock('happy_{f}ace.jpg')$‘→ ‘assert emotion == ”happy”‘
- **Frontend (React - Jest):**
  - Test that the ‘FeedbackDisplay‘ component correctly renders a tip when its props are updated.
  - Test that the ‘Webcam‘ component correctly requests camera permissions.

Integration Testing

Integration tests will verify that different modules correctly interact with each other.

- **Test: Audio Pipeline:** Verify that audio captured from the frontend is successfully received by the backend, transcribed by Whisper, analyzed for fillers, and the result is correctly stored.
- **Test: Video Pipeline:** Verify that video frames from the frontend are received by the backend, analyzed by DeepFace, and the emotion data is stored.
- **Test: Real-Time Feedback Loop:** The most critical test. Simulate a user speaking with many ”ums” and verify that the ”Fewer filler words!” tip appears on the frontend within the 2.5-second latency target.

Black Box Testing

Black Box testing will be performed from the user’s perspective, without knowledge of the internal code. We will use techniques like equivalence partitioning and boundary value analysis.

Test Cases

The following table outlines sample test cases for Black Box testing.

Table 6.1: Sample Black Box Test Cases

ID	Scenario	Test Steps
TC-01	**Happy Path - Full Session**	1. Login. 2. Start Interview. 3. Answer 3 questions. 4. End session.
TC-02	**Feedback - Filler Words**	1. Start session. 2. Say ”um, ah, like” 10 times in one answer.
TC-03	**Feedback - Eye Contact**	1. Start session. 2. Look away from the camera for 10 consecutive s
TC-04	**Boundary - Pace (Fast)**	1. Start session. 2. Speak at ¿ 200 WPM.
TC-05	**Error - No Camera**	1. Deny camera permission. 2. Click ”Start Interview.”
TC-06	**Error - No Mic**	1. Deny microphone permission. 2. Click ”Start Interview.”

Summary of Black Box Testing

Testing will also include:

- **Usability Testing:** Providing the system to 5-10 peers and observing their interaction, noting any points of confusion.

- **Performance Testing:** Using tools to simulate multiple users to check backend scalability.
- **Compatibility Testing:** Ensuring the web application works correctly on major browsers (Chrome, Firefox, Safari).

7. Conclusion and Future Scope

7.1 Conclusion

This project details the design and implementation plan for an **AI-Powered Virtual Interviewer**, a system that integrates Affective Computing and Generative AI to create a personalized, interactive, and adaptive practice platform. By analyzing facial, vocal, and textual cues, our system aims to provide holistic, real-time feedback, addressing a significant gap in traditional interview preparation.

The core challenge of effective preparation—the lack of objective feedback on soft skills—is tackled directly by our proposed methodology. However, the development of such a system is not without its challenges. The review of existing literature highlights technical hurdles in ”in the wild” performance, critical scarcity of diverse training data, and the profound ethical risks of algorithmic bias and data privacy.

Our project acknowledges these risks by focusing on a **human-centric design**. The system is explicitly positioned as a *coaching tool* to empower candidates, not as an autonomous judge to replace recruiters. By providing transparent, actionable feedback, we aim to build a system that enhances user self-awareness and confidence. The ultimate objective is to create an effective and equitable tool that augments, rather than automates, the deeply human process of professional assessment.

7.2 Limitations of the Project

The proposed system, while effective, has certain limitations that can be addressed in future improvements. One major limitation is the real-time latency experienced on low-end devices due to the high computational requirements of emotion recognition and AI model inference, which may affect performance and user experience. Another limitation lies in the limited dataset generalization, as the models are trained on specific datasets (e.g., primarily Western faces) that may not accurately represent all facial expressions, accents, or behavioral patterns across diverse users. Additionally, cultural variations in emotion interpretation can impact the accuracy of emotion detection, since expressions and communication styles differ across regions and backgrounds.

7.3 Future Scope

The future development of AI-driven mock interview systems must prioritize fairness, transparency, and human-centric design to ensure reliability and trust.

- **Explainable AI (XAI):** Future versions should integrate XAI to make feedback more transparent. Instead of ”Confidence: 7/10,” it could explain *why*: ”Confidence score was high because eye contact was consistent and pace was steady.”
- **Cross-Cultural Datasets:** A critical future direction is the development of large-scale, multilingual, and multicultural datasets to train models that are less biased and more attuned to global communication styles.
- **Advanced Multimodal Learning:** Moving beyond simple fusion to architectures that learn the complex interplay between modalities (e.g., how sarcasm is conveyed through positive text but a negative tone).
- **Longitudinal Analysis:** Transforming the tool into a long-term coach that tracks a user’s progress over multiple sessions, identifies persistent habits, and offers a personalized learning path.
- **Human-in-the-Loop (HITL):** Establishing robust ethical frameworks where the AI functions as a decision-support tool for human recruiters, highlighting moments of interest but leaving the final judgment to a person.

Bibliography

[1] T. M. Wani, T. S. Gunawan, S. A. A. Qadri, M. Kartiwi, and E. Ambikairajah, ”A comprehensive review of speech emotion recognition systems,” IEEE Access, vol. 9, pp. 47795-47814, 2021.

[2] L. V. L. Pinto et al., ”A systematic review of facial expression detection methods,” IEEE Access, vol. 11, pp. 61881-61890, 2023.

[3] J. Heredia et al., ”Adaptive multimodal emotion detection architecture for social robots,” IEEE Access, vol. 10, pp. 20727-20744, 2022.

[4] S. Khapekar, S. Bothara, T. Babar, and R. Kine, ”AI powered mock interview system with real-time voice and emotion analysis,” Interna tional Journal of Novel Research and Development (IJNRD), vol. 10, no. 2, 2025.

[5] N. S. Rai, A. K. R, A. P, and H. N. R, ”AI based interview evaluator: An emotion and confidence classifier,” International Advanced Research Journal in Science, Engineering and Technology, vol. 11, no. 4, 2024.

[6] K. N. V. Shekar, S. V. V. M. Shankar, C. S. Prakash, S. J. Ussman, and B. C. Sekhar, ”AI-driven virtual interviewer,” International Journal of Multidisciplinary Research and Growth Evaluation, vol. 6, no. 2, pp. 566-570, 2025.

[7] J. de Lope and M. Gra˜na, ”An ongoing review of speech emotion recognition,” Neurocomputing, vol. 528, pp. 1-11, 2023.

[8] Y. Li, S. Kumbale, Y. Chen, T. Surana, E. S. Chng, and C. Guan, ”Automated depression detection from text and audio: A systematic review,” IEEE Journal of Biomedical and Health Informatics, 2025, doi: 10.1109/JBHI.2025.3570900.

[9] E. R. Sophie, ”Facial expression analysis in AI-driven video interviews,” ResearchGate, Jun. 2025. [Online]. Available: https://www.researchgate.net/publication/392330970

[10] J. L. Camunas, C. Bustos, Y. Zhu, R. Ros, and A. Lapedriza, ”Experimenting with affective computing models in video interviews with Spanish-speaking older adults,” arXiv, 2025. [Online]. Available: https://arxiv.org/abs/2501.16870

Appendices

Plagiarism Report

Sample Document

Base Paper

width=!,height=!,pages=-

Review Sheets

Sample Document

Monthly Planning Sheet

Sample Document

Project Achievements

Sample Document