VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI – 590018

PROJECT REPORT ON

"AI BASED INTERVIEW EVALUATOR: AN EMOTION AND CONFIDENCE CLASSIFIER"

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

PROJECT PHASE –II [18CSP83]

Submitted By

ABHIRAM K R	**4JK20IS002**
ADITHYA P	**4JK20IS005**
HRITHIK N R	**4JK20IS020**

Under the guidance of

Prof. Navya S Rai
Assistant Professor
Department of Information Science & Engineering

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
A.J. INSTITUTE OF ENGINEERING & TECHNOLOGY
NH-66, KOTTARA CHOWKI, MANGALURU – 575006

Accredited By NBA (BE: CV, CSE, ECE, ISE & ME)

2023 – 2024

A. J. INSTITUTE OF ENGINEERING & TECHNOLOGY

NH – 66, Kottara Chowki, Mangaluru - 575006

A Unit of Laxmi Memorial Education Trust (R)

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)

Accredited By NBA (BE: CV, CSE, ECE, ISE & ME)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"AI BASED INTERVIEW EVALUATOR: AN EMOTION AND CONFIDENCE CLASSIFIER"** carried out by Mr. **ABHIRAM K R**, USN: **4JK20IS002**, Mr. **ADITHYA P**, USN: **4JK20IS005** and Mr. **HRITHIK N R**, USN: **4JK20IS020**, as bonafide students of A.J. Institute of Engineering & Technology, Mangaluru, in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science and Engineering of Visvesvaraya Technological University, Belgavi** during the year 2023-2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Prof. Navya S Rai	**Prof . R Sahaya Shamini**	**Dr Suresha D**
Project Guide	**Project Coordinator**	**Head of the Department**

Dr Shantharama Rai C

Principal

Examiners | **Signature with Date**

1.

2.

ACKNOWLEDGEMENT

First and foremost, we thank our parents for what we are and where we are today, without whose hard work and sacrifice we would not be here today.

We deem it a privilege to place on record the deep sense of gratitude to our Project Guide **Prof. Navya S Rai, Assistant Professor,** Department of Information Science and Engineering, and Project coordinator **Prof. R Sahaya Shamini, Assistant Professor,** Department of Information Science and Engineering, who always stood behind us and supported in each step of the project work.

We are grateful to **Dr. Suresha D, Head of the Department,** Information Science and Engineering for his support and encouragement.

We are indebted to our respected Dean Academics **Dr P Mahabaleshwarappa,** Principal **Dr. Shantharama Rai C,** beloved Vice President **Mr. Prashanth Shetty** and the management of **A. J. Institute of Engineering and Technology, Mangaluru** for providing all the facilities that helped us in timely completion of this project report.

Finally, we would like to thank all the teaching and non-teaching staff of Department of Information Science and Engineering for their valuable help and support.

Abhiram K R - 4JK20IS002
Adithya P - 4JK20IS005
Hrithik N R - 4JK20IS020

ABSTRACT

Traditional interview processes suffer from inherent human biases and subjectivity, leading to inconsistent candidate evaluations and potentially flawed hiring decisions. This project presents an AI-based interview evaluator system that aims to revolutionize the recruitment process by providing an objective, data-driven, and comprehensive assessment of candidates during job interviews. The system leverages cutting-edge technologies in machine learning, computer vision, speech recognition, and natural language processing to analyze video and audio inputs from candidates, enabling thorough evaluations of their emotions, confidence levels, and knowledge.

At the core of the system lies a multi-modal approach that integrates facial emotion recognition using deep learning models like Deepface, speech-to-text conversion through APIs like Google Speech Recognition, and audio processing techniques such as Mel-Frequency Cepstral Coefficients (MFCCs) extraction. The facial analysis module classifies candidates' emotions into seven categories, while the speech analysis component evaluates their responses' similarity to expected answers and predicts confidence levels using machine learning classifiers trained on confident and non-confident datasets.Furthermore, the system incorporates a neural network-based chatbot trained on JSON intents, providing an interactive and engaging conversational experience for candidates. The results and insights generated by the system, including emotions, confidence levels, similarity scores, and chatbot interactions, are stored in a MongoDB database, enabling efficient data storage and retrieval for further analysis and reporting.The AI-based interview evaluator system offers a scalable and efficient solution for organizations, enabling consistent and unbiased evaluations of a large number of candidates. By eliminating subjective assessments and providing data-driven insights, the system empowers organizations to make more informed hiring decisions, ultimately leading to the acquisition of top talent and improved organizational performance.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

An innovative project at the nexus of cutting-edge technology and human skill development is the AI Interview Evaluator: an Emotion and Confidence Classifier, which stands out in the quickly changing field of artificial intelligence and its many applications. The project's clear goal is to redefine interview preparation techniques by utilizing cutting-edge AI capabilities to offer an extensive and dynamic training environment. Essentially, the AI Interview Evaluator uses state-of-the-art methods from affective computing to perform the combined roles of interviewer and evaluator. This software goes above and beyond traditional interviewing practices with its combination of speech modulation assessment and facial expression analysis. This project is in line with the current movement to use artificial intelligence to mimic real-world situations, Providing interviewees with an exclusive and indispensable resource. The creation of models that can identify human emotions from speech patterns and facial expressions is an intriguing field of research in this field. Deep learning techniques are used in this multidisciplinary discipline to develop reliable and effective emotion detection systems. There are many uses for emotion detection; earlier techniques frequently depended on feature engineering and explicit rules, but deep learning has completely changed the field. Deep neural networks are perfect for capturing complicated patterns in speech signals and facial expressions because they can automatically learn hierarchical representations from raw input.

The AI Interview Evaluator project capitalizes on the latest advancements in affective computing, computer vision, and speech recognition to create a comprehensive and immersive interviewing experience. By integrating state-of-the-art models for emotion recognition, speech analysis, and confidence evaluation, the system provides candidates with a simulated interview environment that closely mimics real-world scenarios.One of the key components of the project is the facial emotion recognition module, which employs deep learning techniques to analyze the candidate's facial expressions during the interview. This module leverages advanced computer vision algorithms and pre-trained models, such as Deepface and HaarCascade classifiers, to accurately classify facial expressions into seven categories: neutral, happy, sad, surprise, angry, disgust, and fear. By capturing and interpreting these subtle emotional cues, the system can provide valuable insights into the candidate's emotional state and potential areas of improvement. In addition to facial analysis, the AI Interview Evaluator incorporates a sophisticated speech processing module that converts the candidate's spoken responses into text using speech-to-text technologies like Google Speech Recognition or SpeechRecognizer libraries. This text is then analyzed for similarity to expected answers, enabling the system to evaluate the candidate's knowledge and understanding of the subject matter. Moreover, the project employs advanced audio processing techniques, such as feature extraction using Mel-Frequency Cepstral Coefficients (MFCCs) and machine learning models like Random Forest Classifiers, to predict the candidate's confidence level during the interview. By analyzing the audio signals and extracting relevant features, the system can provide feedback on the candidate's confidence, which is a crucial aspect of successful interviews.

1.1 Motivation

The traditional job interview process has long been plagued by inherent biases and subjectivity, leading to inconsistent and potentially flawed hiring decisions. Human evaluators, despite their best efforts, are susceptible to unconscious biases influenced by factors such as appearance, gender, cultural background, or personal preferences. These biases can result in overlooking qualified candidates or selecting unsuitable ones, ultimately hindering an organization's ability to acquire the best talent. Moreover, manual evaluation of candidates during interviews can be time-consuming, resource-intensive, and prone to human error, particularly when dealing with a large volume of applicants. The lack of standardized and objective evaluation criteria can lead to inconsistencies in assessing candidates, making it challenging to compare their performances fairly.The AI-based interview evaluator project aims to address these challenges by leveraging the power of artificial intelligence, machine learning, and data-driven approaches.

1.2 Project Modules

- Flask: A lightweight Python web framework used for building the backend of the system and handling HTTP requests and responses.
- OpenCV: A computer vision library used for face detection and video processing tasks, specifically for integrating the HaarCascade classifier.
- Deepface: A deep learning-based facial analysis framework used for facial emotion recognition and classifying facial expressions into seven categories.
- speech_recognition: A library for performing speech recognition, with support for several engines, including Google Speech Recognition, used for converting audio input into text.
- sounddevice: A cross-platform audio I/O library used for recording audio input from the candidate.

- soundfile: A library for reading and writing audio files, used in conjunction with sounddevice to save the recorded audio.
- librosa: A Python library for audio and music analysis, used for feature extraction from audio data, specifically Mel-Frequency Cepstral Coefficients (MFCCs).
- scikit-learn: A machine learning library for Python, used for implementing the Random Forest Classifier for confidence level prediction and other tasks such as cross-validation and hyperparameter tuning.
- Keras: A high-level neural networks API used for building and training the neural network-based chatbot.
- TensorFlow: A powerful open-source library for numerical computation and machine learning, used as the backend for the Keras-based chatbot.
- NLTK (Natural Language Toolkit): A suite of libraries and programs for working with human language data, used for text preprocessing tasks in the chatbot.
- difflib: A library in Python's standard library used for computing the similarity between sequences, specifically using the SequenceMatcher class for similarity evaluation between the candidate's responses and expected answers.
- MongoDB: A NoSQL database used for storing interview results, including emotions, confidence levels, similarity scores, and chatbot interactions.
- Flask-Session: An extension for Flask that adds server-side session support, used for managing user sessions in the web application.
- NumPy: A fundamental package for scientific computing with Python, used for numerical operations and data manipulation.
- Matplotlib: A plotting library for creating static, animated, and interactive visualizations in Python, used for generating visualizations of audio waveforms and confidence level graphs.
- Tkinter: A standard Python interface to the Tk GUI toolkit, used for building the graphical user interface of the application.
- threading: A module in Python's standard library for creating and managing threads, used for running multiple tasks concurrently, such as video processing, audio recording, and analysis.

1.3 Advantages

- Objective and Unbiased Evaluation: The system eliminates inherent human biases by relying on data-driven analysis of video and audio inputs, providing a fair and consistent evaluation process. By leveraging machine learning and AI techniques, the system can objectively assess candidates without being influenced by factors such as appearance, gender, cultural background, or personal preferences.

- Comprehensive Assessment: The system evaluates multiple aspects of a candidate's performance during the interview, including emotions, confidence levels, knowledge, and communication skills. This holistic approach provides a more comprehensive understanding of a candidate's capabilities and potential fit for the role, compared to traditional interview processes that may focus on a limited set of criteria.
- Scalability and Efficiency: The automated nature of the system allows for efficient evaluation of a large number of candidates, reducing the time and resources required compared to traditional manual evaluation processes. This scalability can be particularly beneficial for organizations with high recruitment volumes, enabling them to streamline their hiring processes and make faster decisions.

1.4 Disadvantages

- Data Quality and Bias in Training Data: The accuracy and fairness of the system's assessments heavily depend on the quality and representativeness of the training data used for emotion recognition, confidence evaluation, and similarity matching. If the training data itself is biased or not representative of the diverse candidate pool, the system may perpetuate or amplify those biases, leading to unfair or inaccurate evaluations.
- Lack of Human Interaction and Context: While the system provides objective evaluations, it may lack the nuanced understanding and interpersonal aspects of traditional interviews. Factors such as body language, tone, and cultural context can be challenging for AI systems to fully capture and interpret. This lack of human interaction may result in missing important contextual information that could influence the overall assessment of a candidate.
- Ethical Concerns: The use of AI in hiring processes raises ethical concerns regarding privacy, bias, transparency, and accountability. There may be concerns about the collection and use of personal data (video and audio recordings), the potential for algorithmic biases, and the lack of transparency in how the system arrives at its evaluations.

1.5 Organization of the Report

The organization of a report refers to the structure and layout of its contents. It involves arranging information in a logical and coherent manner to facilitate understanding and readability.

- Chapter 1 of a report serves as an introduction, offering an overview of the report's purpose, scope, and background. It outlines the current situation or challenges being addressed and summarizes the main findings.
- Chapter 2 concentrates on survey papers that were referenced for the project, analysing them to gain insights into the research problem. Properly referencing these papers is crucial for readers to understand the project's basis and the literature informing it.
- Chapter 3 outlines the models and the specific software requirements, such as operating system compatibility, required software libraries and frameworks, and programming languages used to build the system.
- In Chapter 4, you'll find the problem statement that defines the issue being addressed, the project objectives outlining specific goals, and the expected outcomes, which anticipate the results of achieving those objectives.
- Chapter 5 presents the system design and architectural design of the project, detailing how the system is structured and how its components interact. It also includes flowcharts that visually depict the flow of processes within the system.
- Chapter 6 consists of the conclusion, summarizing the findings and insights from the project, and future enhancements, outlining potential improvements or developments for the project.

CHAPTER 2

LITERATURE SURVEY

This chapter focus on the survey papers that were referred in order to carry out this project. These survey papers were reviewed and analysed to gain insights and understanding related to the research problem. Properly referencing the survey papers used is essential for the reader to understand the basis of the project and the relevant literature used to inform it.

Rohan Patil, Akash Butte , Sahil Temgire ,Varun Nanekar [1], proposed “Real Time Mock Interview using Deep Learning” research , which entails creating a deep learning-powered real-time mock interview system to help individuals improve their interviewing techniques. Acknowledging the current trend toward virtual interviews, the system offers users access to a web application that allows them to rehearse and mimic virtual interviews. Convolutional neural networks (CNNs) are used for facial expression recognition, speech-to-text conversion is used for grammar checking, and deep learning techniques are integrated for detailed feedback on a variety of aspects, such as facial expressions, head nodding, reaction time, speaking rate, and volume. Through graphical comparisons, the technology makes it easier for users to track their progress across several mock interviews. The literature review covers relevant research on automatic personality recognition, action selection, grammatical specification language, dialog state tracking, and text-based emotion detection. demonstrating the relevance of these concepts to the proposed system. The proposed system aims to enhance social skills, particularly relevant to job interviews, through a real-time social cue recognition system. The project demonstrates potential benefits for users in improving their interview performance and receives positive feedback from participants.

Dulmini Yashodha Dissanayake , Venuri Amalya , Raveen Dissanayaka , Lahiru Lakshan [2] ,Proposed “ AI-based Behavioral Analyzer for Interviews/Viva” ,which represents the drawbacks of online interviews in terms of catching subtle nonverbal clues, the system utilizes machine learning and deep learning models to provide a thorough behavioral analysis. With an emphasis on important components like mood, eye movement, smile, and head movements, the system reaches astonishing accuracy levels of over 85%. Moreover, the use of a Big Five trait-based personality model facilitates a comprehensive assessment of applicants. To ensure the robustness and reliability of each component, the project uses a variety of datasets for training. The method enables group comparisons and environment-based evaluations in addition to offering insights into the behavior of individual interviewees through rigorous analysis and model training. The findings from surveys with professionals underscore the significance of the system in addressing challenges associated with virtual interviews. Overall, this project contributes substantially to the field, offering a sophisticated tool for behavior analysis that goes beyond existing systems' capabilities.

Muhammad Laiq, Oscar Dieste [3], Proposed Chatbot-based Interview Simulator: A Feasible Approach to Train Novice Requirements Engineers, This study examines the inadequate preparation provided to beginning requirements engineers in requirements engineering (RE) courses, with a specific focus on interview skills improvement. An AI-based interview simulator created using the Design Science Methodology for Information Systems is the suggested remedy. The goal of the six-cycle refined simulator is to improve interviewing abilities through natural dialogue with requirements engineers. It highlights appropriate interview techniques, including the use of summaries, context-free questioning, and the introduction of common natural language errors like ambiguity and incompleteness. Positive feedback from student testing indicates that the research is producing favorable results. Context-free, context-related, and meta-questions are all understood by the simulator, and it reacts accordingly. Furthermore, a generalized architecture for RE tools utilizing IBM Watson technology has been delineated. The paper concludes with plans to further test the simulator in a real RE course, with the intention of eventually making it available for free use within the RE community. The project addresses a gap in interview training for requirements engineers and leverages AI and natural language processing technologies to provide an effective educational tool.

Ionut Cosmin, Duta Li, Liu Fan, Zhu Ling Shao [4], these authors proposed "Improved Residual Networks for Image and Video Recognition," which describes the notable advancements made to the Residual Network (ResNet) architecture, a popular convolutional neural network (CNN) configuration. The three main aspects of ResNets that are the focus of the suggested improvements are the projection shortcut, the residual building block's structure, and the information flow across network levels. The authors show that across a range of tasks and datasets, including as ImageNet, CIFAR-10, CIFAR-100, COCO, Kinetics-400, and Something-Something-v2, they consistently improve accuracy and learning convergence. Introducing a stage-based network design, optimizing the projection shortcut to minimize information loss, and creating a building block with additional spatial channels to learn more potent spatial patterns are among the main achievements. The suggested method enables very deep training. The experiments showcase the effectiveness of the proposed architecture by successfully training a 404-layer deep CNN on ImageNet and a 3002-layer network on CIFAR-10 and CIFAR-100, surpassing the capabilities of the baseline ResNet. The results suggest that the proposed approach overcomes optimization challenges associated with increasing network depth, establishing new milestones in the depth of CNNs. The authors provide code and models for reproducibility, and the improvements are achieved without a significant increase in model complexity, making the proposed Residual Network architecture a valuable contribution to the field of deep learning.

Ashlin Deepa R N, Prathyusha Karlapati, Mrunhaalhini Reddy Mulagondla, Pavitra Amaranayani [5], proposed “An Innovative Emotion Recognition and Solution Recommendation Chatbot**”,** Here, it presents a web-based chatbot that recognizes emotions and recommends solutions to help people manage them on their own. The system uses artificial intelligence, natural language processing, and machine learning algorithms to address the common mental health issues caused by work-related stress, unstable relationships, and other factors. Users engage with the chatbot by submitting images, choosing a category, and providing text data that describes their mood. Machine learning algorithms, in particular Random Forest, analyze the user's text description to recognize emotions accurately, achieving an impressive accuracy and F1 score of 97.55% and 0.969, respectively. The chatbot functions as a virtual therapist, suggesting customized solutions, such as the system's methodology involves front-end components for user interaction and back-end processes for emotion analysis and recommendation generation. The project demonstrates the potential of AI-driven chatbots in providing accessible and personalized mental health support, with Random Forest identified as the optimal algorithm for emotion recognition.

Chinmayi R, Narahari Sreeja, Aparna S Nair, Megha K Jayakumar, Gowri R,Akshat Jaiswal [6] ,proposed “Emotion Classification Using Deep Learning”**,** The goal of this research project is to improve emotion recognition by using a model framework that combines deep learning methods for speech and facial expression analysis. The main goal is to improve the interaction between humans and computers by correctly recognizing emotions including happiness, sadness, rage, hatred, surprise, and fear. The dataset used in the study consists of 90 samples from the Amrita Emote database for testing and 25,838 samples from the FER2013 image database for training. The speech dataset consists of 20,000 instances spread over four distinct datasets. Convolutional neural networks (CNNs) are used in the proposed model for image processing, while recurrent neural networks (RNNs) are used for speech processing. The study emphasizes how important emotion detection is to overall human wellbeing, especially when it comes to identifying emotional disorders. The system employs various techniques, including pre-emphasis filtering, framing, cepstral conversion, and Mel-frequency Cepstral Coefficients (MFCC) for feature extraction from speech signals.

Guilherme G. Andrade, Geovana R. S. Silva, Francisco, C. M. Duarte Jr [7], proposed “A Conversation-Driven Approach for Chatbot Management", this project presents the Chatbot Management Process (CMP), an all-encompassing technique intended to tackle the difficulties associated with maintaining and modifying chatbot content once it has been deployed. Three major phases make up the CMP, which is centered on developing the Evatalk chatbot for the Brazilian Virtual School of Government: manage, create, and evaluate. The approach places a strong emphasis on user interaction-driven, cyclical content evolution and human-supervised learning. It helps people with different skill sets to collaborate by clearly defining roles for the chatbot team. Positive results from the validation of the Evatalk project included a 160% increase in knowledge base examples, a decrease in human hand-off rates, and stable user satisfaction. Version control, content management, and model training are made easier with the help of tools like EvaTalk Admin, Data Repository, and Model Trainer included in the suggested design. The project's use of containerization enhances scalability, reproducibility, and simplifies the deployment process. Overall, the project underscores the significance of post-deployment content management and offers a systematic approach to enhance chatbot performance and user experience.

Yashwanth Adepu, Vishwanath R Boga ,Sairam U [8],proposed “Interviewee Performance Analyzer Using Facial Emotion Recognition and Speech Fluency Recognition“, In order to analyze interviewee performance, this research develops two multiclass classification models and offers an automated method. The suggested system makes use of video that was taken during interviews, from which frames are extracted to recognize facial emotions and audio to recognize speech fluency. By combining the HaarCascade classifier, Gabor filters, and Convolution Neural Network, the facial emotion identification model achieves higher accuracy in less training time. Using Mel Frequency Cepstral Coefficient characteristics and logistic regression, the speech fluency recognition model classifies speech into four categories: fluent, stuttering, cluttering, and pauses. The interviewee's performance is rated based on the combined forecasts of the two models. The system's increased accuracy is a result of the usage of logistic regression for speech fluency recognition and Gabor filters for facial emotion recognition. The research includes a literature survey, dataset details, and a detailed explanation of the proposed system's components and flow. The datasets comprise FER2013 and ck+ for facial emotion recognition and various speech datasets for speech fluency recognition. The system is presented as a prototype, acknowledging the need for additional metrics such as posture and gestures for a comprehensive interview analysis in future research.

Anushree Deshmukh ,Arti Mishra,Abhishek Adivarekar ,Rohit Pathar [9], proposed “Human Emotion Recognition using Convolutional Neural Network in Real Time**”,** this study employs convolutional neural networks (CNNs) to recognize emotions in real time from facial photos. The study tackles the problem of teaching machines to analyze emotions similarly to humans. The suggested multi-class classifier divides face photos into seven categories based on emotion: anger, happiness, fear, sorrow, disgust, surprise, and neutral. It was trained on the fer2013 dataset. In order to reduce overfitting, the experiment investigates various CNN architectures, including both shallow and deep networks, and uses dropout. The deep network attains an astounding 89.98% accuracy rate. The study goes so far as to use a webcam in real time to demonstrate the model's accuracy in identifying emotions for several faces at once. The model's distinctiveness and effectiveness are enhanced by the employment of swish activation mechanisms in completely connected layers. The study concludes with insights into dataset considerations and future efforts to enhance accuracy further, leveraging advanced GPU capabilities and addressing specific challenges, such as improving recognition of certain emotions like disgust.

Ruhul Amin Khalil, Edward Jones, Mohammad Inayatullah Babar [10], proposed “Speech Emotion Recognition using Deep Learning Techniques” , It employs Deep Learning approaches for Speech Emotion Recognition (SER), addressing the difficulties and developments in this field. In reviewing conventional SER methods, the research highlights how crucial precise emotion identification is to HCI (human-computer interaction). As an alternative to conventional models, it presents Deep Learning techniques, such as Deep Boltzmann Machines (DBMs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Neural Networks (DNNs). The review discusses classification, feature extraction, and the benefits of deep learning—like its capacity to handle intricate structures without the need for manual feature extraction. Using databases like IEMOCAP, Emo-DB, and SAVEE, the study examines the application of these methodologies in the recognition of emotions from speech signals. It provides a comparative analysis of traditional algorithms and deep learning, demonstrating the superior performance of Deep Convolutional Neural Networks (DCNNs). The text emphasizes the significance of deep learning in SER for HCI systems and concludes with a discussion on future directions and challenges in the field.

Pasindu Senarathne, Malaka Silva, Ama Methmini, Dulaj Kavinda, Prof.Samantha Thelijjagoda [11], Proposed "Automate Traditional Interviewing Process Using Natural Language Processing and Machine Learning", The Smart Interviewing System (SIS) is an advanced software application that uses deep learning and contemporary Natural Language Processing (NLP) methods to completely transform the conventional interviewing procedure. The system, which was created with the Python programming language and the ReactJS framework, can handle both written and oral interviews. It functions by translating spoken language into text-based inputs, which enables automatic assessment of applicants' answers. The algorithm predicts scores for each response using sophisticated evaluation parameters derived from deep learning principles, giving interviewers a more precise and effective way to choose eligible applicants. Through a significant reduction in the time and effort needed for candidate selection,Through its innovative approach, the system addresses common flaws in traditional interviewing processes that involve human biases and time-intensive operations. The research methodology includes the use of voice recognition, keyword matching, and machine learning models such as Word2Vec and Doc2Vec for evaluating written test answers. Overall, the SIS represents a cutting-edge solution to enhance Human Resources Management, offering a reliable and automated approach to identify the most suitable candidates for various job positions.

Nuruldelmia Idris, Cik Feresa Mohd Foozy, Palaniappan Shamala [12], proposed "A Generic Review of Web Technology: DJango and Flask" , This research offers a thorough analysis of web technology, emphasizing its historical development, present state, and potential future developments. It highlights the importance of bandwidth and high-speed internet in underlining the role of web services in daily life. The course explores HTML5, CSS, and JavaScript, the core languages of web development, and how important they are to the creation of attractive and useful websites. This article covers the history, benefits, and present and future applications of Python as a programming language and traces the development of web technology back to Sir Tim Berners-Lee's conception. It also looks at using the Bootstrap library and frameworks like Django and Flask for web application development. The functionality, speed, and other aspects of Flask and Django are compared., and authorization is presented, providing insights for developers. The article concludes by underscoring the enduring popularity and versatility of Python in web development, acknowledging the varied strengths of Django and Flask in addressing different development needs.

Li Wang, Ting Liu, Bing Wang, Jie Lin [13], Proposed "Object Tracking Via Imagenet Classification Scores", The goal of this research effort was to improve tracking performance when there is occlusion. The authors tackled the difficult challenge of object tracking in computer vision. Convolutional Neural Network (CNN) classifiers pretrained on the ImageNet dataset provided high-level semantic category responses, which they used to propose a novel occlusion detection technique. A multi-model tracking system that holistically utilizes several tracking models trained on hierarchical convolutional features was also presented by the researchers. Based on occlusion estimate scores, they selected features from several convolutional layers to improve resilience. To effectively re-identify missing target objects, a linear motion model was also suggested. The experimental findings proved the efficacy of the suggested strategy when assessed against the OTB benchmark .showcasing its ability to handle occlusion and improve overall tracking performance compared to state-of-the-art trackers.

R. Mandal, P. Lohar, D. Patil, A. Patil and S.Wagh [14] "AI -Based mock interview evaluator: An emotion and confidence classifier model," 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 521-526, doi: 10.1109/ICISCoIS56541.2023.10100589.Abstract: Interviews are extremely important for a candidate because it is the time when all your hard work is put on the line to get some desired fruitful outcomes in life. It is extremely important in our educational system and recruitment process since they aid in the selection of the right candidate based on the required skills. Mock interviews can boost our confidence and communication skills which can help to perform better. This paper proposed an AI-based mock interview platform that would operate as an intermediatory between the actual interview and preparation mode. Our system will assess the user based on an aggregation of three parameters called emotions, confidence, and knowledge base. Emotion is judged based on facial expression using deep learning CNN algorithm which will classify the emotion among the 7 categorical emotions and confidence evaluation is based on speech recognition using natural language processing and Pydub audio python libraries. For knowledge assessment, keyword mapping, semantic analysis technique is used and web scraping module will extract keywords from received replies and map them to online resources. Hence this system will not only lower the stress and anxiety before an actual job interview but also improve the candidate's confidence.

Nadia Zenati, Cherif Larbas,Mahmoud Belhocine, Nouara Achour [15], proposed “Emotion Recognition via Facial Expressions”, The goal of this study project was to answer the increasing demand for emotion recognition in a variety of fields, including video games, virtual reality, human-computer interaction, and health monitoring. The study used a dataset gathered from 17 participants displaying six different emotional states to propose a unique geometrical facial feature-based technique for recognising facial emotions. An RGB HD camera and the depth sensors Kinect (vI) and Kinect (v2) were used to collect the data. The suggested method used K nearest neighbors (k-NN) as the primary classification methodology and included positional information, including a mix of angle and distance features, for training the classifier. When RGB and RGB-D data were examined in the study, RGB-D features from Kinect (v2) performed better. The paper highlighted the limitations of 2D images for facial emotion recognition due to sensitivity to surrounding conditions. The conclusion emphasized the robustness of RGB-D data in capturing essential geometrical features, leading to higher precision and preservation of facial details across different conditions. The research aimed to contribute to the development of real-time systems for facial expression recognition.

Swayam Badhea, Sangita Chaudhari [16], “Deep Learning based Facial Emotion Recognition”, this shows the importance of facial expressions in nonverbal communication which is examined in this study project, which also highlights the increasing need for automated facial expression identification in fields including market research, AI-based gaming, medical diagnostics, and human-machine interface. With four convolution layers and two fully linked layers, the proposed Convolutional Neural Network (CNN) model seeks to outperform current models in accuracy while requiring less complexity. While it is easy for humans to infer emotions from facial expressions, machines find this task challenging. The study examines many methods for identifying and detecting emotions while emphasizing the drawbacks of particular strategies. The suggested approach entails using a dataset with seven emotion classes—Neutral, Anger, Disgust, Fear, Happy, Sadness, and Surprise—to train the CNN model and assess its output. The model exhibits strong accuracy for most classes but encounters occasional misclassification, particularly for fear and disgust. The research contributes to advancing automated facial expression recognition systems, emphasizing real-time capabilities and ongoing efforts to enhance accuracy across all emotion classes, except for occasional misclassification of fear and disgust.

James Pao [17],proposed “Emotion Detection Through Facial Feature Recognition” which represents a hybrid approach for facial expression recognition, utilizing Viola-Jones cascade object detectors and Harris corner key-points for face and facial feature extraction. To train a multi-class predictor for classifying the seven basic human face emotions, the technique combines principal component analysis (PCA), linear discriminant analysis (LDA), support vector machines (SVM), and histogram-of-oriented-gradients (HOG) feature extraction. The hybrid approach enables reliable prediction through more in-depth analysis using HOG and SVM for complex expressions like disdain, after which effective initial classification using PCA and LDA is possible, especially for "easy-to-distinguish" emotions. Compared to utilizing HOG alone, the technique achieves a 20% runtime reduction with a respectable accuracy of 81%. The paper emphasizes the potential uses of automated facial expression recognition in a variety of contexts and makes recommendations for future developments, highlighting the need for more advanced detection techniques and training data.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf [18] present "DeepFace, a face recognition system that closes the performance gap with human-level accuracy". They introduce a novel approach that combines 3D face modeling with a deep neural network (DNN) to improve alignment and representation. The DNN, with over 120 million parameters, is trained on a large dataset of four million facial images from 4,000 identities. This method achieves a remarkable 97.35% accuracy on the Labeled Faces in the Wild (LFW) dataset, reducing the error of the current state-of-the-art by more than 27%.DeepFace stands out from other approaches by using DL for feature extraction instead of engineered features. The system's success lies in its ability to generalize well to new datasets, even those from different populations, with minimal adaptation. Additionally, DeepFace produces a compact face representation, unlike other systems that rely on tens of thousands of appearance features. The key to its performance is a rapid 3D alignment step that fixes the location of facial regions at the pixel level, allowing the network to learn directly from raw pixel RGB values.In summary, the contributions of DeepFace include the development of an effective DNN architecture and learning method, an advanced facial alignment system based on explicit 3D modeling, and a significant advancement in face recognition performance on challenging datasets like LFW.

2.1 Gap in the literature survey

1. Real-Time Feedback System:

- Incorporating real-time feedback distinguishes the project from existing works.
- This feature allows for immediate user interaction and response, enhancing the overall user experience.

2. Chatbot Integration with CNN:

- The project's chatbot integration, powered by CNN and keras, enhances user engagement.
- This unique feature sets the project apart from existing solutions, providing seamless access to information and support.

3. Flask for User Interface:
    - Flask is used to empower the project's user interface with innovative design and lightweight flexibility.
    - The choice of Flask addresses literature gaps and aims to redefine the user experience by providing an intuitive interface.

4. Real-Time Interview Experience:
    - The inclusion of a real-time interview experience addresses a literature gap.
    - This feature provides students with a valuable platform to practice job interviews online, enhancing their skills and confidence.

5. Technologies and Modules Used:
    - The project integrates Deepface for facial recognition, lybrosa for speech analysis, and CNN for multitasking purposes, ensuring comprehensive data processing.
    - Additional functionalities include a chatbot interface and an suggestion system, enhancing user interaction and experience.
    - Confidence levels are provided through speech analysis utilizing MFCCs, enriching the project's capabilities.

6. Overall Impact and Uniqueness:
    - The project's combination of real-time feedback, chatbot integration, Flask UI, and interview experience sets it apart as a comprehensive and innovative solution.
    - By addressing literature gaps and incorporating advanced technologies, the project aims to make a significant impact in its domain.

CHAPTER 3

REQUIREMENT SPECIFICATION

Requirements specifications (RS) are extensive documents that list all of the features and attributes of the system that has to be developed. A series of use examples that outline every interaction user will have with the software are included. These use cases, sometimes referred to as functional requirements, specify the precise commands and replies that the system is anticipated to provide. The Requirements Specification includes use cases as well as non-functional requirements that specify the characteristics of the system, like security, scalability, and performance.

3.1 Python

Python is a widely used programming language in machine learning (ML) because of its ease of use, adaptability, and robust library and tool ecosystem. Below is the quick synopsis of Python:

1. Ease of Use: Writing and maintaining machine learning code is made simple for developers by Python's legible and clear syntax. Its ease of use is especially advantageous for ML novices.
2. Rich Ecosystem: TensorFlow, PyTorch, scikit-learn, and Keras are just a few of the many modules and frameworks made expressly for machine learning that Python has. These libraries offer preconfigured tools for a range of machine learning activities, including preparing data and deploying models.
3. Community Support: Because of Python's sizable and vibrant developer community, ML practitioners may easily find a wealth of information, tutorials, and discussion boards to assist and advise them.
4. Flexibility: Python is a multifaceted language that facilitates several programming paradigms, enabling machine learning engineers to select the best method for their projects, be it procedural, object-oriented, or functional programming.
5. Integration Capabilities Python is a good choice for developing end-to-end machine learning pipelines that include data processing, model training, and deployment since it can be readily integrated with other languages and technologies.

3.2 Machine Learning

Artificial intelligence (AI) has a branch called machine learning (ML) that focuses on the giving systems the ability to learn from experience and get better without explicit programming. Python's ease of use, adaptability, and availability of robust libraries and frameworks specifically designed for machine learning applications have made it one of the most popular programming languages for ML. Here are some essential features of Python machine learning: Libraries and Frameworks: Python offers a rich ecosystem of machine learning libraries and frameworks like TensorFlow, Keras, PyTorch, librosa and scikit-learn, providing tools for building, training, and deploying models for various ML tasks.

- Ease of Use: Python's clean syntax is beginner-friendly and ideal for quick prototyping and experimentation, crucial in ML where iterative development is common.
- Community and Support: Python has a large community of developers, data scientists, and researchers contributing to open-source projects, providing ample support through forums, online communities, and extensive documentation.
- Integration Capabilities: Python integrates seamlessly with tools like Jupyter notebooks, Matplotlib, Seaborn, and various cloud platforms, facilitating interactive development, data visualization, and scalable deployment.
- Flexibility: Python's flexibility allows practitioners to use a wide range of algorithms and techniques for different ML problems, including supervised, unsupervised, and deep learning methods.
- Scalability: While not as performant as lower-level languages, Python can be optimized using libraries like NumPy and by integrating with optimized code in other languages. Its ecosystem also supports distributed computing and parallel processing for scaling ML applications. Below Figure shows the working of Machine learning model.

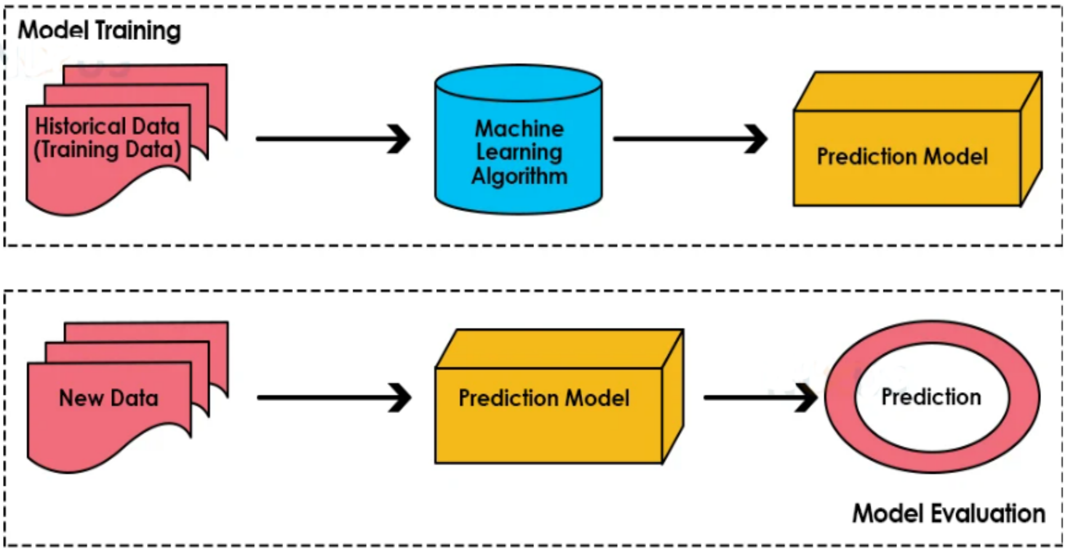


Figure 3.1: Working of the Machine learning model

Figure 3.1 shows the Working of the Machine learning model, during model training, historical data (training data) is used to train the machine learning algorithm, which adjusts its internal parameters to learn patterns and relationships in the data. Once trained, the algorithm forms a prediction model capable of making predictions or classifications. When new data is inputted, it directly goes to the prediction model, which applies the learned patterns to the new data to make predictions without retraining the entire algorithm. This separation allows for efficient use of computational resources and enables the model to adapt to new data without repeating the training process.

3.3 Deepface model

The DeepFace Model, developed by Facebook, utilizes a Convolutional Neural Network (CNN) architecture and is trained using a vast dataset. This model is adept at recognizing emotions by comparing triplets of images (anchor, positive, and negative) and adjusting its parameters to minimize intra-class variations while maximizing inter-class differences.

1. DeepFace represents a significant departure from traditional models by leveraging deep learning techniques to generate facial representations in a high-dimensional space, enabling comprehensive analysis of emotional expressions during interviews. This advanced approach provides a more nuanced understanding of candidates' emotional intelligence and their suitability for roles requiring strong interpersonal skills.
2. The real-time analysis capability of DeepFace facilitates instant feedback to both candidates and interviewers, enhancing the interview experience by enabling adjustments during the interview process. This iterative feedback loop fosters better communication and rapport between the parties involved, ultimately leading to more informed hiring decisions and improved candidate selection processes.

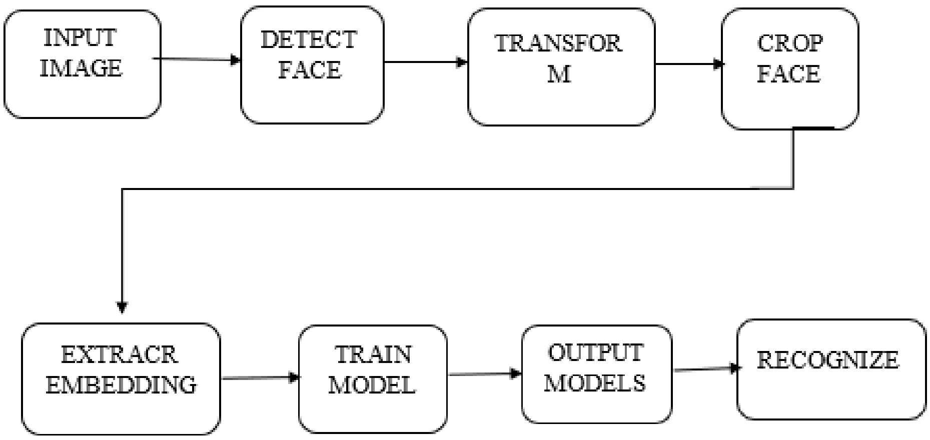


Figure 3.2: Architecture of DeepFace Model

The DeepFace Model, developed by Facebook, is a state-of-the-art face recognition system that utilizes a Convolutional Neural Network (CNN) architecture. Trained on a vast dataset, deep face is adept at recognizing emotions by comparing triplets of images: an anchor (representing a base emotion), a positive sample (representing a similar emotion), and a negative sample (representing a different emotion). By updating its parameters to minimize intra-class variations while maximizing inter-class differences, the network learns to produce emotion embeddings for new face images based on its learned representations.

Step 1: Batches of face images expressing various emotions are provided as input.
Step 2: The input images undergo processing through convolutional layers within the CNN architecture for feature extraction.
Step 3: Extracted features undergo normalization to ensure consistent representation across different faces and expressions, enhancing the model's robustness.
Step 4: Normalized features contribute to creating embeddings that capture unique facial expression features, facilitating a rich representation of emotional cues.
Step 5: Leveraging a Triplet Loss Function, the network learns by contrasting triplets of images, facilitating nuanced understanding of emotional nuances.
Step 6: Distances between embeddings of anchor, positive, and negative images are computed to assess the relative similarity and dissimilarity of emotions.
Step 7: A loss function is computed to minimize variations within similar emotions and maximize disparities between different emotions, guiding the network's learning process.
Step 8: The computed loss is used to update the network's parameters through backpropagation, iteratively refining its ability to discern subtle emotional cues.
Step 9: Through training on a labelled dataset, the network learns to recognize emotions across diverse facial expressions, enhancing its accuracy and generalization capabilities.
Step 10: Once trained, the network can produce emotion embeddings for new face images, enabling real-world applications of emotion recognition in various contexts.

3.4 Haar Cascade Classifier

The Haar Cascade Classifier architecture comprises a cascade of simple classifiers arranged in sequential stages, each tailored to detect specific features like edges, corners, or textures indicative of the target object. This cascading structure enables swift rejection of non-object regions in the image, concentrating computational efforts on areas more likely to contain the target. By iteratively evaluating regions against increasingly stringent criteria, the cascade progressively refines candidate regions, optimizing the detection process. This approach effectively narrows down the search space to regions closely resembling the target object, enhancing efficiency and accuracy, particularly in real-time applications where speed is critical. Overall, the cascade architecture streamlines computational workload by prioritizing the examination of promising regions, bolstering the classifier's object detection performance. Proposed steps are as follows:

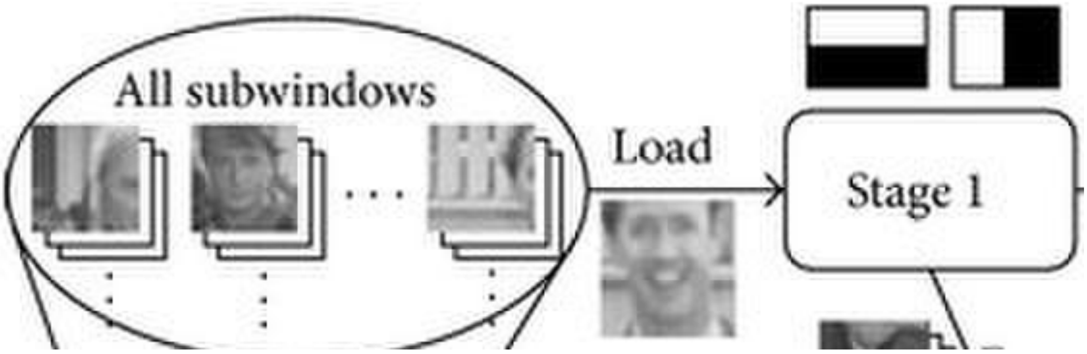


Figure 3.3: Architecture of Haar Cascade Classifier

Figure 3.3 shows the architecture of the Haar Cascade Classifier, depicting its stepwise process:

Step 1: The classifier initiates the detection process by evaluating simple Haar-like features at various scales and positions across the image.

Step 2: At each stage of the cascade, a subset of features is selected and combined using a weighted sum to form a feature vector.

Step 3: The feature vector undergoes scrutiny by a decision tree or another classifier to determine whether the region contains the target object.

Step 4: If the region surpasses the classification threshold at a specific stage, it is considered a potential detection and forwarded to subsequent stages for further refinement.

Step 5: As the image progresses through the cascade, regions failing to meet the classification criteria are swiftly dismissed, reducing the computational burden.

Step 6: After traversing all stages of the cascade, regions that have successfully passed all classification thresholds are confidently identified as detections of the target object, thus completing the classification process.

3.5 Audio Feature extraction using MFCC

- Feature Extraction with MFCC: The librosa.feature.mfcc function is used to extract Mel-frequency cepstral coefficients (MFCCs) from audio files. MFCCs are commonly used in speech and audio processing as they capture the characteristics of the audio signal.
- Random Forest Classifier: The RandomForestClassifier from scikit-learn is used

for classification. Random forests are an ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees.

- Cross-Validation: The StratifiedKFold method from scikit-learn is used for cross-validation. It splits the dataset into n_splits consecutive folds while ensuring that each fold is a good representation of the whole dataset by preserving the percentage of samples for each class.
- Predicting Confidence Level: The predict_proba method of the trained classifier is used to predict the probability of the input features belonging to each class. In this case, the confidence level is calculated as the predicted probability of the 'confident' class multiplied by 10 to scale it to a 0-10 range.
- Visualization: Matplotlib is used to visualize the audio waveform and the confidence level graph for each audio file.

3.6 Random forest classifier

A Random Forest classifier employs ensemble learning with decision trees, training each tree on a subset of data and aggregating predictions through averaging (regression) or voting (classification). It enhances performance by introducing diversity with bootstrapping and random feature selection. Random Forests excel in accuracy, scalability, and handling high-dimensional datasets, while being robust against overfitting.

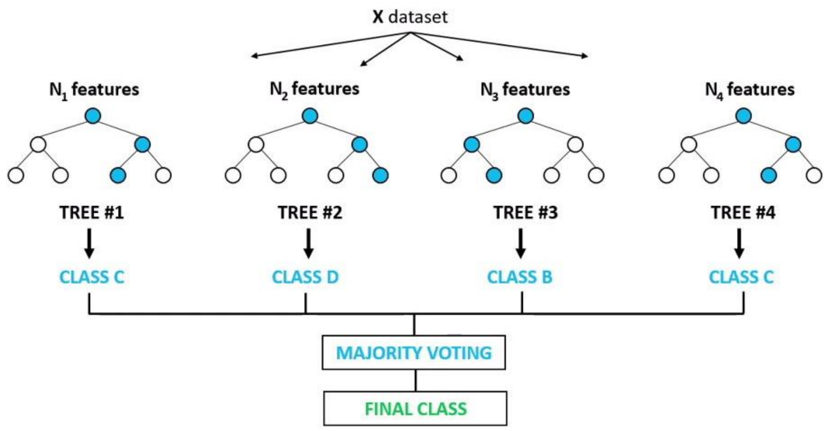


Figure 3.4: Random forest classifier

Above figure 3.4 shows the Random Forest classifier which is a machine learning model that utilizes an ensemble technique called "bagging" to enhance the performance of decision tree classifiers. It is employed for both classification and regression tasks. The model comprises a collection of decision trees, with each tree trained on a subset of the training data. The final prediction is made by aggregating the predictions of all trees, either by averaging (for regression) or voting (for classification). To introduce diversity among the trees, random subsets of the training data are sampled with replacement for each tree (bootstrapping). Additionally, at each node of the decision tree, a random subset of features is considered for splitting. This random feature selection further enhances diversity and helps prevent overfitting. In classification tasks, the class predicted by each tree is treated as a "vote," and the class with the most votes is predicted. For regression tasks, the predictions of all trees are averaged to obtain the final prediction. Random Forests are favored for their high accuracy, scalability, and ability to handle large datasets with high dimensionality. They are also robust against overfitting due to the randomness introduced during training.

3.7 Chatbot using Neural Network:

A loaded intentions knowledge base is available. This file's data is pre-processed before being used. We'll keep all of the intents' tags in a list named labels that will be used for training purposes. Train the sequential model following the numerical transformation of the labels and patterns. The model was evaluated using 20% of the data after training on 80% of the data. If the model doesn't produce good results, keep repeating this training.

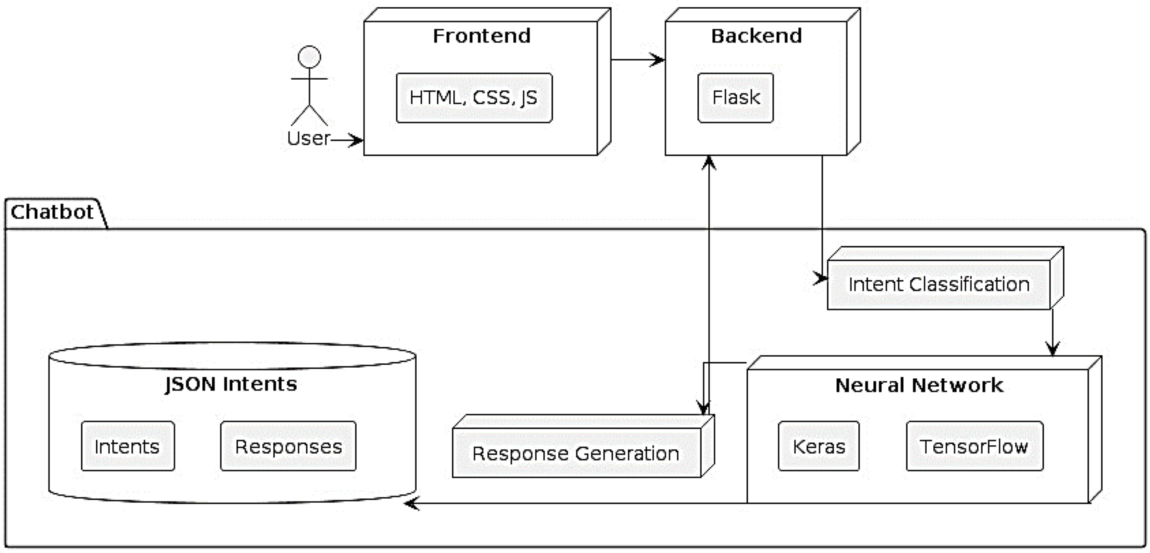


Figure 3.5: Architecture of Chatbot

Figure 3.5 illustrates the chatbot Architecture diagram of the chatbot, The chatbot architecture incorporates a neural network model trained on JSON intents, enabling natural language understanding and response generation. The modular design allows for easy integration with the overall AI-based interview evaluator system, providing an interactive conversational experience for users.

The key components of the chatbot architecture are:

1. **Frontend**: The user interface, built with HTML, CSS, and JavaScript, where users can interact with the chatbot.

2. **Backend**: The Flask backend serves as an intermediary between the frontend and the chatbot component.
3. **Chatbot**: The core chatbot module responsible for handling user inputs, intent classification, and response generation.
4. **Intent Classification**: This module classifies the user's input into predefined intents using natural language processing techniques.
5. **Neural Network**: A neural network model, built using Keras and TensorFlow, is used for intent classification and response generation.
6. **Response Generation**: Based on the classified intent, this module generates an appropriate response using the trained neural network model.
7. **JSON Intents Database**: A database containing predefined intents and corresponding responses in JSON format, used for training the neural network model.

3.8 Functional Requirements:

Functional requirements for a face and speech-based emotion detection system using deep learning encompass the features and capabilities that the system must possess to meet its objectives. Here's a list of functional requirements:

User Authentication and Authorization:

- Users (candidates and interviewers) must be able to register and create accounts.
- Secure login and logout functionality must be implemented.

Video Processing:

- The system must be able to capture and process video input from the candidate during the interview.
- Facial detection and tracking using techniques like HaarCascade classifiers must be implemented.
- Facial expressions analysis and emotion classification using the Deepface model must be conducted.
- Real-time feedback on the candidate's emotions must be displayed.

Audio Processing:

- Audio input from the candidate must be recorded during the interview.
- Speech-to-text conversion using Google Speech Recognition API or SpeechRecognizer library must be implemented.
- Audio feature extraction using libraries like librosa must be conducted.
- The candidate's confidence level must be evaluated based on audio features and a trained Random Forest Classifier.

Similarity Evaluation:

- The system must be able to compare the candidate's responses with expected answers.
- The similarity between the candidate's responses and expected answers must be computed.
- A score or metric representing the candidate's knowledge and understanding must be provided.

Chatbot Integration:

- A neural network-based chatbot must be integrated for natural language understanding and response generation.
- Candidates must be able to interact with the chatbot during the interview process.
- Chatbot interactions must be recorded and analyzed for further evaluation and feedback.

Data Storage and Retrieval:

- Interview data must be stored in a MongoDB database.
- Mechanisms must be implemented to retrieve and analyze stored data for reporting and analytics purposes.

Suggestion Generation:

- Personalized feedback and suggestions for candidates must be generated based on the analysis results.
- Recommendations for improvement and areas to focus on for future interviews must be provided.

User Interface:

- A user-friendly web interface must be developed for candidates and interviewers using HTML, CSS, JavaScript, and frameworks like Bootstrap.
- Candidates must be able to initiate and participate in interviews through the web interface.
- Real-time analysis results, feedback, and suggestions must be displayed during the interview process.

Security and Privacy:

- Appropriate security measures must be implemented to protect user data and ensure confidentiality and integrity.
- Compliance with relevant data protection regulations and privacy laws must be ensured.

- Transparency and control over the use of personal data must be provided for candidates and interviewers.

3.9 Software Requirements:

- Python 3.11.2
- Flask (web framework)
- TensorFlow/Keras (for neural network-based chatbot)
- OpenCV (for computer vision tasks)
- Librosa (for audio processing)
- Scikit-learn (for machine learning algorithms)
- NumPy (for numerical operations)
- Matplotlib (for data visualization)
- MongoDB (for data storage)

3.10 Hardware Requirements:

- RAM: 8GB
- Processor: AMD Ryzen 5 10th generation
- Hard disk: compatible
- A computer or server with sufficient processing power and memory to handle video and audio processing.
- A webcam or video input device for capturing candidate video during the interview.
- A microphone or audio input device for recording candidate audio during the interview.

CHAPTER 4

PROBLEM STATEMENT

The traditional job interview process is plagued by inherent biases and subjectivity, leading to inconsistent and potentially flawed hiring decisions. Human evaluators are susceptible to unconscious biases influenced by factors such as appearance, gender, cultural background, or personal preferences. Additionally, manual evaluation of candidates during interviews can be time-consuming, resource-intensive, and prone to human error, particularly when dealing with a large volume of applicants. The lack of standardized and objective evaluation criteria can lead to inconsistencies in assessing candidates, making it challenging to compare their performances fairly.

4.1 Objectives

The primary objectives of AI Based Interview Evaluator include:

1. **Real-Time Interview Experience**: Our system provides candidates with a real-time, interactive interview environment that closely mimics the experience of an actual interview. Candidates can engage with the system as they would with a human interviewer, creating a more natural and immersive experience.
2. **Feedback Generation**: We generate customized feedback messages to offer constructive advice to candidates. This helps them address nervousness and improve their interview performance. The feedback is tailored to each candidate's responses and behavior during the interview.
3. **Interactive Support**: Our system offers interactive support through a chatbot interface. Candidates can ask questions and seek advice, fostering a dynamic and personalized learning experience. The chatbot provides real-time assistance, enhancing the overall interview experience.
4. **Facial Expression Analysis**: We use computer vision techniques to detect and analyze facial expressions. Deep learning models, such as Convolutional Neural Networks (CNNs), are trained to recognize emotions like happiness, sadness, anger, surprise, fear, and disgust. This analysis provides valuable insights into candidates' emotional states during the interview.
5. **Speech-Based Emotion Detection**: Our system extracts emotional information from speech signals using signal processing and machine learning techniques. We analyze intonation, pitch and other acoustic features in speech to classify emotional states. This helps us understand candidates' emotions and reactions during the interview.

6. **Objective Evaluation**: We analyze video and audio inputs objectively using techniques like facial expression recognition, speech analysis, and confidence level assessment. This ensures unbiased evaluation of candidates and provides a fair assessment of their performance.
7. **Comprehensive Assessment**: Our system evaluates multiple aspects including emotions, confidence levels, knowledge, and communication skills. This holistic approach offers a complete view of candidates' performance, helping to identify strengths and areas for improvement.
8. **Efficient Evaluation Process**: We automate the assessment process to reduce time and resources compared to manual evaluation. This enables efficient and scalable evaluation of a large number of candidates, streamlining the interview process.
9. **Standardized Evaluation Criteria**: We implement standardized and consistent evaluation criteria to ensure a fair comparison of candidates' performances. This helps maintain fairness and objectivity in the evaluation process.
10. **Chatbot Interface**: Our system integrates a chatbot interface for natural language understanding. The chatbot provides interactive feedback to candidates during the interview process, enhancing their experience and providing valuable insights.
11. **Personalized Feedback**: We generate personalized feedback and suggestions for candidates based on their performance. This helps candidates improve their interview skills and performance in future interviews, tailoring the feedback to their specific needs.
12. **Data Privacy and Security**: We ensure data privacy, security, and compliance with relevant regulations. Our system provides transparency and control over the use of personal data, ensuring that candidates' information is protected.
13. **Audio Inputs Analysis**: We analyze audio inputs using techniques like Mel Frequency Cepstral Coefficients (MFCCs) and Random Forest Classifiers. This helps evaluate candidates' confidence levels and provides additional insights into their performance.
14. **Facial Recognition and Tracking**: Our system implements facial recognition and tracking using techniques like HaarCascade classifiers and the Deepface model. This allows us to analyze candidates' facial expressions and reactions during the interview.
15. **Neural Network-Based Chatbot**: We integrate a neural network-based chatbot for natural language understanding. This chatbot enables candidates to interact during the interview process, providing a more engaging and personalized experience.
16. **Support for Different Interview Domains**: Our system supports various interview domains such as Frontend Web Development, Backend Web Development, AI and

Machine Learning, and Android App Development. This allows us to cater to a wide range of candidates with different skill sets and backgrounds.

17. **User-Friendly Web Interface**: We develop a user-friendly Flask-based web interface for candidates and interviewers. The interface displays real-time analysis results, feedback, suggestions, and output graphs, making it easy to navigate and use.
18. **Evaluation of Response Similarity**: We evaluate the similarity between candidates' responses and expected answers, providing a score or metric for knowledge and understanding. This helps us assess candidates' comprehension and analytical skills.
19. **User Authentication and Authorization**: Our system implements secure sign-in and sign-up functionalities, ensuring user authentication and authorization. This helps protect candidates' information and maintain the integrity of the interview process.
20. **Data Storage**: We store interview data in a MongoDB database, including emotions, confidence levels, similarity scores, and chatbot interactions. This allows us to keep track of candidates' performance and progress over time.
21. **Separate Chatbot API**: We integrate a separate chatbot API for website navigation, providing an additional layer of interactive support to users. This enhances the user experience and makes it easier to access information and resources.
22. **Display Interview History**: Our system displays the history of previous interviews conducted by candidates, including detailed analysis and evaluation metrics. This helps candidates track their progress and identify areas for improvement.
23. **Learning Section**: We provide a learning section with educational materials and resources to assist candidates in improving their interview skills and performance. This helps candidates prepare for future interviews and enhance their professional development.
24. **Scalability and Security**: We ensure system scalability and efficiency to handle a large number of candidates and interviews. We adhere to security protocols for data confidentiality and privacy compliance, ensuring that candidates' information is protected at all times.

CHAPTER 5

SYSTEM DESIGN

System design, where a project's architectural blueprint is painstakingly constructed, is an important stage in the software development lifecycle. It entails a thorough understanding of the requirements and their translation into an effective, scalable structure. The design integrates different parts, modules, and how they work together to guarantee smooth operation. Data storage, processing techniques, and the system's general capacity to manage user inputs and external dependencies are important factors to take into account. A well-planned system design considers future scalability and adaptability in addition to meeting the objectives of the project now. It acts as the development team's cornerstone, directing the implementation stage in the direction of a reliable and maintainable software solution. In this situation, using best practices in system design as a software engineer is essential to producing a high-quality and sustainable product. System design is indeed a crucial stage in the software development lifecycle. It serves as the blueprint for the entire project, guiding how different components will interact and ensuring that the system is not only functional but also scalable , maintainable in the long run. A well-planned system design considers future scalability and adaptability in addition to meeting the objectives of the project now.

5.1 Architectural Design

A crucial step in the software development process is architectural design, which defines the general arrangement and structure of a system. It entails choosing carefully which system parts to include, how to arrange them, and what systems to use to make connections easier. The foundation for a software solution that is durable, scalable, and maintainable is laid at the architectural design stage. To guarantee peak performance, system modularity, data flow, and communication protocols are carefully taken into account. Within the framework of our project, architectural design decisions are essential to effectively accomplishing the project's objectives. Following well-established architectural principles—such as modularity and separation of concerns—the design seeks to improve flexibility, encourage code reuse, and expedite the development process. This approach ensures that the resulting system not only meets current requirements but also allows for seamless adaptation to future enhancements and changes. By focusing on these aspects, architectural design sets the stage for a software solution that not only meets current requirements but also adapts and evolves to meet future needs effectively.

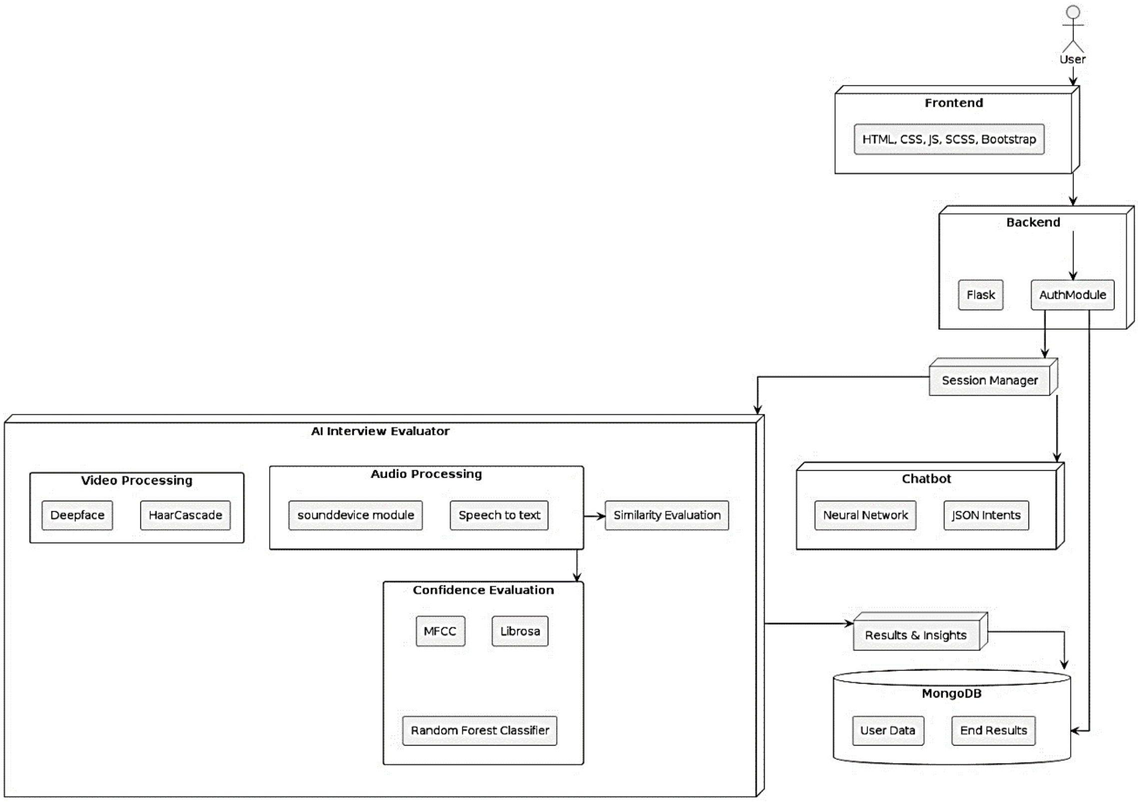


Figure 5.1: Architectural diagram of AI Based Interview Evaluator

Figure 5.1 Shows the Architectural diagram of proposed system which is AI Based Interview Evaluator: An Emotion and Confidence Classifier makes use of a Flask backend and artificial intelligence components , the components are as follows

- User: This block represents the person using the interview simulator.
- Frontend: This block represents the user interface of the interview simulator. It is likely built with HTML, CSS, JavaScript, SCSS, and Bootstrap, which are all web development technologies.
- Backend: This block represents the server-side of the interview simulator. It is built with Flask, which is a Python web framework.
- Deepface: This block is likely referring to a deep learning library used for facial recognition. In the context of the interview simulator, it might be used to track the user's eye gaze or facial expressions.
- HaarCascade: This block is likely referring to a computer vision technique used to detect objects in images and videos. In the context of the interview simulator, it might be used to track the user's body language.
- sounddevice module: This is a Python library used for recording and playing audio. In the interview simulator, it might be used to record the user's responses to interview questions.

- Speech to text: This block refers to a technology that converts spoken language into text. In the interview simulator, it might be used to transcribe the user's responses to interview questions.
- Similarity Evaluation: This block compares the user's responses to a set of pre-defined criteria.
- Neural Network: This is a type of machine learning algorithm that is inspired by the structure of the human brain. In the interview simulator, it might be used to evaluate the user's responses for sentiment or key phrases.
- JSON Intents: JSON (JavaScript Object Notation) is a lightweight data format for human-readable text interchange. Intents are a way to specify what a user is trying to achieve with a language interaction. In the interview simulator, JSON intents might be used to define the goals of the interview and the criteria for a successful response.
- Confidence Evaluation: This block might evaluate the user's confidence level based on their speech or facial expressions.
- MFCC: Mel-frequency cepstral coefficients (MFCCs) are a feature extraction technique used in speech recognition. In the interview simulator, MFCCs might be used to extract features from the user's speech that can be used to evaluate their communication skills.
- Librosa: This is a Python library for audio and music analysis. In the interview simulator, it might be used to extract features from the user's speech.
- Results & Insights: This block represents the output of the interview simulator. It might include a score for the user's interview skills, as well as feedback on their strengths and weaknesses.
- MongoDB: This is a NoSQL database that is often used for storing large amounts of data. In the interview simulator, MongoDB might be used to store user data, such as their interview results.
- Random Forest Classifier: This is a machine learning algorithm that can be used for classification tasks. In the interview simulator, it might be used to classify the user's responses as good, bad, or neutral.
- User Data: This block represents the data that is collected about the user during the interview simulation. This data might include the user's responses to interview questions, as well as their facial expressions and body language.
- End Results: This block represents the final output of the interview simulator, which could be a score, a ranking, or other insights.

5.2 Flow Diagram

Flow diagrams are an essential tool for project documentation since they help to visually depict the logical flow of processes inside a project. These diagrams give a clear and succinct summary of the interactions, decision-making stages, and sequential steps in a system. Using flow charts in project documentation as a software engineer helps team members, stakeholders, and other project participants communicate effectively.

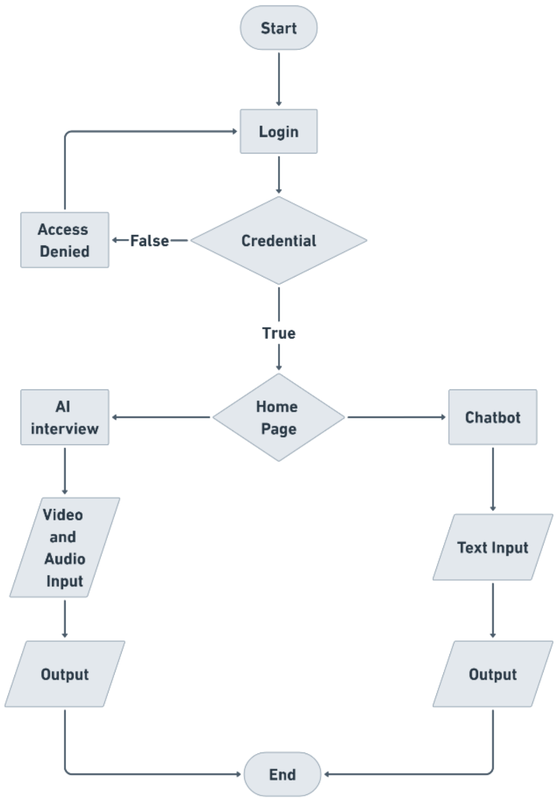


Figure 5.2: Flow Chart

Figure 5.2 shows the flowchart for the AI-Based Interview Evaluator system which illustrates the sequential process initiated by a user logging into the system, with subsequent pathways dependent on the validation of the provided credentials, if the credentials are confirmed as correct, the user is then directed to the system's homepage, where they are presented with the option to choose between engaging in an AI-based interview that incorporates both video and audio inputs, leading to an output after processing, or alternatively, interacting with a chatbot through text input, which similarly results in an output after processing. However, in the event that the login credentials are determined to be incorrect, access to the system is denied, thereby preventing further progression. Here the chatbot is trained using neural network and built using keras and the AI interview evaluator consists audio analysis using lybrosa through MFCC,s of the audio.

5.3 Usecase Diagram

A use case diagram is a visual representation of the interactions between actors (users or external systems) and a system under consideration to achieve specific goals. It illustrates the functionalities or features of a system from an external point of view.

Figure 5.3: Usecase Diagram

The diagram features two stick figures on opposite sides, likely representing the interviewee or candidate and the interviewer or system administrator. This suggests a clear distinction between the roles or entities involved in the interview process. The system receives video and audio data as input from the candidate, indicating a multimedia input source for the evaluation process. Voice and audio datasets are used as training data for the machine learning models, suggesting the system employs machine learning techniques for analysis and evaluation. Pre-processing steps, such as noise removal, are undertaken on the input data, indicating a data cleaning or preparation phase to enhance the quality of the input data for analysis. Face detection algorithms are applied to identify and track the candidate's face in the video, highlighting a visual analysis component within the evaluation process. Audio processing techniques are employed to extract features from the audio data, suggesting that both visual and audio cues are considered in the evaluation process. The system then measures and analyzes various aspects of the video and audio data, such as emotions and confidence levels, indicating a multi-dimensional analysis of the candidate's performance. The detected emotions and confidence levels are presented as outputs or results, suggesting the system provides feedback or evaluation based on these metrics. Additional outputs, such as recommendations or a final evaluation score for the candidate, may be generated, indicating a comprehensive evaluation process with potential feedback mechanisms.

5.4 Sequence Diagram

A sequence diagram is a type of interaction diagram that shows how objects interact in a particular scenario of a use case. It visualizes the interactions between objects in a sequential order, depicting the messages exchanged between them over time. Each object is represented by a vertical line, and the messages exchanged between objects are represented by horizontal arrows between the lines. The sequence diagram also shows the order in which the messages are sent and the lifeline of each object, indicating the duration of its existence during the interaction. Sequence diagrams are useful for visualizing the flow of control in a system and understanding the interactions between different components.

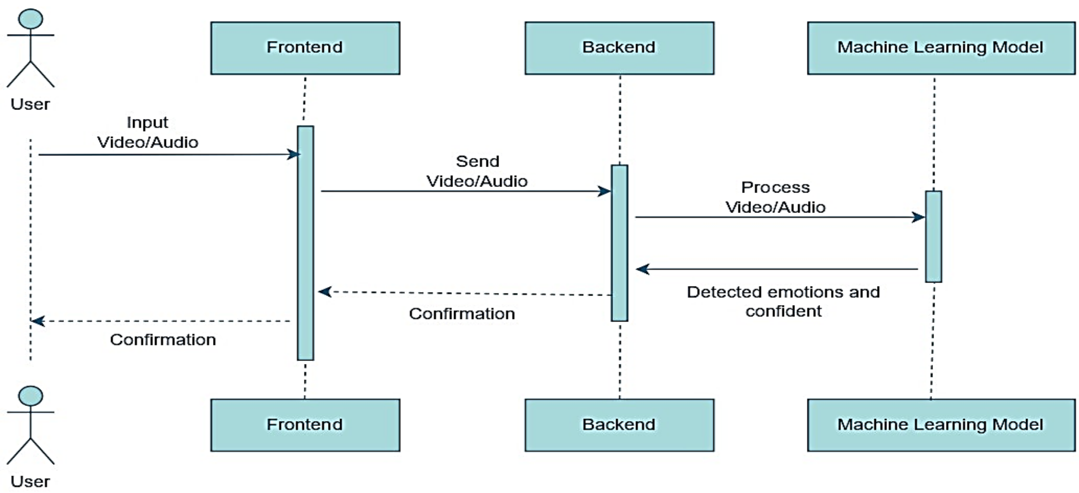


Figure 5.4: Sequence diagram

Figure 5.4 depicts a system with three key components: a user interface, a backend, and a machine learning model. Users provide video and audio input, which is processed by the backend and analyzed by the model. Analyzed emotions and confidence levels are then displayed back to the user. This diagram showcases the overall flow of data and interaction between components for user input processing and feedback generation. The image depicts an architecture diagram for an emotion detection system involving three main components:

Frontend, Backend, and Machine Learning Model.

The process flow is as follows:

1. The user inputs a video or audio file through the Frontend.
2. The Frontend sends the video/audio file to the Backend.
3. The Backend processes the video/audio file and sends it to the Machine Learning Model.
4. The Machine Learning Model analyzes the video/audio and detects emotions along with a confidence score.
5. The detected emotions and confidence score are sent back to the Backend.
6. The Backend sends a confirmation to the Frontend.
7. The Frontend displays the confirmation to the user.

This architecture separates the responsibilities into different components: the Frontend handles user interaction, the Backend manages data processing and communication with the Machine Learning Model, and the Machine Learning Model performs the actual emotion detection analysis on the video/audio input.

CHAPTER 6

SYSTEM TESTING

System testing is an important phase in the software development lifecycle because it allows for a thorough evaluation of the integrated system. It entails evaluating the system's functionality, performance, dependability, and security to ensure that it meets the required specifications. The process begins with test planning, which defines objectives and strategies, followed by test case design and execution, which validates the system against different scenarios. Defect tracking allows for the documentation and resolution of discrepancies, while regression testing ensures that existing functionality remains intact during changes. Performance and security testing are critical components for assessing responsiveness, scalability, and data protection measures. Finally, system testing seeks to instill confidence in stakeholders regarding the software's quality and readiness for deployment.

6.1 White Box Testing

White box testing examines software's internal structure, allowing for comprehensive coverage of code paths and logic. It enables early defect detection and code optimization, thereby improving overall software quality. However, it requires skilled testers and may be time-consuming. Maintenance overhead can arise as a result of code changes that affect test cases. While effective at detecting implementation flaws, it may overlook issues with user interaction or external dependencies.

Advantages:

1. Detailed coverage of code paths and logic.
2. Enables early defect detection.
3. Optimizes and improves code quality.
4. Improves security testing capabilities.
5. Provides insight into the inner workings.
6. Customises test cases for specific code segments.

Disadvantages:

1. Requires skilled testers.
2. Time-consuming testing process.
3. Dependence on internal implementation knowledge.
4. Code changes result in increased maintenance overhead.
5. Due to limited scope, external factors may be overlooked

6.2 Black Box Testing

Black box testing assesses software functionality without knowledge of internal code and focuses on user scenarios. It ensures that requirements are met while also providing a satisfactory user experience. While it is user-focused and does not require coding knowledge, it may overlook complex code interactions. Significant challenges include the reliance on documentation and the potential for redundancy in test cases. Despite its limitations, it is critical for ensuring system behavior and compatibility.

Advantages:

1. It simulates real-world user scenarios.
2. No prior knowledge of internal code is required.
3. Focuses on the user experience and requirements.
4. End-user-focused tests.
5. Ideal for testing third-party or proprietary software.

Disadvantages:

1. Limited insight into internal operations.
2. You may overlook certain code paths or logic.
3. Less effective in detecting implementation flaws.
4. The testing scope may not include all scenarios.
5. Rely heavily on documentation and specifications.

Testing Methodologies

1. Unit Testing
2. Integration Testing
3. System Testing

6.2.1 Unit Testing

Unit testing is the process of validating individual units or software components in isolation, usually at the function or method level. Its goal is to ensure that each unit performs exactly as specified. To detect defects early, developers frequently automate it during the coding phase. Unit tests are run using testing frameworks to verify small, specific pieces of functionality. They improve code maintainability and make regression testing more efficient.

6.2.2 Integration Testing

Integration testing investigates the interactions and interfaces between integrated units or components to ensure that they work as intended. It verifies interoperability and communication between system components. It is typically performed after unit testing to detect interface flaws, data flow issues, and integration errors early in the development process. Integration tests can be automated or manual, and their goal is to ensure that components integrate smoothly. They play an important role in ensuring the overall stability and reliability of the software.

6.2.3 System Testing

System testing assesses the entire integrated software system to ensure that it meets the required specifications and quality standards. It assesses the system's functionality, performance, reliability, and security. It is typically performed after integration testing to validate the system's behavior under a variety of real-world scenarios. System tests may include functional, non-functional, and regression testing to ensure complete coverage. The results help to determine the system's readiness for deployment and user acceptance.

CHAPTER 7

RESULTS

The result of a project is the outcome achieved after completing its tasks, representing the desired change or deliverable. It encompasses tangible products, like software applications, and intangible benefits, such as improved processes or increased efficiency, demonstrating the project's success in meeting its objectives. Project results are often measured against predefined criteria to determine their success and impact. They can also include documentation, training materials, or other artifacts produced during the project. Ultimately, the result of a project should align with the goals and expectations set forth in its initial planning stages.

7.1Screenshots

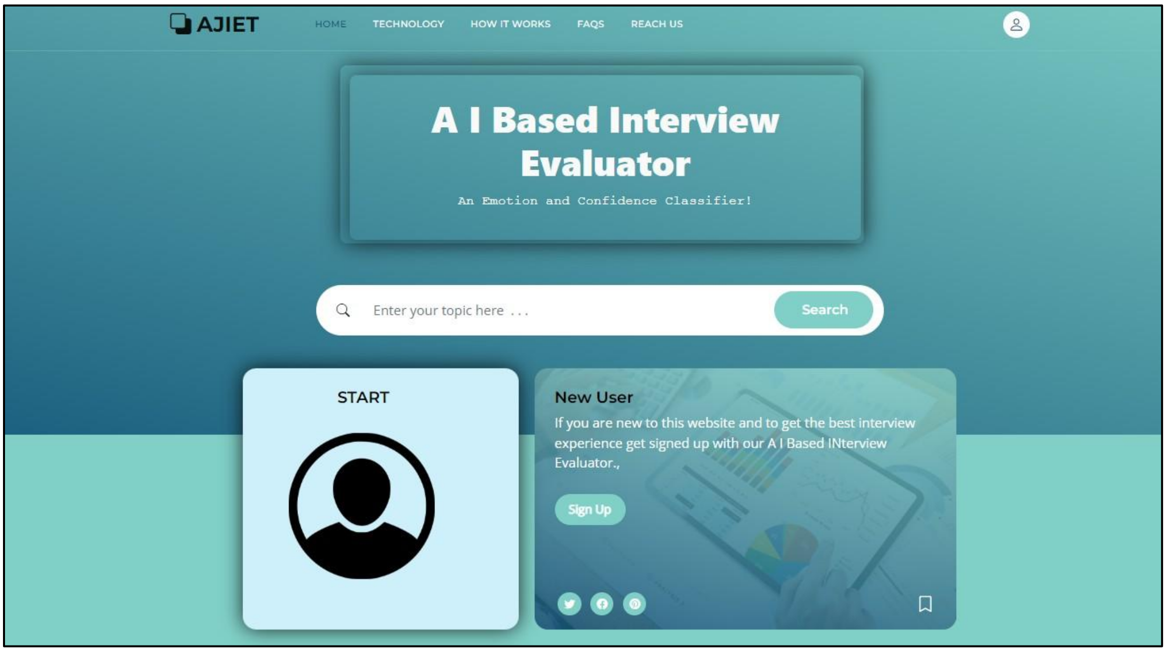


Figure 7.1: Frontpage of A I Based Interview Evaluator

Figure 7.1 shows the homepage of "A I Based Interview Evaluator," which is a website designed for evaluating interviewees using artificial intelligence. The banner highlights its function as both an "A I Based Interview Evaluator" and an "Emotion and Confidence Classifier," indicating its focus on assessing emotional states and confidence levels. Users are greeted with a search bar to enter interview topics and options to either start using the tool or register as a new user. The website's clean and professional design enhances usability, although it's crucial to recognize that while AI may discern emotional cues and confidence levels, its ability to accurately assess a candidate's job suitability, skills, or experience remains a topic of debate in the scientific community.

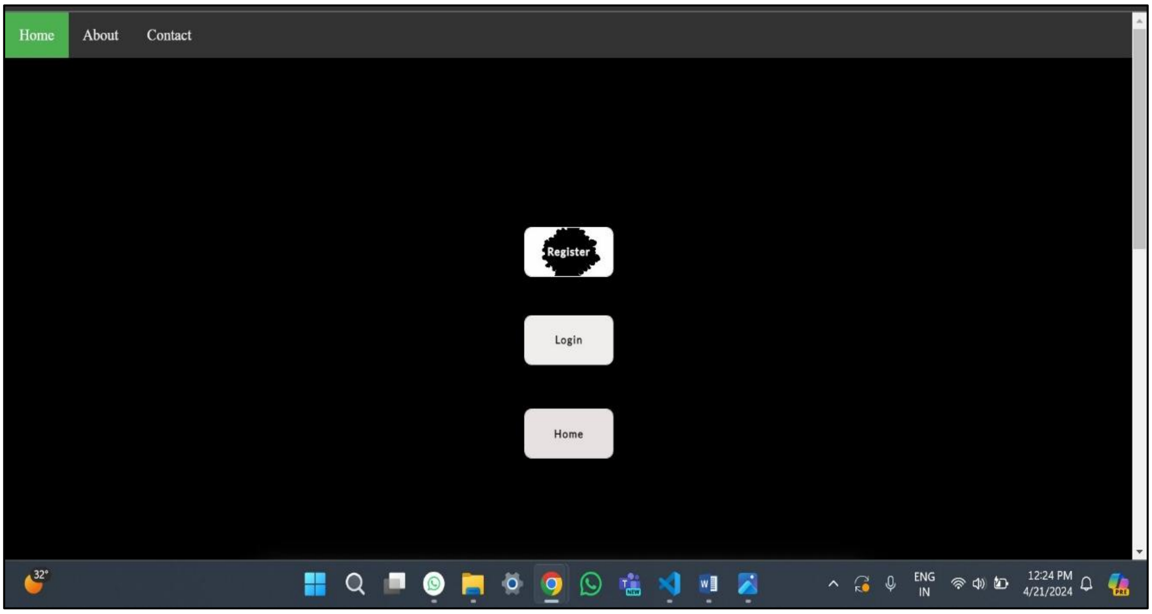


Figure 7.2: Registration page of A I Based Interview Evaluator

Figure 7.2 shows the registration page of the AI-Based Interview Evaluator platform. It likely includes fields for users to input their personal information and create an account, along with options for password creation and agreement to terms of service.

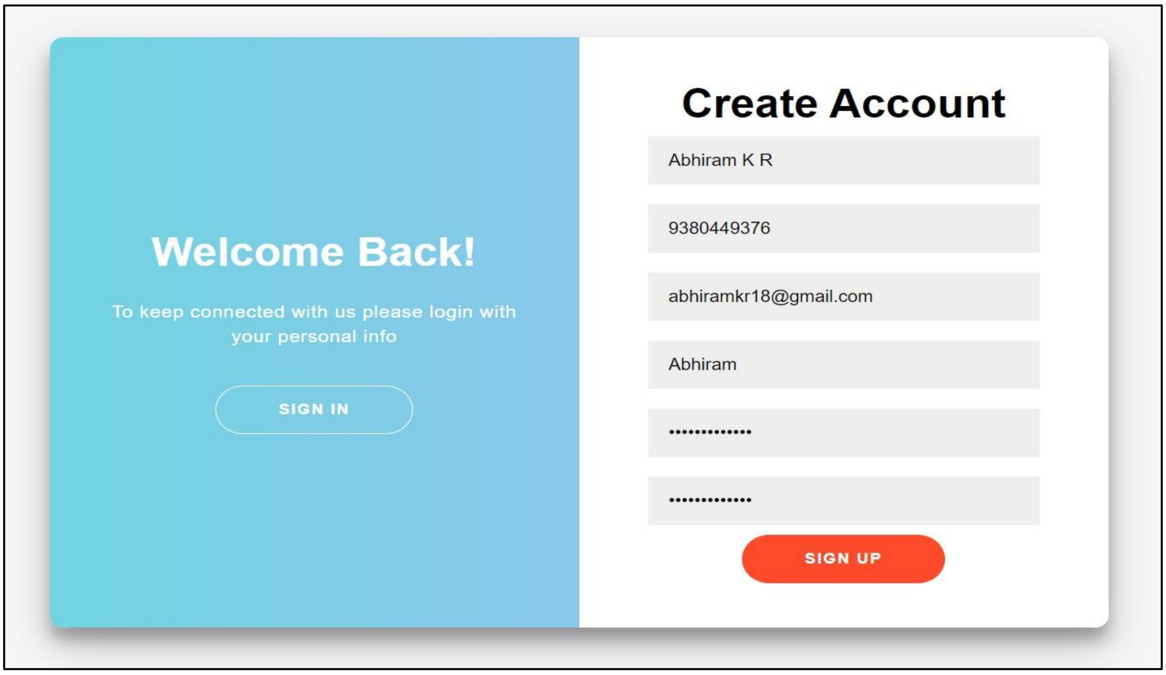


Figure 7.3: Sign up page of A I Based Interview Evaluator

Figure 7.3 shows the sign-up page of an AI-Based Interview Evaluator platform. It likely includes fields for users to input their information and create an account, along with options for password creation and user agreement acceptance. The data of the registered users will be stored in the mongoDb database , and this data will also be accessed and checked while login/signin into the website.

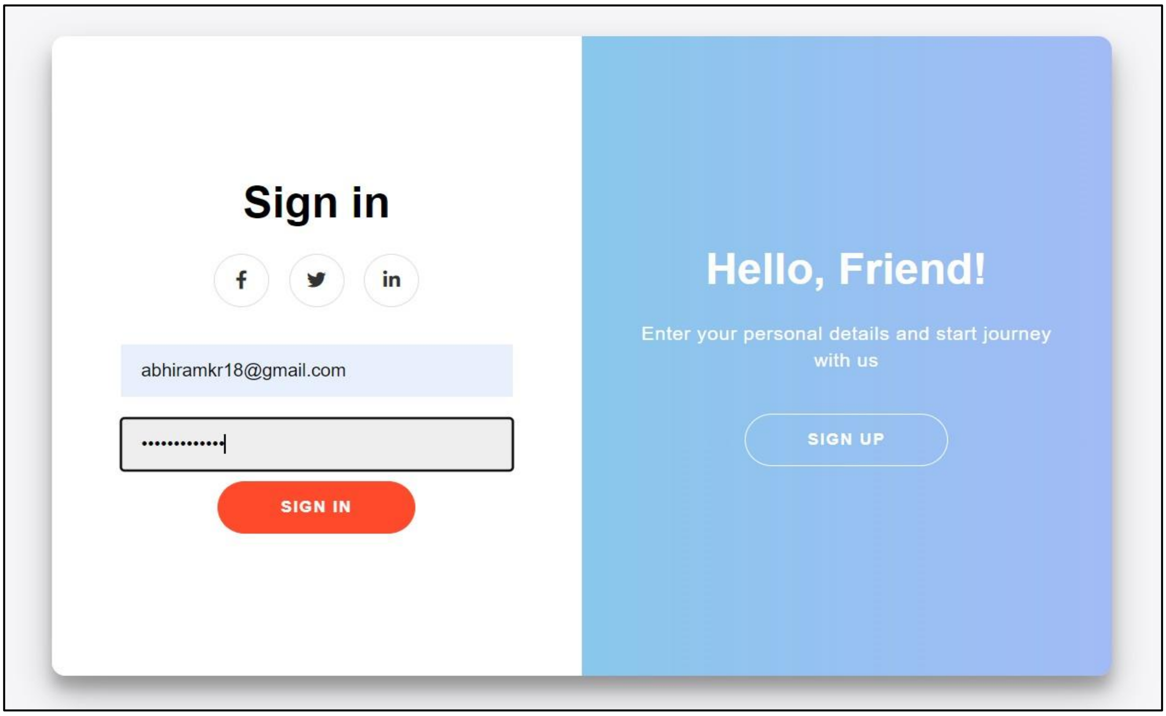


Figure 7.4: Sign in page of A I Based Interview Evaluator

Figure 7.4 shows the sign-in page of an AI-Based Interview Evaluator platform. It probably features fields for users to input their credentials, such as username and password, along with options for password recovery and account creation if users are new to the platform.

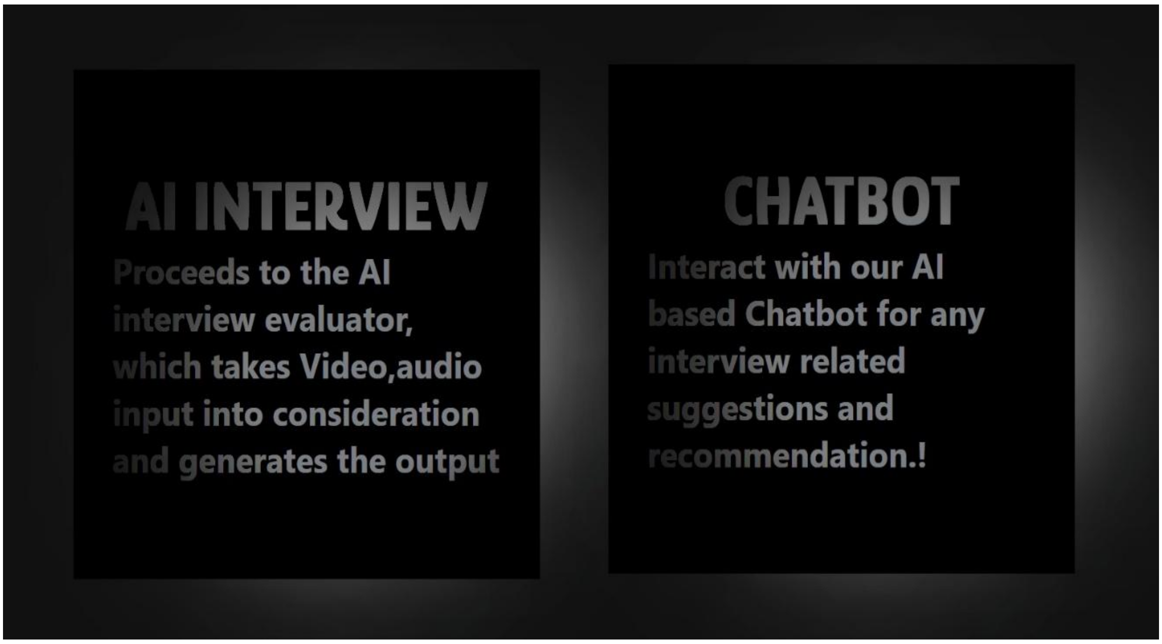


Figure 7.5: Interview and Chatbot selection button

Figure 7.5 represents the interface element for selecting interviews and chatbots within the AI-based Interview Evaluator Platform. This button likely allows users to choose between initiating an interview session or engaging with a chatbot for specific evaluation purposes.

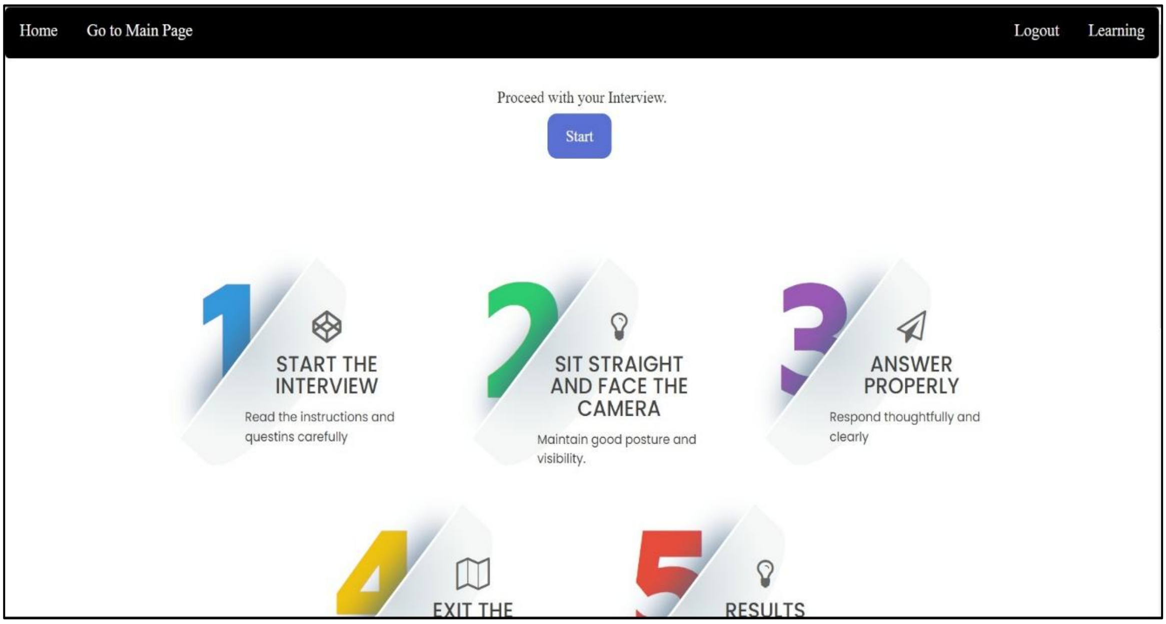


Figure 7.6: StartPage of A I Based Interview Evaluator

Figure 7.6 showcases the start page of the AI-Based Interview Evaluator platform, providing users with options to initiate interviews, access evaluation tools, or navigate sections. It serves as the gateway for users to engage with the platform's functionalities and begin their evaluation journey.

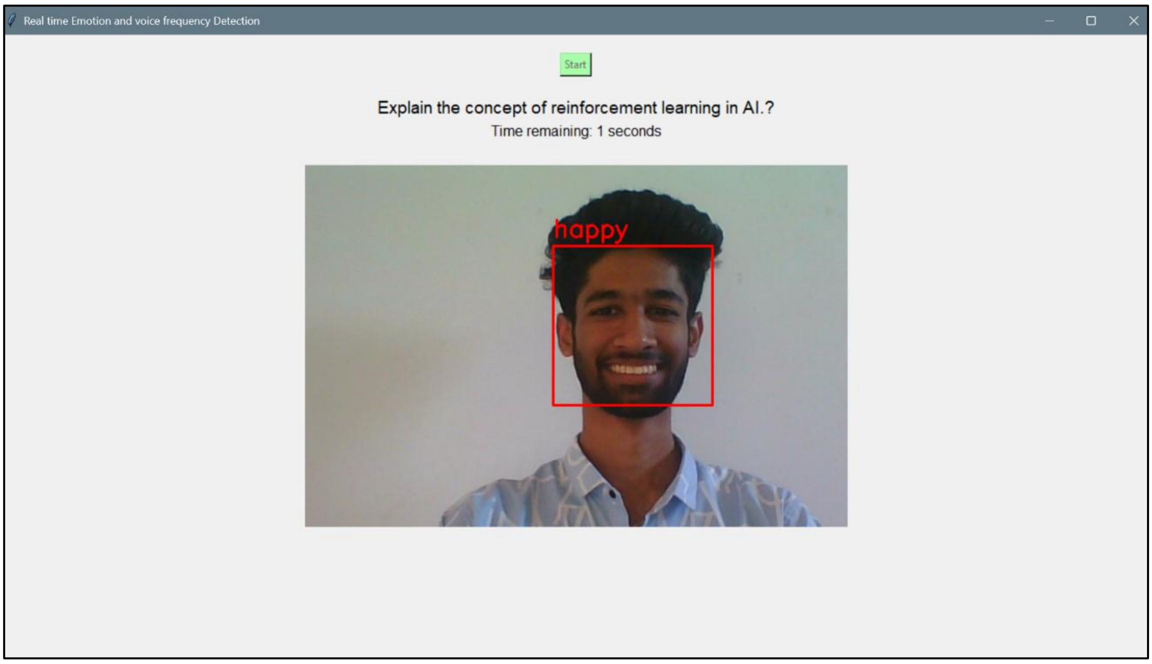


Figure 7.7: Happy emotion detected by Interview Evaluator

Figure 7.7 demonstrates the software's capability to detect a happy emotion. This detection likely indicates the successful recognition of positive emotional cues in a user's expression, showcasing the software's effectiveness in emotion detection.

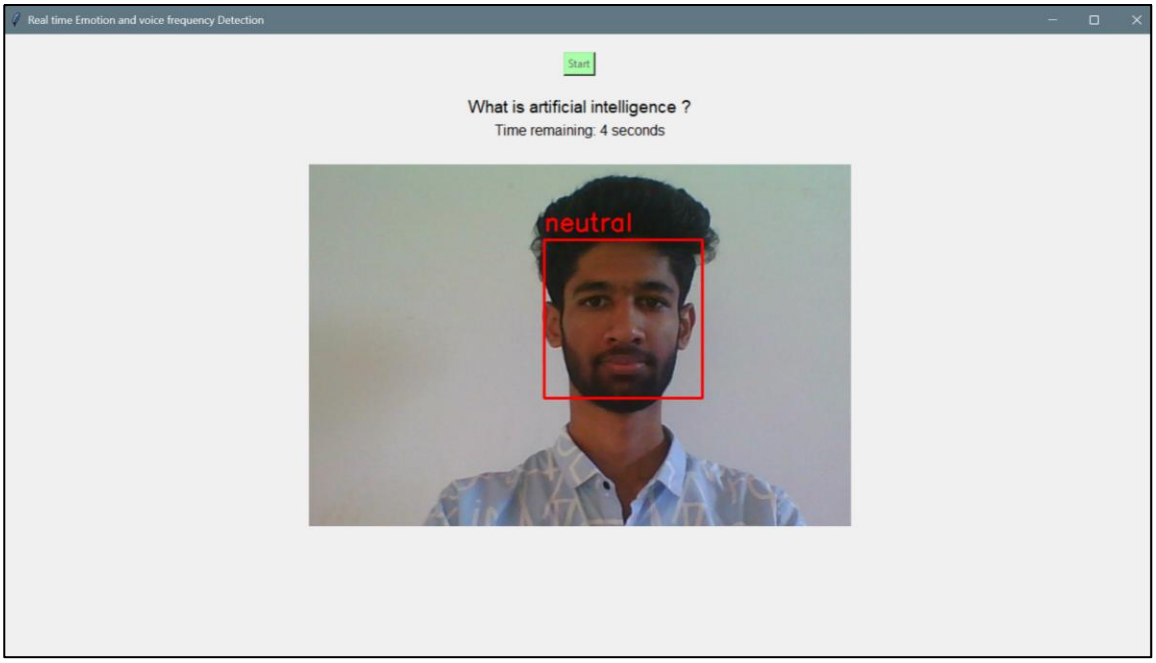


Figure 7.8: Neutral emotion detected by Interview Evaluator

Figure 7.8 indicates the detection of a neutral emotion by the Interview Evaluator software. This detection suggests the recognition of a neutral emotional state in a user's expression, showcasing the software's ability to discern various emotional cues during evaluations.

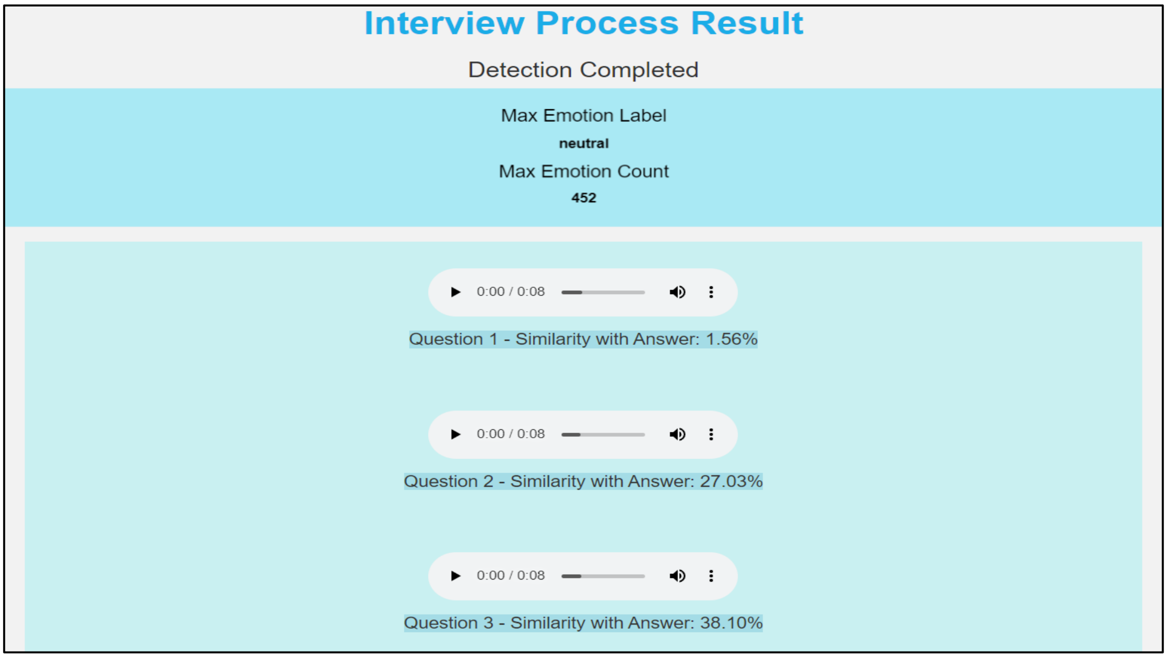


Figure 7.9: Result of A I based Interview Evaluator

Figure 7.9 displays the result of the AI-based Interview Evaluator, providing an assessment of the interviewee's performance. It includes factors such as emotional cues and confidence levels, offering valuable feedback for decision-making and performance improvement.

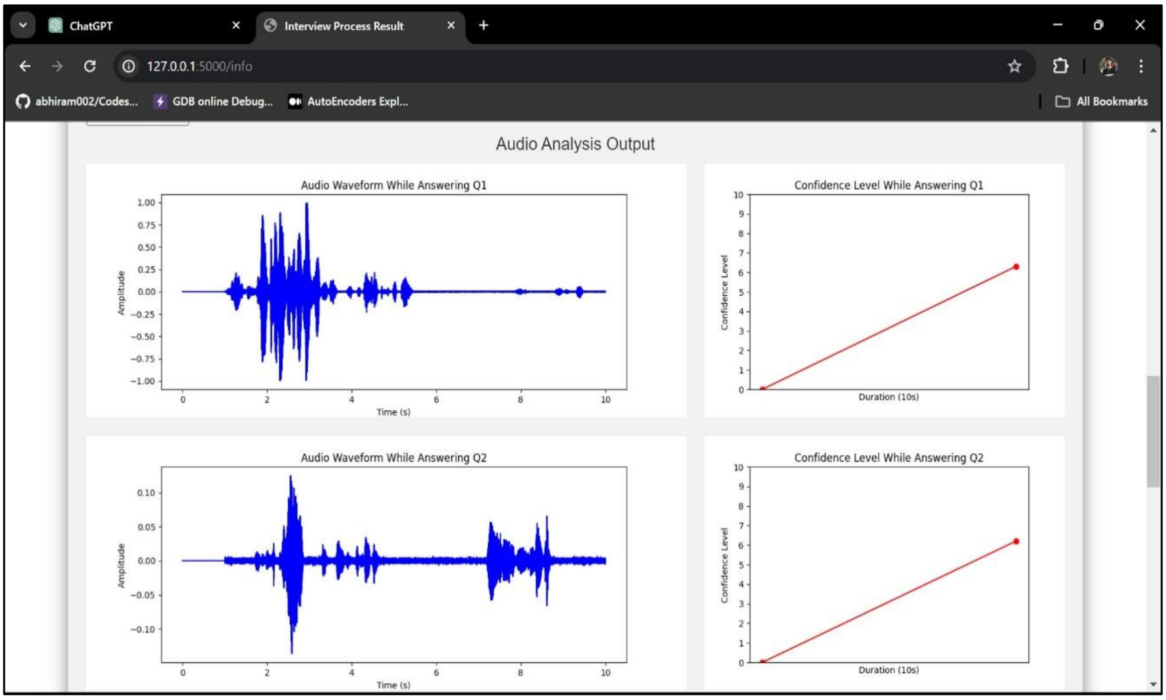


Figure 7.10: Audio analysis output of A I based Interview Evaluator

Figure 7.10 shows the audio analysis output of the AI-based Interview Evaluator, offering verbal feedback on the interviewee's performance. This provides convenient access to evaluation results and additional insights for users.

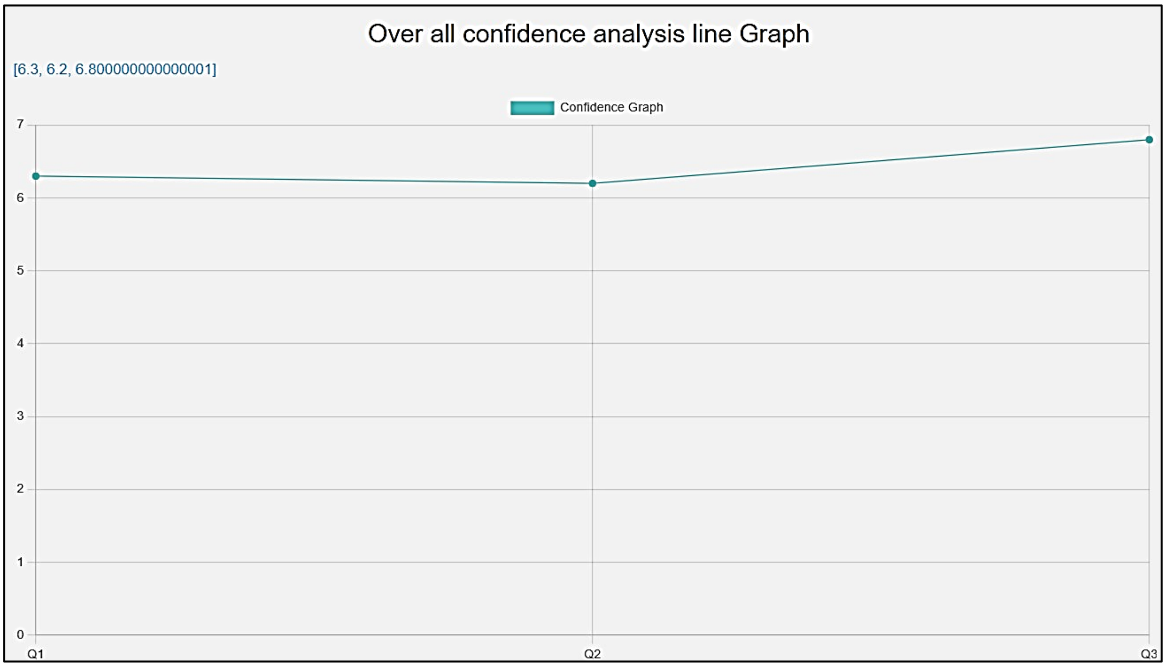


Figure 7.11: Overall confidence analysis graph of A I based Interview Evaluator

Figure 7.11 shows the overall confidence analysis graph generated by the AI-based Interview Evaluator. It provides insights into the interviewee's confidence levels throughout the interview process, aiding in understanding their performance.

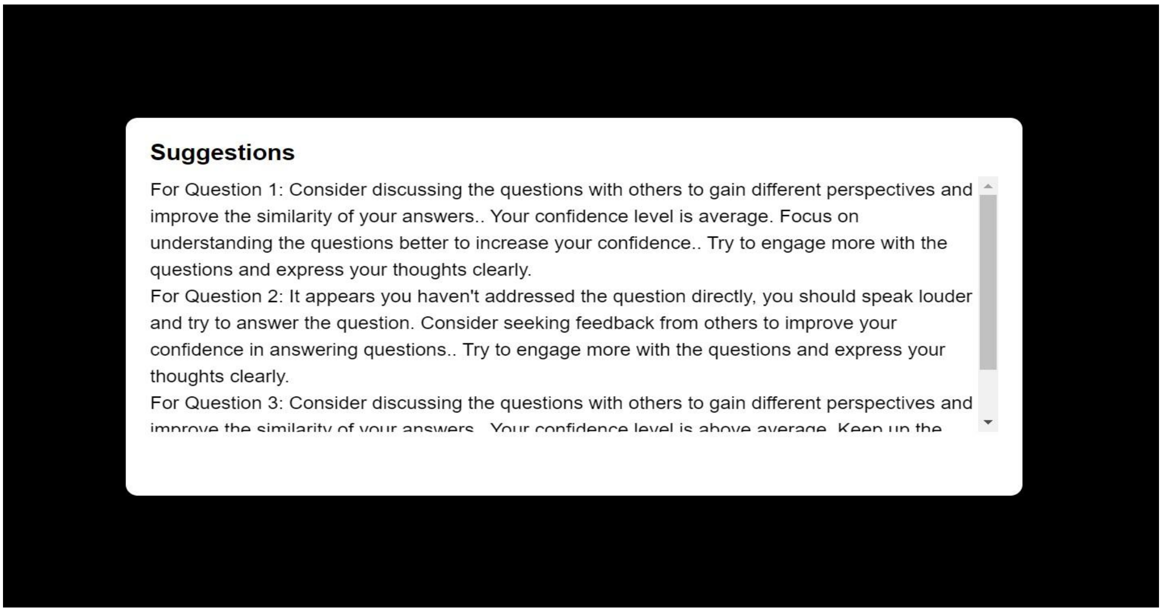


Figure 7.12: Suggestion given by A I based Interview Evaluator

Figure 7.12 shows the suggestions provided by the AI-based Interview Evaluator for each question posed during the interview. These suggestions likely offer guidance or feedback on improving responses, enhancing interview performance. This feature aids interviewees in refining their answers and addressing areas of improvement.

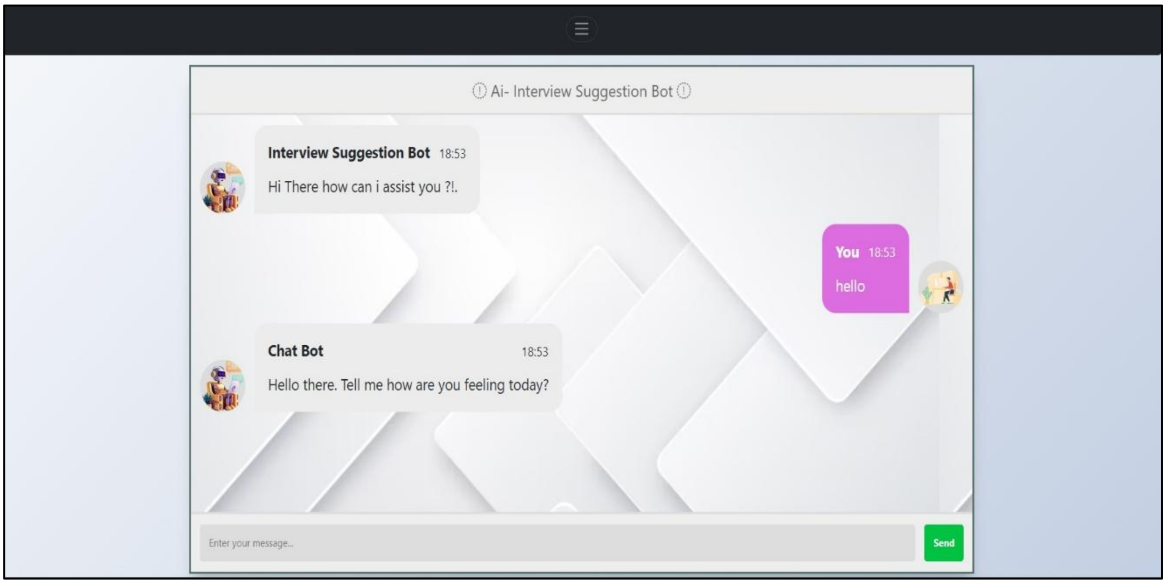
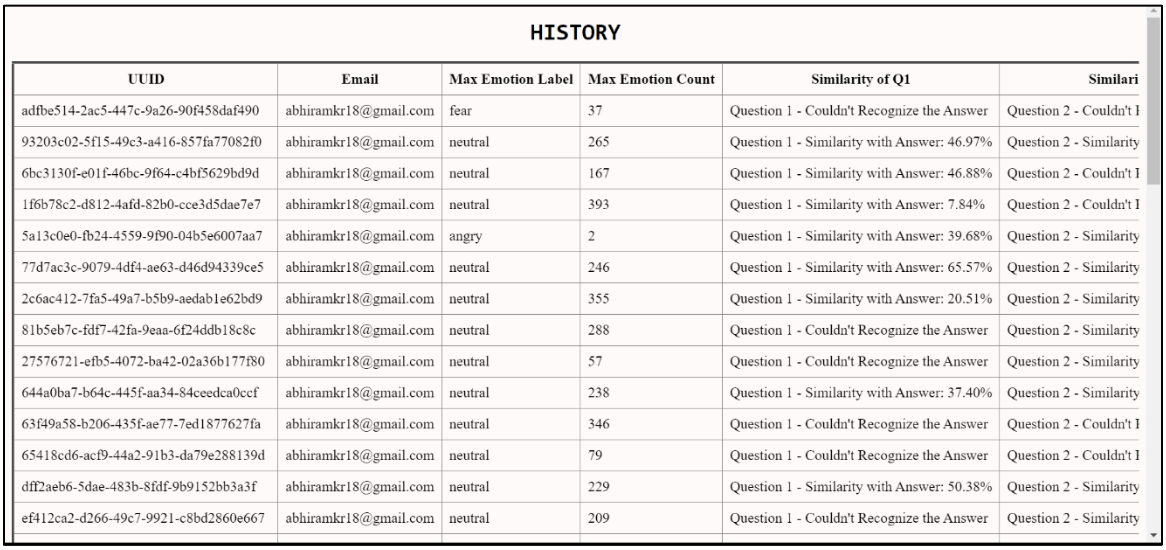


Figure 7.13: Chatbot integrated to A I based Interview Evaluator

Figure 7.13 illustrates the integration of a chatbot into the AI-based Interview Evaluator platform. This feature likely allows users to interact with the chatbot to receive assistance, guidance, or feedback during the interview process. The chatbot serves as a valuable tool for interviewees, providing real-time support and enhancing their overall interview experience.

Figure 7.14: Learning material provided by A I based Interview Evaluator

Figure 7.14 depicts the learning material provided by the AI-based Interview Evaluator. This material likely includes resources such as articles, tutorials, or guides aimed at assisting users in improving their interview skills and performance. By offering educational content, the platform supports continuous learning and development for interviewees.



HISTORY

UUID	Email	Max Emotion Label	Max Emotion Count	Similarity of Q1	Similari
adfbe514-2ac5-447c-9a26-90f458daf490	abhiramkr18@gmail.com	fear	37	Question 1 - Couldn't Recognize the Answer	Question 2 - Couldn't I
93203c02-5f15-49c3-a416-857fa77082f0	abhiramkr18@gmail.com	neutral	265	Question 1 - Similarity with Answer: 46.97%	Question 2 - Similarity
6bc3130f-e01f-46bc-9f64-c4bf5629bd9d	abhiramkr18@gmail.com	neutral	167	Question 1 - Similarity with Answer: 46.88%	Question 2 - Couldn't I
1f6b78c2-d812-4afd-82b0-cce3d5dae7e7	abhiramkr18@gmail.com	neutral	393	Question 1 - Similarity with Answer: 7.84%	Question 2 - Couldn't I
5a13c0e0-fb24-4559-9f90-04b5e6007aa7	abhiramkr18@gmail.com	angry	2	Question 1 - Similarity with Answer: 39.68%	Question 2 - Similarity
77d7ac3c-9079-4df4-ae63-d46d94339ce5	abhiramkr18@gmail.com	neutral	246	Question 1 - Similarity with Answer: 65.57%	Question 2 - Similarity
2c6ac412-7fa5-49a7-b5b9-aedab1e62bd9	abhiramkr18@gmail.com	neutral	355	Question 1 - Similarity with Answer: 20.51%	Question 2 - Similarity
81b5eb7c-fdf7-42fa-9eaa-6f24ddb18c8c	abhiramkr18@gmail.com	neutral	288	Question 1 - Couldn't Recognize the Answer	Question 2 - Similarity
27576721-efb5-4072-ba42-02a36b177f80	abhiramkr18@gmail.com	neutral	57	Question 1 - Couldn't Recognize the Answer	Question 2 - Similarity
644a0ba7-b64c-445f-aa34-84ceedca0ccf	abhiramkr18@gmail.com	neutral	238	Question 1 - Similarity with Answer: 37.40%	Question 2 - Similarity
63f49a58-b206-435f-ae77-7ed1877627fa	abhiramkr18@gmail.com	neutral	346	Question 1 - Couldn't Recognize the Answer	Question 2 - Couldn't I
65418cd6-acf9-44a2-91b3-da79e288139d	abhiramkr18@gmail.com	neutral	79	Question 1 - Couldn't Recognize the Answer	Question 2 - Couldn't I
dff2aeb6-5dae-483b-8fdf-9b9152bb3a3f	abhiramkr18@gmail.com	neutral	229	Question 1 - Similarity with Answer: 50.38%	Question 2 - Similarity
ef412ca2-d266-49c7-9921-c8bd2860e667	abhiramkr18@gmail.com	neutral	209	Question 1 - Couldn't Recognize the Answer	Question 2 - Similarity

Figure 7.15 : History of the current interviewee

The history page of the interview evaluator tool shows a record of previous interviews conducted using the system. Each row in the table corresponds to a single interview UUID is a unique identifier that is assigned to each interview. Email is the email address of the person who was interviewed. Max Emotion Label indicates the most prominent emotion that was detected during the interview. Possible emotions include fear, neutral, angry, etc. Max Emotion Count shows how many times the most prominent emotion was detected during the interview. Similarity of Q: This indicates how similar the user's response to the first question was to a predefined answer. Confidence level indicates the interviewee's confidence level for each question.

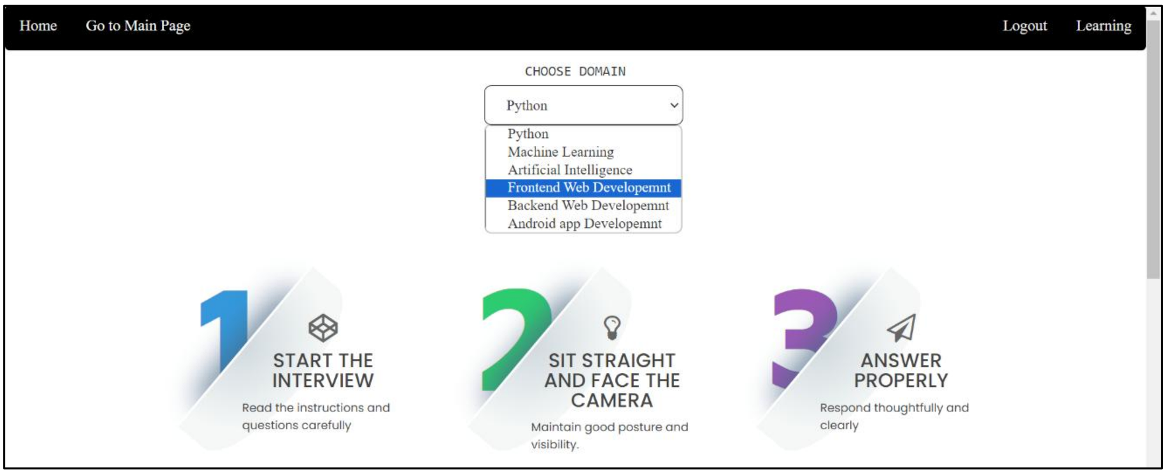


Figure 7.16: Choose Domain Section

The above figure 7.16 shows that we can choose the domain for the interview. The domain you choose will determine the questions that are generated for the interview. Here are the domains listed on the page:

- Python
- Machine Learning
- Artificial Intelligence
- Frontend Web Development
- Backend Web Development
- Android App Development

To select a domain, click on the desired domain name. Once you have selected a domain, you can then click the "START THE INTERVIEW" button to begin the interview process.

Figure 7.17: Login details of user stored in database

Figure 7.17 delves into the critical aspect of user login details within the AI-based Interview Evaluator platform. This database storage mechanism ensures secure access by registered users. Usernames, passwords, and potentially other authentication data are meticulously stored, enabling the platform to verify user identities and grant appropriate access levels. This secure storage plays a vital role in maintaining platform integrity and safeguarding user information. Additionally, storing login details within a database allows for efficient user management and streamlined access control processes.


```
_id: ObjectId('6623823a3b58e89471485fb0')
uuid: "72500b3d-7ad7-4c35-be1d-8f0041aa1625"
email: "ppp@gmail.com"
max_emotion_label: "neutral"
max_emotion_count: 166
similarity of Q1: "Question 1 - Couldn't Recognize the Answer"
similarity of Q2: "Question 2 - Similarity with Answer: 65.93%"
similarity of Q3: "Question 3 - Similarity with Answer: 56.79%"
quiz_time: "08:52:10"
timestamp: "2024-04-20T14:22:10.297696"
confidence level from Q1: 5.4
confidence level from Q2: 6.4
confidence level from Q3: 6.800000000000001
```


Figure 7.18: Results stored in MongoDB database

Figure 7.18 demonstrates the storage of results in a MongoDB database. This likely includes data such as timestamps, user identifiers, and specific emotional analysis details, facilitating further analysis and retrieval of information. The use of MongoDB allows for efficient management and retrieval of emotion-related data within the Interview Evaluator platform.

The storage of results related to neutral emotions in a MongoDB database likely includes data such as timestamps, user identifiers, and specific emotional analysis details, facilitating further analysis and retrieval of information. For instance, the record for user "ppp@gmail.com" shows a neutral emotional label with a count of 166. This could indicate that the user remained composed throughout most of the interview. Additionally, timestamps associated with emotional data can be used to track emotional trends over the course of an interview. This emotional data can then be used to improve the effectiveness of the interview evaluator by identifying areas where users struggle and tailoring the interview experience accordingly. The use of MongoDB, a NoSQL database known for its scalability and flexibility, allows for efficient management and retrieval of emotion-related data within the Interview Evaluator platform, even as the volume of data grows.

CHAPTER 8

FUTURE WORK

Future work for the AI based interview evaluator: an emotion and confidence classifier model may include several avenues of improvement and expansion

8.1 Future work

- **Multimodal Analysis**: Incorporate additional modalities such as body language, gestures, and tone of voice analysis to provide a more comprehensive understanding of the candidate's nonverbal cues and overall demeanor during the interview.
- **Personalized Interview Customization**: Develop a system to tailor the interview questions and scenarios based on the candidate's background, experience, and the specific job requirements, ensuring a more personalized and relevant evaluation process.
- **Integration with Existing HR Systems**: Develop seamless integration with existing human resources management systems and applicant tracking systems to streamline the recruitment process and provide a more cohesive candidate experience.
- **Explainable AI**: Implement techniques to make the system's decision-making process more transparent and explainable, providing insights into the reasoning behind the evaluations and recommendations.
- **Privacy and Ethics Considerations**: Continuously assess and address privacy and ethical concerns related to the use of AI in hiring processes, ensuring compliance with relevant regulations and adhering to best practices for responsible AI.
- **Multi-language Support**: Extend the system to support multiple languages for speech recognition, text analysis, and chatbot interactions, enabling a more inclusive and diverse recruitment process.
- **Virtual Reality (VR) Integration**: Explore the potential of integrating virtual reality technologies to create immersive and realistic interview scenarios, allowing for more comprehensive evaluation of candidates' skills and behaviors in simulated environments.
- **Gamification and Skill Assessment**: Incorporate gamification elements and interactive skill assessments into the interview process to evaluate candidates' problem-solving abilities, critical thinking, and decision-making skills in a more engaging and interactive manner.

CHAPTER 9

CONCLUSION

The AI-based interview evaluator system represents a significant step towards revolutionizing the traditional job interview process. By leveraging cutting-edge technologies in machine learning, computer vision, and natural language processing, the system offers an objective and data-driven approach to candidate evaluation. Through the integration of facial emotion recognition, speech-to-text conversion, confidence level prediction, and similarity analysis, the system provides comprehensive insights into a candidate's emotions, knowledge, and overall preparedness during the interview.

The system's ability to eliminate inherent biases and subjectivity in the evaluation process ensures fairness and consistency, enabling organizations to make more informed and unbiased hiring decisions. Additionally, the incorporation of a neural network-based chatbot enhances the overall interview experience, making it more interactive and engaging for candidates. While the current implementation demonstrates the potential of AI in the hiring process, there are areas for further improvement and expansion. Continuous refinement of the underlying models, incorporation of additional modalities such as body language analysis, and seamless integration with existing HR systems can further enhance the system's capabilities. Moreover, addressing ethical concerns related to privacy, bias, and transparency should be a priority to ensure the responsible deployment of such AI-based systems.

Overall, the AI-based interview evaluator project represents a significant advancement in the field of talent acquisition, paving the way for more efficient, fair, and data-driven hiring practices. By combining cutting-edge technologies with a focus on ethical considerations, this project has the potential to transform the way organizations identify and acquire top talent, ultimately driving organizational success and growth.

REFERENCES

[1] Sahil Temgire , Akash Butte , Rohan Patil , Varun Nanekar, Shivganga Gavhane, 2021, Real Time Mock Interview using Deep Learning, International Journal Of Engineering Research & Technology (Ijert) Volume 10, Issue 05 (May 2021).

[2] Dulmini Yashodha Dissanayake, Venuri Amalya, Raveen Dissanayaka, Lahiru Lakshan,2021, AI-based Behavioural Analyser for Interviews/Viva, The Journal of Machine Learning Research, vol. 10, pp. 1755–1758.

[3] Laiq Muhammad & Dieste Tubío, Óscar. (2020). Chatbot-based Interview Simulator: A Feasible Approach to Train Novice Requirements Engineers. 1-8. 10.1109/REET51203.2020.00007.

[4] Ionut Cosmin Duta, Li Liu ,Fan Zhu and Ling Shao Improved Residual Networks for Image and Video Recognition, (2021) International Conference on Pattern Recognition (ICPR) Milan, Italy, pp. 37-41, doi: 10.1109/ICIET55102.2022.9778990.

[5] Ashlin Deepa R N, Prathyusha Karlapati,Mrunhaalhini Reddy Mulagondla, Pavitra Amaranayani, An Innovative Emotion Recognition and Solution Recommendation Chatbot, March 2022, ICACCS, 10.1109/ICACCS54159.2022.9785269.

[6] R. Chinmayi, N. Sreeja, A. S. Nair, M. K. Jayakumar, R. Gowri and A. Jaiswal, "Emotion Classification Using Deep Learning," *2020* Third International Conference on Smart Systems and Inventive Technology *(ICSSIT)*, Tirunelveli, India, 2020, pp. 1063-1068, doi: 10.1109/ICSSIT48917.2020.9214103.

[7] de Andrade, Guilherme & Silva, Geovana & Duarte, Francisco & Santos, Giovanni & Javidi da Costa, Joao Paulo & de Sousa Junior, Rafael. (2022). A Conversation-Driven Approach for Chatbot Management. IEEE Access. PP. 1-1. 10.1109/ACCESS.2022.3143323.

[8] Adepu, Yashwanth & Boga, Vishwanath & Utukuru, Sairam. (2020). Interviewee Performance Analyzer Using Facial Emotion Recognition and Speech Fluency Recognition. 1-5. 10.1109/INOCON50539.2020.9298427.

[9] Pathar, Rohit & Adivarekar, Abhishek & Mishra, Arti & Deshmukh, Anushree. (2019). Human Emotion Recognition using Convolutional Neural Network in Real Time. 1-7. 10.1109/ICIICT1.2019.8741491.

[10] Ruhul Amin Khalil , Edward Jones , Mohammad Inayatullah Babar, Tariqullah Jan ,Mohammad Haseeb Zafar , And Thamer Alhussain6, Speech Emotion Recognition using Deep Learning Techniques,2019.

[11] Senarathne, Pasindu & Silva, Malaka & Methmini, Ama & Kavinda, Dulaj & Thelijjagoda, Samantha. (2021). Automate Traditional Interviewing Process Using Natural Language Processing and Machine Learning. 1-6. 10.1109/I2CT51068.2021.9418115.

[12] Idris, N., Mohd Foozy, C. F., & Shamala, P. (2020). A Generic Review of Web Technology: DJango and Flask. International Journal of Advanced Computing Science and Engineering, 2(1), 34-40. ISSN 2714-7533.

[13] L. Wang, T. Liu, B. Wang, J. Lin, X. Yang and G. Wang, Object Tracking Via ImageNet Classification Scores, 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 2101-2105, 10.1109/ICIP40778.2020.9191129.

[14] R. Mandal, P. Lohar, D. Patil, A. Patil and S. Wagh, "AI -Based mock interview evaluator: An emotion and confidence classifier model," 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 521-526, 10.1109/ICISCoIS56541.2023.10100589.

[15] Paweł Tarnowski, Marcin Kołodziej, Andrzej Majkowski, Remigiusz J. Rak, 2017, Emotion recognition using facial expressions, International Conference on Computational Science, ICCS 2017, 12-14 June 2017.

[16] Swayam Badhea , Sangita Chaudhari, Deep Learning based Facial Emotion Recognition, ITM Web of Conferences 44, 03058 (2022).

[17] James Pao, Emotion Detection Through Facial Feature Recognition, International Journal of Multimedia and Ubiquitous Engineering 12(11):21-30, 10.14257/ijmue.2017.12.11.03.