

INTERVIEW WITH AI USING MACHINE LEARNING

Submitted in partial fulfillment of the required for the award of

Masters of Computer Science

By

MARY PRIYA A

(REGISTER NO :42734010)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024



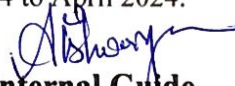
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A++" by NAAC

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
MARY PRIYA A (42734010) who carried out the project entitled
"INTERVIEW WITH AI USING MECHINE LEARNING" under our
supervision from January 2024 to April 2024.


Internal Guide

Ms. AISHWARYA B M.C.A


Dean And Head Of The Department

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on 20.04.2024


Internal Examiner


External Examiner

DECLARATION

I am, **MARY PRIYA A (42734010)** hereby declare that the Project Report entitled **“INTERVIEW WITH AI USING MECHINE LEARNING”** done by us under the guidance of **Ms. AISHWARYA B M.C.A** Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfilment of the requirements for the award of Master of Science degree in Computer Science.

DATE: 20.04.2024

PLACE: CHENNAI

Mary Priya .A
MARY PRIYA A

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Ms.AISHWARYA B M.C.A**Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project

MARY PRIYA A

ABSTRACT

Interview process of the bigger organization with more candidates is very tedious. Sometimes the shortlisted employees for the final face to face interview are more. So the HR has to handle each candidate and it is time consuming process. So Digitalization and artificial intelligence (AI) are a growing part of everyday lives. The aim of this project is to review the artificial process automation in the recruitment of employee for bigger organization and its practical implications. Face recognition machine learning technique is used to detect the status of the candidate confident level and the fear for the interview. If the concept is strong then the candidate will be very confident and the level of fear is also low so it is important to identify the confident and fear of the candidate. So machine learning techniques are used to give the training dataset of the confident and fear and test the candidate confident by using AI Classification algorithm. For face recognition HAAR cascade algorithm is used to detect the emotional state of the candidate and Recurrent Neural Network (RNN) is used to classify the correct answers of the candidates in interview. This project discussed possible applications for the recruitment industry caused due to the adoption of Artificial intelligence process automation in the recruitment process. Here the Machine learning applications like the face recognition, emotional analysis and NLP text processing is used to take the interview process. This project can be done by using python t-kinder framework with MYSQL back end connectivity.

Chapter No	TITTLE	Page No
	ABSTRACT	i
1	INTRODUCTION	1
1.1	AI BASED RECRUITMENT	1
1.2	RECRUITMENT PROCESS	1
1.3	AI POWERED SYSTEMS	2
1.4	MODELS OF REQUIREMENT	2
1.5	IMPLICATION	3
2	LITERATURE SURVEY	4
3	SYSTEM DESIGN	8
4	SYSTEM ARCHITECTURE	13
5	SYSTEM IMPLEMENTATION	15
5.1	MODULES	15
5.2	MODULES DESCRIPTION	15
6	CODING AND SCREENSHOT	19
6.1	SAMPLE CODING	19
6.2	SAMPLE SCREENSHOT	34
7	CONCLUSION AND FUTURE WORK	47
7.1	CONCLUSION	47
7.2	FUTURE ENHANCEMENT	47

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO
3.1	USE CASE DIAGRAM	9
3.2	SEQUENCE DIAGRAM	10
3.4	ACTIVITY DIAGRAM	11
4.1	SYSTEM ARCHITECTURE	14
5.1	CLASSIFIER SYSTEM	16
5.2	HAAR CASCADE FACE DETECTION	17
6.1	HOME PAGE	34
6.2	ADMIN LOGIN	35
6.3	COMPANY HOME	36
6.4	COMPANY ADD JOB DETAILS	37
6.5	COMPANY QUESTIONS TO BE ASKED	38
6.6	COMPANY VIEW CANDIDATE DETAILS	39
6.7	USER REGISTRATION	40
6.8	USER LOGIN	40
6.9	USER HOME PAGE	41
6.10	USER JOB SEARCH	42
6.11	JOB TITLE	43
6.12	STAFF INTERVIEW START	44
6.13	EMOTION RECOGNITION WITH ANSWERING SYSTEM	45
6.14	SCORE ANALYSIS	46

1.INTRODUCTION

1.1 AI BASED RECRUITMENT

For a rapidly changing business environment with less defined and unstable jobs, the conventional recruitment methods based on psychometric principles might no longer be effective. Additionally, traditional sourcing methods such as job applications and printed mediums are shrinking in favor of internet sources and e-recruitment where the variation in the candidate accessibility skill is significantly declining. Human resources (HR) is one of the most important assets of an organization as highly skilled, experienced, and qualified employees play a vital role in the growth, competitive advantage, and innovative transformation of an organization. That is why organizations compete to acquire the right skills and qualifications, investing a lot of time, money, and effort to get and retain the best candidates.

1.2 RECRUITMENT PROCESS

Recruitment is the process of searching and selecting the right candidate for a job which starts with analyzing the job requirements by the organization, then attracting the potential candidates by announcing the job positions followed by screening and selecting the candidates and finally hiring and welcoming the hired ones into the organization. Continuous innovation in HR and recruitment is shaping the future of HR and as a result, AI-based solutions are being adopted to overcome any challenges. The increased complexity of the new technological content and its association with the rapidly changing business environment has changed the conventional method of employee selection by making it more technologically dependent. In the area of recruitment, AI has been strengthening its position by increasing the productivity of employees, reducing cost and time, and eliminating human errors and biases. It helps to automate the candidate sourcing and search, matching the candidate with the employer, real-time communication with the applicants, providing feedback to the recruiters of the applicant's experience, interviewing, facial expression analysis, candidate selection, and diverse hiring

1.3 AI POWERED SYSTEMS

AI-powered systems are programmed to avoid unconscious biases like age, gender, race, or names. Unconscious human bias in recruitment occurs when an individual or a group of individuals gets a favor based on age, race, color, education, or ethnicity. Many studies, have been conducted on human biases that may affect hiring decisions, reviewed 49 studies on gender biases and concluded that during the hiring process, recruiters tend to bias against female candidates. In their study about bias against ethnic groups held that white names candidates receive 50 percent more calls for interviews than for African-American names. AI eliminates errors and biases associated with humans during the screening and selection of candidates. The algorithm used in AI captures the applicant's characteristics to construct a profile that predicts if the person has the required competency and expertise that the company is looking for irrespective of age, gender, ethnicity, and race, hence revolutionizing recruitment and creating diversified, rapport and vigorous workplaces.

1.4 MODELS OF RECRUITMENT

Several models for the recruitment process have been mentioned by different authors. In one of the recruitment process models proposed by the author focused on both the organization and the applicant, perspectives. The first three-steps of the model describe the organization's perspective, fourth steps involve in the applicants' view and the final step is to capture the whole recruitment process. The first step for this model is to set some recruitment objectives like attracting suitable candidates according to the job description to fill the empty position. The second step is to develop a strategy of recruitment by generating some questions in mind such as whom to recruit, what is nature of the job offer, what is the budget constraints, and so on. The third step is recruitment activities which require different types of activities posting jobs in specific sites, extend the recruitment time if needed; selecting the recruiter's types, and basically, it involves the types of recruitment methods that will use in this step. The fourth step of the model is to observe the interest of job applicants such as the interest of accepting the job, expectation from the job, applicant self insights, and also the decision-making process of the applicant. And the fifth and final step is to evaluate the whole recruitment results by going back to the recruitment objective

and measuring the outcome of the results which help to learn from experiences and improve the effectiveness of future recruitment processes

1.5 IMPLICATION

Two relevant algorithms are used in the detector engagement system: the Haar Cascade Algorithm; and a Convolution Neural Network.

a) Haar Cascade Algorithm

The Viola & Jones object detection algorithm uses the so-called Haar Cascade algorithm to extract features from images in a rapid and efficient way, and it had become one of the most popular methods currently being used to this end. It needs a lot of positive and negative images to be trained for the cascade function to work properly.

After getting trained, it can detect other images according to the previous training. In this work we use this algorithm to detect the student's frontal face in the image and locate the eyes region within the face.

b) Convolution Neural Network (CNN)

CNN are distinct from traditional Artificial Neural Networks because they have the ability to encode relevant image features directly from the raw input images, making them more efficient to implement and reducing the number of parameters in the network. In this work a CNN was trained on eye images to detect if the student is facing the webcam ("Focused") or not ("Distracted"), performing a binary classification in these two categories.

2.LITERATURE SURVEY

2.1 TITLE: An Automated System for the Assesment of Interview Performance through Audio & Emotion Cues

AUTHOR: P.KalaivaniS. Mohamed Mansoor Roomi, K.Priya, M.G.Sethuraman, P.Shanmugavadivu

DESCRIPTION:

Here the research defines about the Automatic analysis and performance evaluation of interviewees is a largely unexplored and challenging problem. The proposed work provides a computational structure to enumerate the performance of the interviewee in the context of communication to give their performance feedback from the analysis of multimodal signals such as facial images and voice. A video captured during the interview is split into audio and visual frames. From the visual frames, the face is detected and their emotions are analyzed with the help of Histogram of Oriented Gradients. The facial expressions are classified by using Support Vector Machine. The classified facial expressions are happy, fear, sad, neutral, surprise, disgust and angry. From the audio cues,, the Frequency Coefficient features are extracted and this is categorized as fluent, non-fluent-pause and non-fluent-stammer. Both emotion and fluency of the candidate are fused to find performance score. This automated analysis provides the ratings for interview behavior such as poor, medium, high performance.

2.2TITTLE Toward Personalized Emotion Recognition: A Face Recognition Based Attention Method for Facial Emotion Recognition

AUTHOR: Jin Tang; Jiani Li, Mostafa Shahabinejad and Yang Wang; Yuanhao Yu.

DESCRIPTION:

This paper aims to address the subject-dependent challenge of the facial emotion recognition (FER) task. To accomplish this, we propose a novel face recognition based attention

FER (FRA-FER) framework which propagates subtle face recognition (FR) features through the FER network. Particularly, first a spatial attention map from the feature maps of an FR convolutional neural network (CNN) is created and then it is fused into the FER-CNN. By doing this FR feature propagation, the FER network is personalized as it takes the advantage of the FR features learned from large-scale face recognition datasets. Experiments on the two challenging datasets Affect Net and AFEW demonstrate the superiority of our proposed FRA-FER network to the state-of-the-art work.

2.3TITLE: APPLICATION OPTIMIZATION OF NLP SYSTEM UNDER DEEP LEARNING TECHNOLOGY IN TEXT SEMANTICS AND TEXT CLASSIFICATION

AUTHOR: Gang Jin

DESCRIPTION:

Gang jin describes about How to use advanced deep learning technology to build an effective and powerful text classification model, extract text semantic attributes and achieve good classification results on large-scale test datasets has always been the main problem and severe challenge of text classification tasks. The purpose of this paper is to study the application optimization of NLP system in text semantics and text classification based on deep learning technology. A WC-GCN classification algorithm combining GCN and Attention mechanism is proposed. The related tasks of text classification are discussed, including three typical text classification tasks, including sentiment analysis, topic analysis, and news classification. The algorithm model proposed in this paper is verified from the experimental point of view. Compared with other deep learning-based text classification methods, the proposed algorithm model performs better on the public datasets of three text classification tasks. The news text also achieved a high accuracy rate.

2.4TITLE: RELATIONSHIPS BETWEEN BEHAVIORS BY APPLICANTS AND INTERVIEWER IN INTERVIEW

AUTHOR: Eiji Watanabe, Takashi Ozeki and Takeshi Kohama

DESCRIPTION:

In interviews, the evaluation by the interviewer is affected by not only the reply but also behaviors (eye and face movements) by applicants. Moreover, the evaluation by the interviewer is communicated with applicant as the behavior by the interviewer. Therefore, the interaction between the behaviors by the interviewer and the applicant occurs. In this paper, we show that the evaluation by the interviewer is strongly influenced by behaviors by applicants and the relationship between the evaluation by the interviewer and the behaviors by applicants.

2.5 TITLE: Face Detection and Recognition Using Machine Learning

AUTHOR: Dibya jyoti Bora, Kaberi Kakoty, Raktim Nath and Udari Welipitiya

DESCRIPTION:

Face recognition has been a rapidly growing and intriguing region progressively applications. A huge number of face recognition calculation have been produced in a long time ago. In this paper, for face detection we are using HOG (Histogram of oriented Gradient) based face detector which gives more accurate results rather than other machine learning algorithms like Haar Cascade. In recognition process we are using CLAHE (Contrast Limited Adaptive Histogram equalization) for preprocessing than we are using HOG which is a standard technique for features extraction. HOG features are extracted for the test image and also for the training images. And finally for classification we are using SVM (support vector machine). SVM will classify the HOG features. The result of this paper show the liability and productiveness in better face recognition performance.

2.6 TITLE: Human Emotion Detection using Machine Learning Techniques

AUTHOR: Ameer Shathali M, Inba S,Pavithra K.S and Punidha Angusamy

DESCRIPTION:

Image processing is a method to convert an image into digital form and perform some operations on it. This is done to enhance the image or to extract useful information from it. Facial expressions are nonverbal form of communication. There are eight universal facial

expressions which include: neutral, happy, sadness, anger, contempt, disgust, fear, and surprise. So it is very important to detect these emotions on the face. A monitoring system for elderly people which is based on technology involving recognizing emotions from video image. Our proposed system includes video analysis technology which includes the data from video is adopted to realize monitoring elders' living conditions in real time. In case of emergency, the system will alert their relatives and children by sending a message

3.SYSTEM DESIGN

3.1 UML DIAGRAMS

A UML (Unified Modeling Language) diagram is a graphical representation of a system or process using standardized symbols and notation. UML diagrams are widely used in software engineering and other fields to visually depict various aspects of a system, such as its structure, behavior, interactions, and architecture.

There are several types of UML diagrams, each serving a different purpose and focusing on different aspects of the system being modeled. Some common types of UML diagrams include:

Unified Modeling Language (UML) diagram for the proposed automated college timetable scheduling system. Here's an example of a high-level UML class diagram for the system:

3.1.1 Use Case Diagram

A Use Case Diagram is a type of UML (Unified Modeling Language) diagram that illustrates the interactions between actors (users or external systems) and a system under consideration. It provides a high-level view of the functionalities or features of a system from the perspective of its users. Use case diagrams are often used during the requirements analysis phase of software development to capture and communicate the system's behavior and requirements.

This diagram provides a high-level overview of the interactions between users, administrators, and the animal detection system, as well as the main functionalities supported by the system.

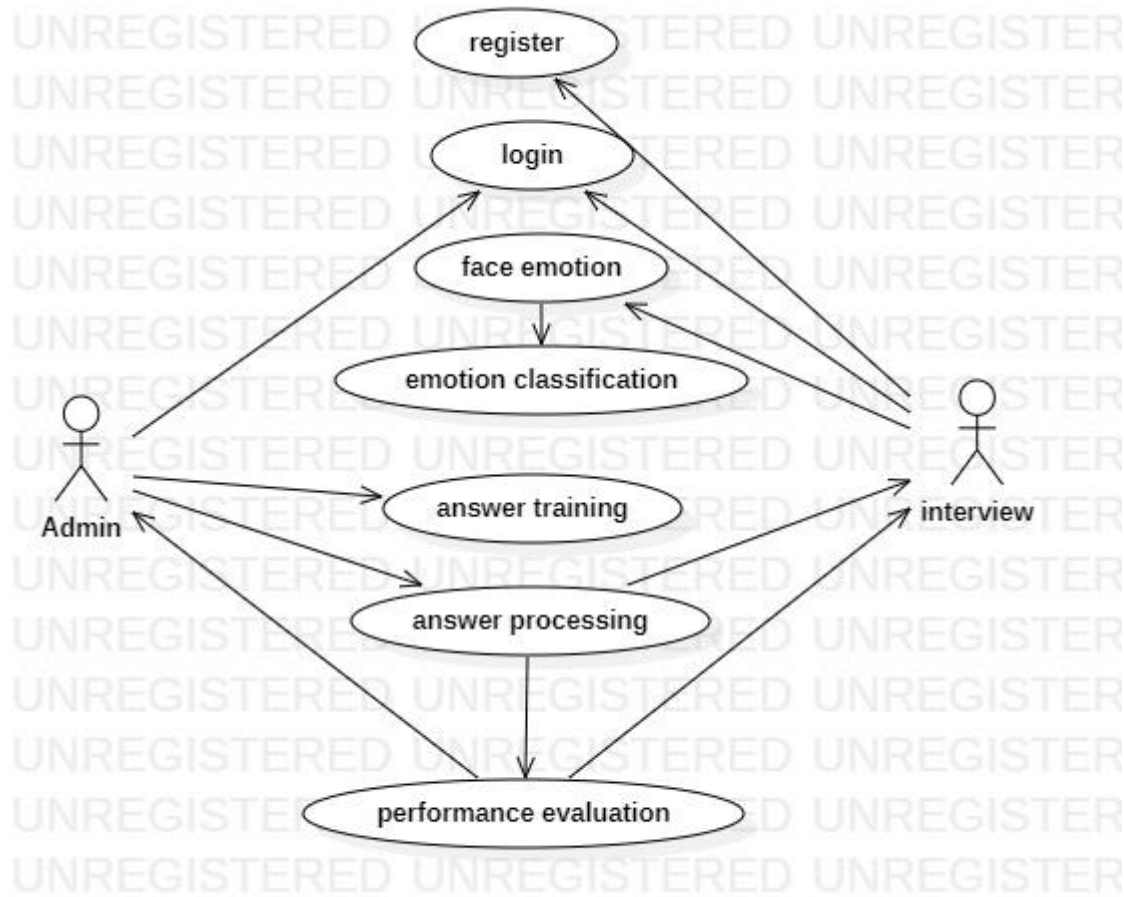


Fig 3.1 Use case diagram

3.1.2 Sequence Diagram

A Sequence Diagram is a type of UML (Unified Modeling Language) diagram that depicts the interactions between objects or components within a system over time. It illustrates the sequence of messages exchanged between these objects to accomplish a specific behavior or scenario. Sequence diagrams are particularly useful for visualizing the flow of control and collaboration among objects during the execution of a use case or scenario.

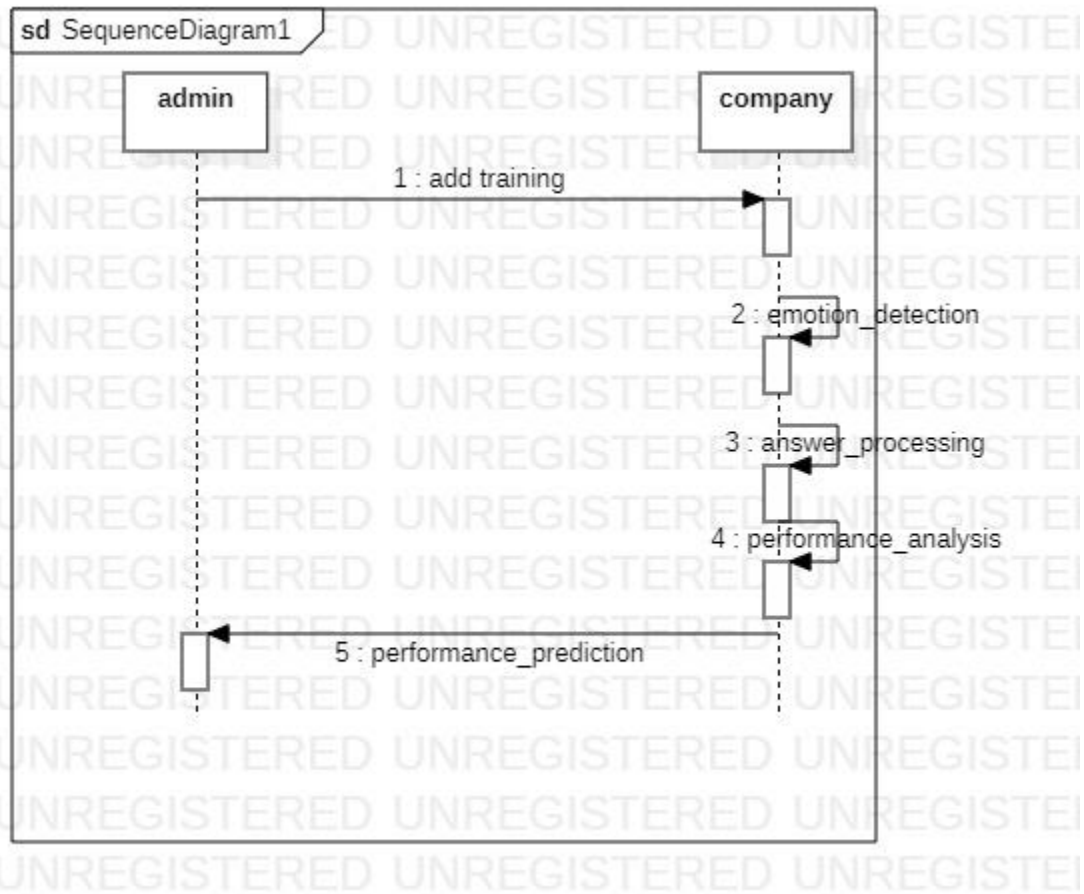


Fig 3.2 Sequence Diagram

3.2.3 Activity Diagram

An Activity Diagram is a type of UML (Unified Modeling Language) diagram that illustrates the flow of activities or actions within a system or a specific business process. It visually represents the sequence of actions, decisions, and transitions from one activity to another, allowing stakeholders to understand the workflow or behavior of a system or process.

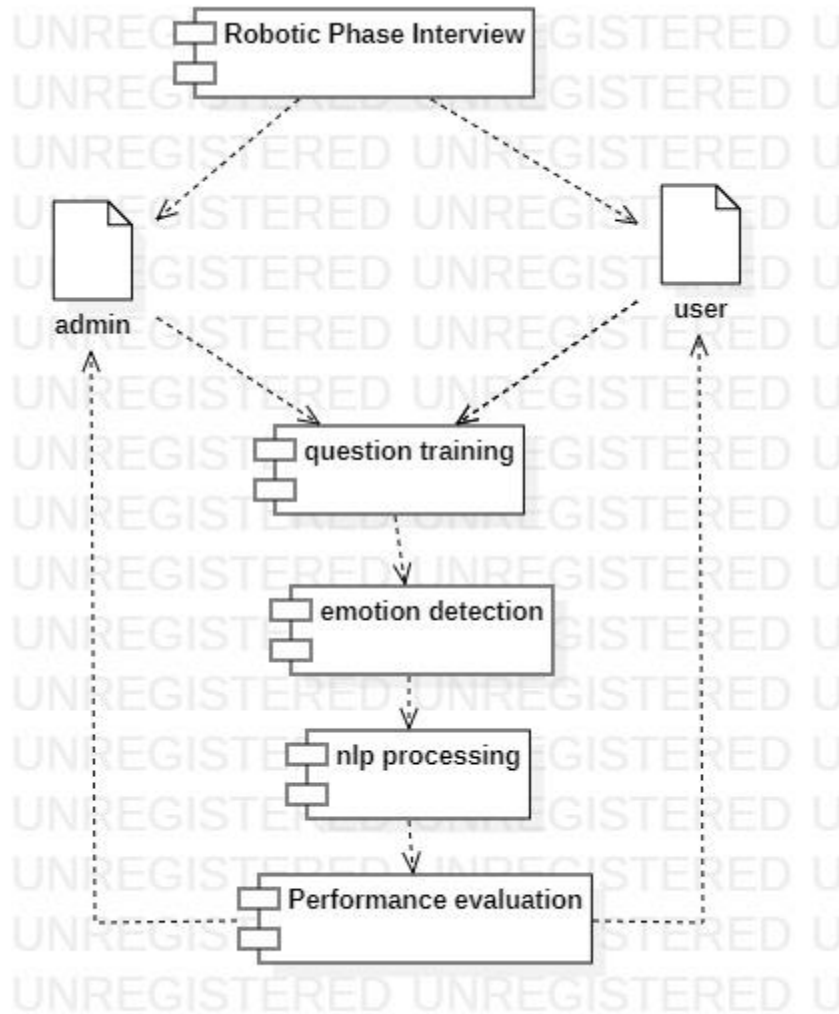


Fig 3.4 Activity Diagram

3.2.4 Class Diagram

A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that represents the structure and relationships of classes within a system. It provides a visual representation of the classes, their attributes, methods, and the associations between them.

Here's a breakdown of the components typically found in a class diagram:

Class: A class is represented as a rectangle with three compartments:

The top compartment contains the class name.

The middle compartment contains the attributes (data members) of the class.

The bottom compartment contains the methods (member functions) of the class.

Association: An association represents a relationship between two classes. It is depicted as a line connecting the classes, with optional multiplicity and role names indicating the nature of the association.

Generalization (Inheritance): Generalization represents an "is-a" relationship between classes, where one class (subclass or child) inherits attributes and behaviors from another class (superclass or parent). It is depicted as a solid line with a hollow arrowhead pointing from the subclass to the superclass.

Dependency: Dependency represents a relationship where one class depends on another class. It is typically shown as a dashed line with an arrowhead pointing from the dependent class to the independent class.

Aggregation and Composition: Aggregation and composition represent "whole-part" relationships between classes. Aggregation implies a weaker form of ownership, where the parts can exist independently of the whole, while composition implies a stronger form of ownership, where the parts are destroyed when the whole is destroyed. They are represented using diamond-shaped lines with filled or unfilled diamonds.

Multiplicity: Multiplicity indicates the number of instances of one class that are associated with a single instance of another class. It is denoted using numbers or symbols near the ends of association lines.

Class diagrams are commonly used during the design phase of software development to model the structure of object-oriented systems. They provide a high-level overview of the system's architecture, aiding in communication among stakeholders, software designers, and developers. Additionally, class diagrams serve as a blueprint for generating code and implementing the system.

4.SYSTEM ARCHITECTURE

The system architecture for animal detection typically involves a combination of hardware and software components designed to identify and classify recommendatio based on input data, such as images or sensor readings. Below is a conceptual overview of the system architecture for virtual blind people detection:

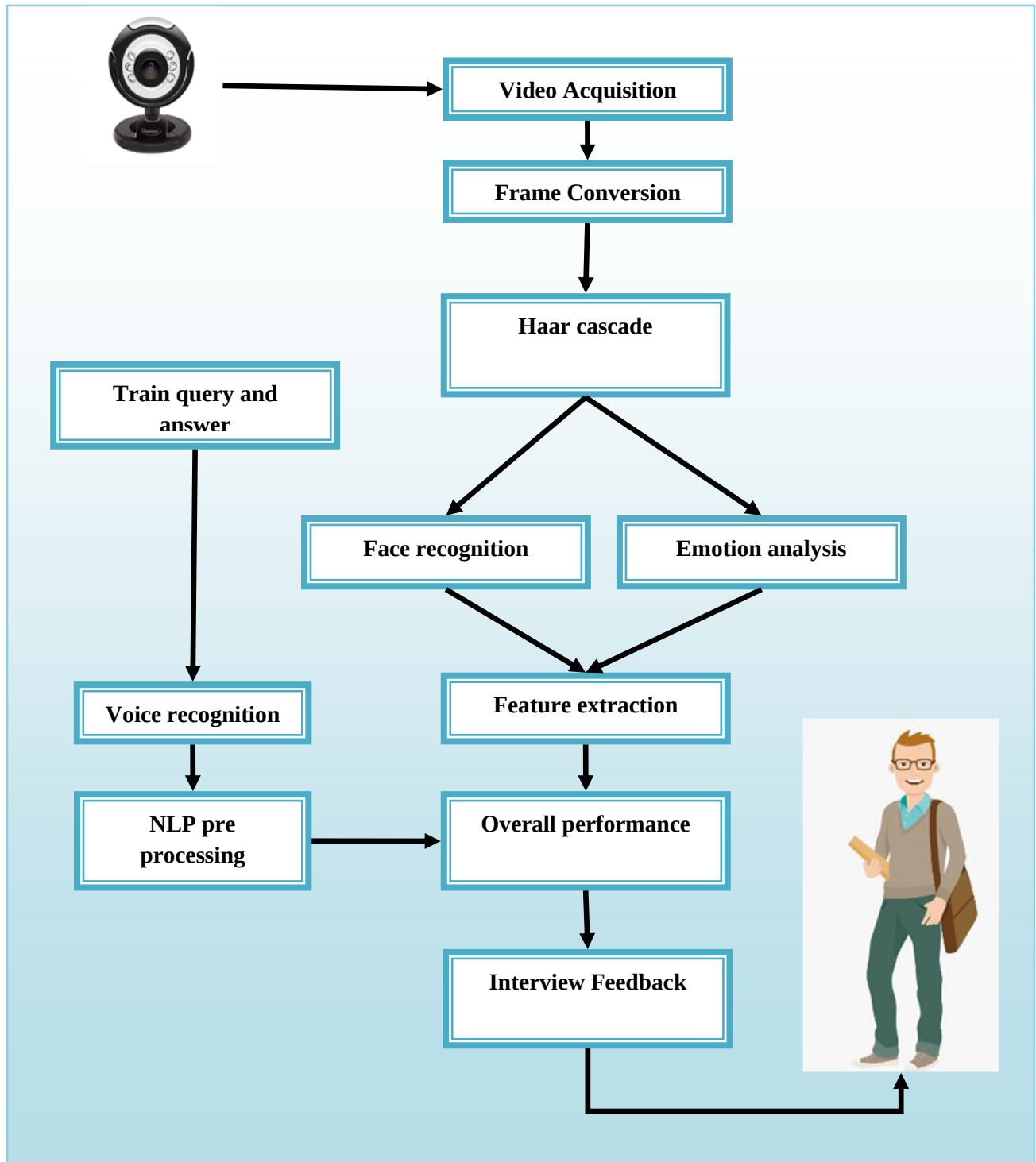


Fig 4.1 System Architecture

5.SYSTEM IMPLEMENTATION

5.1 MODULES

- Video acquisition
- Face detection
- Feature extraction
- Emotion detection
- Train interview query and answer
- Voice recognition
- NLP preprocessing
- Overall performance
- Interview feedback

5.2MODULES DESCRIPTION

5.2.1 VIDEO ACQUISITION

Video acquisition is the process where the video capturing unit in which the normal CCTV based surveillance functional is done. The capture videos are converted as frames where the image based analysis reduces the complexity

5.2.2 FACE DETECTION

Facial detection via the Haar Cascades algorithm is a common method used due to its high detection rate and fast processing speed. The algorithm can be summed up in four steps: feature selection, feature evaluation, feature learning to create a classifier, and cascading classifiers. Simple features are used, inspired by Haar basis functions, which are essentially rectangular features in various configurations. A two-rectangle feature represents the difference between the sum of the pixels in two adjacent regions of identical shape and size. This idea can be extended to the three-rectangle and four-rectangle features. In order to quickly compute these rectangle

features, an alternate representation of the input image is required, called an integral image.

5.2.3 ALGORITHM USED

Haar Classifier is generated by training few hundred sample views of a particular image, called positive examples, that are scaled to same size, and negative examples- arbitrary image of the same size. After a classifier is trained, it is applied to region of interest in an input image. The classifiers outputs as “1” if the region is likely to show the object and “0” otherwise. Here the word ‘cascade’ means that final classifier consists of various simple classifier that are applied one after the other to the area of interest till the detected face is rejected or passed completely.

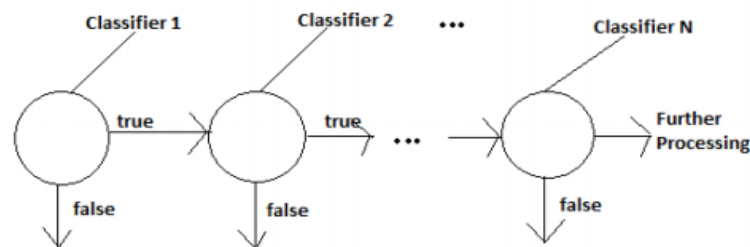


Fig 5.1 Classifier system

Haar cascade classifier is an open cv algorithm. It makes classification between images with an object (i.e face) and images without an object (i.e with non-faces). Initially, several hundreds of images with face and several hundreds of images with non-faces have been given to this classifier. This classifier was then trained by applying machine learning methods like the neural networks to recognize human faces. It then extracted Haar Features from those images and stored them in an xml file.

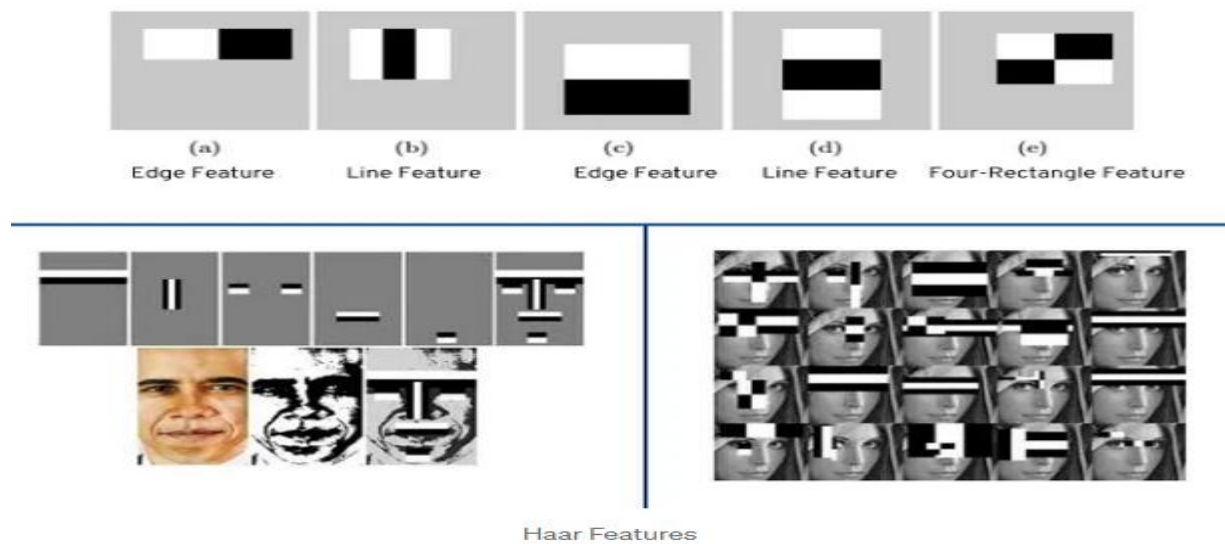


Fig 5.2 HAAR cascade face detection

Haar features are white and black pixels on the face. Also, we know that the grayscale image of a face does not have completely white and black pixels but here we are considering an ideal case where white pixels are lighter pixels and black pixels are darker pixels. In the above figure, eyebrow contains darker pixels and forehead contains lighter pixels. Similarly, it goes for eyes, nose and lips.

5.2.4 FEATURE EXTRACTION

The feature extraction points are done with the extraction of the maintenance approach in which the face feature point to detect the emotions are done. The point will be extracted on the basis of the pixel point maintenance

5.2.5 EMOTION DETECTION

The precise analysis of human facial expressions is necessary for a better understanding of human behavior. With the recent progress in deep learning, Convolution Neural Network (CNN) with its enhanced complex architecture is capable of emotion detection in a much better and more efficient way. The candidate

emotion is detected in the interview process for analyzing the emotion state of the student while giving the interview.

5.2.6 TRAIN INTERVIEW QUERY AND KEYWORD

The process starts with the training of the company trends and real time process and the related question and answer to the system to process the AI mock interview session by analyzing the answer. Based on the information gathered, AI receives a question as input, uses natural language processing (NLP) to understand the question, and generates a response in the form of an answer. AI models can provide different responses based on the context, phrasing, and quality of the input.

5.2.7 VOICE RECOGNITION

This module is done by the system to process the answer given by the candidate to analyze the voice to the text format conversion.

5.2.8 NLP PREPROCESSING

An NLP pipeline for text classification might include steps such as sentence segmentation, word tokenization, lowercasing, stemming or lemmatization, stop word removal, and spelling correction. After processing the student given answer it will map with the trained keyword.

5.2.9 OVERALL PERFORMANCE

The overall performance based on the answer and the face expression and the emotion state is analyzed and the overall performance result is analyzed. The result of all the above process is given as a result and gives the feedback based on the overall performance to improve their confident.

6.CODING AND SCREENSHOT

6.1 SAMPLE CODING

Back end code

```
import pymysql
class master_flask_code:
    def __init__(self):
        self.user = 'root'
        self.password = ""
        self.host = 'localhost'
        self.database = 'python_ai_interview'
    def find_max_id(self,table):
        conn = pymysql.connect(user=self.user, password=self.password,
host=self.host, database=self.database,charset='utf8')
        cursor = conn.cursor()
        cursor.execute("SELECT id FROM "+table)
        data = cursor.fetchall()
        maxin = len(data)
        if maxin == 0:
            maxin = 1
        else:
            maxin += 1
        return maxin
    def insert_query(self,qry):
        conn = pymysql.connect(user=self.user, password=self.password,
host=self.host, database=self.database,charset='utf8')
        cursor = conn.cursor()
```

```

        result=cursor.execute(qry)
        conn.commit()
        conn.close()
        return result
def select_login(self,qry):
    conn = pymysql.connect(user=self.user, password=self.password,
host=self.host, database=self.database,charset='utf8')
    cursor = conn.cursor()
    cursor.execute(qry)
    data = cursor.fetchall()
    check = len(data)
    if check == 0:
        return 'no'
    else:
        return 'yes'
def select_single_colum(self, table, colum):
    conn = pymysql.connect(user=self.user, password=self.password,
host=self.host, database=self.database,charset='utf8')
    qry1 = ("select " + colum + " from " + table)
    cursor = conn.cursor()
    cursor.execute(qry1)
    data = cursor.fetchall()
    return data
def select_entire_colum(self,table,colum):
    conn = pymysql.connect(user=self.user, password=self.password,
host=self.host, database=self.database,charset='utf8')
    qry1=("select * from "+table+" where symptoms LIKE '%" +colum+"%'")

```

```

        cursor = conn.cursor()
        cursor.execute(qry1)
        data = cursor.fetchall()
        return data

def select_direct_query(self,qry):
    conn = pymysql.connect(user=self.user, password=self.password,
host=self.host, database=self.database,charset='utf8')

    cursor = conn.cursor()
    cursor.execute(qry)
    data = cursor.fetchall()
    return data

```

Exam Data code

```

import csv
from difflib import SequenceMatcher

```

```

class exam_demo:
    qid=0
    job_name = ""
    job_company = ""
    load_question=0
    mark=0
    average=0
    ar_check_sts=0
    question_list = []
    answer_list = []

```

```

file = 'dataset.csv'

def __init__(self):
    with open(exam_demo.file) as f:
        reader = csv.DictReader(f, delimiter=',')
        for row in reader:

            t1 = row['company']
            t2 = row['job']
            t3 = row['question']
            t4 = row['answer']

            print(exam_demo.job_company, exam_demo.job_name)
            if exam_demo.job_name == t2:
                exam_demo.question_list.append(t3)
                exam_demo.answer_list.append(t4)
            # print(exam_demo.question_list)

def read_question(self,qid):
    if qid<10:
        question=exam_demo.question_list[qid]
        answer=exam_demo.answer_list[qid]
        return question,answer
    else:
        return 'no','no'

def match_answer(self,user_inut,original_answer):
    return SequenceMatcher(None, user_inut, original_answer).ratio()

```

```

# answer='demo sample'
# qid=5
# arn=exam_demo()
# question,answer=arn.read_question(qid)
# # print(question)
# user_input=input("Enter Your Answer")
# mark=arn.match_answer(user_input,answer)
# print(mark)
#

```

Main Code

```

import csv
import os
import shutil
import time
from tkinter import Tk, messagebox, ttk
from tkinter import *
from tkinter.ttk import Treeview

import cv2
from PIL import Image, ImageTk

import speech_recognition as sr
from PIL import ImageTk
import ar_master

from tkinter_video_voice_input import DomainOperations

```

```

class tk_master:
    care_taker = "
    user = "
    emotion = "
    company = "
    job_list=[]
    job_id=""
    job_name=""
    job_company=""
    job_skills=""

    def __init__(self):
        self.master = 'ar_master'
        self.title = 'FACE INTERVIEW'
        self.titlec = 'FACE INTERVIEW'
        self.background_color = '#2F4F4F'
        self.text_color = '#FFF'
        # self.text_color = '#c0c0c0'
        self.background_image = 'images/background_hd1.jpg'

    def get_title(self):
        return self.title

    def get_titlec(self):
        return self.titlec

    def get_background_color(self):

```

```
return self.background_color
```

```
def get_text_color(self):
```

```
    return self.text_color
```

```
def get_background_image(self):
```

```
    return self.background_image
```

```
def admin_login(self):
```

```
    admin_login_root = Toplevel()
```

```
    admin_login_root.attributes('-topmost', 'true')
```

```
    get_data = tk_master()
```

```
    w = 780
```

```
    h = 500
```

```
    ws = admin_login_root.winfo_screenwidth()
```

```
    hs = admin_login_root.winfo_screenheight()
```

```
    x = (ws / 2) - (w / 2)
```

```
    y = (hs / 2) - (h / 2)
```

```
    admin_login_root.geometry('%dx%d+%d+%d' % (w, h, x, y))
```

```
    image = Image.open('images/background_hd1.jpg')
```

```
    img = image.resize((w, h))
```

```
    my_img = ImageTk.PhotoImage(img)
```

```
    admin_login_root.resizable(False, False)
```

```
    canvas1 = Canvas(admin_login_root, width=200, height=300)
```

```
    canvas1.create_image(0, 0, image=my_img, anchor=NW)
```

```
    canvas1.pack(fill="both", expand=True)
```

```

        canvas1.create_text(390, 20, text=get_data.get_title(), font=("Times New
Roman", 24),
                            fill=get_data.get_text_color())
    ##
    admin_id2 = canvas1.create_text(390, 100, text="ADMIN LOGIN",
font=("Times New Roman", 24),
                                fill=get_data.get_text_color())
    admin_id2 = canvas1.create_text(300, 200, text="Username", font=("Times
New Roman", 24),
                                fill=get_data.get_text_color())
    admin_id2 = canvas1.create_text(300, 300, text="Password", font=("Times
New Roman", 24),
                                fill=get_data.get_text_color())
    e1 = Entry(canvas1, font=('times', 15, ' bold '))
    canvas1.create_window(470, 200, window=e1)
    e2 = Entry(canvas1, font=('times', 15, ' bold '), show="*")
    canvas1.create_window(470, 300, window=e2)

def exit_program():
    a = e1.get()
    b = e2.get()
    if (a == ""):
        messagebox.showinfo(title="Alert", message="Enter Username",
parent=admin_login_root)
    elif (b == ""):
        messagebox.showinfo(title="Alert", message="Enter Password",
parent=admin_login_root)

```

```

        elif ((a == "admin") and (b == "admin")):
            messagebox.showinfo("Result", "Login Success",
parent=admin_login_root)
            admin_login_root.destroy()
            tt = tk_master()
            # tt.admin_home()
            tt.company_home()
        else:
            messagebox.showinfo("Result", "Login Failed",
parent=admin_login_root)

        b1 = Button(canvas1, text="Login", command=exit_program, font=('times',
15, 'bold '))
        canvas1.create_window(470, 400, window=b1)
        admin_login_root.mainloop()

def admin_home(self):
    admin_home_root = Toplevel()

    admin_home_root.attributes('-topmost', 'true')

    messagebox.showinfo("Result", "Login Success",
parent=company_login_root)
    company_login_root.destroy()
    tt = tk_master()
    tt.company_home()

```

```

b1=Button(canvas1, text="Login", command=exit_program, font=('times', 15,
' bold '))

canvas1.create_window(470, 400, window=b1)

company_login_root.mainloop()

def company_registration(self):
    company_registration_root = Toplevel()
    company_registration_root.attributes('-topmost', 'true')
    get_data = tk_master()
    w = 780
    h = 500
    ws = company_registration_root.winfo_screenwidth()
    hs = company_registration_root.winfo_screenheight()
    x = (ws / 2) - (w / 2)
    y = (hs / 2) - (h / 2)
    company_registration_root.geometry('%dx%d+%d+%d' % (w, h, x, y))
    image = Image.open('images/background_hd1.jpg')
    img = image.resize((w, h))
    my_img = ImageTk.PhotoImage(img)
    company_registration_root.resizable(False, False)
    canvas1 = Canvas(company_registration_root, width=200, height=300)
    canvas1.create_image(0, 0, image=my_img, anchor=NW)
    canvas1.pack(fill="both", expand=True)
    # canvas1.create_text(390, 20, text=get_data.get_title(), font=("Times New
Roman", 24),
    #
    fill=get_data.get_text_color())
    ##

```

```

admin_id2 = canvas1.create_text(390, 20, text="COMPANY
REGISTRATION", font=("Times New Roman", 24),
                                fill=get_data.get_text_color())
admin_id2 = canvas1.create_text(300, 70, text="Name", font=("Times New
Roman", 24),
                                fill=get_data.get_text_color())
admin_id2 = canvas1.create_text(300, 120, text="Contact", font=("Times
New Roman", 24),
                                fill=get_data.get_text_color())
admin_id2 = canvas1.create_text(300, 170, text="Email", font=("Times New
Roman", 24),
                                fill=get_data.get_text_color())
admin_id2 = canvas1.create_text(300, 220, text="Address", font=("Times
New Roman", 24),
                                fill=get_data.get_text_color())
admin_id2 = canvas1.create_text(300, 270, text="Username", font=("Times
New Roman", 24),
                                fill=get_data.get_text_color())
admin_id2 = canvas1.create_text(300, 320, text="Password", font=("Times
New Roman", 24),
                                fill=get_data.get_text_color())

e1 = Entry(canvas1, font=('times', 15, ' bold '))
canvas1.create_window(480, 70, window=e1)
e2 = Entry(canvas1, font=('times', 15, ' bold '))
canvas1.create_window(480, 120, window=e2)
e3 = Entry(canvas1, font=('times', 15, ' bold '))

```

```

scrollbary.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)
tree.heading('id', text="id", anchor=W)
# tree.heading('company', text="company", anchor=W)
tree.heading('job_title', text="job_title", anchor=W)
tree.heading('skills', text="skills", anchor=W)
tree.bind('<ButtonRelease-1>', selectItem)
tree.column('#0', stretch=NO, minwidth=0, width=0)
tree.column('#1', stretch=NO, minwidth=0, width=200)
tree.column('#2', stretch=NO, minwidth=0, width=200)
# tree.column('#3', stretch=NO, minwidth=0, width=200)
tree.pack()
for x in date_list:
    tree.insert("", 0, values=(x))

job_search_root.attributes('-topmost', 'true')
get_data = tk_master()
w = 880
h = 600
ws = job_search_root.winfo_screenwidth()
hs = job_search_root.winfo_screenheight()
x = (ws / 2) - (w / 2)
y = (hs / 2) - (h / 2)
job_search_root.geometry('%dx%d+%d+%d' % (w, h, x, y))
image = Image.open('images/background_hd.jpg')

```

```

img = image.resize((w, h))
my_img = ImageTk.PhotoImage(img)
job_search_root.resizable(False, False)
canvas1 = Canvas(job_search_root, width=200, height=300)
canvas1.create_image(0, 0, image=my_img, anchor=NW)
canvas1.pack(fill="both", expand=True)
# canvas1.create_text(390, 20, text=get_data.get_title(), font=("Times New
Roman", 30),
#           fill=get_data.get_text_color())
##
admin_id2 = canvas1.create_text(390, 20, text="JOB SEARCH",
font=("Times New Roman", 24),
           fill=get_data.get_text_color())
admin_id2 = canvas1.create_text(100, 100, text="Job Title", font=("Times
New Roman", 24),
           fill=get_data.get_text_color())
e1 = Entry(canvas1, font=('times', 15, ' bold '))
canvas1.create_window(280, 100, window=e1)

b1 = Button(canvas1, text="Search", command=view_data, font=('times',
15, ' bold '))
canvas1.create_window(490, 100, window=b1)
job_search_root.mainloop()
def dd2():
    dd=0
    b1 = Button(canvas1, text="Job Search", command=job_search, font=('times',
15, ' bold '))

```

```

canvas1.create_window(420, 150, window=b1)
###
b2 = Button(canvas1, text="Logout", command=dd2, font=('times', 15, ' bold
'))
canvas1.create_window(420, 200, window=b2)
###
# b3 = Button(canvas1, text="Received", command=dd3, font=('times', 15, '
bold '))
# canvas1.create_window(420,
# .250, window=b3)

user_home_root.mainloop()
def job_search_1(self):
    job_search_1_root = Toplevel()
    job_search_1_root.attributes('-topmost', 'true')
    get_data = tk_master()
    w = 780
    h = 500
    ws = job_search_1_root.winfo_screenwidth()
    hs = job_search_1_root.winfo_screenheight()
    x = (ws / 2) - (w / 2)
    y = (hs / 2) - (h / 2)
    job_search_1_root.geometry('%dx%d+%d+%d' % (w, h, x, y))
    image = Image.open('images/background_hd1.jpg')
    img = image.resize((w, h))
    my_img = ImageTk.PhotoImage(img)
    job_search_1_root.resizable(False, False)

```

```
canvas1 = Canvas(job_search_1_root, width=200, height=300)
canvas1.create_image(0, 0, image=my_img, anchor=NW)
canvas1.pack(fill="both", expand=True)
canvas1.create_text(390, 20, text=get_data.get_title(), font=("Times New
Roman", 24),
                    fill=get_data.get_text_color())

ar = tk_master()
root = ar.set_window_design()
```

6.2 SAMPLE SCREENSHOT

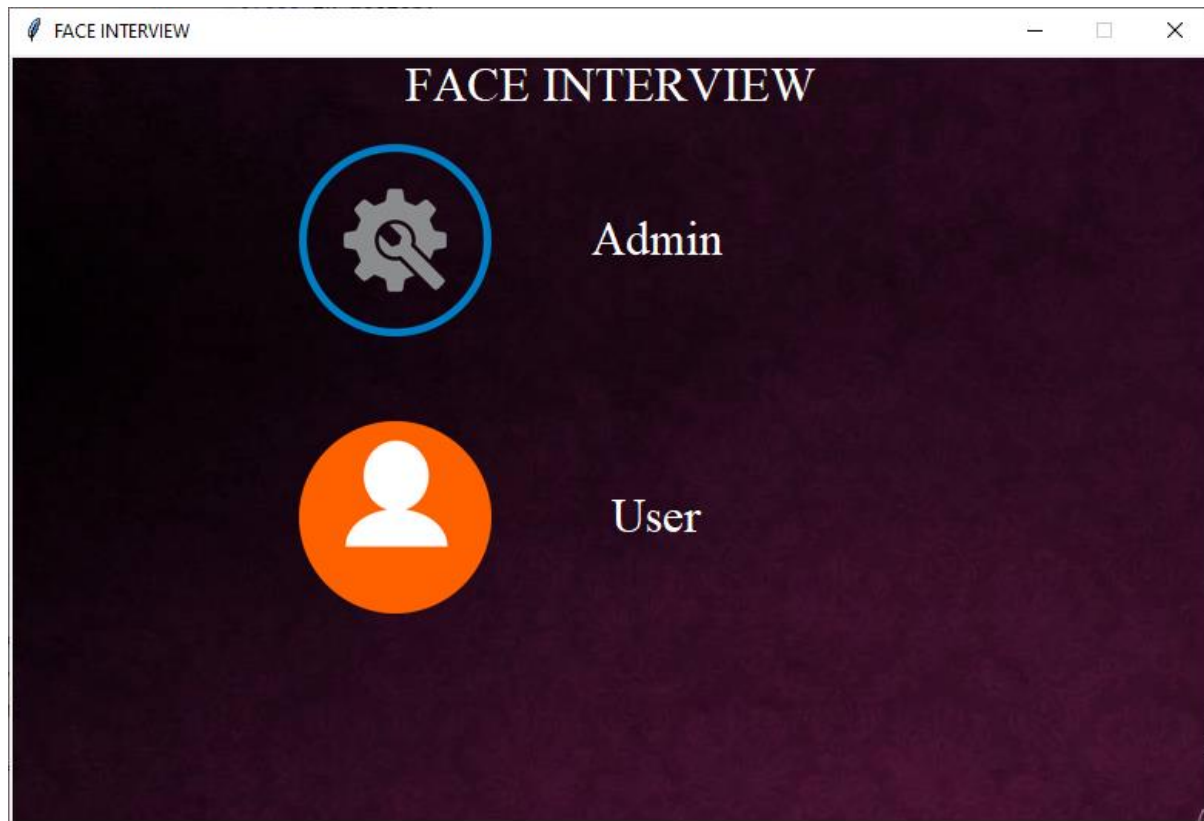


Fig 6.1 Home Page

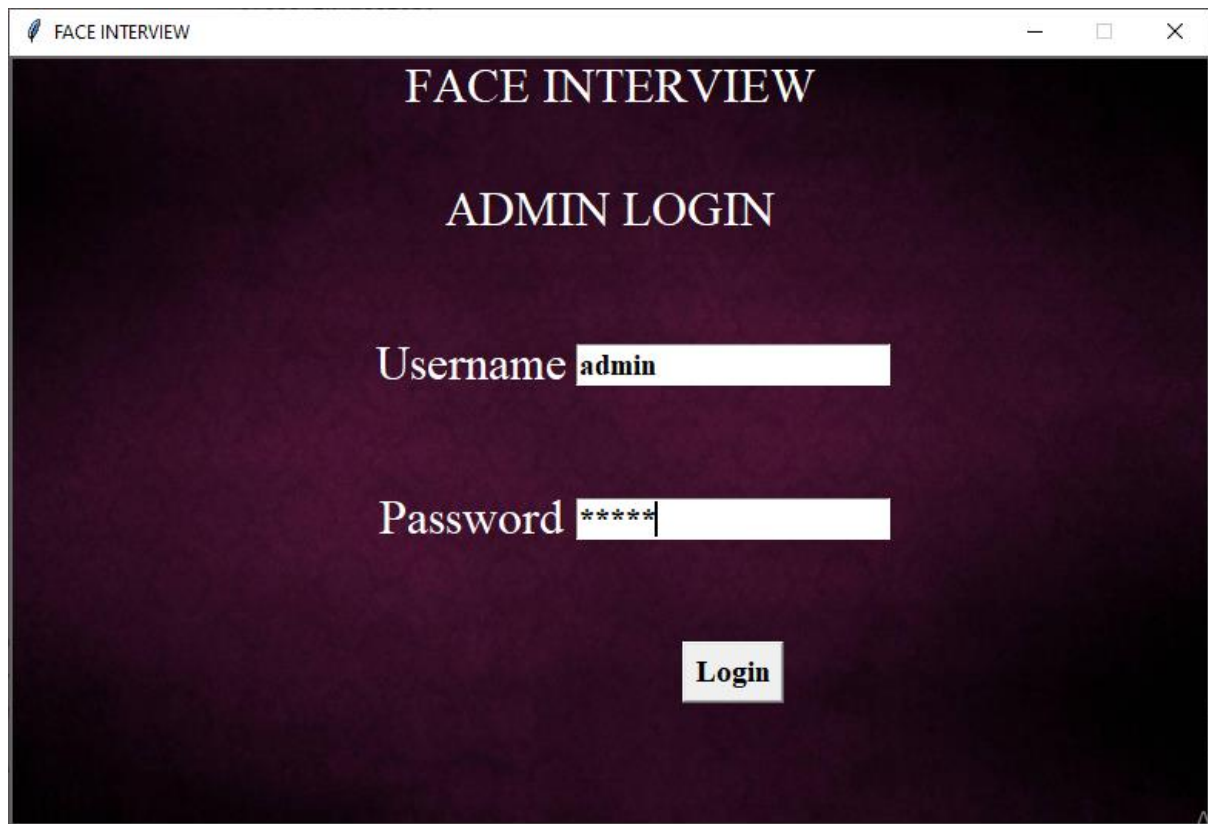


Fig 6.2 Admin login

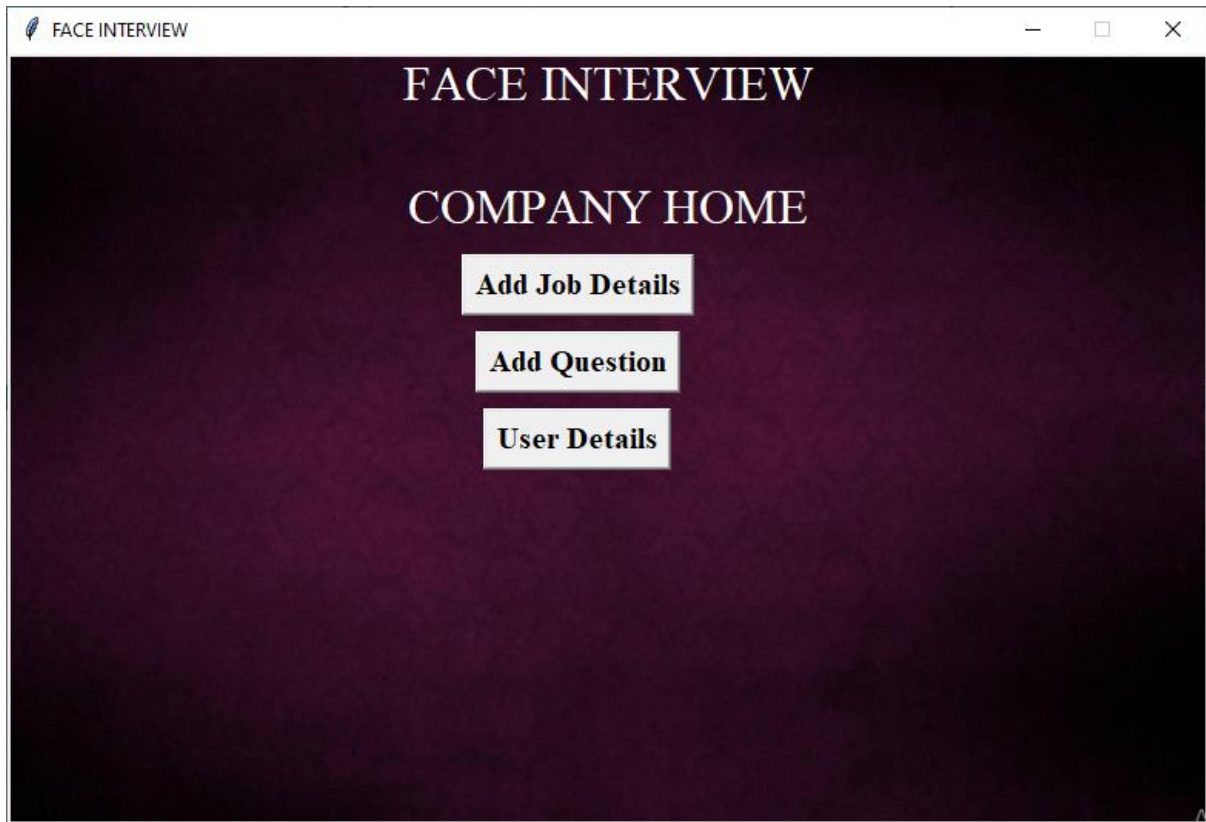


Fig 6.3 Company Home

FACE INTERVIEW

FACE INTERVIEW

ADD JOB

Job Title	js developer
Skills Required	js
Salary	8000
Description	js dddd

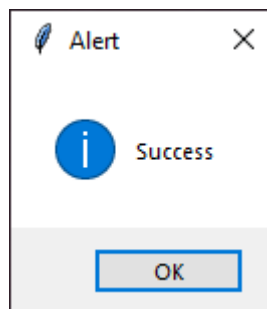


Fig 6.4 Company Add job details

FACE INTERVIEW

FACE INTERVIEW

ADD QUESTION

Company	hcl
Job Title	js developer
Question	es present in javascript
Answer	use the typeof operator

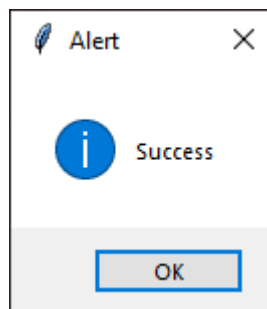


Fig 6.5 Company Questions to be asked

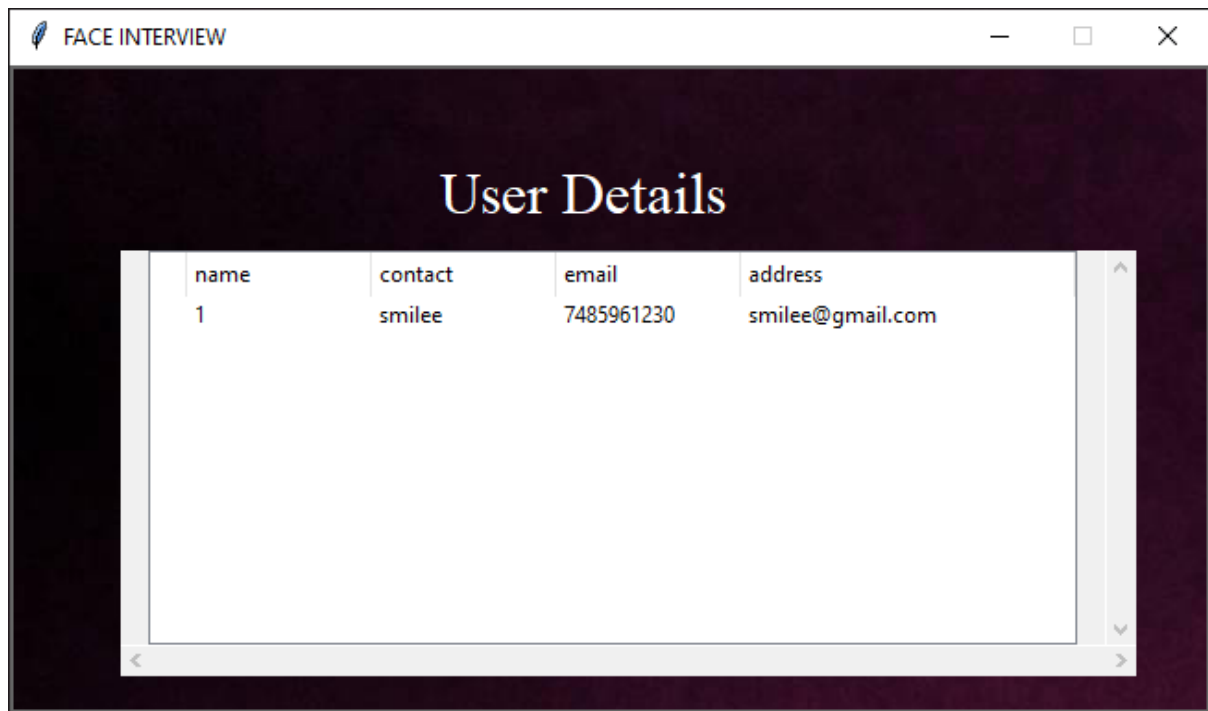


Fig 6.6 Company view candidate details

The screenshot shows a window titled "FACE INTERVIEW" with a dark blue background. The main heading is "USER REGISTRATION" in a large, white, serif font. Below the heading is a registration form with the following fields and values:

- Name: hendry
- Contact: 9876543210
- Email: hendry
- Address: trichy
- Username: hendry
- Password: 123

At the bottom of the form is a button labeled "Register".

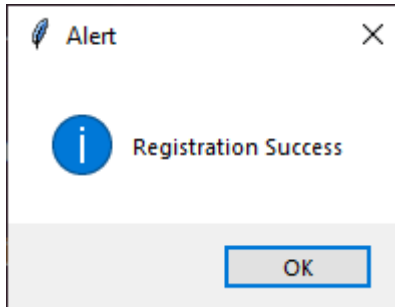


Fig 6.7 User Registration

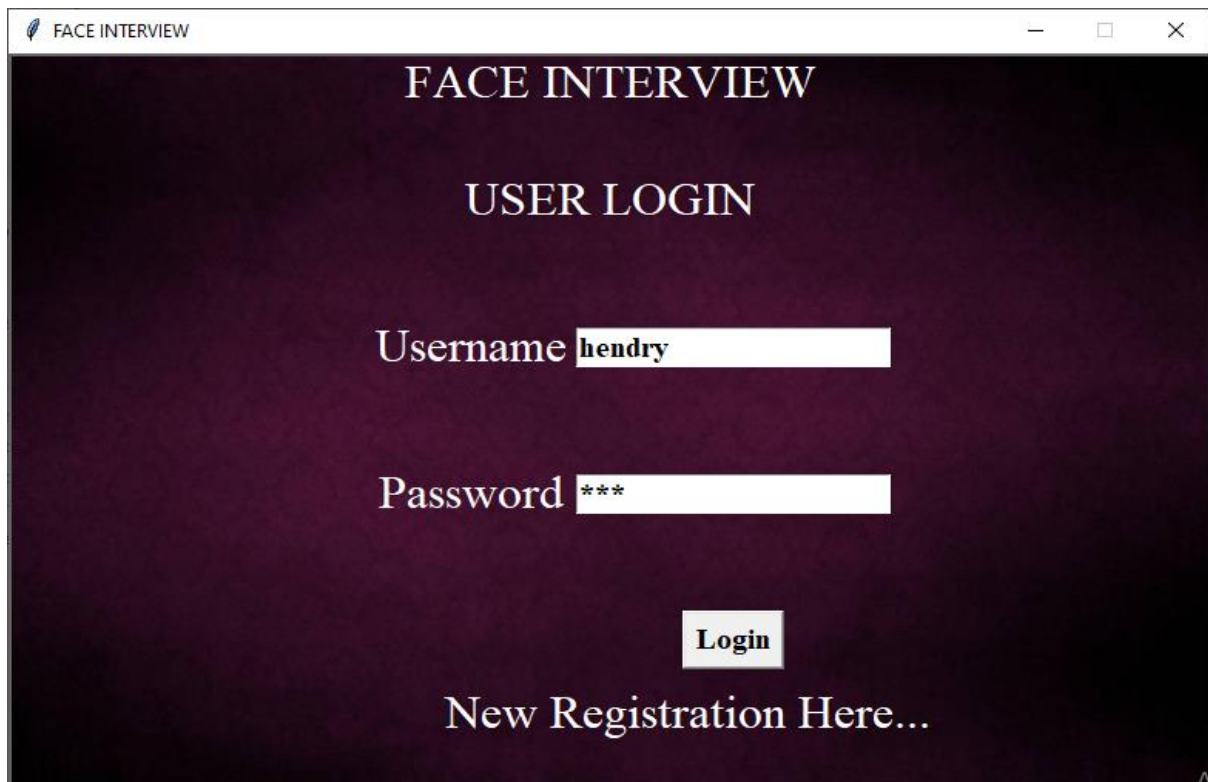


Fig 6.7 User Login

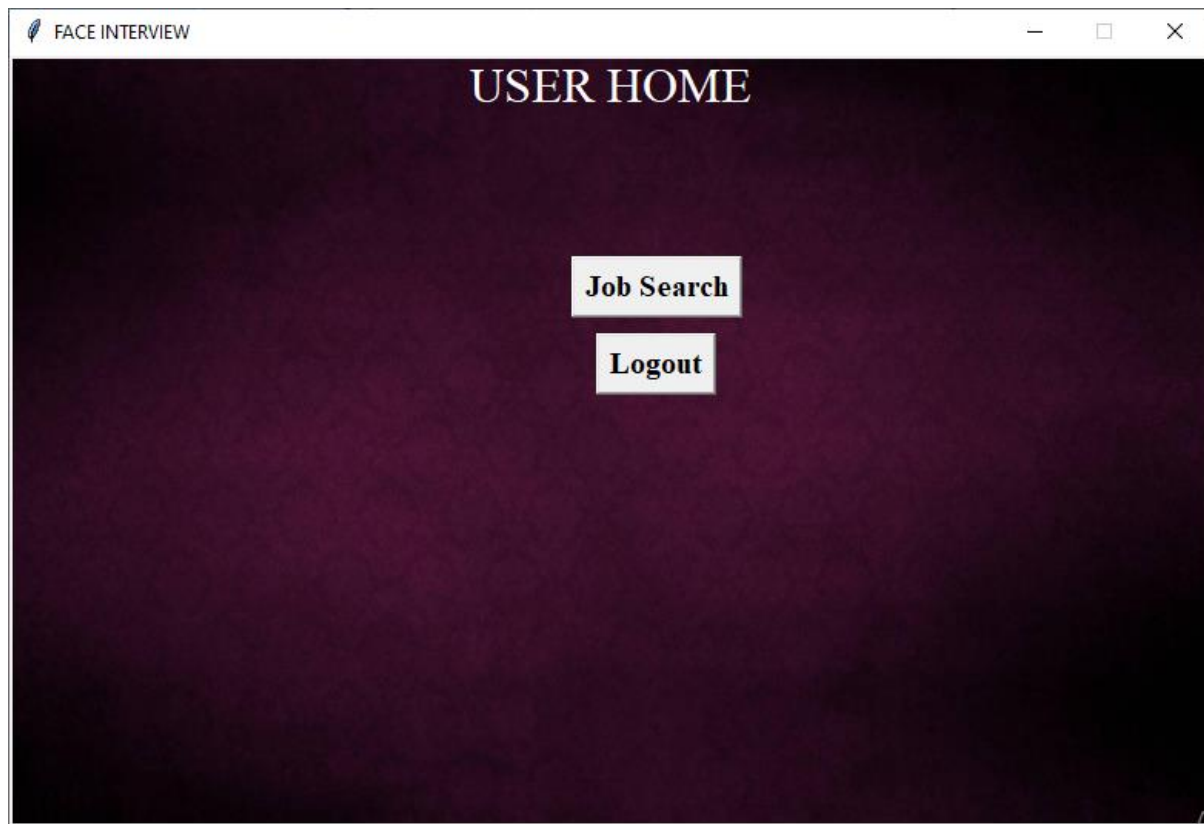


Fig 6.8 User home page



Fig 6.9 User Job search

FACE INTERVIEW

—

□

×

JOB SEARCH

Job Title

js developer

Search

id	job_title	skills
1	js developer	js

Activat

Go to Se

Fig 6.10 Job title



Fig 6.11 Staff interview start

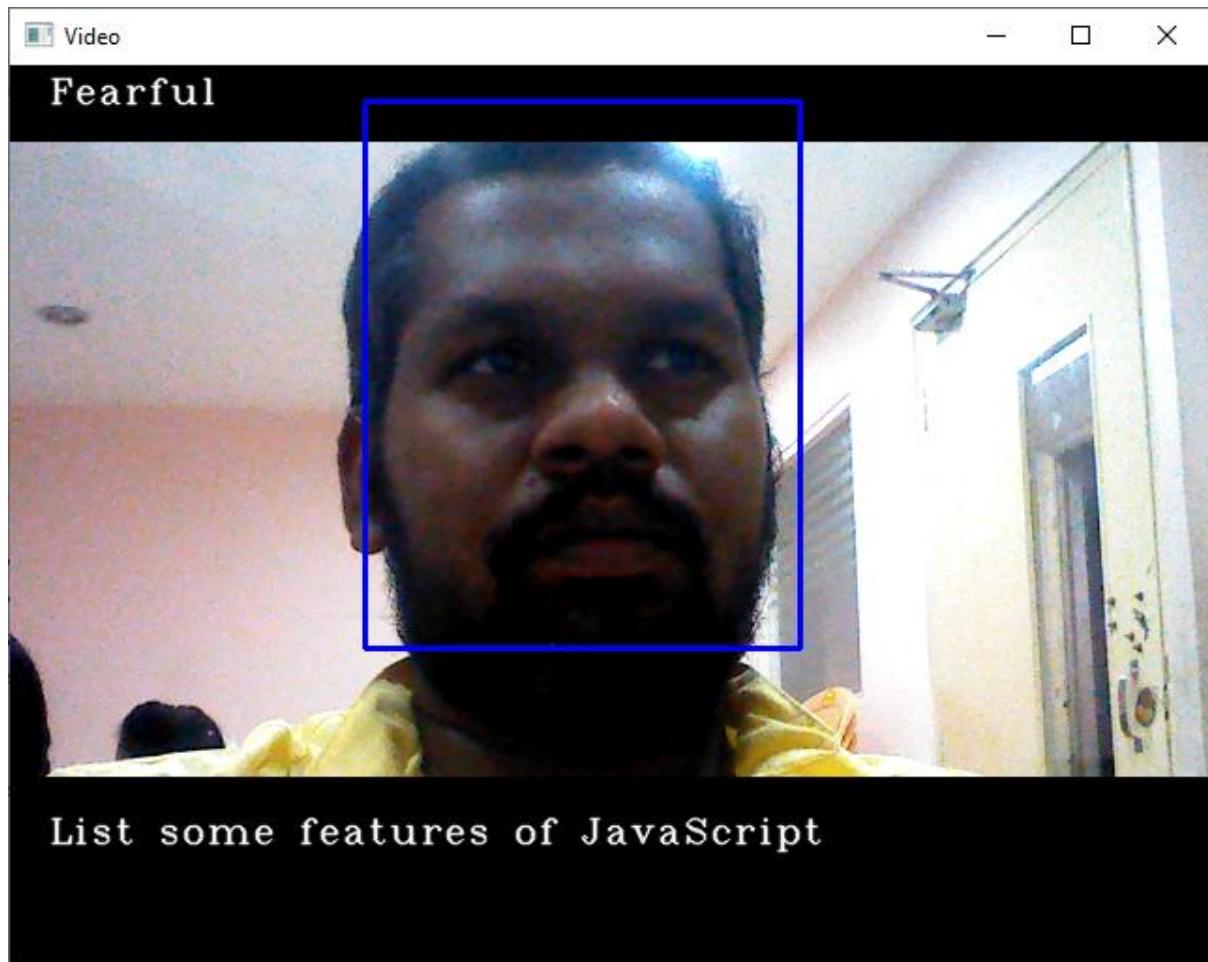


Fig 6.12 Emotion recognition with answering system

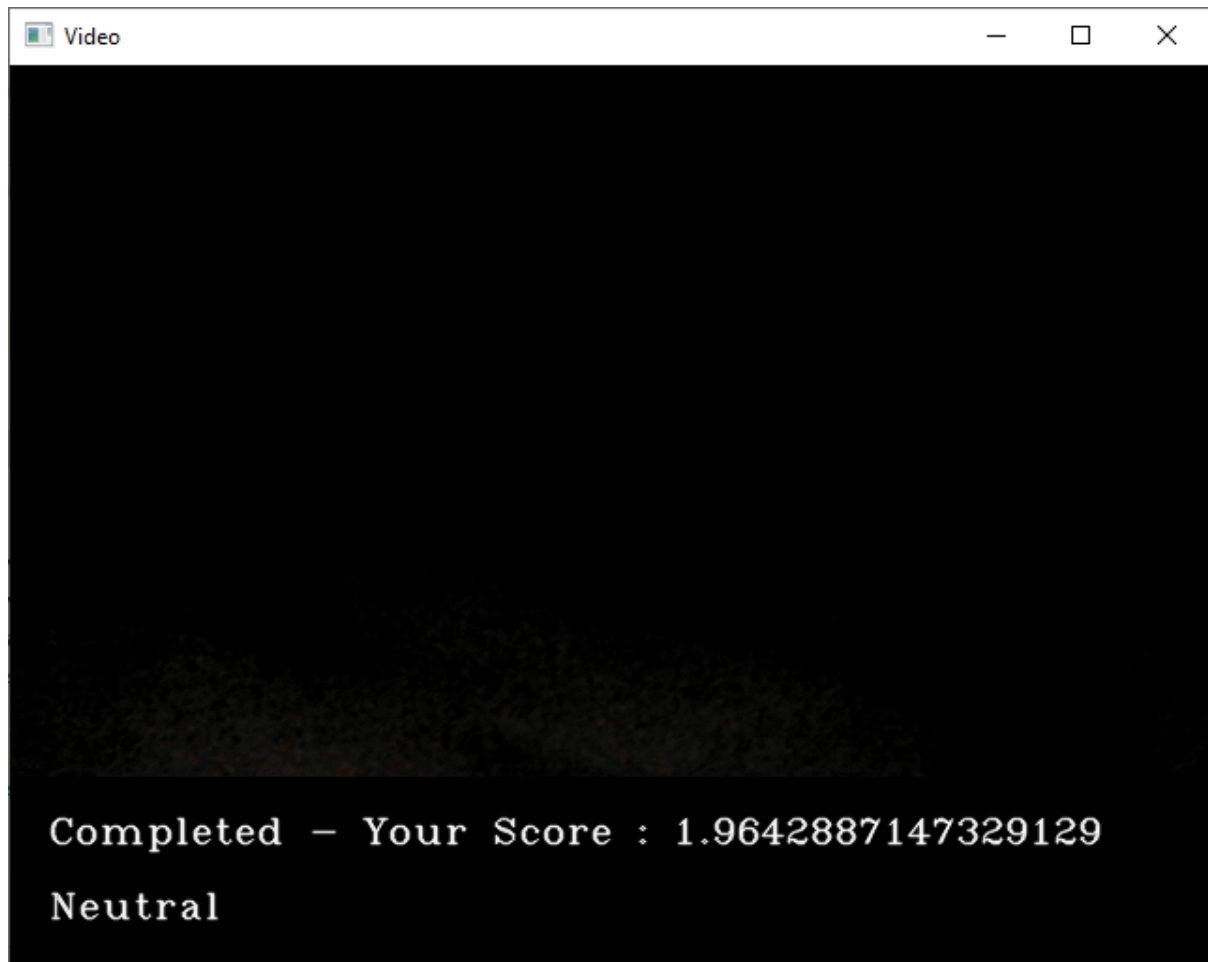


Fig 6.13 Score analysis

7.CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

The main purpose of AI assessment for companies for recruiting candidate is to save the employer time in the interview process without losing too much quality. Then again, it also makes life a lot more comfortable for the candidate. The applicant can choose to have the interview wherever they like, whether that's at home or somewhere else. As the applicable range of AI-based solutions is expanding to the general area of personnel management with its time and cost efficiency, as well as reliability and fairness recognized, the deep learning-based job interview solution proposed by the present study needs to be applied widely to written examinations and personality and aptitude tests. AI-based job interviews were developed to support the reasonable selection of outstanding candidates and the decision-making aspect in the employment process. It is expected that the design specifications and solution application results presented.

7.2 FUTURE ENHANCEMENT

In future resume based short listing who all applied for the system can be done to reduce some more burdens. The system can be carried with website itself with online recruitments

REFERENCES

- [1] Blacksmith, N., & Poeppelman, T., 2014. Video-based technology: The next generation of recruitment and hiring. *The Industrial-Organizational Psychologist*, 52, 84-88.
- [2] Muthukumaran, C.K., 2014. Recruitment process: a study among the employees at information technology (it) industry in Chennai. *International Journal of Management Research and Reviews*, 4(1), p.91.
- [3] Holm, A., 2010. The effect of e-recruitment on the recruitment process: Evidence from case studies of three Danish MNCs. In *Proceedings of the 3rd European academic workshop on electronic human resource management* (pp. 91-111).

- [4] Louw, G.J., 2013. Exploring recruitment and selection trends in the Eastern Cape. SA Journal of Human Resource Management, 11(1), pp.10-pages.
- [5] Cowan, D., 1985. Artificial intelligence at Edinburgh University. Computer-Aided Design, 17(9), pp.465-469.
- [6] Ghosh, S. and Mitra, I., 2017. Message from PwC.
- [7] Mansfield, Wooster, & Marion, 2016, Staffing Decisions. Artificial Intelligence and Human Resources.
- [8] Yeruva Ramana Reddy, Applications of Artificial Intelligence and Machine Learning in Geotechnical Engineering. International Journal of Emerging Technologies and Innovative Research, 2017, 4(2), pp. 425-428
- [9] Didi Faouzi, Nacereddine Bibi Triki and Ali Chermitti, Optimizing The Greenhouse Micro–Climate Management by The Introduction of Artificial Intelligence Using Fuzzy Logic, International Journal of Computer Engineering and Technology, 7(3), 2016, pp. 78–92.
- [10] Yeruva Ramana Reddy, An Empirical Study on The Methods And Instruments For Automated Geotechnical Monitoring. International Journal of Creative Research Thoughts (IJCRT), 2016, 4(3), pp.542-546.