INDEX

CHAPTER -1: INTRODUCTION

1.1 INTRODUCTION

Face is the most common biometric used by humans. Recognize the face from a general view point under different illumination conditions, facial expressions. A face recognition system is a technology capable of matching a human face from digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.

Face detection and Face recognition are often employed in various real-world application fields. The purpose of the examination monitoring system using face recognition is to ease the attendance process which consumes lot of time and efforts, it is a convenient and easy way for students and examiner.

The system will capture the images of the students and using face recognition algorithm mark the attendance in the sheet. This way the examiner will get their attendance marked without actually spending time in traditional attendance marking. The identification process to determine the presence of a person in a room or building is currently one of the routine security activities. Every person who will enter a room or building must go through several authentication processes first, that later these information's can be used to monitor every single activity in the room for a security purpose. Authentication process that is being used to identify the presence of a person in a room or building still vary. It detects the face and mark attendance accordingly.

There are four steps in face recognition process like acquiring a sample, extracting feature, compression template, declare a match etc. It is easier and user-friendly human-machine interaction to authenticate the examinee by his facial features which are extracted using the photo which he/she submitted during the registration process of the exams. A machine can detect and recognize a person's face, resulting in the authentication of the candidate. It has become more difficult for face detection because of some random attributes. For example: - if the person changing his features like eyeglasses and a beard will have effect in detecting effectiveness.

The face recognition to authentication of the examinee with minimal human interaction.The process varies from writing a name and signatures in the attendance list, using an identity card, or using biometric methods.

Biometrics is a developing technology, which has been widely used in forensics, safe access and prison security. Biometrics systems are basically a pattern recognition system that recognizes someone by determining. Authentication using different biological features, namely fingerprints, retinal scanning, facial recognition that causes physiological biometrics and behavioral characteristics. Controlling over all students during the exam period. Face recognition is an necessary to make a whole complete part of biometrics.

The aim of this method is to study prototype of an online exam application design in order to recognize students who will take the exam with face recognition methods.

Facial features are extracted and implemented through algorithms which are efficient. They are a great diversity in the way facial appearance is interpreted for recognition by an automatic system. Multi model biometric is also done in which more than one biometric grouped together and compare with the existing databases.

Face recognition is gaining more popularity and has been widely used. Face verification is an matching process, it compares face image against the pattern face images. Widely used in commercial system to perform real-time face detection, image registration and image matching.

1.2 KEY FEATURES:

The key features of an online examination face recognition system using Python can include:

1. Face Detection: The system should be able to detect and locate human faces in images or video streams. This is usually done using techniques such as Haar cascades, HOG (Histogram of Oriented Gradients), or deep learning-based approaches like Convolutional Neural Networks (CNNs).

2. Face Recognition: The system should have the ability to recognize and identify individual faces. It can utilize methods like eigenfaces, Fisher faces, or modern deep learning techniques such as Deep Face or Face Net. This allows the system to match the captured faces with the pre-registered faces of the examination participants.

3. Liveness Detection: To prevent fraudulent behaviour, the system can incorporate liveness detection to ensure that the detected face is a live person and not a photograph or a video. Techniques like eye blinking detection, head movement analysis, or texture analysis can be employed to detect liveness.

4. User Enrolment: The system should provide a mechanism for user enrolment, where participants' faces can be registered and stored in a database. This allows for accurate recognition and verification during the examination.

5. Authentication and Verification: Before granting access to the online examination, the system should authenticate and verify the identity of the participant based on their face. This ensures that only the authorized individuals can take the examination.

6. Real-time Monitoring: The system can include real-time monitoring capabilities to track participants during the examination. This can help detect any suspicious activities or attempts to deceive the system.

7. Robustness and Performance: The face recognition system should be robust to variations in lighting conditions, facial expressions, and pose. It should be capable of delivering accurate results in real-time scenarios with a reasonable level of performance.

8. Privacy and Security: Privacy and security should be given utmost importance. The system should comply with relevant data protection regulations and ensure the secure storage and handling of participant's facial data.

9. Scalability and Flexibility: The system should be designed to handle a large number of participants and should be scalable to accommodate future growth. Additionally, it should be flexible enough to be easily integrated with different online examination platforms or custom applications.

1.3 ARCHITECTURE AND TECHNOLOGIES:

To implement an online examination system with face recognition using Python, you would need to combine various technologies and architectures. Here's a suggested architecture and the technologies you can use:

1. Web Application Framework: You can use a web framework such as Flask or Django to build the front-end and back-end components of the application. These frameworks provide tools for handling HTTP requests, routing, and rendering HTML templates.

2. Front-end Development: HTML, CSS, and JavaScript are essential for designing the user interface of the online examination system. You can use libraries like Bootstrap or Material-UI to create responsive and visually appealing web pages.

3. Database: Choose a database management system like MySQL, PostgreSQL, or SQLite to store user information, examination questions, and results. These databases allow you to query and manage data efficiently.

4. Face Recognition Library: To implement face recognition, you can use a Python library such as OpenCV or dlib. These libraries provide methods for face detection, face recognition, and facial landmarks identification.

5. Machine Learning Model: To recognize faces, you'll need a trained machine learning model. One popular approach is to use deep learning-based models, such as convolutional neural networks (CNNs), pretrained on large face recognition datasets like VGGFace or FaceNet. You can use libraries like TensorFlow or PyTorch to train and utilize these models.

6. User Authentication: Implement a secure user authentication system to handle user registration, login, and session management. You can use libraries like Flask-Login or Django's built-in authentication system to handle user authentication.

7. API Development: Create APIs to communicate between the front-end and back-end components. These APIs can handle user requests, validate user inputs, and interact with the database and face recognition components.

8. Deployment: Deploy your application to a web server or cloud platform, such as Heroku, AWS, or Azure, to make it accessible online. Ensure that the necessary dependencies and libraries are installed, and the environment is set up correctly.

Chapter – 2: LITERATURE SURVEY

2.1 LITERATURE SURVEY

Face Verification is a process of recognizing and matching faces. The use of biometrics for recognition systems have the aim of increasing human comfort and security in the scope of personal privacy and in a wider scope such as for an agency, the advantages of biometrics have many benefits and advantages compared to traditional systems such as manual signing, use of passwords, PINs, cards and the key that has been applied to entrance access, attendance, ATM machines and others. According to Syed Navaz & Mazumder Face identification systems also have advantages such as: Accurate, Cost-effective, non-invasive, using legacy data, the only biometrics suitable for use and made as a backup mechanism.

According to Dhavalsinh the face is the mind index. This is a complex multidimensional structure and requires computational techniques to recognize. When using an automatic system for facial recognition, computers are easily confused with changes in illumination, variations in poses and changes in the angle of the face. This attendance application with face recognition uses the Eigenface method to carry out facial recognition processes.

Face recognition shares a rich common literature with many of the domains. For recognition faces in video, face tracking is necessary, they are many dimensions while recognizing the faces there are problems we are going estimate these different head poses. The camera should be installed in the examination hall which detects the faces.

The model focuses on how face recognition incorporated with Radio Frequency Identification(RFID) detect the authorized students and counts as they get in and get out form the classroom. The system keeps the authentic record of every registered student. The system also keeps the data of every student registered for a particular course in the attendance log and provides necessary information according to the need.

A face recognition system based on local binary patterns and support vector machine for home security service robot is a detailed explanation of LBP and working of face recognition is a very flexible and mobile due to use of robot.

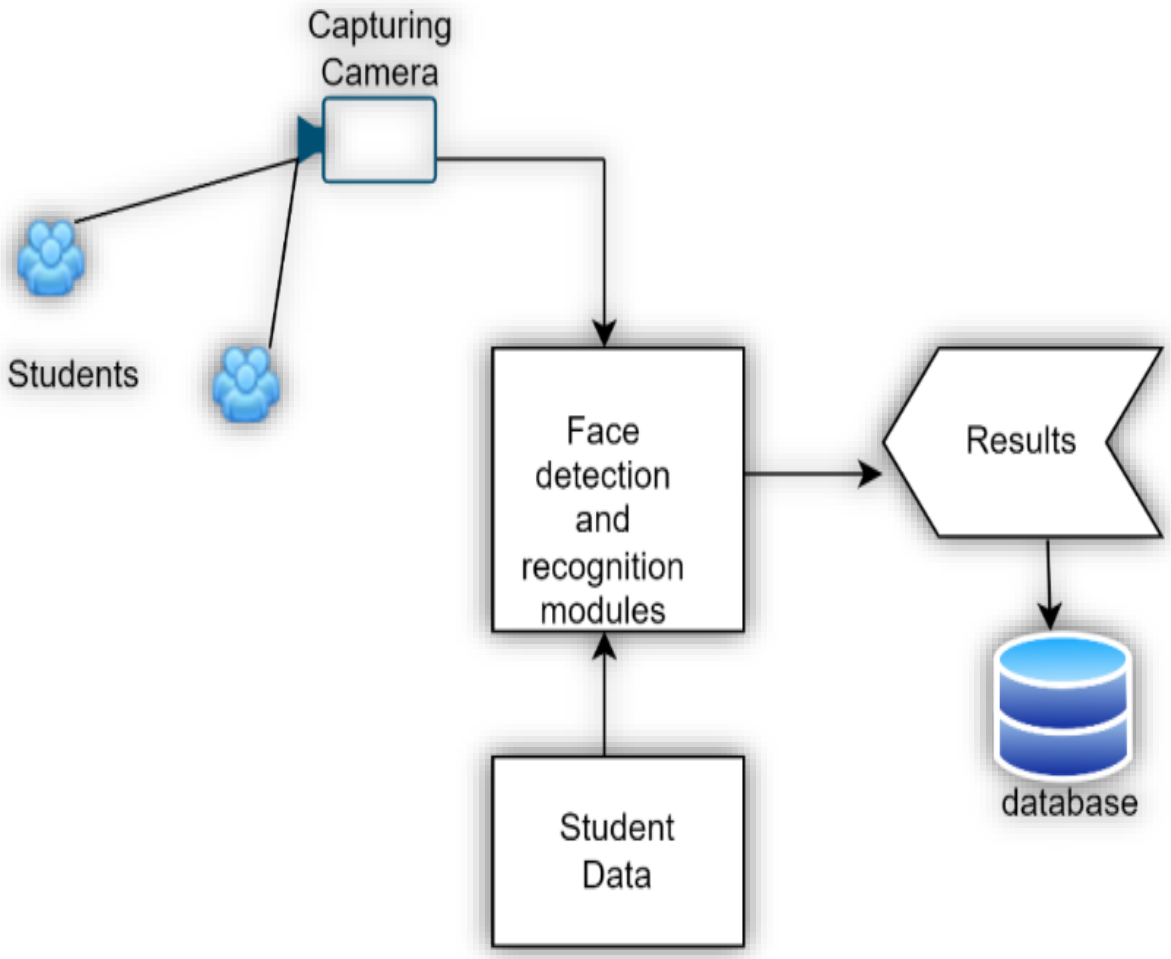
The power requirement is must for this system, no explanation for hardware and flow of system, additional cost for robotic components. An integrated face and face expression recognition system with high recognition rate was introduced. It is a two stage process, in the first stage users face is recognized and facial expression is recognized by using his/her personal expression database, this method can give excellent face recognition performance under various conditions including extreme expressions, strong non uniform lighting and partial occlusions.

2.2 EXISTING SYSTEM

The present system of attendance marking i.e., manually calling out the roll call by the faculty have quite satisfactorily served the purpose. With the change in the educational system with the introduction of new technologies in classroom such as virtual classroom, the traditional way of taking attendance may not be viable anymore. Even with rising number of course ofstudy offered by universities, processing of attendance manually could be time consuming.
Hence, in our project we aim at creating a system to take attendance using facial recognition technology in classroom and creating an efficient database to record them.

2.3 PROPOSED SYSTEM

Face is the most common biometric used by humans. Recognize the face from a general view point under different illumination conditions, facial expressions. A face recognition system is a technology capable of matching a human face from digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.
Face detection and Face recognition are often employed in various real-world application fields Facial detection is influenced by clarity of the image, coloured or black and white images. It can only support frontal detection of images.
The system will capture the images of the students by using face recognition algorithm mark the attendance in the sheet. It is a convenient and easy way for students and teacher.



2.4 FEASIBILITY STUDY

A feasibility study is an important step in determining the viability and practicality of a project or idea before investing significant time, effort, and resources into its development. In the context of an online examination system with face recognition using Python, here are some key aspects to consider in a feasibility study:

1. Technical Feasibility: Assess the technical requirements and challenges involved in implementing the proposed system. Consider factors such as the availability of necessary technologies, programming languages, libraries, and APIs required for face recognition. Evaluate if the required hardware (such as cameras) and software resources are readily available and compatible.

2. Financial Feasibility: Examine the estimated costs associated with developing and maintaining the system. This includes expenses related to software development, database management, hosting, and any licensing or subscription fees for third-party libraries or services. Compare these costs with the available budget or resources to determine financial feasibility.

3. Market Feasibility: Conduct market research to understand the demand and potential user base for an online examination system with face recognition. Identify the target audience (educational institutions, recruitment agencies, etc.) and assess their willingness to adopt such a system. Consider competitive analysis and market trends to determine if there is a viable market for the proposed system.

4. Legal and Ethical Feasibility: Consider legal and ethical aspects related to face recognition technology, privacy concerns, and data protection regulations. Evaluate if the proposed system complies with relevant laws and regulations, and if there are any potential ethical implications or risks associated with using face recognition for examination purposes.

5. Operational Feasibility: Assess the practicality and ease of implementing and managing the system on an operational level. Consider factors such as user experience, scalability, security measures, system performance, and the availability of technical support or expertise for maintaining and updating the system.

6. Time Feasibility: Evaluate the time required for system development, including research, design, implementation, testing, and deployment. Consider any project dependencies or constraints that may impact the development timeline. Assess if the project can be completed within a reasonable timeframe considering the available resources and project objectives.

By conducting a thorough feasibility study, you can identify potential challenges, risks, and benefits associated with developing an online examination system with face recognition using Python. This will help you make informed decisions and determine the overall feasibility and viability of the project before proceeding further.

2.5 TOOLS AND TECHNOLOGIES USED

Here are some of the tools and technologies commonly used in the development of an online examination system with face recognition using Python:

1. Python: Python is a popular programming language known for its simplicity and versatility. It provides a wide range of libraries and frameworks that facilitate various aspects of web development and machine learning, making it a suitable choice for implementing the system.

2. OpenCV: OpenCV (Open Source Computer Vision Library) is a powerful computer vision library that offers various functions for image and video processing, including face detection and recognition. It provides pre-trained models and algorithms for face-related tasks.

3. Deep Learning Libraries: TensorFlow and PyTorch are widely used deep learning libraries that provide efficient tools for training and deploying neural networks. These libraries can be used for building and training face recognition models using convolutional neural networks (CNNs) or other deep learning architectures.

4. Database Management Systems: MySQL, PostgreSQL, or SQLite are commonly used database management systems for storing and managing user information, examination questions, and results. These databases offer efficient querying and data management capabilities. In this project we used Excel to store data.

5. HTML, CSS, JavaScript: These web technologies are used for designing and developing the user interface of the online examination system. HTML defines the structure, CSS handles the styling, and JavaScript provides interactivity and dynamic functionality.

6. Version Control: Version control systems like Git are essential for managing source code and collaborating with other developers. They enable tracking changes, branching, and merging code.

7. API Development: Flask-RESTful or Django REST framework can be used to develop APIs for communication between the front-end and back-end components of the system. These frameworks simplify the process of creating RESTful APIs and handling HTTP requests.

2.6 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE:

1. Minimum 5 GB HDD space
2. Intel core 2 duo or above processor
3. 4 GB RAM required or above
4. X86 64-bit CPU (Intel/AMD architecture)
5. Power supply for backup

SOFTWARE:

1. Microsoft Windows 11
2. PyCharm 10.1

Chapter – 3

Software Requirements Specification

3.1 MODULE EXPLANATION

Face Recognition : This module incorporates face recognition capabilities to verify the identity of the test-takers. It uses computer vision algorithms and machine learning models to detect and match faces against the registered profiles. It ensures that the person taking the exam is the authorized user.

3.2 FUNCTIONAL REQUIREMENTS

Here are some functional requirements for the module of online exam attendance taking by face recognition using Python:

1. **User Registration:** Users should be able to register for the online examination system. They need to provide their personal information such as name and a unique username.

2. User Authentication: The system should authenticate users during the attendance taking process. Users should be required to log in with their registered username and password.

3. Face Enrollment: Users should have the option to enroll their faces into the system. This involves capturing their facial features using a camera and storing them securely in the database.

4. Face Recognition: The system should be able to recognize and match the enrolled faces during the attendance taking process. It should compare the captured face with the enrolled faces to verify the user's identity.

5. Attendance Recording: When a user attempts to take an online exam, the system should capture their face and record their attendance. The attendance record should include the user's name, timestamp, and the status of the face recognition process (e.g., successful or unsuccessful).

6. Attendance Validation: The system should validate the captured face against the enrolled faces to ensure accuracy. If the face recognition process fails to match the captured face with any enrolled face, the attendance should be flagged for manual verification.

3.3 NON-FUNCTIONAL REQUIREMENTS AND CONSTRAINTS

Non-functional requirements and constraints outline the quality attributes and limitations of the system. For an online exam attendance system using face recognition with Python, here are some non-functional requirements and constraints to consider:

1. Performance: The system should be able to handle a large number of users simultaneously and provide real-time face recognition for attendance. The response time should be fast, ensuring a seamless user experience.

2. Accuracy: The face recognition algorithm should have a high level of accuracy in identifying and verifying individuals' faces. The system should minimize false positives and false negatives to ensure accurate attendance records.

3. Security: The system should prioritize data security and protect the privacy of users' facial data. It should implement robust security measures to prevent unauthorized access, data breaches, and misuse of personal information.

4. Scalability: The system should be designed to accommodate future growth in terms of user numbers and increased usage. It should be scalable to handle additional users, exams, and concurrent sessions without significant performance degradation.

5. Usability: The system should have a user-friendly interface that is intuitive and easy to navigate. Users, including both administrators and test-takers, should be able to interact with the system effortlessly and understand its functionality.

6. Reliability: The system should be reliable and available for use during scheduled exam sessions. It should minimize downtime and ensure that users can access the system when needed. Backup and recovery mechanisms should be in place to safeguard against data loss.

7. Compatibility: The system should be compatible with a range of devices and browsers to allow users to access the platform using their preferred devices. It should support popular operating systems and browsers, ensuring a seamless user experience across different platforms.

8. Regulatory Compliance: The system should comply with relevant privacy laws, data protection regulations, and ethical guidelines. It should handle user data appropriately and obtain necessary consents for capturing and processing facial data.

Constraints:

1. Hardware Constraints: The system's face recognition functionality may require specific hardware, such as cameras or webcams, capable of capturing high-quality facial images. The availability and compatibility of such hardware may be a constraint.

2. Internet Connectivity: The system relies on internet connectivity for users to access the platform and perform face recognition attendance. Users should have a stable internet connection to ensure smooth operation.

3. System Requirements: The system's performance and functionality may depend on the hardware and software requirements, such as processing power, memory, and operating system compatibility. These requirements should be clearly defined and communicated to users.

4. Integration with Existing Infrastructure: If the system needs to integrate with existing systems or databases, there may be constraints related to data compatibility, data synchronization, and security protocols.

It is important to note that the specific non-functional requirements and constraints may vary based on the project's scope, organizational policies, and specific user needs.

Chapter – 4: SYSTEM DESIGN

INPUT DESIGN:

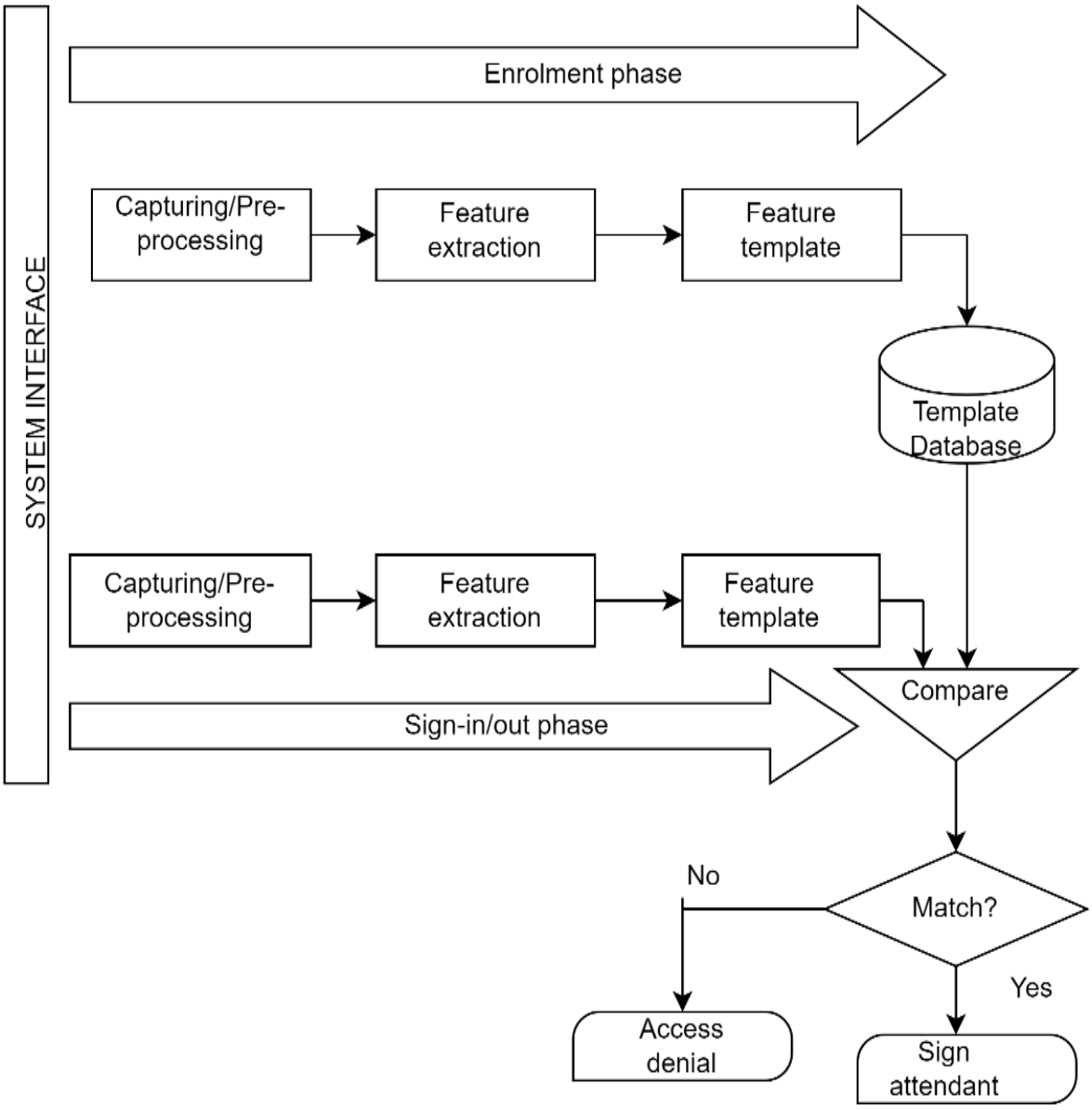
Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc. It should serve specific purpose effectively such as storing, recording,and retrieving the information.

OUTPUT DESIGN:

It is the immediate result of design activity. They are the documents and files that describe a design sufficiently that someone else can manufacture or actualise it exactly as intended. A design output is a drawing or specification or manufacturing instruction. Design outputs describe all the components, parts, and pieces that go into your medical device. Design outputsdescribe all assemblies and subassemblies of your product. Many of the toolbox functions arePYTHON M-files, a series of PYTHON statements that implement specialized image processing algorithms.

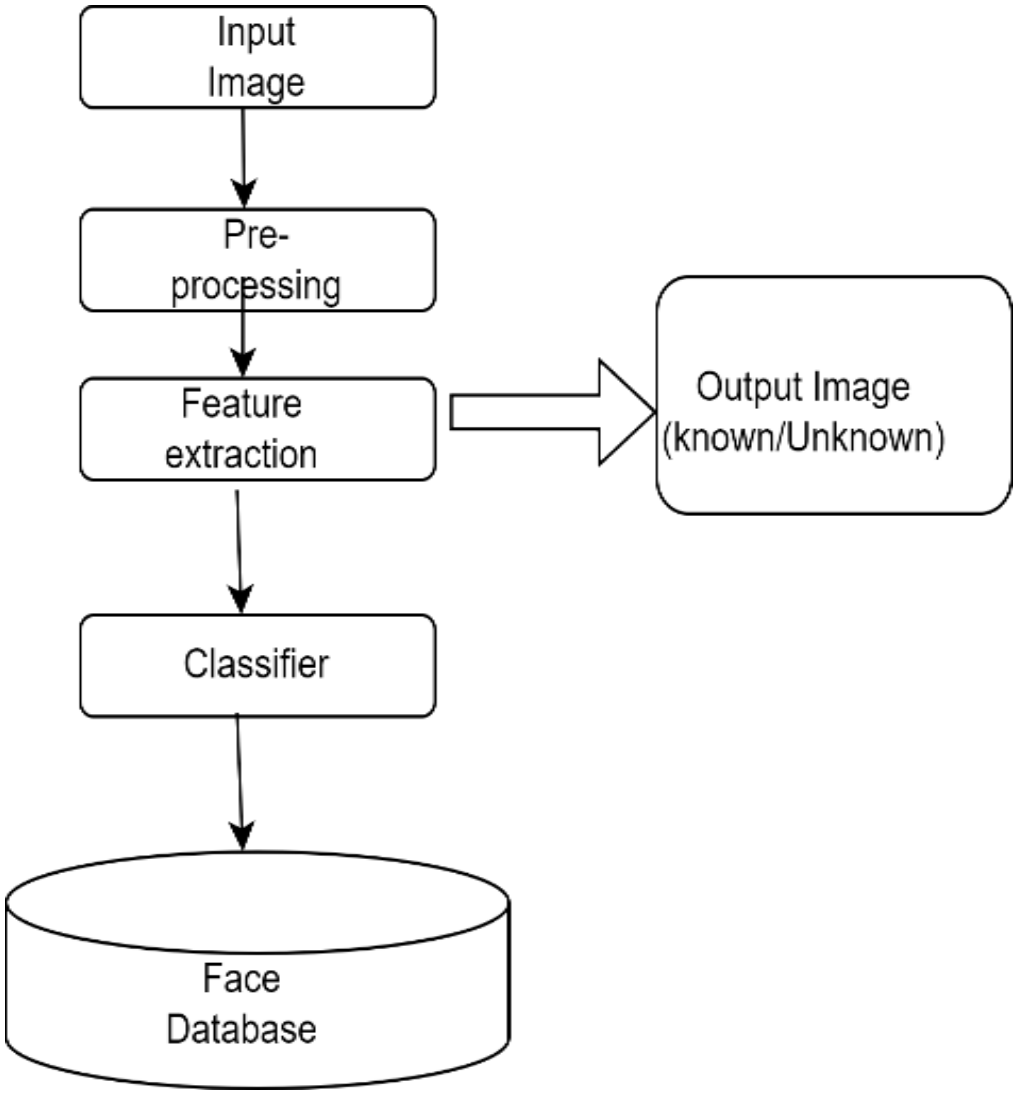
4.1 SYSTEM ARCHITECTURE

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and processflow diagrams.



4.2 CONTEXT DIAGRAM and Level -1 DFD

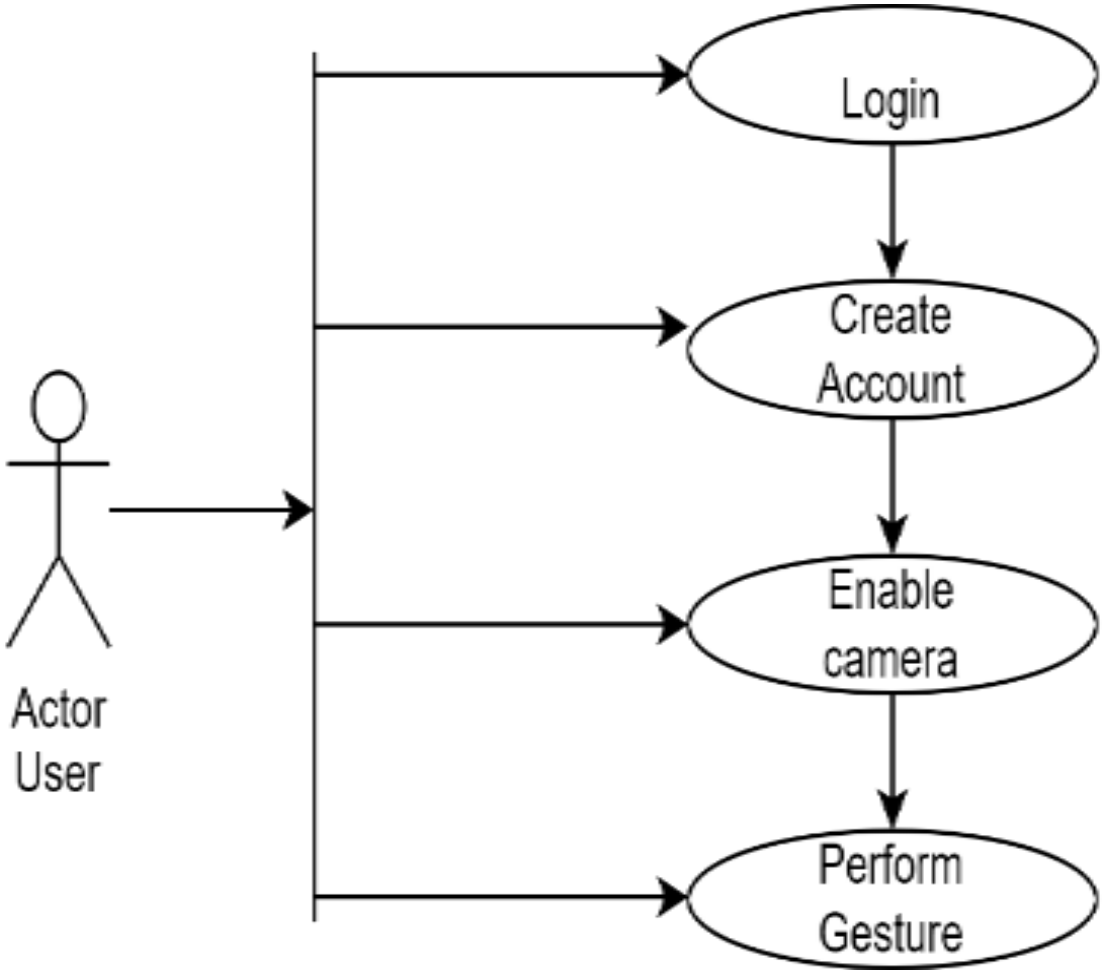
A data-flow diagram is a way of representing a flow of data through a process or a system. TheDFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM).



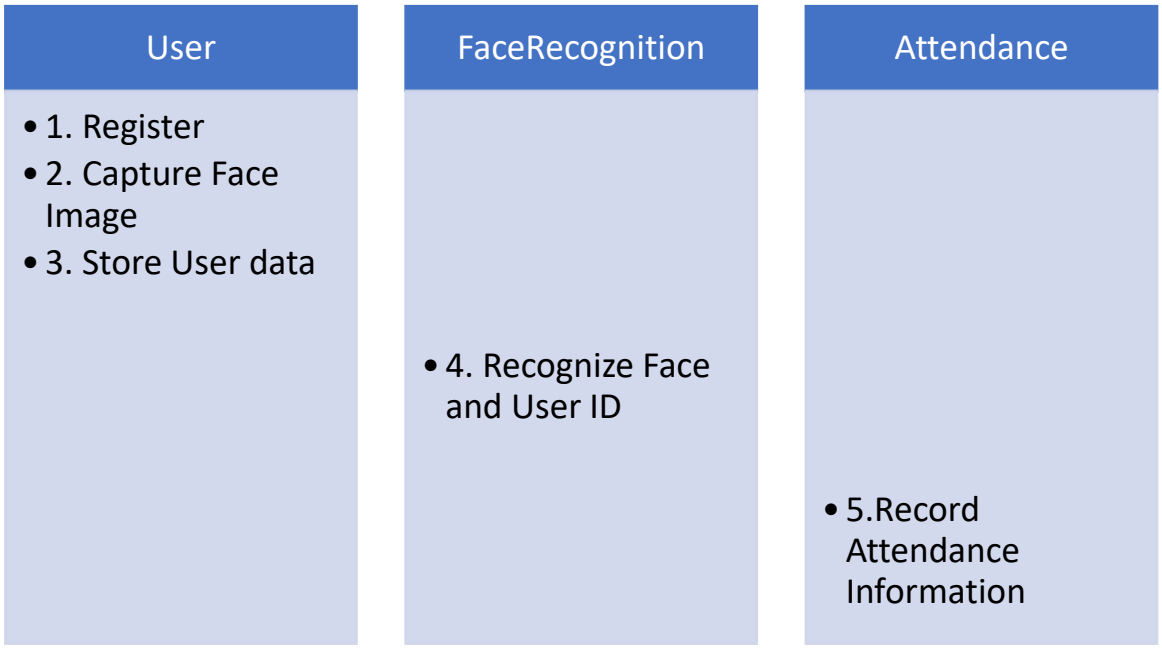
Chapter – 5 : Detailed Design.

5.1 USE CASE DIAGRAM

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the systemoperates internally.



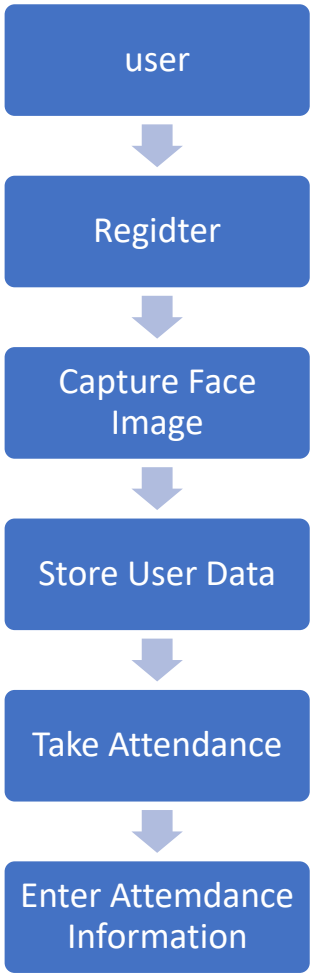
5.2 SEQUENCE DIAGRAMS



In the sequence diagram above, the interactions between the User, FaceRecognition, and Attendance components are shown. Here's a step-by-step explanation of the diagram:

1. The User component initiates the registration process.
2. The User captures their face image.
3. The User sends the captured face image to the Face Recognition component.
4. The Face Recognition component recognizes the face and identifies the user ID.
5. The Face Recognition component sends the recognized user ID back to the User.
6. The User component stores the user data, including the user ID and username.
7. The User component initiates the attendance taking process.
8. The Face Recognition component recognizes the face and identifies the user ID again.
9. The Face Recognition component sends

5.3 ACTIVITY DIAGRAM



In the activity diagram above, the User component is responsible for the registration and attendance taking process. Here's a step-by-step explanation of the diagram:

1. The User initiates the registration process.

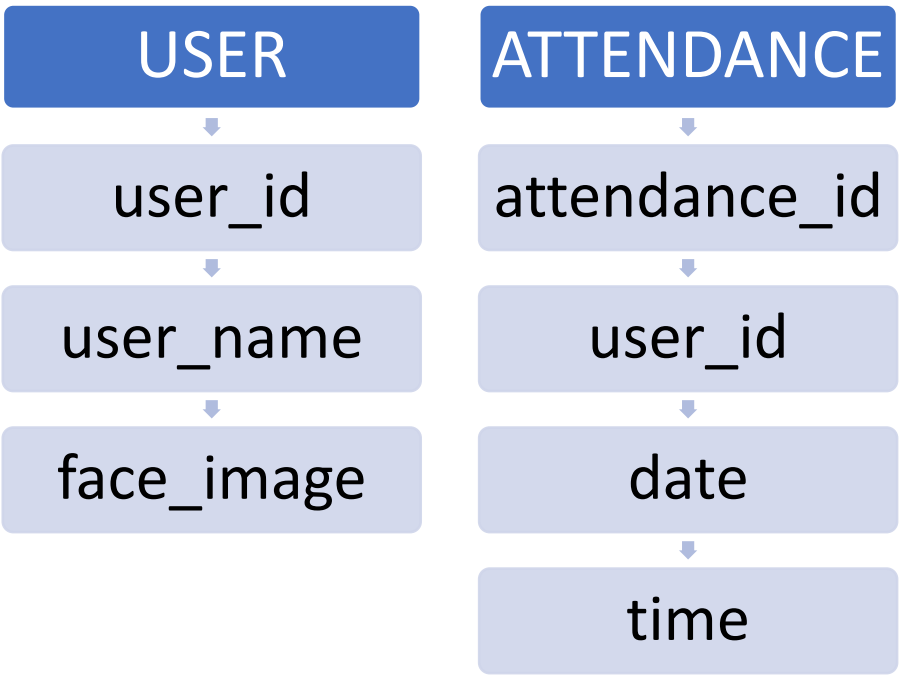
2. The User captures their face image.

3. The User stores their user data, including the user ID and username.

4. The User initiates the attendance taking process.

5. The User enters the attendance information, including the user ID, username, date, and time.

5.4 Class Diagrams



In the class diagram above, we have two classes: User and Attendance.

- The User class represents a user in the system. It has private attributes:
  - user id : Represents the unique identifier for each user.
  - username: Stores the name of the user.
  - face_image: Stores the face image of the user.

- The Attendance class represents an attendance record. It has private attributes:
  - attendance_id: Represents the unique identifier for each attendance record.
  - user_id: Stores the user ID of the attendee.
  - date: Stores the date of attendance.
  - time: Stores the time of attendance.

Both classes encapsulate their attributes, meaning they are accessed through methods (getters and setters) rather than directly accessing the attributes.

The User class is responsible for managing user information such as registration, capturing the face image, and providing access to user data.

The Attendance class handles attendance-related operations, including recording attendance by storing the user ID, username, date, and time.

Chapter – 6: Database Design

6.1 Database Design

To design the database for the online exam attendance system using face recognition in Python with an Excel database, you can consider the following structure:

User Data in Excel:
- Column A: user_id
- Column B: username
- Column C: password
- Column D: face_image_filename

Attendance Data in Excel:
- Column A: attendance_id
- Column B: user_name
- Column C: attendance _date
- Column D: attendance_time

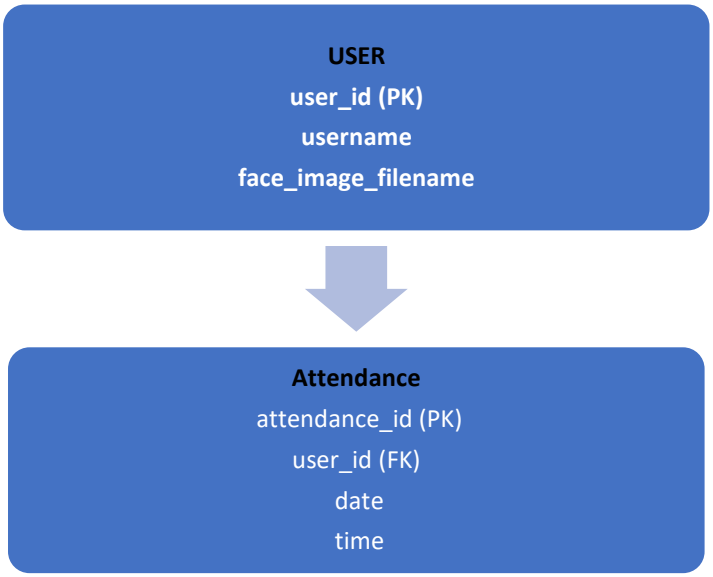
In this design, user data (including username, password, and the filename of the face image) will be stored in one Excel sheet, while attendance data (including attendance ID, user ID, exam ID, and attendance time) will be stored in another Excel sheet.

The user data sheet will have columns for user information, including their unique user ID, username, password, and the filename of their corresponding face image. Each row represents a user in the system.

The attendance data sheet will have columns for attendance information, including the attendance ID, user ID (foreign key referencing the user data sheet), exam ID, and the attendance time. Each row represents an attendance record for a specific user and exam.

By using Excel as the database, you can store and manage user data and attendance records in separate sheets. The face image files can be stored externally, with the file names referenced in the user data sheet.

6.2 E-R Diagram



In the updated E-R diagram above, we have two entities: User and Attendance.

- The User entity represents the users participating in the system. It has attributes like user_id (primary key), username, and face_image_filename. The face_image_filename is the filename of the user's face image stored externally.

- The Attendance entity represents the attendance records for users. It includes attributes like attendance_id (primary key), user_id (foreign key referencing User), date, and time.

The relationships between the entities are as follows:

- Each Attendance record is associated with a User.

- Each Attendance record corresponds to a single User.

Chapter – 7: IMPLEMENTATION

7.1 Model Code

MAIN

```
################## IMPORTING ###################

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox as mess

import tkinter.simpledialog as tsd

import cv2,os

import csv

import numpy as np

from PIL import Image

import pandas as pd

import datetime

import time

###################### FUNCTIONS################

def assure_path_exists(path):

    dir = os.path.dirname(path)

    if not os.path.exists(dir):

os.makedirs(dir)
```

```
def tick():
    time_string = time.strftime('%H:%M:%S')
    clock.config(text=time_string)
    clock.after(200,tick)
######################################################################
def contact():
    mess._show(title='Contact us', message="Please contact us on :
'prempeter07@gmail.com' ")
######################################################################
def check_haarcascadefile():
    exists = os.path.isfile("haarcascade_frontalface_default.xml")
    if exists:
        pass
    else:
        mess._show(title='Some file missing', message='Please contact us for help')
        window.destroy()
######################################################################
def save_pass():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
```

```
        key = tf.read()
    else:
        master.destroy()
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered successfully!!')
            return
    op = (old.get())
    newp= (new.get())
    nnewp = (nnew.get())
    if (op == key):
        if(newp == nnewp):
            txf = open("TrainingImageLabel\psd.txt", "w")
            txf.write(newp)
        else:
```

```
            mess._show(title='Error', message='Confirm new password again!!!')
            return
    else:
        mess._show(title='Wrong Password', message='Please enter correct old password.')
        return
    mess._show(title='Password Changed', message='Password changed successfully!!')
    master.destroy()
####################################################################################
def change_pass():
    global master
    master = tk.Tk()
    master.geometry("400x160")
    master.resizable(False,False)
    master.title("Change Password")
    master.configure(background="white")
    lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, ' bold '))
     lbl4.place(x=10,y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='*')
```

```
old.place(x=180,y=10
default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
    exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists1:
        df = pd.read_csv("StudentDetails\StudentDetails.csv")
    else:
        mess._show(title='Details Missing', message='Students details are missing, please
check!')
        cam.release()
        cv2.destroyAllWindows()
        window.destroy()
    while True:
        ret, im = cam.read()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, 1.2, 5)
        for (x, y, w, h) in faces:
            cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
            serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
```

```
if (conf < 50):
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
            timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
            ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
            ID = str(ID)
            ID = ID[1:-1]
            bb = str(aa)
            bb = bb[2:-2]
            attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]
        else:
            Id = 'Unknown'
            bb = str(Id)
        cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
    cv2.imshow('Taking Attendance', im)
    if (cv2.waitKey(1) == ord('q')):
        break
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")
if exists:
```

```
    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
        writer = csv.writer(csvFile1)
        writer.writerow(attendance)
    csvFile1.close()
else:
    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
        writer = csv.writer(csvFile1)
        writer.writerow(col_names)
        writer.writerow(attendance)
    csvFile1.close()
with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:
    reader1 = csv.reader(csvFile1)
     for lines in reader1:
        i = i + 1
        if (i > 1):
            if (i % 2 != 0):
                iidd = str(lines[0]) + '   '
                tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))
csvFile1.close()
 cam.release()
cv2.destroyAllWindows()
```

```
################## USED STUFFS ########################################
global key
key = ''
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
day,month,year=date.split("-")
mont={'01':'January',
      '02':'February',
      '03':'March',
       '04':'April',
      '05':'May',
      '06':'June',
      '07':'July',
      '08':'August',
      '09':'September',
      '10':'October',
       '11':'November',
       '12':'December'
      }
```

```
########## GUI FRONT-END ###########################

window = tk.Tk()

window.geometry("1280x720")

window.resizable(True,False)

window.title("Attendance System")

window.configure(background='#fce3cf')

frame1 = tk.Frame(window, bg="#fcefd2")

frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#fcefd2")

frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance System"
,fg="white",bg="#780101" ,width=55 ,height=1,font=('times', 29, ' bold '))

message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")

frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")

frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+"  |  ",
fg="white",bg="#031980" ,width=55 ,height=1,font=('times', 22, ' bold '))

datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="white",bg="#031980" ,width=55 ,height=1,font=('times',
22, ' bold '))
```

```
clock.pack(fill='both',expand=1)

tick()

head2 = tk.Label(frame2, text="                For New Registrations                ",
fg="black",bg="#90e5fc" ,font=('times', 17, ' bold ') )

head2.grid(row=0,column=0)

head1 = tk.Label(frame1, text="                For Already Registered                ",
fg="black",bg="#90e5fc" ,font=('times', 17, ' bold ') )

head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter ID",width=20  ,height=1  ,fg="black"
,bg="#fcefd2" ,font=('times', 17, ' bold ') )

lbl.place(x=80, y=55)

txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))

txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name",width=20  ,fg="black"  ,bg="#fcefd2"
,font=('times', 17, ' bold '))

lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ')  )

txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images  >>>  2)Save Profile"
,bg="#fcefd2" ,fg="black"  ,width=39 ,height=1, activebackground = "yellow"
,font=('times', 15, ' bold '))

message1.place(x=7, y=230)
```

```
message = tk.Label(frame2, text="" ,bg="#fcefd2" ,fg="black"  ,width=39,height=1,
activebackground = "yellow" ,font=('times', 16, ' bold '))

message.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance",width=20  ,fg="black"  ,bg="#fcefd2"
,height=1 ,font=('times', 17, ' bold '))

lbl3.place(x=100, y=115)

res=0

exists = os.path.isfile("StudentDetails\StudentDetails.csv")

if exists:

    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:

        reader1 = csv.reader(csvFile1)

        for l in reader1:

            res = res + 1

    res = (res // 2) - 1

    csvFile1.close()

else:

    res = 0

message.configure(text='Total Registrations till now  : '+str(res))


###################### MENUBAR ################################

menubar = tk.Menu(window,relief='ridge')

filemenu = tk.Menu(menubar,tearoff=0)
```

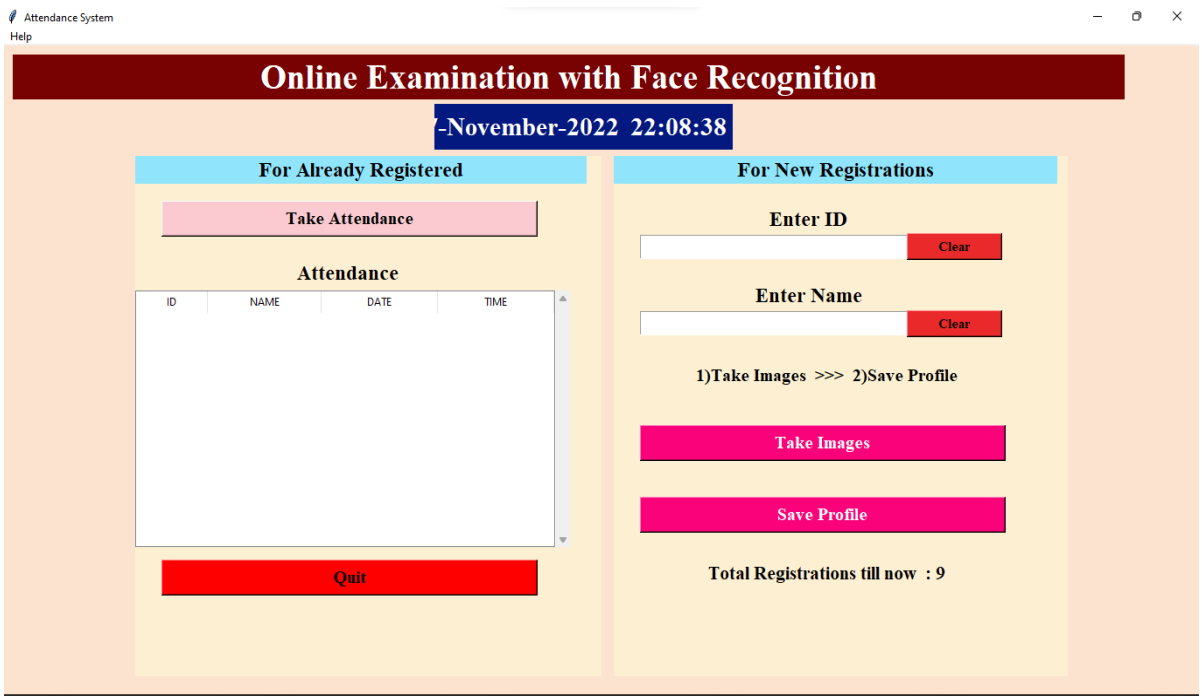
```
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)
################## TREEVIEW ATTENDANCE TABLE ####################
tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=82)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ='ID')
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')

######################## SCROLLBAR ################################
scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)

###################### BUTTONS ##################################
```

```
clearButton = tk.Button(frame2, text="Clear", command=clear  ,fg="black"
,bg="#ea2a2a"  ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

clearButton.place(x=335, y=86)

clearButton2 = tk.Button(frame2, text="Clear", command=clear2  ,fg="black"
,bg="#ea2a2a"  ,width=11 , activebackground = "white" ,font=('times', 11, ' bold '))

clearButton2.place(x=335, y=172)

takeImg = tk.Button(frame2, text="Take Images", command=TakeImages  ,fg="white"
,bg="#fa027a"  ,width=34  ,height=1, activebackground = "white" ,font=('times', 15, '
bold '))

takeImg.place(x=30, y=300)

trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="white"
,bg="#fa027a"  ,width=34  ,height=1, activebackground = "white" ,font=('times', 15, '
bold '))

trainImg.place(x=30, y=380)

trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages
,fg="black"  ,bg="#facad0"  ,width=35  ,height=1, activebackground = "white"
,font=('times', 15, ' bold '))

trackImg.place(x=30,y=50)

quitWindow = tk.Button(frame1, text="Quit", command=window.destroy  ,fg="black"
,bg="red"  ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold
'))

quitWindow.place(x=30, y=450)
```

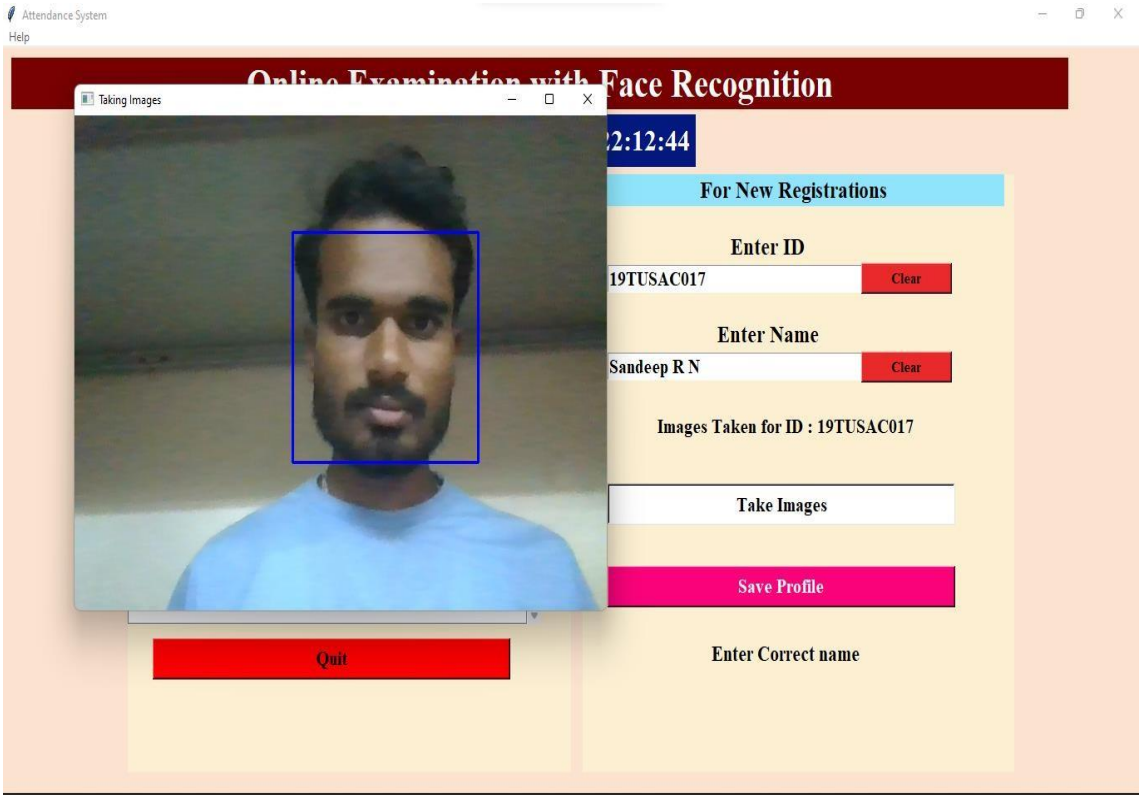
###################### END ######################################

7.2 Screen Shots

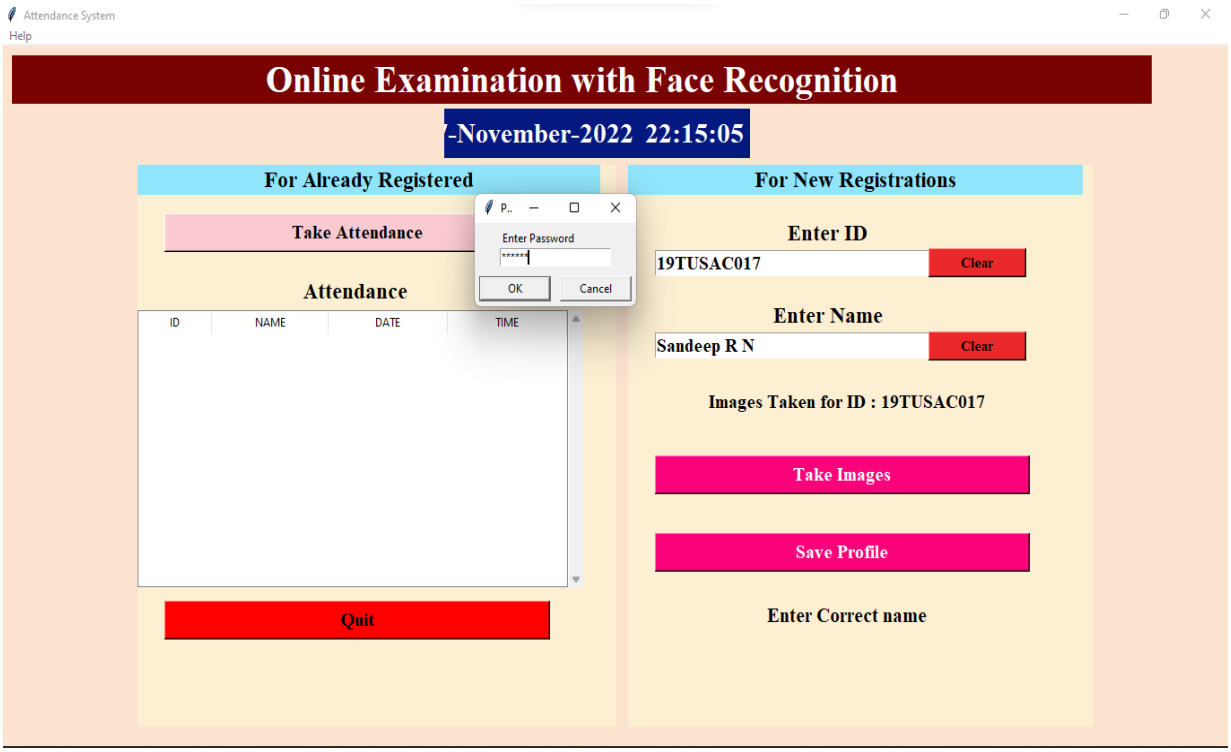
1.HOME PAGE



2.TAKEIMAGES

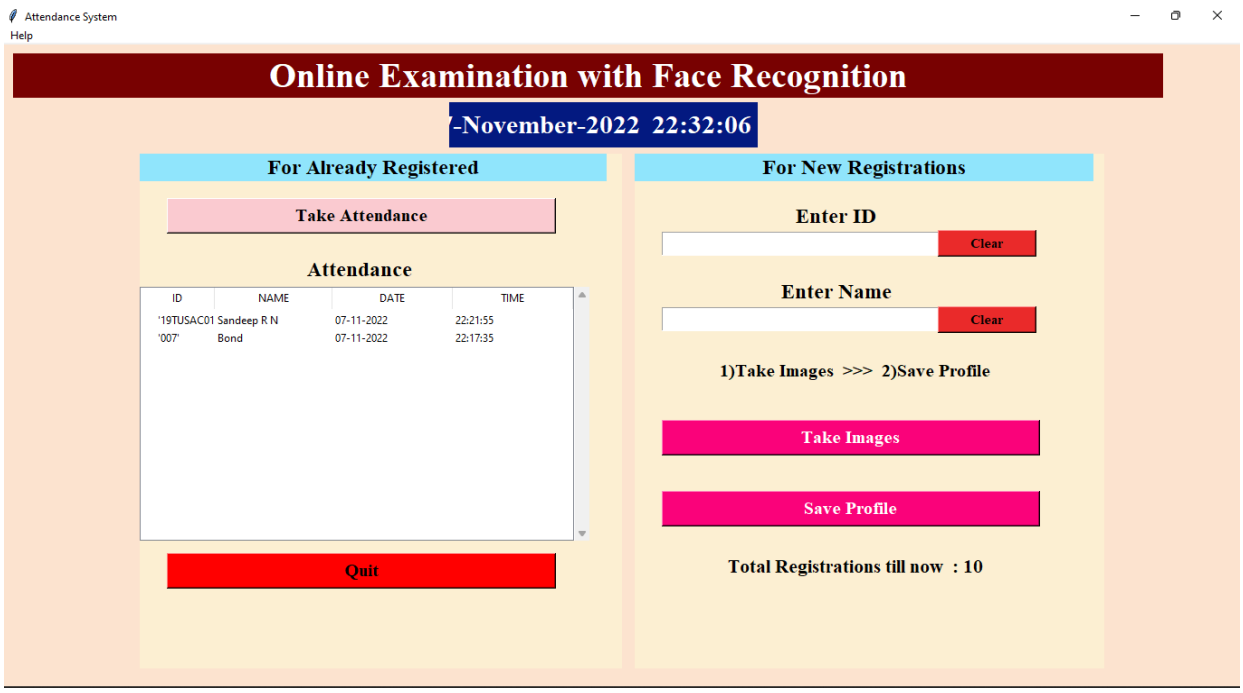


3.PASSWORD PAGE

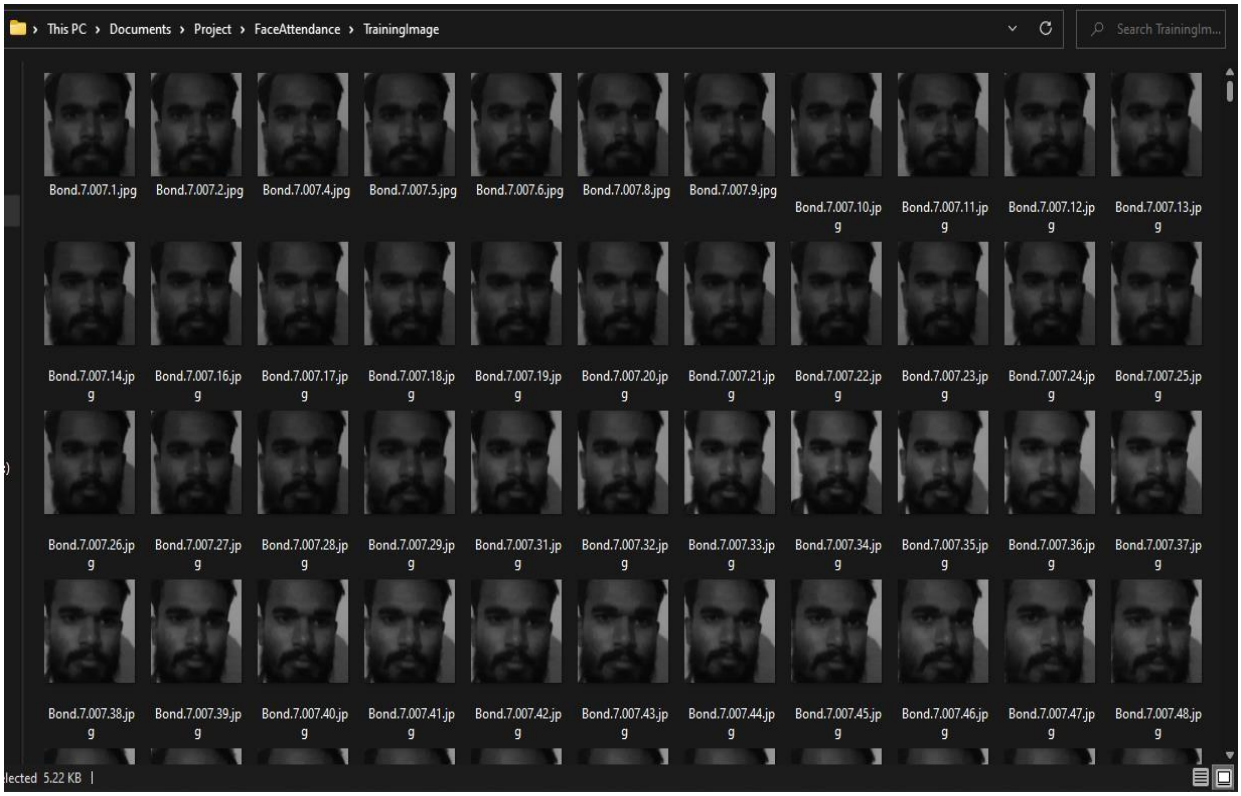


4.TAKE ATTENDENCE

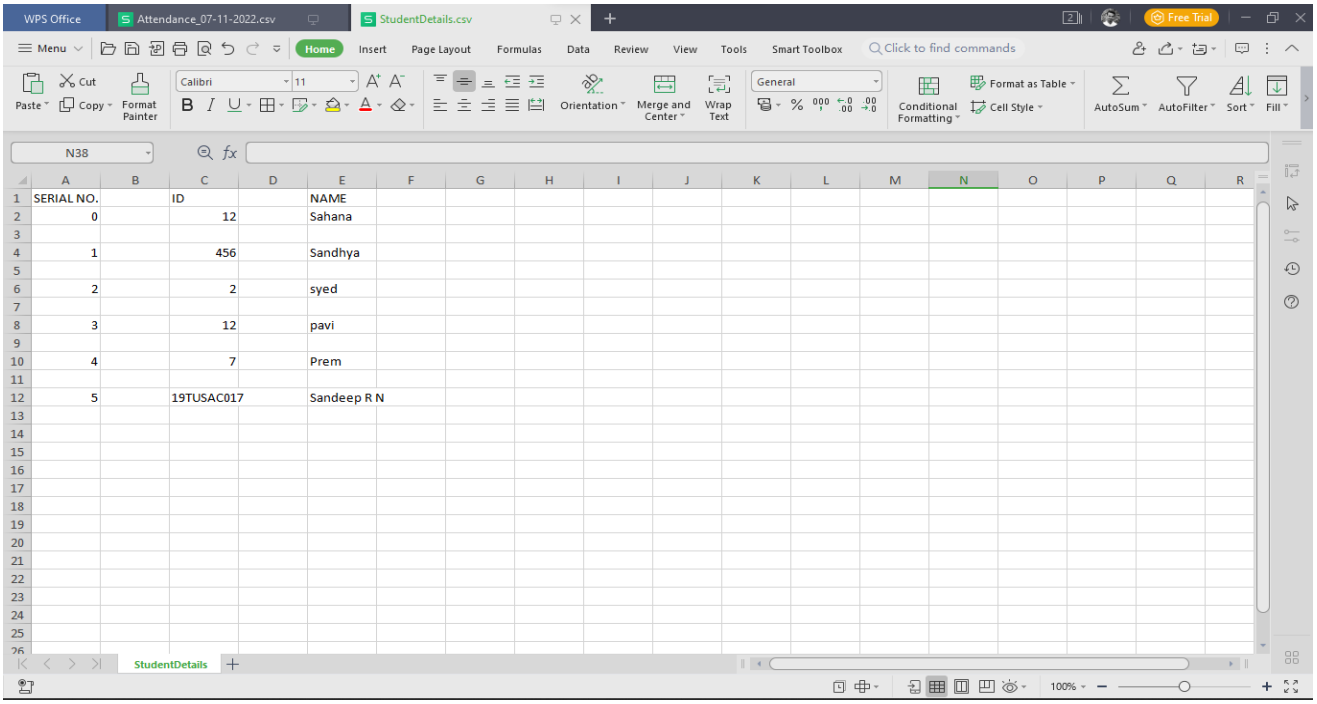
5.ATTENDENCE LIST



6.TRAINING IMAGES

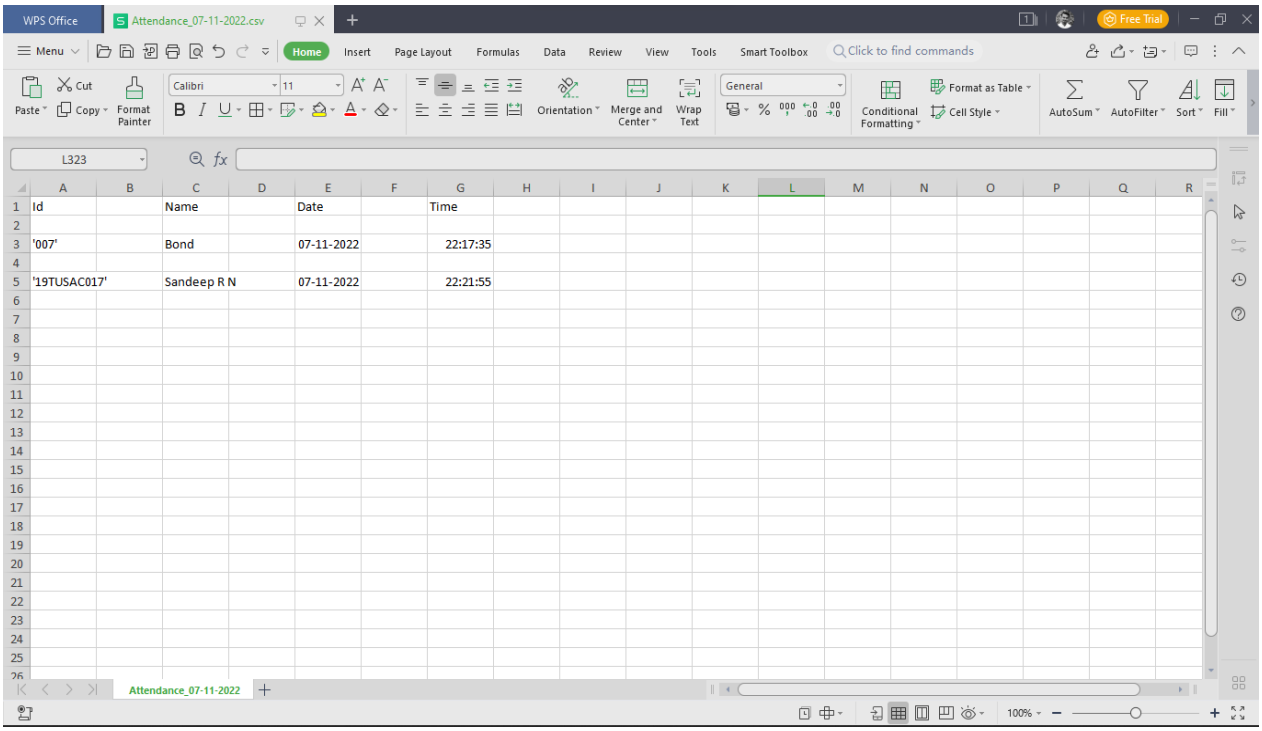


7.STUDENT LIST XML



SERIAL NO.	ID	NAME
0	12	Sahana
1	456	Sandhya
2	2	syed
3	12	pavi
4	7	Prem
5	19TUSAC017	Sandeep R N

8.ATTENDENCE LIST XML



Id	Name	Date	Time
'007'	Bond	07-11-2022	22:17:35
'19TUSAC017'	Sandeep R N	07-11-2022	22:21:55

Chapter – 8 : SOFTWARE TESTING

8.1 Testing

Software testing is an essential part of developing a reliable and robust online exam attendance system using face recognition in Python. Here are some key aspects of software testing for this type of system:

1. Unit Testing: Unit testing involves testing individual components or units of code in isolation to ensure they function correctly. In the case of face recognition, you can write unit tests to verify the accuracy and reliability of the face recognition algorithm. Unit tests can be performed on functions or methods responsible for capturing and processing face images, performing face recognition, and validating the results.

2. Integration Testing: Integration testing focuses on testing the interaction between different components or modules of the system. For the online exam attendance system, integration testing ensures that the face recognition module integrates seamlessly with other system components such as user authentication, attendance marking, and exam management. This includes verifying that user data is properly synchronized with the face recognition module and that attendance records are accurately marked.

3. Functional Testing: Functional testing verifies that the system's functionalities work as intended. In the context of the online exam attendance system, functional testing ensures that users can successfully login, capture their face images for recognition, mark attendance, and access exam questions. It involves testing different scenarios and user interactions to ensure the system functions correctly and provides the expected outcomes.

4. Performance Testing: Performance testing evaluates the system's performance under various load conditions. In the case of online exam attendance with face recognition, performance testing can involve simulating a large number of concurrent users attempting to capture their face images and mark attendance simultaneously. It helps identify any performance bottlenecks, such as slow response times or resource limitations, and allows for optimization and scalability improvements.

5. Security Testing: Security testing is crucial to ensure the system's protection against unauthorized access and data breaches. In the online exam attendance system, security testing involves assessing the security measures in place to safeguard user data and facial images. It includes testing for vulnerabilities like SQL injection, session hijacking, and unauthorized access to sensitive information.

6. Usability Testing: Usability testing focuses on evaluating the system's user interface and user experience. For the online exam attendance system, usability testing ensures that the interface is intuitive, easy to navigate, and provides clear instructions for capturing face images and marking attendance. It helps identify any usability issues and provides insights for improving the user experience.

7. Regression Testing: Regression testing is performed to ensure that system updates or modifications do not introduce new issues or cause existing functionalities to fail. Whenever changes are made to the online exam attendance system, regression testing should be conducted to verify that the face recognition module and other functionalities continue to work as expected.

8.2 Case Study:

Implementation of an Online Attendance System Using Face Recognition

1. Introduction:

In this case study, we explore the implementation of an online attendance system using face recognition technology. The aim of the system is to automate the attendance tracking process in an efficient and reliable manner. We will discuss the challenges faced, the solution architecture, and the benefits of using face recognition technology for attendance management.

2. Problem Statement:

Traditional attendance systems, such as manual sign-in sheets or card-based systems, are often time-consuming, prone to errors, and susceptible to misuse. To address these issues, the organization decided to implement an online attendance system that utilizes face recognition technology to accurately and securely track attendance.

3. Challenges:

a. Accuracy: Ensuring high accuracy in recognizing faces and eliminating false positives and negatives.

b. Scalability: Designing a system that can handle a large number of users and process attendance data in real-time.

c. Security: Implementing robust security measures to protect the personal data of individuals and prevent unauthorized access.

d. Usability: Creating a user-friendly interface for both administrators and attendees to interact with the system easily.

4. Solution Architecture:

a. Data Collection: Capture images of individuals during the registration process and store them securely in a database.

b. Face Detection: Use computer vision algorithms to detect and extract faces from the images.

c. Face Recognition: Compare the extracted faces with the enrolled faces in the database to identify individuals accurately.

d. Attendance Tracking: Match the recognized faces with the registered attendees and record their attendance in a centralized system.

e. Real-time Processing: Process attendance data in real-time to generate reports and notifications for administrators.

f. Security Measures: Implement encryption techniques, access controls, and secure storage practices to protect sensitive data.

5. Implementation:

a. Hardware: Deploy cameras or other devices capable of capturing high-quality images for face recognition.

b. Software Development: Develop custom software using machine learning algorithms for face detection and recognition.

c. Integration: Integrate the attendance system with existing infrastructure such as databases, user management systems, and notification systems.

d. Testing and Refinement: Conduct rigorous testing to ensure system accuracy, scalability, and usability. Refine the system based on feedback and performance evaluation.

6. Benefits:

a. Accuracy: Face recognition technology provides a highly accurate method for attendance tracking, minimizing errors and false records.

b. Efficiency: The system automates the attendance process, saving time and effort for both administrators and attendees.

c. Real-time Tracking: Attendance records are updated in real-time, allowing immediate visibility of attendance status.

d. Security: Face recognition enhances security by reducing the possibility of proxy attendance and identity fraud.

e. Data Insights: The system can generate comprehensive reports and analytics, providing valuable insights into attendance patterns and trends.

7. Conclusion:

Implementing an online attendance system using face recognition technology offers significant advantages over traditional methods. It improves accuracy, efficiency, and security while providing real-time tracking and valuable data insights. However, it is crucial to address challenges related to accuracy, scalability, security, and usability during the implementation process. With careful planning, testing, and refinement, organizations can successfully deploy a robust online attendance system using face recognition technology**.**

Chapter - 9 : CONCLUSION

Face recognition technologies have been associated generally with very costly top secure applications. Certain applications of face recognition technology arenow cost effective, reliable and highly accurate. As a result there are no technological or financial barriers for stepping from the pilot project to widespread deployment.

- Face recognition is a difficult problem.
- Pre-processing is very important.
- It is not enough to use only global features.
- Better results can be obtained with different classifications(eigenfeatures).
- A good face detection system may need to use more than one method.

Face recognition has many potential applications

Appendix A : FUTURE ENHANCEMENTS

The system can be modified to detect and recognize a broader class of faces. Gender classification, happiness index and emotion analysis may be obtained.

We also need to discuss the approach of camera planning based on the result of the position estimation in order to improve face detection effectiveness. In further work, we intend to improve face detection effectiveness by using the interaction among our system, the students and the teacher on the other hand, our system can be improved by integrating viedo-streamingservice and lecture archiving system, to provide more profound applications in the field of distance education, Course Management System, Support for Faculty Development(FD).

As a future scope eyebrows also be included which increasing the accuracy of system. Also thebasic method can be modified by getting better results.

Future studies must invest aggregation of data from different sensors.The future scope includestesting of the proposed method larger databases like CK+ are databases.

- Point out the major face features.
- Gives an accurate result.
- Apply some more elements regarding recognition.
- Improve the interface.
- Testing and deployment

Appendix B: User Manual

1. System Requirements:

- Operating System: Windows, macOS, or Linux
- Python 3.x installed
- Webcam or camera connected to the computer

2. Installation:

- Download the system files from the provided source or repository.
- Install the required Python libraries by running the command `pip install -r requirements.txt`.
- Launch the system by running the command `python main.py`.

3. User Registration and Login:

- Open the system and click on the "Register" button to create a new user account.
- Enter a username, password, and capture your face image using the webcam.
- Once registered, click on the "Login" button and enter your username and password.

4. Taking Exam Attendance:

- After logging in, you will see a list of available exams.
- Select the desired exam and click on the "Start Attendance" button.
- Allow access to your webcam for face recognition.
- Position yourself properly in front of the camera and ensure good lighting.
- The system will automatically capture your face image and perform face recognition.
- If your face is recognized, the attendance will be marked, and you will proceed to the exam.
- If face recognition fails, try again by capturing a clear face image.

Appendix C: BIBLIOGRAPHY

Anuradha.S.G, Kavya.B, Akshatha.S, Kothapalli Jyothi, Gudipati Ashalatha, "Automated Face Detection & Recognition for Detecting Impersonation of Candidate in Examination System" –International Journal of Scientific & Engineering Research, vol.7,Issue 3,March-2016.

Akshat Shrivastava, Ashray Parmar, Anurag Sen, Ayush Soun, Suresh Kapare, "Face Recognition

Classification in Proctored Exam" – International Journal of Creative Research Thoughts(IJCRT) ISSN:2320-2882 vol.9,Issue 6, June-2021

FERENCES

- Smitha, Pavithra S Hegde, Afshin, "Face Recognition based Attendance Management System" – International Journal of Engineering Research & Technology(IJERT) ISSN:2278- 0181 vol.9,Issue 05, May-2020
- Ayham Fayyoumi, Anis Zarrad, "Novel Solution Based on Face Recognition to Address Identity Theft and Cheating in Online Examination Systems" – Article in Advances in Internetof Things April-2014
- Akshat Shrivastava, Ashray Parmar, Anurag Sen, Ayush Soun, Suresh Kapare, "Face Recognition Classification in Proctored Exam" – International Journal of Creative Research Thoughts(IJCRT) ISSN:2320-2882 vol.9,Issue 6, June-2021
- Arief Sukmandhani, Indrajani Sutedja, "Face Recognition Method for Online Exams" – Conference Paper, August-2019
- Nikhil Sharma, Aditi Garg, Aviral Aeron, Faraj Chisti, "Online Examination With Face Recognition" – International Journal of Engineering Applied Sciences and Technology, ISSN:2455-2143 vol.4,Issue 12,2020
- Anuradha.S.G, Kavya.B, Akshatha.S, Kothapalli Jyothi, Gudipati Ashalatha, "Automated Face Detection & Recognition for Detecting Impersonation of Candidate in Examination System" – International Journal of Scientific & Engineering Research, vol.7,Issue 3,March- 2016.