

**Summer Project On**  
**Face Recognition Attendance System**  
**By**  
**Chinmay Bhagwat (2021510005)**  
**Rutik Dhere (2021510011)**

Under the guidance of  
**Internal Supervisor**

**Prof. Pooja Raundale**



Department of Master Of Computer Application  
Sardar Patel Institute of Technology  
Autonomous Institute Affiliated to Mumbai University  
2021-22

## **CERTIFICATE OF APPROVAL**

This is to certify that the following students

**Chinmay Bhagwat (2021510005)**  
**Rutik Dhere (2021510011)**

Have satisfactorily carried out work on the project  
entitled

**“Face Recognition Attendance System”**

Towards the fulfilment of project, as laid down  
by

Sardar Patel Institute of Technology  
during year  
2021-22.

Project Guide:  
Pooja Raundale

## **PROJECT APPROVAL CERTIFICATE**

This is to certify that the following students

**Chinmay Bhagwat (2021510005)**  
**Rutik Dhere (2021510011)**

Have successfully completed the Project report on

**“Face Recognition Attendance System”,**

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,  
ANDHERI (W), MUMBAI

INTERNAL EXAMINER

EXTERNAL EXAMINER

HEAD OF DEPARTMENT

PRINCIPAL

# Contents

<b>Abstract</b>	i
<b>Objectives</b>	i
<b>List Of Figures</b>	ii
<b>List Of Tables</b>	ii
<b>1 Introduction</b>	1
1.1 Problem Definition . . . . .	1
1.2 Objectives and Scope . . . . .	1
1.2.1 Objectives . . . . .	1
1.2.2 Scope . . . . .	1
1.3 Existing System . . . . .	2
1.4 Proposed System . . . . .	2
1.5 System Requirements . . . . .	3
<b>2 Software Requirement Specification (SRS) and Design</b>	4
2.1 Purpose . . . . .	4
2.2 Definition . . . . .	4
2.3 Overall Description . . . . .	4
2.3.1 Product Functions . . . . .	4
2.3.2 User Characteristics . . . . .	5
<b>3 Project Implementation and Testing</b>	8
3.1 Register . . . . .	8
3.2 Login . . . . .	9
3.3 Homepage . . . . .	10
3.4 Students Section . . . . .	11
3.5 Attendance section . . . . .	12
3.6 Face recognition window . . . . .	13
3.7 Train data . . . . .	14
3.8 View Added data . . . . .	15
3.9 Code 1 . . . . .	16
3.10 Code 2 . . . . .	16
3.11 Code 3 . . . . .	17
<b>4 Test Cases</b>	18
<b>5 Limitations</b>	19
<b>6 Future Enhancements</b>	19
<b>7 User Manual</b>	20

<b>8 Bibliography</b>	<b>21</b>
8.1 Web References . . . . .	21

## **Abstract**

The Attendance App is an desktop and PC software application which can be used for making the advanced way for marking attendance of students by storing their data and marking attendance by recognising face.

The User(teacher) can view and keep track of the attendance of the students through face recognition in the application such that there is no misplaced any record between both the students and teachers during the important sessions.

The app will provide Teachers and coordinators the required accurate attendance and students details such as Personal details, Educational details, etc. for academic processes

## **Objectives**

The windows desktop based Application "Attendance App" is used

- To provide students a user friendly platform for the attendance process.
- To provide a convenient communication between Students-Teacher-HOD.
- To provide a facility to view records, details of students for academic process easily.
- To provide a platform to keep personal information updated

---

## List of Figures

3.1.1 Register user . . . . .	8
3.2.1 Login View . . . . .	9
3.3.1 Homepage menu . . . . .	10
3.4.1 Adding and viewing students records . . . . .	11
3.5.1 View attendances and download as CSV from here . . . . .	12
3.6.1 Recognise registered face. . . . .	13
3.7.1 Training the added dataset of face images . . . . .	14
3.8.1 Photos data . . . . .	15

## List of Tables

1.5.1 Hardware Requirements on Server Side . . . . .	3
1.5.2 Hardware Requirements on Client Side . . . . .	3
1.5.3 Software Requirements on Server Side . . . . .	3
1.5.3 Software Requirements on Client Side . . . . .	3
4.2.1 Use Case Table - Register . . . . .	6
4.2.2 Use Case Table - Login . . . . .	6
4.2.3 Use Case Table - Update Profile . . . . .	6
4.2.4 Use Case Table - Update attendance . . . . .	7
4.2.5 Use Case Table - Train data . . . . .	7
4.2.6 Use Case Table - Helpdesk . . . . .	7
4.2.7 Use Case Table - Recognise face . . . . .	7
6.1 Test Case - Login and Register . . . . .	18
6.2 Test Case - Others . . . . .	18

Project Name

Student Name (UCID)

---

## 1 Introduction

### 1.1 Problem Definition

To eliminate redundancy in collecting data in current physical attendance marking process and to squash the time gap and miscommunication in the current process.

### 1.2 Objectives and Scope

#### 1.2.1 Objectives

The windows based application "Attendance App" is

- To provide records saving system in which communication, application and execution of the attendance process is uncomplicated.
- To keep required details regarding students handy
- To provide students a easy and convenient process of marking attendance for the classroom sessions.
- To provide a better connection between the teacher and students.

#### 1.2.2 Scope

The student can provide his/her details in the profile and view for various notifications attendance is also shown they can add their faces and train the data on the app.

In the application the students/user must enter his/her personal and educational details in the profile section.

Our System is being made for reducing the information loss and smoothening the communication between Teacher and students so that they both have all the required information in their hand.

### 1.3 Existing System

Currently no such kind of application exists for the Teacher students intervention. Each time a google form is circulated to get the information about the candidates and a sheet has to be maintained for the same.

Some of the disadvantages of existing system are as follows :

- Time consuming, Manual work  
Every time teachers have to maintain several excel sheets regarding the various available data of the student.
- Redundancy  
Many times the details of the same student is taken multiple times causing redundancy. Same google forms are to made again and again. All this is very redundant and tiring.
- Loss of information  
Many times students or teachers might miss some information regarding a particular session due to scattering of the information in various places.

### 1.4 Proposed System

The User is the Teacher marking the attendance or Student who will register (only using SPIT email address) and access the system features. This System will be connected to the HOD office from where the academic updates on upcoming sessions will be updated on regular intervals which can be accessed by the user to perform further teaching and learning activities.

The HOD and Teacher will be communicating company updates through this application with the student(end user) which will initially help the student for the attendance procedure. This will avoid the miscommunication between the Teachers and Students as Teachers will be keeping track of the record of the students directly.

You can also reset your password if forgotten. Upon entering your UID, your Year and Branch will automatically get stored.

Some of the advantages of our system are as follows :

- User Friendly  
It provides attractive interface to the user for navigation through a dynamic flow of placement process in the app.  
The items of the application are already connected to each other using side navigation bar.
- Companies in a place  
Users can view list of companies as per their categories or all together as per their requirement in a single list view.

Project Name

Student Name (UCID)

## 1.5 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Device	Android Device with Touch Screen minimum 5" inch Display
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	OS Independent
Database	Firestore

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	Windows desktop/PCs
Server	Not Required

## 2 Software Requirement Specification (SRS) and Design

### 2.1 Purpose

The purpose of our project is to develop an UI application that can help user (students) to easily access their Attendance sheets and to keep all the records handy.

This can save lots of efforts of Teachers and the students and it will be easier to store the each and every single and small detail regarding each session. This app will keep track of all the information regarding the present people in the sessions including student information and the academic updates.

### 2.2 Definition

To build a Attendance Application so the students can have an easy to go placement process.

### 2.3 Overall Description

#### 2.3.1 Product Functions

The product function includes:

1. Authentication: Users are required to Sign-up and Log-in (only using SPIT id). Users will get a verification email for successful registration.
2. Profile: This will contain information regarding candidate.
3. Marks: This will contain information regarding candidate's marks.
4. View All student details: It is only viewed by Teachers, to check information of each and every candidate.
5. Add Notifications: It is viewed by all, to add updates about the academics.
6. View Attendance CSV: To view the attendance records as per the categories or view all of them together.

Project Name

Student Name (UCID)

---

### **2.3.2 User Characteristics**

There are two types of users:

- Teachers: Once HOD's email id is encountered via login, Teacher is set on in database and he/she gets all the special privileges and can make others' Teachers as well.
- Non -Teachers: He/She can only update his own profile and marks and can view updates regarding Sessions and mark the attendance.

Project Name

Student Name (UCID)

---

Use Cases:

1. Register
2. Login
3. Update Profile
4. Update Marks
5. View Marks
6. Add Company
7. View Company

Table 4.2.1: Use Case Table - Register

Use Case ID	1
Use Case Name	Register
Actor	Teacher and Student
Pre-Condition	They must register themselves first
Post-Condition	User can login
Flow of events	Login,Register or Edit details and attendance or export and view attendance

Table 4.2.2: Use Case Table - Login

Use Case ID	2
Use Case Name	Login
Actor	Teacher and Students
Pre-Condition	They must register themselves first
Post-Condition	User can view its details and add face and view and upload attendance with the option import
Flow of events	Login,Register or Edit details and upload and download attendance data sheets

Table 4.2.3: Use Case Table - Update Profile

Use Case ID	3
Use Case Name	Update Profile
Actor	Teacher and Students
Pre-Condition	Login
Post-Condition	User can view or edit the profile and add the face data

Project Name

Student Name (UCID)

---

Table 4.2.4: Use Case Table - Update attendance

Use Case ID	4
Use Case Name	Update Attendance
Actor	Teacher only
Pre-Condition	Login
Post-Condition	User can view and edit the attendance manually in exceptional cases

Table 4.2.5: Use Case Table - Train data

Use Case ID	5
Use Case Name	Train data
Actor	Students
Pre-Condition	Login
Post-Condition	Can Train data

Table 4.2.6: Use Case Table - Helpdesk

Use Case ID	6
Use Case Name	Helpdesk
Actor	TPC
Pre-Condition	Login
Post-Condition	Can contact on helpdesk if any needed assistance

Table 4.2.7: Use Case Table - Recognise face

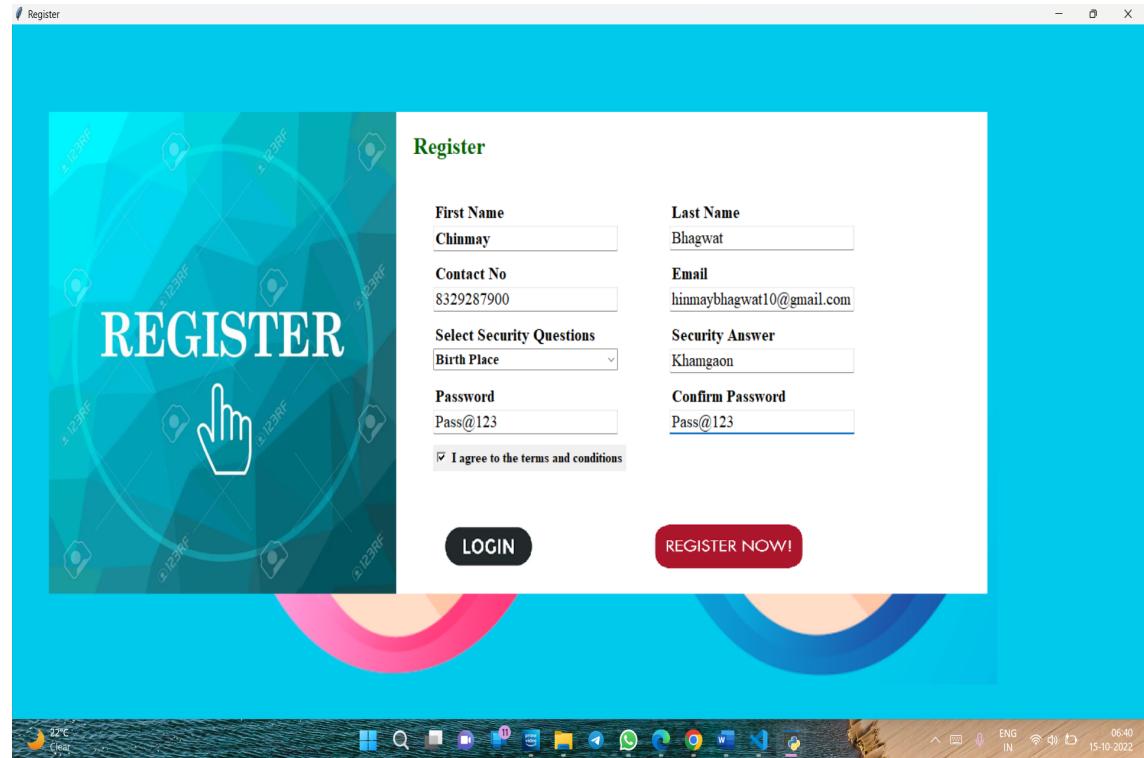
Use Case ID	7
Use Case Name	Recognise face
Actor	TPC
Pre-Condition	add face
Post-Condition	Recognise face and fetch the recorded details with the matching face.

Project Name

Student Name (UCID)

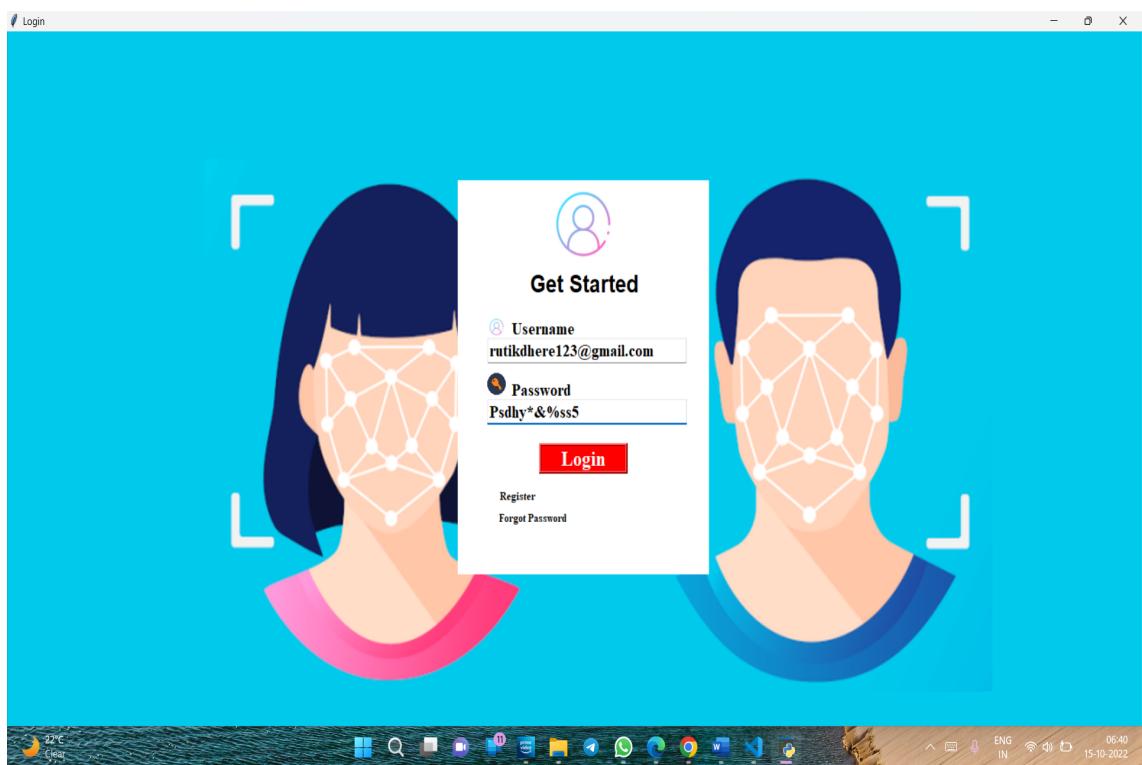
### 3 Project Implementation and Testing

#### 3.1 Register



3.1.1: Register user

### 3.2 Login



3.2.1: Login View

### 3.3 Homepage



3.3.1: Homepage menu

### 3.4 Students Section

**STUDENTS INFO**

[Course details]

Department : EXTC Course : M.Tech

Academic Year: 2022-23 Semester : Sem-II

[Personal details]

Registration No.: 001 Full Name: Shrikrushna Bhagwat

College UCID: 2021510005 Division: B

Gender: Male DOB: 13051999

email: ybhagwat@spit.ac.in mobile: 8329287900

Mentor: Pooja Raundale Address: 1, GB rd, Thane, 400601

Take Photo Sample  NO Photo Sample

**STUDENT DETAILS**

[Search Bar]

Search by: Choose search  SEARCH SHOW ALL

Department	Course	Year	Semester	Id	Name	UCID
IT	FE	2020-2021	1	1	Chinmay	B

**UPDATES**

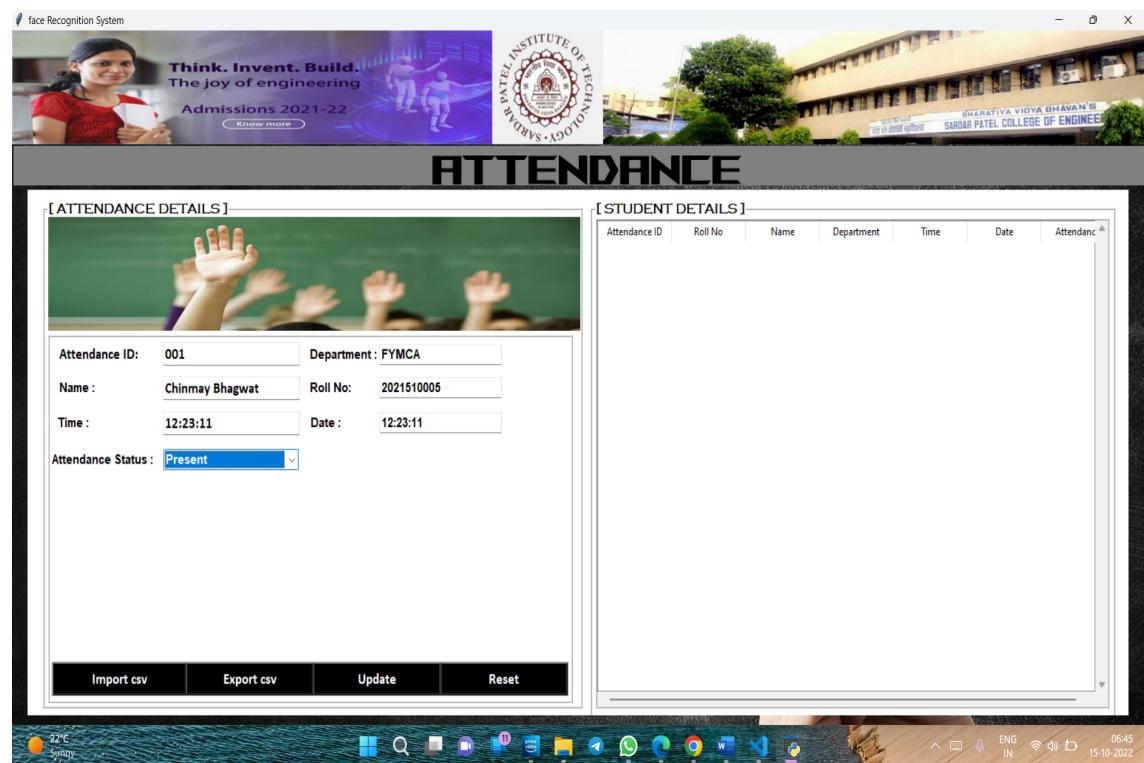
22°C  
Polluted air... 06:42  
15-10-2022

3.4.1: Adding and viewing students records

# Project Name

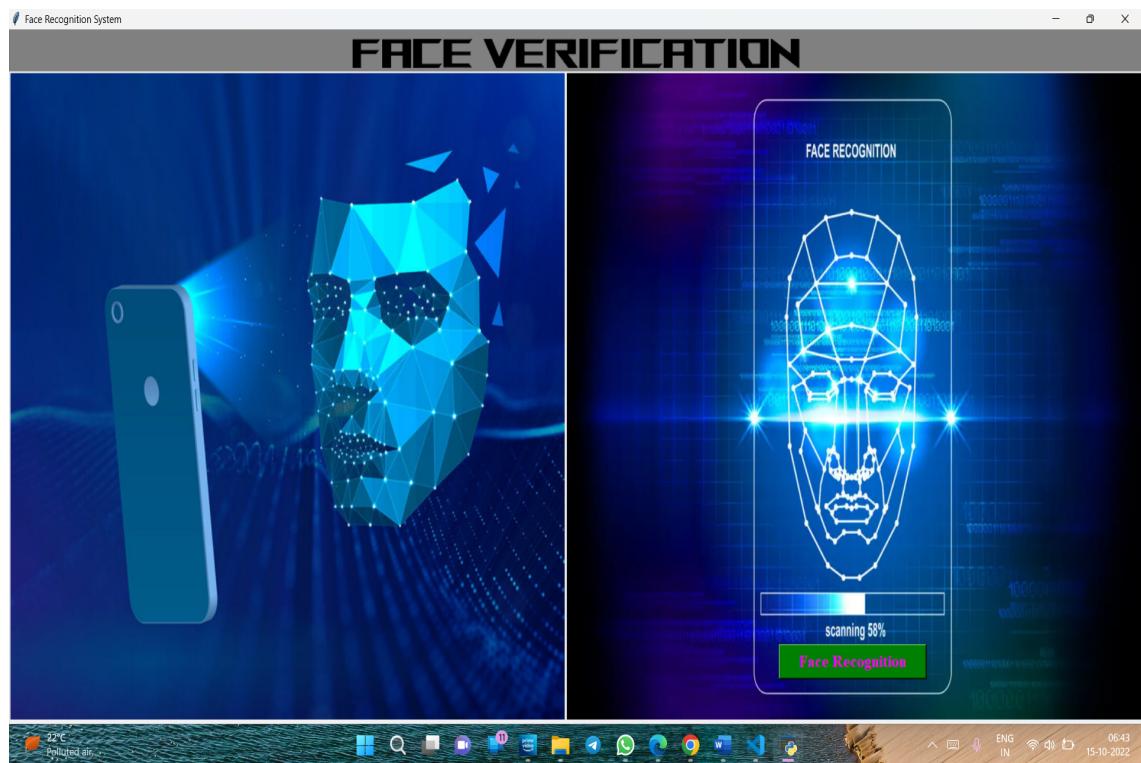
Student Name (UCID)

### 3.5 Attendance section



3.5.1: View attendances and download as CSV from here

### 3.6 Face recognition window

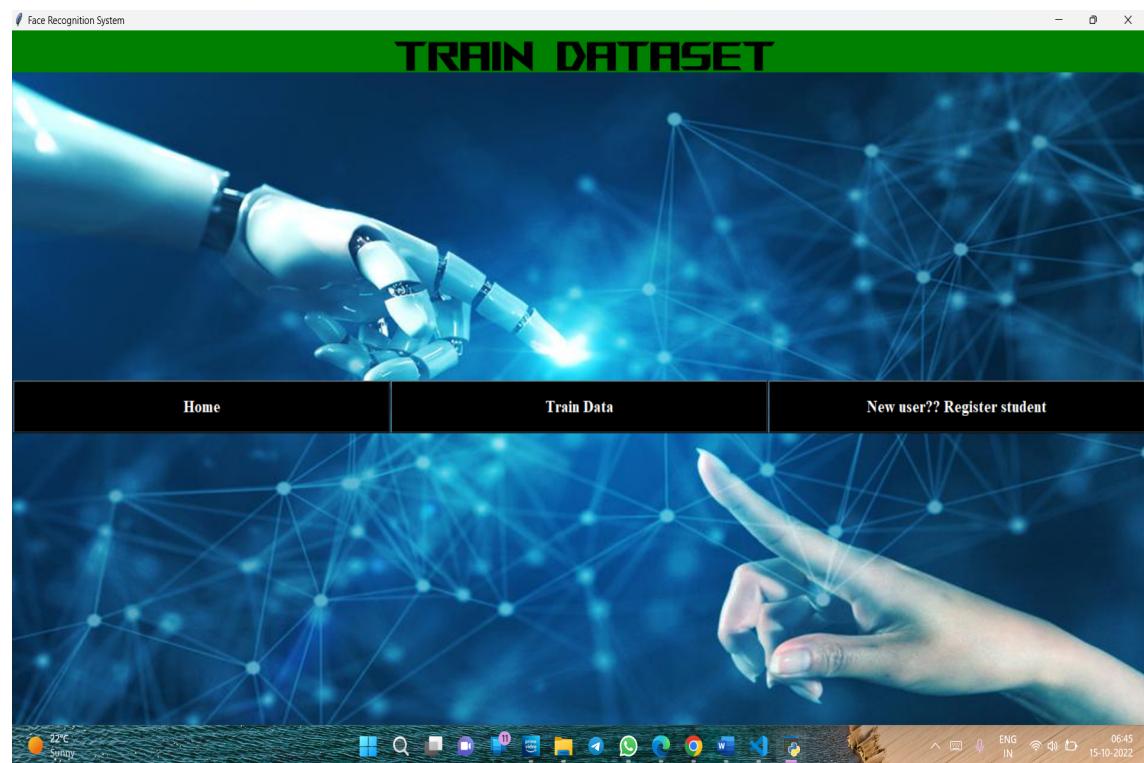


3.6.1: Recognise registered face.

Project Name

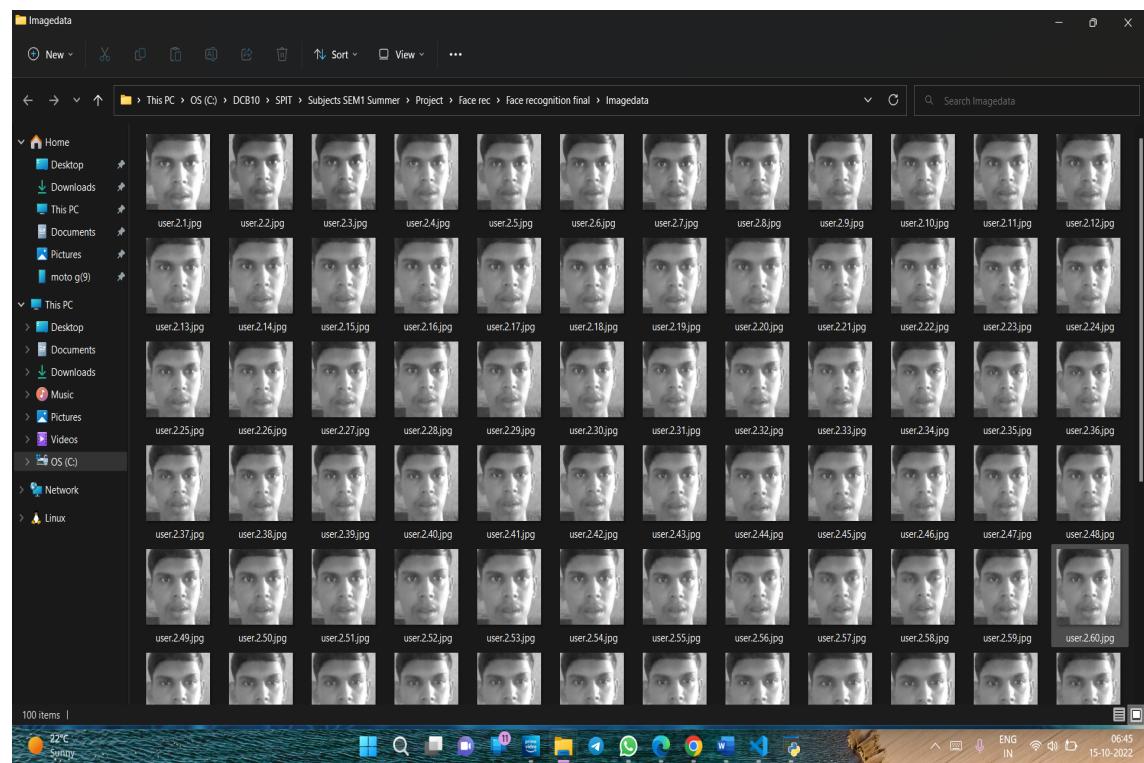
Student Name (UCID)

### 3.7 Train data



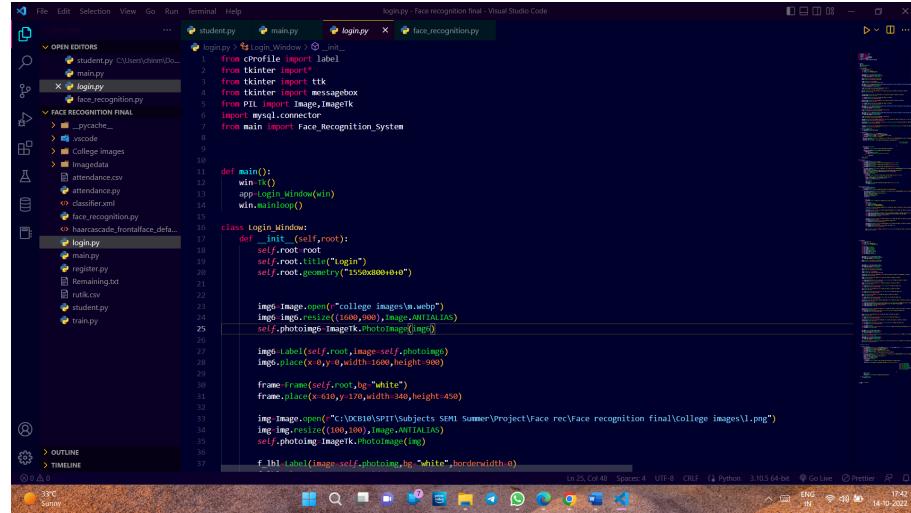
3.7.1: Training the added dataset of face images

### 3.8 View Added data



3.8.1: Photos data

### 3.9 Code 1



```

File Edit Selection View Go Run Terminal Help
... student.py login.py ...
login.py - Face recognition final - Visual Studio Code
...
from tkinter import *
from PIL import Image,ImageTk
import mysql.connector
from main import Face_Recognition_System
...
def main():
    win = Tk()
    win.title("Login")
    win.geometry("1500x800+0+0")

    class Login_Window:
        def __init__(self,root):
            self.root = root
            self.root.title("Login")
            self.root.geometry("1500x800+0+0")

            img = Image.open("college_images\univ.jpg")
            img = img.resize((1000,300),Image.ANTIALIAS)
            self.photologo = ImageTk.PhotoImage(img)

            img_label(self.root,image=self.photologo)
            img_label.place(x=0,y=0,width=1000,height=300)

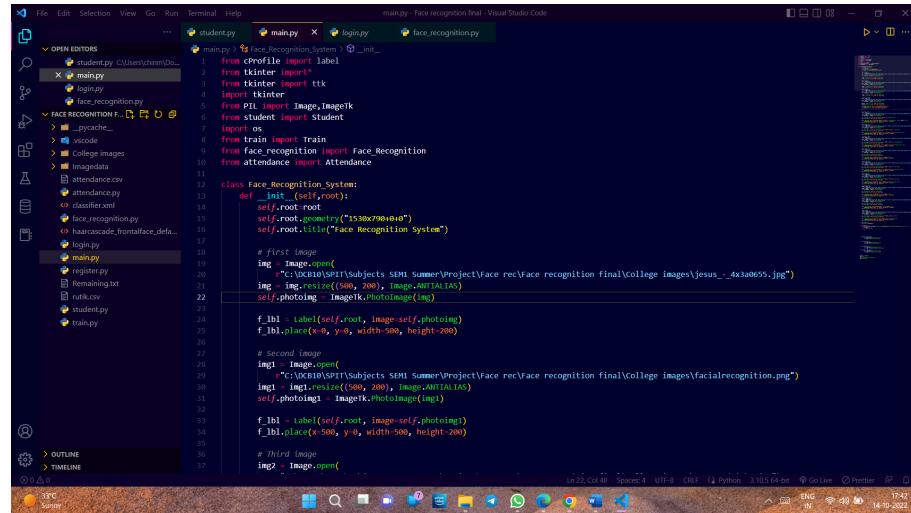
            frame = Frame(self.root,bg="white")
            frame.place(x=610,y=170,width=340,height=450)

            img = Image.open("C:\DC010\SP1\Subjects SEMI Summer\Project\Face rec\Face recognition final\college images\l.jpg")
            img = img.resize((100,100),Image.ANTIALIAS)
            self.photologo = ImageTk.PhotoImage(img)

            f_lbl = Label(self.root,image=self.photologo,bg="white",borderwidth=0)

```

### 3.10 Code 2



```

File Edit Selection View Go Run Terminal Help
... student.py login.py ...
main.py - Face recognition final - Visual Studio Code
...
from tkinter import *
from PIL import Image,ImageTk
from student import Student
import os
from Face_Recognition import Face_Recognition
from attendance import Attendance
...
class Face_Recognition_System:
    def __init__(self,root):
        self.root = root
        self.root.geometry("1500x900+0+0")
        self.root.title("Face Recognition system")

        # First Image
        img = Image.open("C:\DC010\SP1\Subjects SEMI Summer\Project\Face rec\Face recognition final\college images\jesus_-4x3a0655.jpg")
        img = img.resize((500, 200), Image.ANTIALIAS)
        self.photologo = ImageTk.PhotoImage(img)

        f_lbl = Label(self.root,image=self.photologo)
        f_lbl.place(x=0,y=0, width=500, height=200)

        # Second Image
        img = Image.open("C:\DC010\SP1\Subjects SEMI Summer\Project\Face rec\Face recognition final\college images\Facialrecognition.png")
        img = img.resize((500, 200), Image.ANTIALIAS)
        self.photologo = ImageTk.PhotoImage(img)

        f_lbl = Label(self.root,image=self.photologo)
        f_lbl.place(x=500,y=0, width=500, height=200)

        # Third Image
        img2 = Image.open(

```

# Project Name

Student Name (UCID)

### 3.11 Code 3

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the current workspace, including `student.py`, `main.py`, `register.py`, and `face_recognition.py`.
- Editor:** The `register.py` file is open, displaying Python code for a registration interface. The code includes imports for `atexit`, `register`, `email.mime.image`, `tkinter`, `tk`, `messagebox`, `mysql.connector`, and `PIL`. It defines a `Register` class with methods for initializing the root window and displaying a registration form with labels for name, contact, security type, and password.
- Terminal:** The terminal tab shows the command `register.py - Face Recognition Final - Visual Studio Code`.
- Status Bar:** Shows the current file as `register.py`, line 35, column 1. It also displays system information like CPU: Intel(R) Core(TM) i5-10210U, RAM: 3105.64 MB, and battery status.

## 4 Test Cases

Table 6.1: Test Case - Login and Register

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enter user id and password	Enters the correct user id and password	Log in Successful	Home Page	Pass
2	User enter user id and password	Enters the user id and password	Prompt error	Prompt error	Pass
3	User enter user id and password	Valid user id and password which doesn't exist in Database	Registered Successfully	Login Page	Pass
4	User enter user id and password	Invalid user id and password which contains in Database	Prompt error	Prompt error	Pass

Table 6.2: Test Case - Others

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enters invalid information in marks section	Invalid marks Details	Prompt Message showing error	Successful Updated	Fail
2	User enters valid information in Profile	Valid Profile Details	Profile Updated successful	Profile Updated successful	Pass

Project Name

Student Name (UCID)

---

## 5 Limitations

- It needs internet to be accessed.
- It does not have a feature to connect with TPO directly for queries and support.
- PDFs regarding company can not be uploaded
- It does not have a feature to apply for placements offers directly from application.
- Graphical representation of placement statistics are not shown

## 6 Future Enhancements

- Automatic report generation for TPO.
- Feature of applying for placements directly from app.
- Details in which company the user is placed will be able to be stored
- Graphical representation of placement statistics need to be shown

Project Name

Student Name (UCID)

---

## 7 User Manual

### **Part 1 – Register**

Upon opening the application, user will be greeted with the registration screen. If the user has no account, user can click on register and register self.

After verification of user id, User's account will be saved in our database. The user can now proceed to login.

### **Part 2 – Login**

User needs to enter email first and then password. If there is an active internet connection, user can proceed to login.

### **Part 3 – Profile**

User can access/add/update personal details (eg. Name, Contact No.). User can update this details anytime.

### **Part 4 – Marks Section**

User can access and add educational details (eg. Qualifications, Grades, etc). User cannot update this details once entered.

### **Part 5 – Company Updates**

User can view company updates received from the placement office on basis user can apply for the placement process. system generates a fairly accurate prognosis.

## 8 Bibliography

### 8.1 Web References

- [1.] <https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08>
- [2.] <https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection/>
- [3.] <https://pyimagesearch.com/2021/04/12/opencv-haar-cascades/>
- [4.] [https://www.youtube.com/watch?v=ANeuqpI\\_VcY](https://www.youtube.com/watch?v=ANeuqpI_VcY)
- [5.] <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>
- [6.] <https://labelyourdata.com/articles/facial-recognition-algorithms-for-machine-learning>
- [7.] <https://www.analyticssteps.com/blogs/how-does-facial-recognition-work-deep-learning>