MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

Project Title

Real Time Face Recognition for Library Check in Check out System using opencv

Under the guidance of

Dr . M.SARAVANAMUTHU

PRESENTED BY

T. SULOCHANA

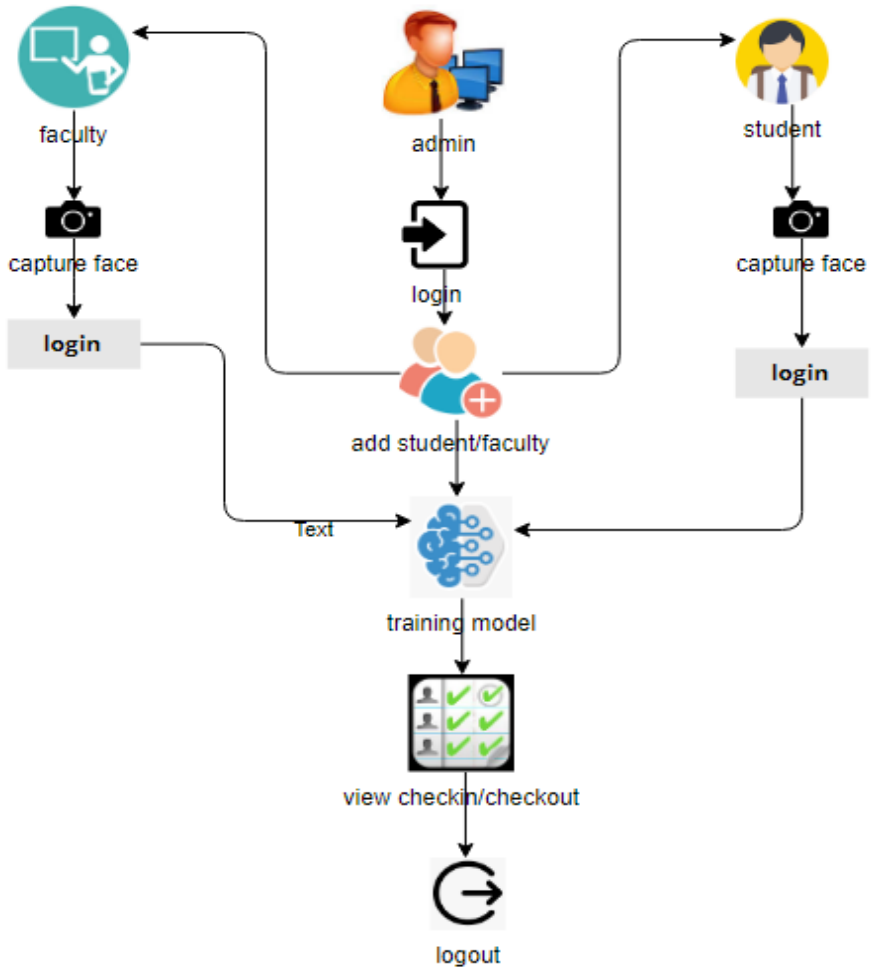
(22691F00G7)

CONTENTS

Abstract

- AWS Cloud-based data storage solutions have seen a spike in attention from academics and business in recent years because to their effective and low-cost administration.
- Service providers must utilise secure data storage and sharing protocols to maintain data confidentiality and service user privacy since they deliver services via an open network.
- In previous days all the library login and logout information is stored in records (books) manually.

ARCHITECTURAL DESIGN FOR PROPOSED SYSTEM



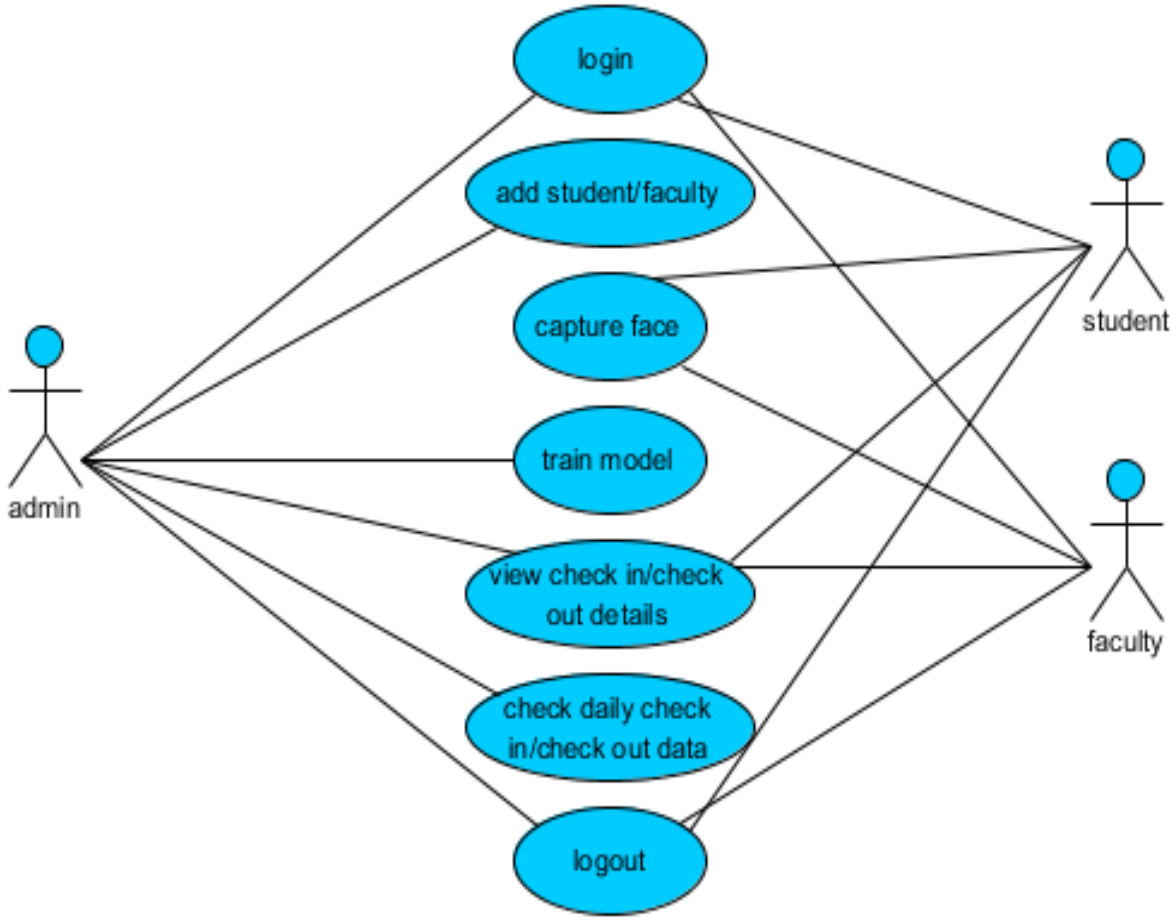
UML DIAGRAMS

- Uml stands for unified modeling language. Uml is a standardized general-purpose modeling language in the field of object-oriented software engineering.
- The standard is managed, and was created by, the object management group.
- The goal is for uml to become a common language for creating models of object oriented computer software.

USECASE DIAGRAM

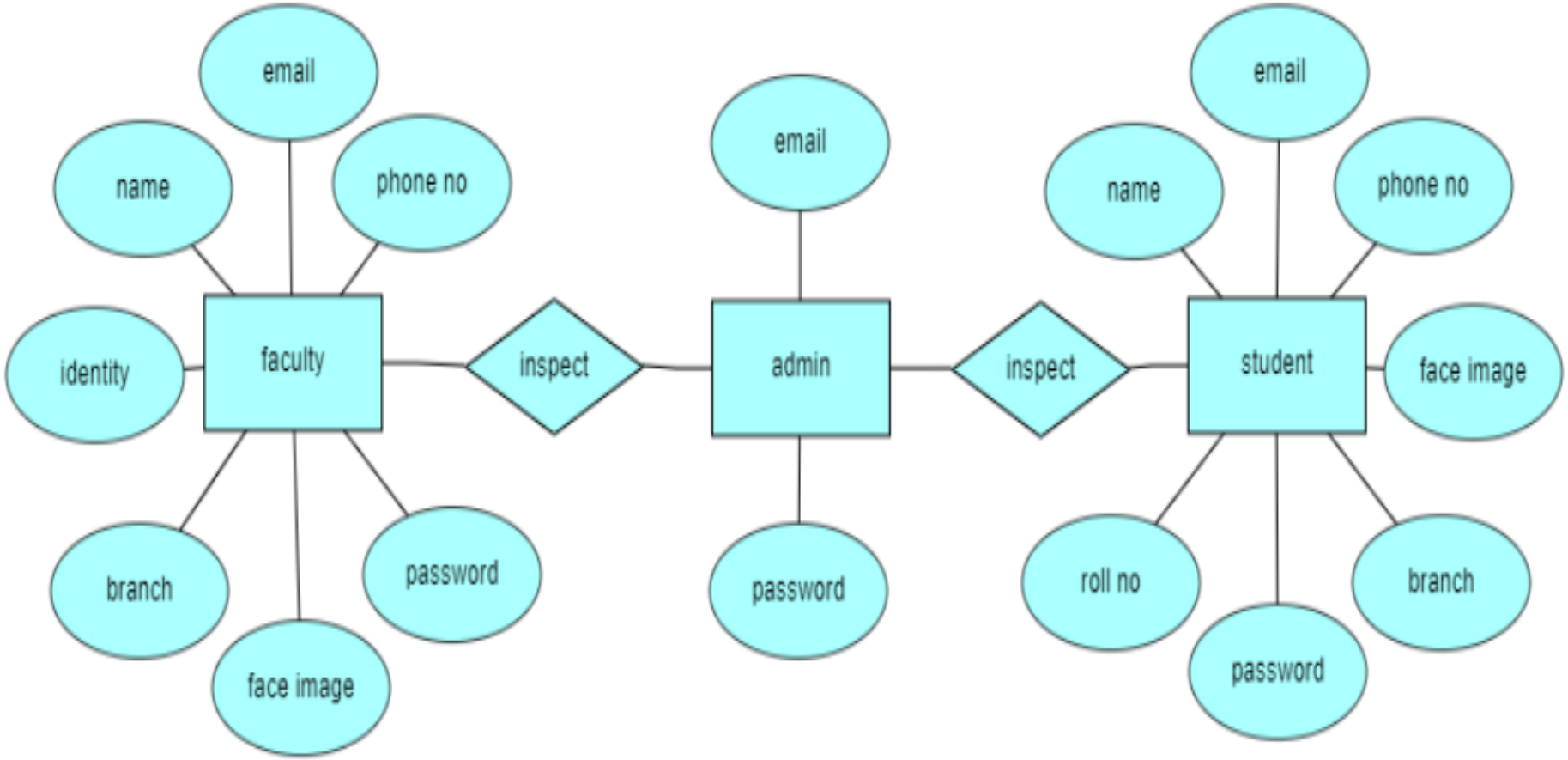
- A use case diagram in the unified modeling language (uml) is a type of behavioral diagram defined by and created from a use-case analysis.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.
- The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

USECASE DIAGRAM



ENTITY RELATIONSHIP DIAGRAM

- An Entity relationship model describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram .
- An ER model is a design or blueprint of a database that can later be implemented as a database.

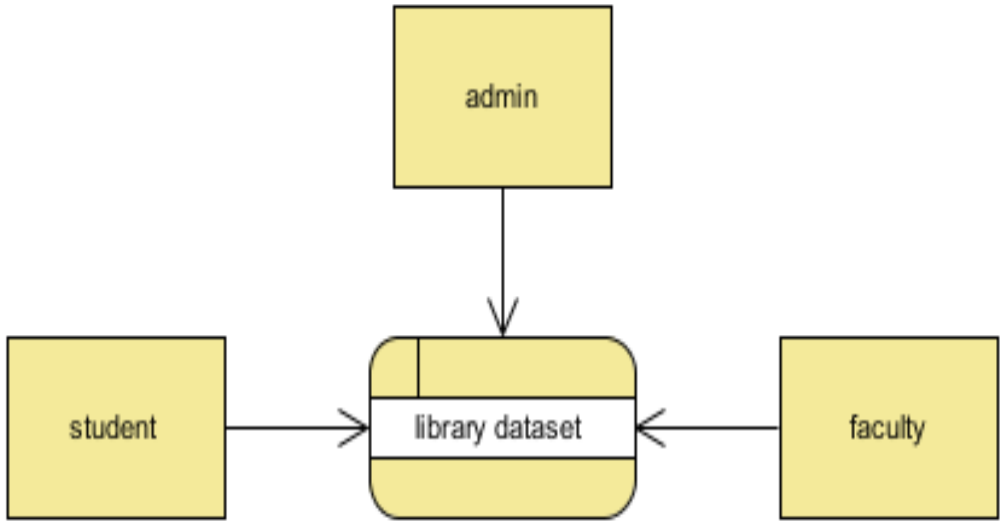

email
name
phone no
identity
faculty
branch
face image
password
inspect
email
admin
password
inspect
email
name
phone no
student
face image
roll no
password
branch

DATA FLOW DIAGRAM

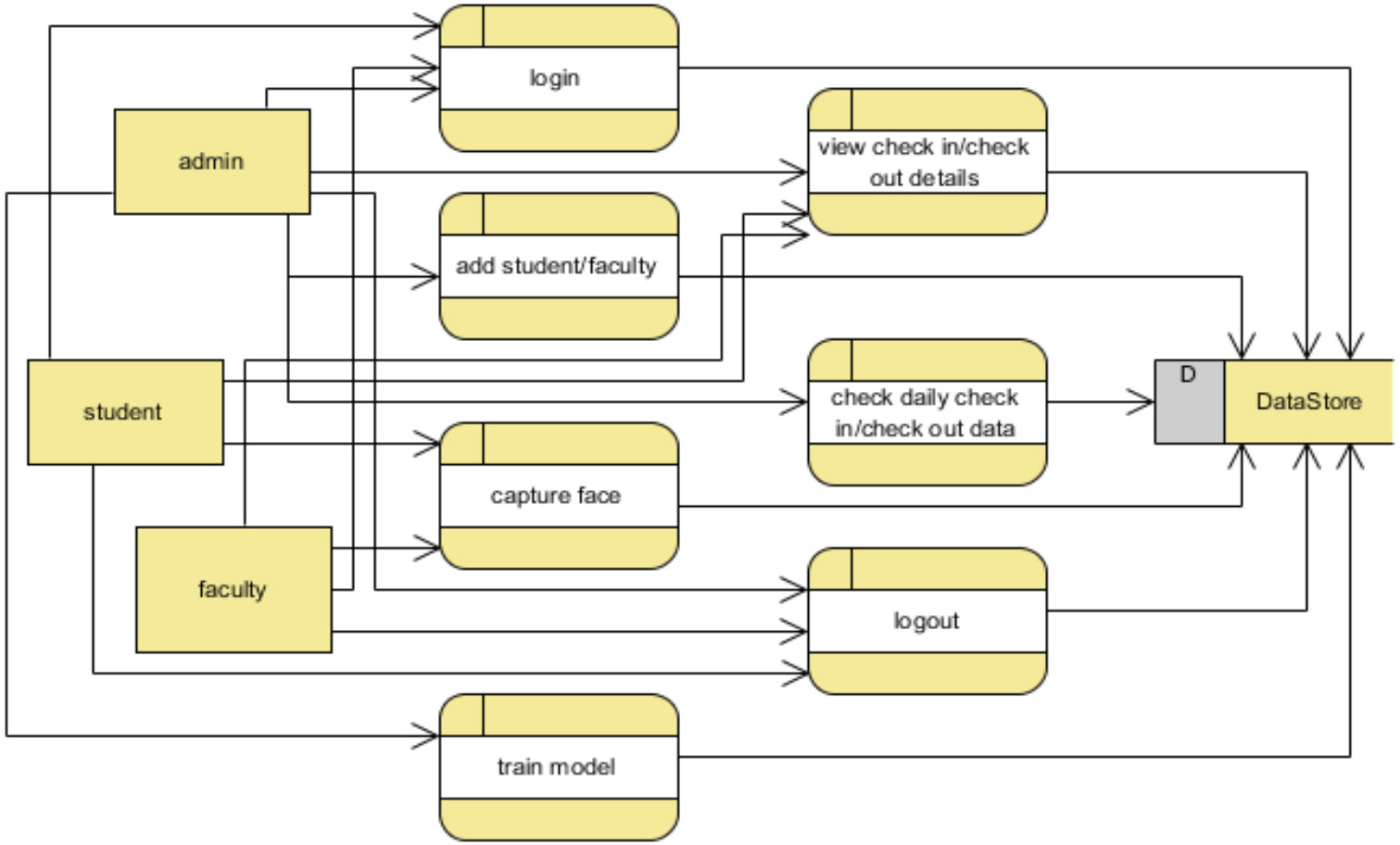
- A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system.
- A neat and clear DFD can depict a good amount of the system requirements graphically.
- It can be manual, automated, or a combination of both.

- It shows how information enters and leaves the system, what changes the information and where information is stored.
- The purpose of a DFD is to show the scope and boundaries of a system as a whole.
- It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

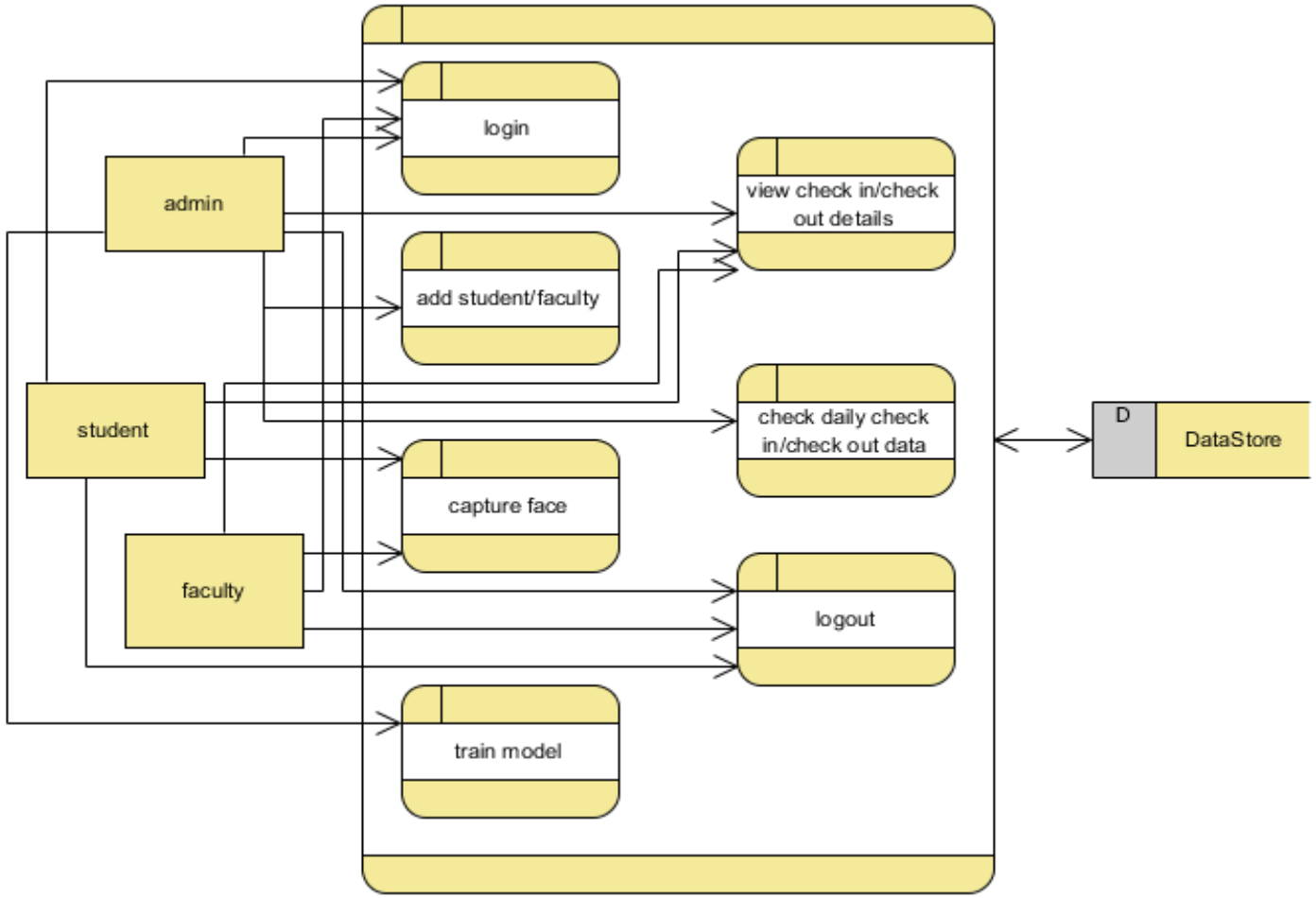
Context Level Diagram:



Level 1 Diagram:



Level 2 Diagram:



TECHNIQUES USED

- Face Detection: Use OpenCV's pre-trained face detection models to locate faces in the video stream.
- Face Recognition: Train a face recognition model using techniques like Eigenfaces or LBPH (Local Binary Patterns Histograms). OpenCV provides implementations for these algorithms.

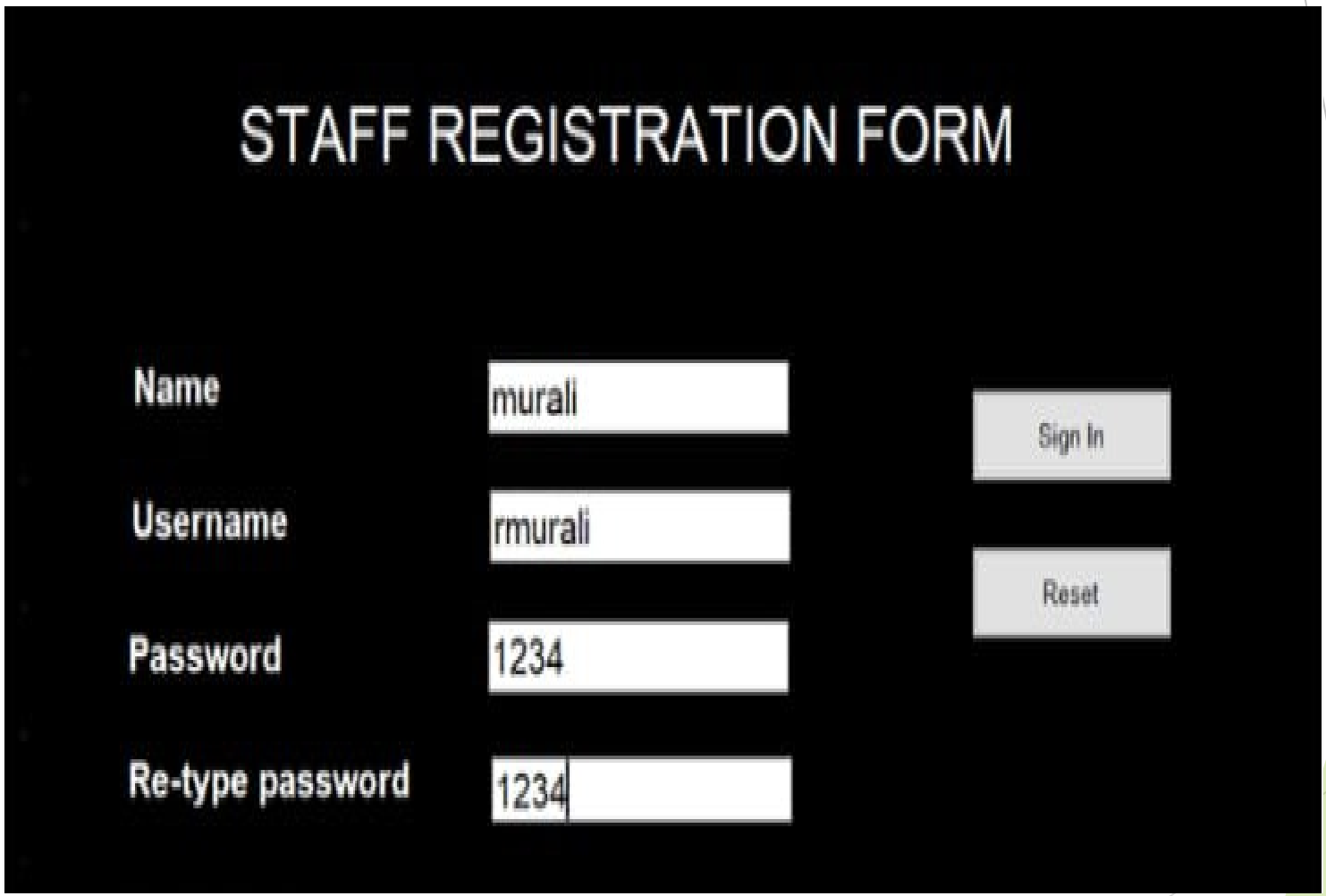
SAMPLE CODE

```
import csv
from sklearn.preprocessing import LabelEncoder
from flask import Flask,render_template,request,redirect,url_for,flash,session
import numpy as np
import mysql.connector
import cv2,os
from email.message import EmailMessage
import pandas as pd
from PIL import Image
import time
from playsound import playsound
import pickle
import smtplib
import imghdr
import datetime
import time
import requests
from PIL import Image, ImageDraw, ImageFont
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
mydb = mysql.connector.connect(host="localhost", user="root", passwd="", database="face_library")
cursor = mydb.cursor()
```

```
        sql = "select * from faculty where email='%s' and pwd='%s' " % (username, password1)
        x = cursor.execute(sql)
        results = cursor.fetchall()
        print(type(results))
        if not results:
            flash("Invalid Email / Password", "danger")
            return render_template('contact.html')

        else:
            # session['email'] = username
            if len(results) > 0:
                flash("Welcome ", "primary")
                session['identity']=results[0][2]
                session['name']=results[0][1]
                flash("Welcome ", "success")
                return render_template('facultyhome.html', msg=results[0][1])
return render_template('contact.html')
```

EXPECTED OUTCOMES(CONT...)


STAFF REGISTRATION FORM
Name
murali
Username
rmurali
Password
1234
Re-type password
1234
Sign In
Reset

REFERENCES

1. H. S. G.Asep and Y. Bandung , “A design of continuous user verification for online exam proctoring on M-learning”, International Conference on Environmental and Energy Engineering (ICEEI), 2019

2. L.K.Musambo and J. Phiri ,” Student facial authentication model based on OpenCV’s object detection method and QR code for Zambian higher institutions of learning” , International Journal of Advanced Computer Science and Application (IJACSA), 2018


Thank
you!