



# **Department of Information Technology**

## **NBA Accredited**

A.P. Shah Institute of Technology

— G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year 2021-2022

A Project Presentation on

# **Developing AI-Based Comprehensive Framework for Online Assessments**

Submitted in partial fulfillment of the degree of

Bachelor of Engineering(Sem-8)

in

**INFORMATION TECHNOLOGY**

By

Swapnil Sapre(18104027)

Kunal Shinde(18104012)

Keval Shetta(18104013)

Under the Guidance of

Prof Vishal Badgujar

# 1. Project Conception and Initiation

---

# 1.1 Abstract

- Online examinations are the way of conducting examinations on the user's mobile devices or laptops rather than actual paper. During situations when physical exams cannot be conducted online exams have been the preferred choice.
- In physical examinations the students doing malpractices reduces by a great amount as there are examiners physically present to monitor every student. The idea of our examination framework is to reduce the malpractices done by the students in the online mode as far as the current platforms in use are concerned.
- We have arrived at conclusion that Smart Online Examination platform is a much viable solution to the existing platforms for conduction of the exams and doing the proctoring.

## 1.2 Objectives

- To set up proctoring mechanism to stop malpractices in exam. To create a system that validates each user well before the exam using facial recognition and validation.
- To perform object tracking (person tracking) throughout the exam.
- To allow the examiner to alert warnings or end the exam if students found performing malpractices.
- To provide a platform for the teachers to set the questions and get detailed reporting.
- To provide tab locking features to not allow students to navigate away from exam screen.
- To send attendance and mis activity report details to teacher's.

## 1.3 Literature Review

Sr. No	Authors	Publication	Findings
1	Walaa M. Abd-Elhafiez, Mohamed Heshmat, and Seham Elaw	<p>Title : Efficient Method for Face Recognition and Its Role in Supporting E-Learning Systems</p> <p>Year : IEEE , 2015</p> <p>Conference : Fifth International Conference on e-Learning (econf), 2015</p>	<ul style="list-style-type: none"><li>• Facial feature detection algorithm which is needed to detect the face of the candidate from several image sets.</li><li>• Detection required calculation of the three components in the face image like eyes, nose, and mouth and the extraction of these features</li></ul>
2	Arief Agus Sukmandhani and Indrajani Sutedja	<p>Title : Face Recognition Method for Online Exams</p> <p>Year : IEEE , 2019</p> <p>Conference : 2019 International Conference on Information Management and Technology (ICIMTech), 2019</p>	<ul style="list-style-type: none"><li>• CNN (Convolutional Neural Networks) a deep learning algorithm helps in face verification and detection.</li><li>• Requires training samples by taking number of sample data of student faces</li></ul>

Sr. No	Authors	Publication	Findings
3	Samuel S. Chua et	<p>Title : Online Examination System with Cheating Prevention Using Question Bank Randomization and Tab Locking</p> <p>Year : IEEE , 2019</p> <p>Conference : 2019 4th International Conference on Information Technology (InCIT)</p>	<ul style="list-style-type: none"> <li>• Tab locking mechanism that prevents candidates from opening other as their exam window is locked while the exam is in progress.</li> <li>• Having randomized system for question papers so students do not get the same questions</li> </ul>
4	Arief Agus Sukmandhani and Indrajani Sutedja	<p>Title : Face Recognition Method for Online Exams</p> <p>Year : IEEE , 2019</p> <p>Conference : 2019 International Conference on Information Management and Technology (ICIMTech),</p>	<ul style="list-style-type: none"> <li>• Deep face is a deep learning python-based framework that can detect faces from images and has already been trained with sample images.</li> </ul>
5	Mansi Mahendru and Sanjay Kumar Dubey	<p>Title : Real Time Object Detection with Audio Feedback using Yolo vs. YOLOv3</p> <p>Year : IEEE , 2021</p> <p>Conference : 2021 11th International Conference on Cloud Computing, Data Science &amp; Engineering (Confluence)</p>	<ul style="list-style-type: none"> <li>• YOLO is a machine learning algorithm that is used to perform object detection</li> <li>• It can be used to detect objects in still images, provided sample video, or from a real-time webcam</li> </ul>

# 1.4 Problem Definition

Problems Identified :

- Inability to conduct in person exams in situations like pandemic.
- Malpractices done by students by taking advantage of no proctoring methods during exam.
- Personal monitoring of students required that takes more time and not easily manageable
- Problem of Impersonation
- Student management in terms of marks record and question bank management in terms of setting question paper.



# 1.5 Scope

- Can be applied in educational institutes like schools and colleges for conduction of tests.
- Can be used anytime and anywhere.
- Can be used by teachers to conduct any kind of tests and having different kinds of question patterns.

# 1.6 Technology stack

- HTML , CSS , Javascript – for building web application.
- Flask – Python web based lightweight framework
- Python – For server side processing.
- YOLO – Machine Learning algorithm for object detection
- DeepFace – For face validation
- MySQL – Xampp : localhost server to run the application.

## 1.7 Benefits for environment & Society

- Proctored testing removes administrative burden on test centers
- Proctoring creates a fair online test environment for students.
- Proctored testing features prevent and deter malpracticing.
- Proctoring online examination avoids the use of pen and paper and saves time.

## 2. Project Design

---

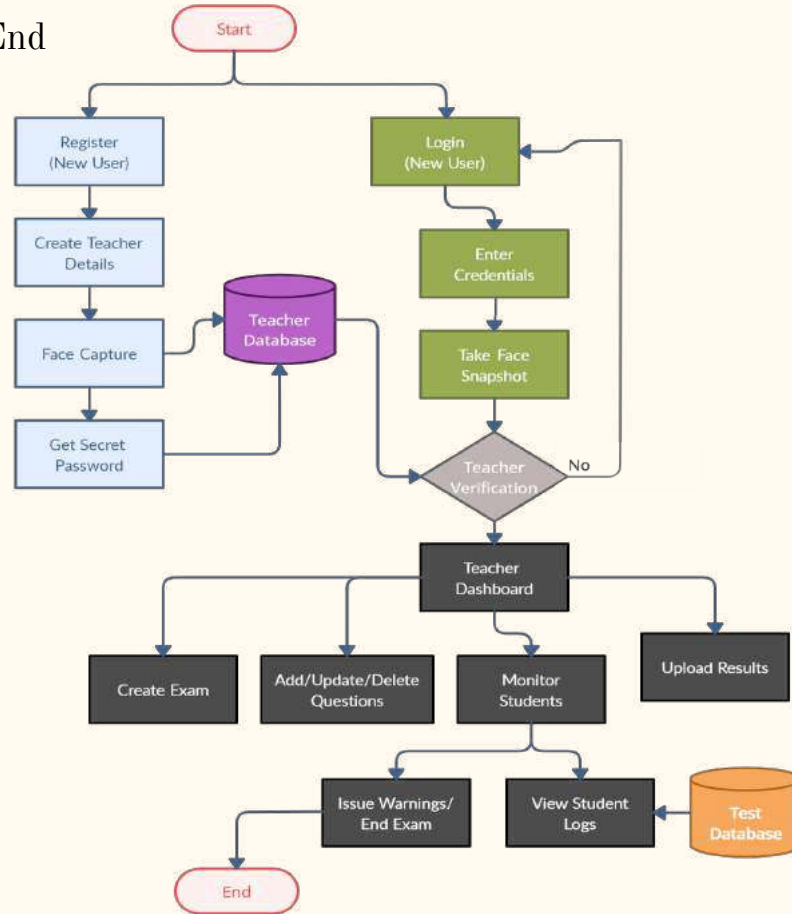
## 2.1 Proposed System

- Student Block/Dashboard
  - To attempt the test.
  - To check the result of the attempted test.
- Examiner Block/Dashboard
  - To create test
  - To add/update/delete questions
- Question Paper Generation Process
  - By importing a document file having test questions in it.

- Object Tracking and Face recognition
  - During login into system , face recognition and validation is performed.
  - During examination, tracking of the student's live environment.
- Tab Locking
  - Display of alert message when student navigates away from the exam window.
  - Exam terminates after 5 attempts.

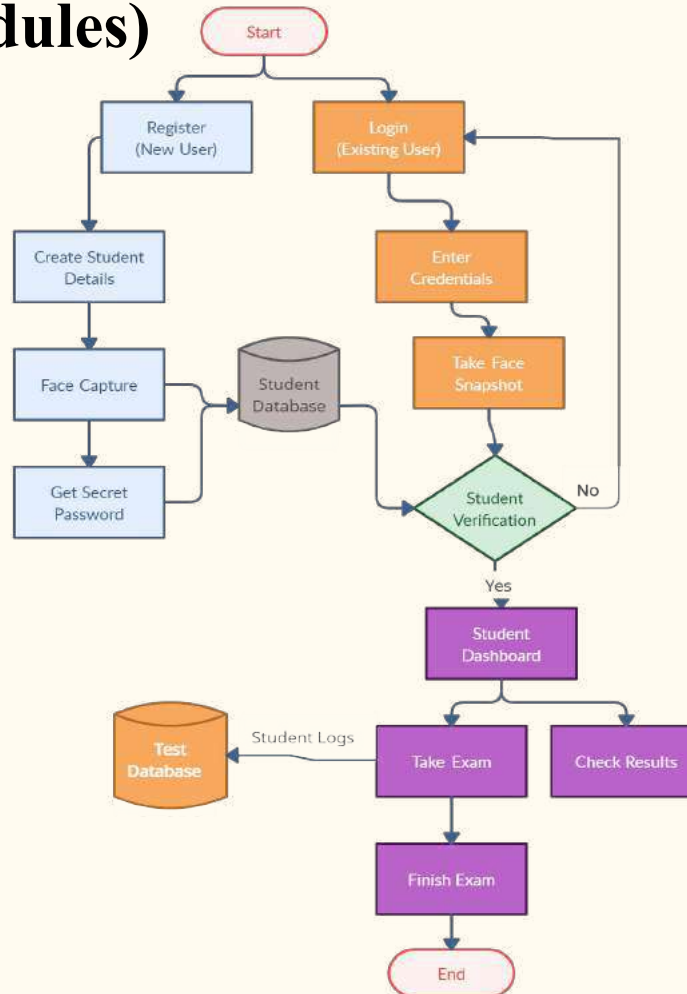
## 2.2 Design(Flow Of Modules)

### 1. Examiner's (Teacher's) End



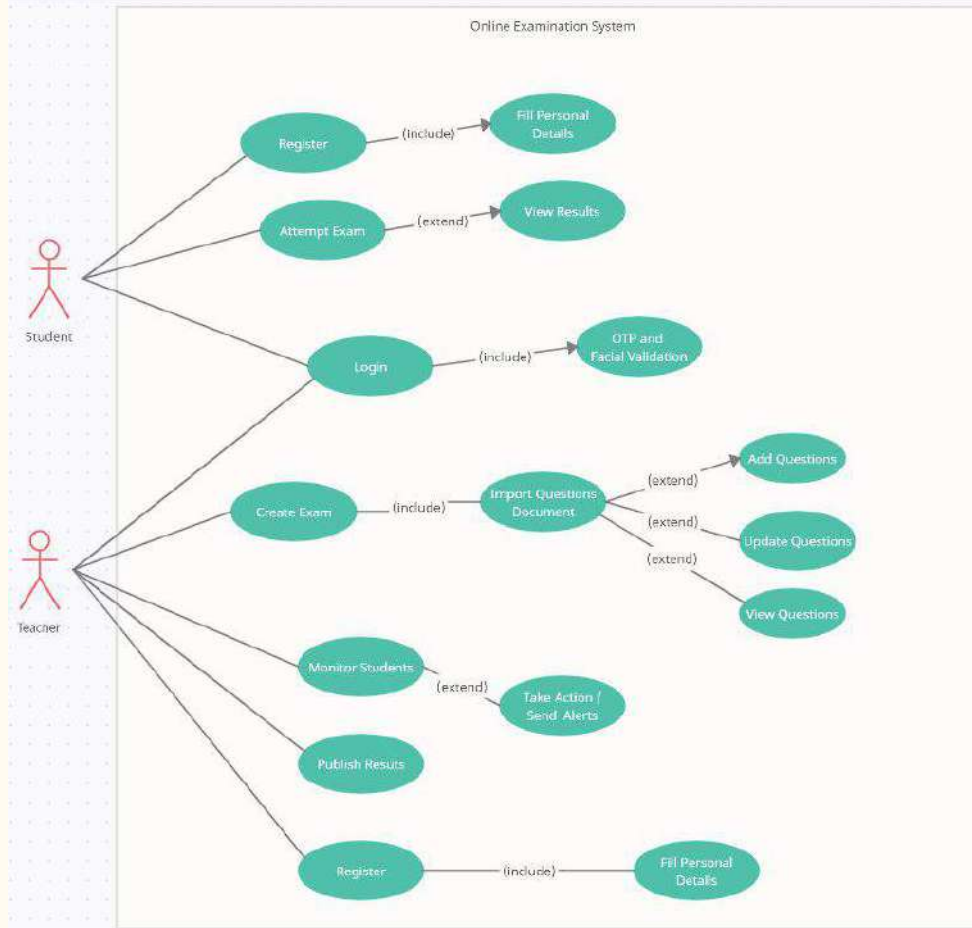
## 2.2 Design(Flow Of Modules)

### 2. Student's End



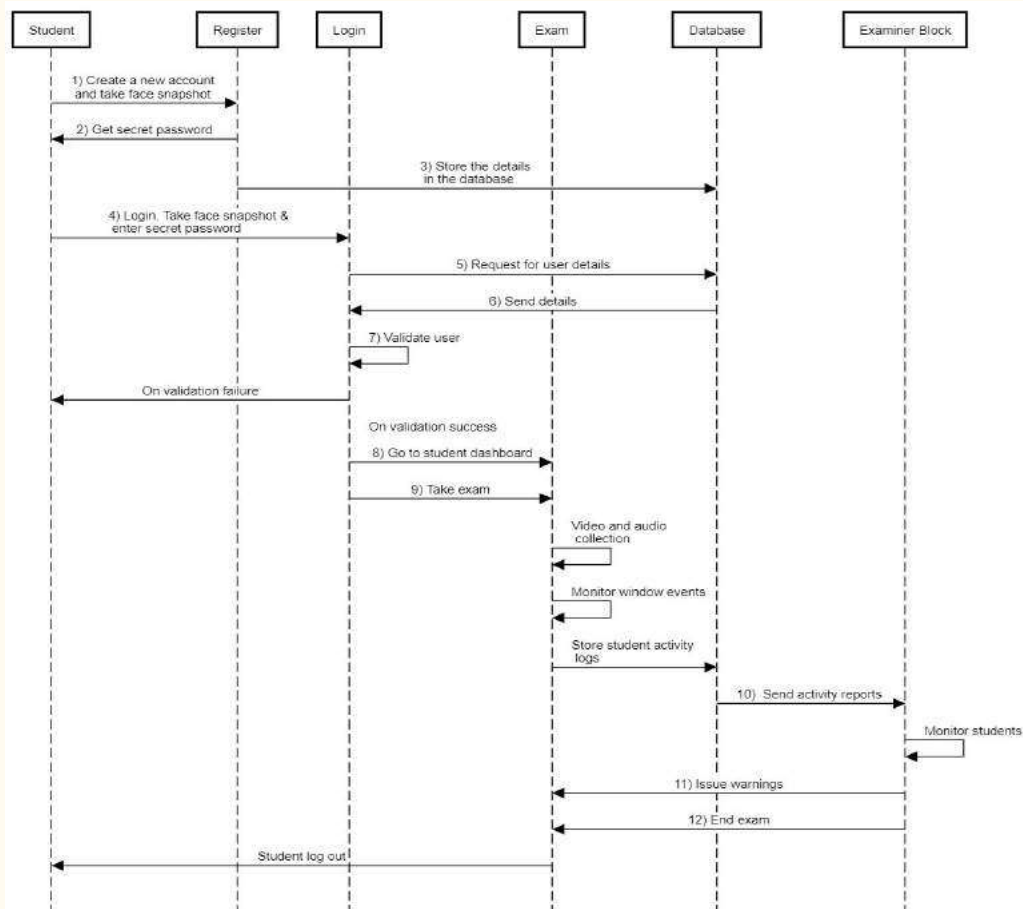


## 2.3 Description Of Use Case



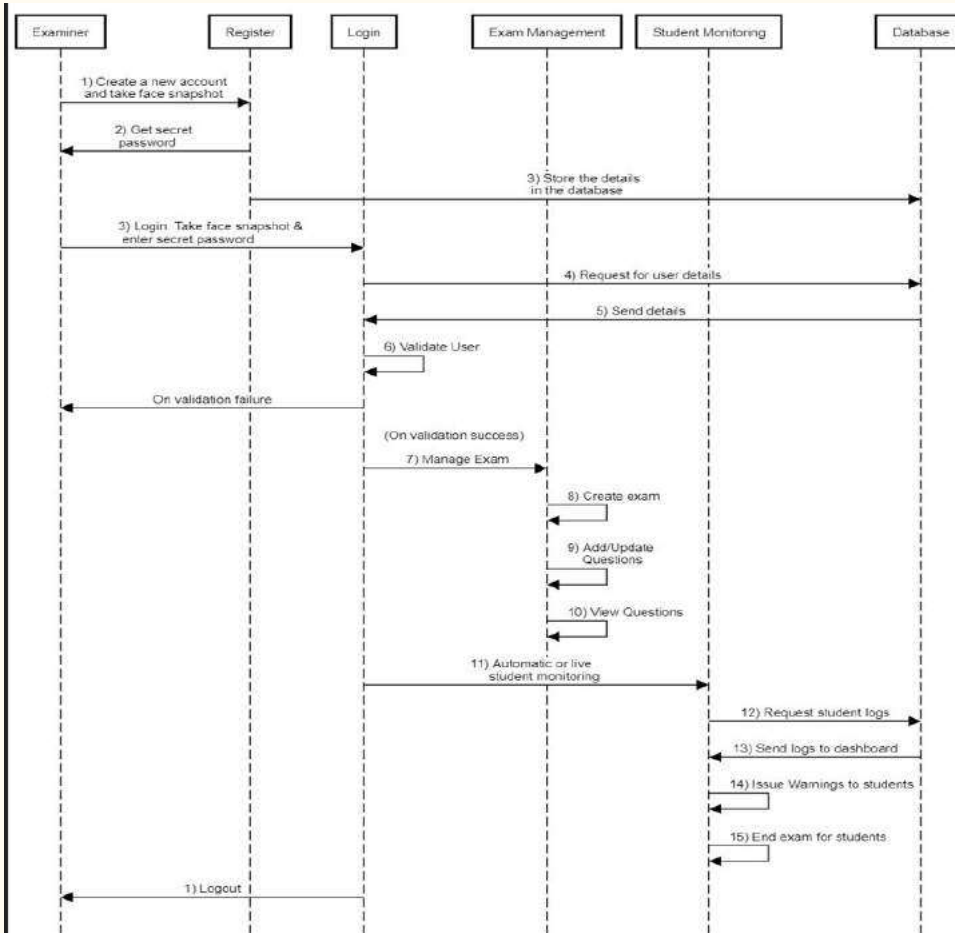
## 2.4 Activity diagram

### 1) Student's End

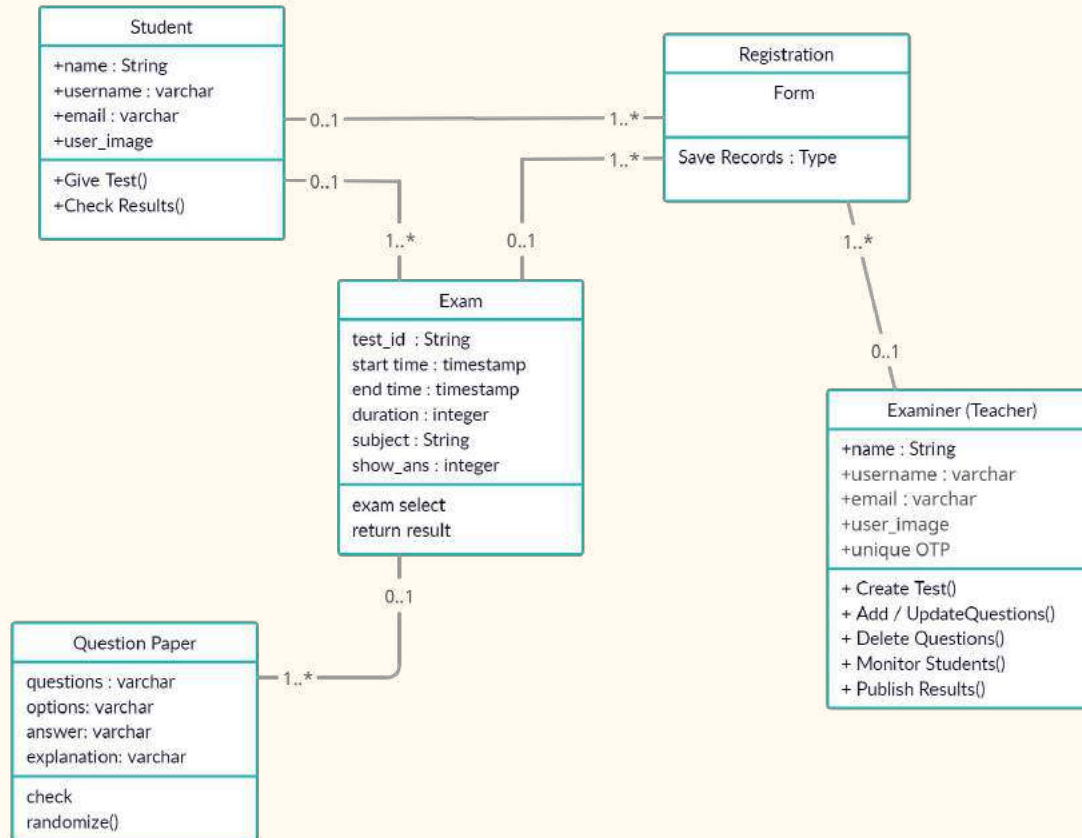


## 2.4 Activity diagram

### 1) Teacher's End



## 2.5 Class Diagram



# 3. Implementation

---

Implementation

```

@app.route('/register', methods=['GET', 'POST'])
def register():
    cur = mysql.connection.cursor()
    #form = RegisterForm(request.form)
    if request.method == 'POST':
        cur = mysql.connection.cursor()
        name = request.form['name']
        email = request.form['email']
        username = request.form['username']
        imgdata1 = request.form['image_hidden']
        ut = request.form['user_type']
        sesOTP = generateOTP()
        session['secretpassword'] = sesOTP
        cur = mysql.connection.cursor()
        cur.execute('SELECT * from users')
        data = cur.fetchone()
        cur.execute('INSERT INTO users(username,name,email,secretpassword,user_type,user_image) values(%s,%s,%s,%s,%s,%s)', (username,name,email,sesOTP,ut,imgdata1))
        msg1 = Message('From Smart E-Exam', sender = sender, recipients = [email])
        msg1.body = "Thanks for registering. Please keep this secret code with you as it is needed for future logins. Your secret code is "+sesOTP+"."
        mail.send(msg1)
        mysql.connection.commit()
        cur.close()
        flash('Thanks for registering! Please check your email to confirm your email address.', 'success')
        # change in login function to redirect to warning page
    elif request.method == 'POST':
        flash('Thanks for registering! Please check your email to confirm your email address.', 'success')
    return render_template('register.html')

```

(1) Register

```

@app.route('/login', methods=['GET','POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        ut = request.form['user_type']
        password_candidate = request.form['secretpassword']
        imgdata1 = request.form['image_hidden']
        ut = request.form['user_type']
        cur = mysql.connection.cursor()
        cur.execute('SELECT * from users where username = %s and user_type=%s', (username,ut))
        data = cur.fetchone()
        if data:
            password = data['secretpassword']
            username = data['username']
            email = data['email']
            name = data['name']
            imgdata2 = data['user_image']
            nparr1 = np.frombuffer(base64.b64decode(imgdata1), np.uint8)
            nparr2 = np.frombuffer(base64.b64decode(imgdata2), np.uint8)
            image1 = cv2.imdecode(nparr1, cv2.COLOR_BGR2GRAY)
            image2 = cv2.imdecode(nparr2, cv2.COLOR_BGR2GRAY)
            img_result = DeepFace.verify(image1, image2, enforce_detection = False)
            if img_result["verified"] == True and password_candidate == password:
                session['logged_in'] = True
                session['username'] = username
                session['name'] = name
                session['user_type'] = ut
                session['email'] = email
                if ut == 'student':
                    return redirect(url_for('student_dashboard'))
                else:
                    return redirect(url_for('teacher_dashboard'))
            else:
                error = 'Either image not verified or Invalid password'
                return render_template('login.html', error=error)
        cur.close()
    else:
        error = 'Username not found'
        return render_template('login.html', error=error)
    return render_template('login.html')

```

## (2) Login

```

app.route('/create-test', methods = ['GET', 'POST'])
@is_logged
def create_test():
    form = UploadForm()
    if request.method == 'POST' and form.validate_on_submit():
        f = form.doc.data
        filename = secure_filename(f.filename)
        f.save('questions/' + filename)
        cur = mysql.connection.cursor()
        d = doctodict('questions/' + f.filename.replace(' ', '_').replace('.', '').replace("'", ''))
        test_id = generate_slug(2)
        test_id = form.testid.data
        try:
            for no, data in d.items():
                marks = data['((MARKS)) (1/2/3...)]
                a = data['((OPTION_A))']
                b = data['((OPTION_B))']
                c = data['((OPTION_C))']
                d = data['((OPTION_D))']
                question = data['((QUESTION))']
                correct_ans = data['((CORRECT CHOICE)) (A/B/C/D)']
                explanation = data['((EXPLANATION)) (OPTIONAL)']

                cur.execute('INSERT INTO questions(test_id,qid,q,a,b,c,d,ans,marks,explanation) values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)',
                    (test_id,no,question,a,b,c,d,correct_ans,marks,explanation))
                mysql.connection.commit()

            start_date = form.start_date.data
            end_date = form.end_date.data
            start_time = form.start_time.data
            end_time = form.end_time.data
            start_date_time = str(start_date) + " " + str(start_time)
            end_date_time = str(end_date) + " " + str(end_time)
            show_result = form.show_result.data
            neg_mark = form.neg_mark.data

            duration = int(form.duration.data)*60
            password = form.password.data
            passp = form.pass_percentage.data
            subject = form.subject.data
            topic = form.topic.data
            proctoring_type = form.proctoring_type.data
            cur.execute('INSERT INTO teachers(username, test_id, start, end, duration, show_ans, password, passp , subject, topic,neg_mark, proctoring type) values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)',
                (id(session['username']), test_id, start_date_time, end_date_time, duration, show_result, password, passp , subject, topic, neg_mark, proctoring_type))
            mysql.connection.commit()
            cur.close()
            flash('Test ID: {test_id}', 'success')
            return redirect(url_for('create_test'))
        except Exception as e:
            print(e)
            flash('Invalid Input File Format', 'danger')
            return redirect(url_for('create_test'))
    return render_template('create_test.html', form = form)

```

### (3) Create Test



```

@app.route('/update/<testid>/<qid>', methods=['GET','POST'])
@is_logged
def update_quiz(testid, qid):
    if request.method == 'GET':
        cur = mysql.connection.cursor()
        cur.execute('SELECT * FROM questions where test_id = %s and qid = %s', (testid,qid))
        uresults = cur.fetchall()
        mysql.connection.commit()
        return render_template("updateQuestions.html", uresults=uresults)
    if request.method == 'POST':
        ques = request.form['ques']
        ao = request.form['ao']
        bo = request.form['bo']
        co = request.form['co']
        do = request.form['do']
        anso = request.form['anso']
        markso = request.form['mkso']
        cur = mysql.connection.cursor()
        cur.execute('UPDATE questions SET q = %s, a = %s, b = %s, c = %s, d = %s, ans = %s, marks = %s where test_id = %s and qid = %s', (ques,ao,bo,co,do,anso,markso,testid,qid))
        cur.connection.commit()
        flash('Updated successfully.', 'success')
        cur.close()
        return redirect(url_for('updatetidlist'))
    else:
        flash('ERROR OCCURED.', 'error')
        return redirect(url_for('updatetidlist'))

```

#### (4) Update Questions

```

@app.route('/give-test/<testid>', methods=['GET','POST'])
@is_logged
def test(testid):
    global duration,marked_ans,proctortype
    tid = testid
    if request.method == 'GET':
        try:
            data = {'duration': duration, 'marks': '', 'q': '', 'a': '', 'b': '', 'c': '', 'd': '' }
            return render_template("quiz.html" ,**data, answers=marked_ans, proctortype=proctortype,tid=tid)
        except:
            return redirect(url_for("give_test"))
    else:
        cur = mysql.connection.cursor()
        flag = request.form['flag']
        if flag == 'get':
            num = request.form['no']
            results = cur.execute('SELECT * from questions where test_id = %s and qid = %s',(testid, num))
            if results > 0:
                data = cur.fetchone()
                del data['ans']
                cur.close()
                return json.dumps(data)
            elif flag == 'mark':
                qid = request.form['qid']
                print(qid)
                ans = request.form['ans']
                print(ans)
                results = cur.execute('SELECT * from students where test_id = %s and qid = %s and username = %s', (testid, qid, session['username']))
                if results > 0:
                    cur.execute('UPDATE students set ans = %s where test_id = %s and qid = %s and username = %s', (ans,testid, qid, session['username']))
                    mysql.connection.commit()
                    cur.close()
                else:
                    cur.execute('INSERT INTO students(username,test_id,qid,ans) values(%s,%s,%s,%s)', (session['username'], testid, qid, ans))
                    mysql.connection.commit()
                    cur.close()
            elif flag == 'time':
                cur = mysql.connection.cursor()
                time_left = request.form['time']
                try:
                    cur.execute('UPDATE studentstestinfo set time_left=SEC_TO_TIME(%s) where test_id = %s and username = %s and completed=0', (time_left, testid, session['username']))
                    mysql.connection.commit()
                    cur.close()
                    return json.dumps({'time': 'fired'})
                except:
                    pass
            else:
                cur = mysql.connection.cursor()
                cur.execute('UPDATE studentstestinfo set completed=1 where test_id = %s and username = %s', (testid, session['username']))
                mysql.connection.commit()
                cur.close()
                flash("Test submitted successfully", 'info')
                return json.dumps({'sql': 'fired'})

```

## (5) Give Test

```
@app.route('/randomize', methods = ['POST'])
def random_gen():
    if request.method == "POST":
        id = request.form['id']
        cur = mysql.connection.cursor()
        results = cur.execute('SELECT count(*) from questions where test_id = %s', [id])
        if results > 0:
            data = cur.fetchone()
            total = data['count(*)']
            nos = list(range(1,int(total)+1))
            random.Random(id).shuffle(nos)
            cur.close()
            return json.dumps(nos)
```

(6) Randomize Questions in test

```
def generateOTP():
    otp=str(randint(00000,99999))
    return otp
```

(7) Generation of secret password (OTP)

```

wf.close()

def draw_outputs(img, outputs, class_names):
    boxes, objectness, classes, nums = outputs
    boxes, objectness, classes, nums = boxes[0], objectness[0], classes[0], nums[0]
    wh = np.flip(img.shape[0:2])
    for i in range(nums):
        x1y1 = tuple((np.array(boxes[i][0:2]) * wh).astype(np.int32))
        x2y2 = tuple((np.array(boxes[i][2:4]) * wh).astype(np.int32))
        img = cv2.rectangle(img, x1y1, x2y2, (255, 0, 0), 2)
        img = cv2.putText(img, '{} {:.4f}'.format(
            class_names[int(classes[i])], objectness[i]),
            x1y1, cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (0, 0, 255), 2)
    return img

yolo_anchors = np.array([(10, 13), (16, 30), (33, 23), (30, 61), (62, 45),
                        (59, 119), (116, 90), (156, 198), (373, 326)],
                        np.float32) / 416

yolo_anchor_masks = np.array([[6, 7, 8], [3, 4, 5], [0, 1, 2]])

```

(8) Applying boundary box  
around captured face

```

def get_frame(imgData):
    nparr = np.frombuffer(base64.b64decode(imgData), np.uint8)
    image = cv2.imdecode(nparr, cv2.COLOR_BGR2GRAY)
    ret = True

    size = image.shape
    font = cv2.FONT_HERSHEY_SIMPLEX
    model_points = np.array([
        (0.0, 0.0, 0.0),      # Nose tip
        (0.0, -330.0, -65.0), # Chin
        (-225.0, 170.0, -135.0), # Left eye left corner
        (225.0, 170.0, -135.0), # Right eye right corner
        (-150.0, -150.0, -125.0), # Left mouth corner
        (150.0, -150.0, -125.0) # Right mouth corner
    ])

    focal_length = size[1]
    center = (size[1]/2, size[0]/2)
    camera_matrix = np.array(
        [[focal_length, 0, center[0]],
         [0, focal_length, center[1]],
         [0, 0, 1]], dtype = "double"
    )

    img = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (320, 320))
    img = img.astype(np.float32)
    img = np.expand_dims(img, 0)
    img = img / 255
    class_names = [c.strip() for c in open("models/classes.txt").readlines()]
    boxes, scores, classes, nums = yolo(img)
    count=0
    mob_status = ""
    person_status = ""
    for i in range(nums[0]):
        if int(classes[0][i] -- 0):
            count +=1
        if int(classes[0][i] -- 67):
            print('Mobile Phone detected')
            mob_status = 1
        else:
            print('Mobile Phone not detected')
            mob_status = 0
    print(mob_status)

```

(9) Mobile detection ML code

```

user_move1=""
user_move2=""
if ret == True:
    faces = find_faces(image, face_model)
    for face in faces:
        marks = detect_marks(image, landmark_model, face)
        image_points = np.array([
            marks[30], # Nose tip
            marks[8], # Chin
            marks[36], # Left eye left corner
            marks[45], # Right eye right corner
            marks[48], # Left Mouth corner
            marks[54], # Right mouth corner
        ], dtype="double")
        dist_coeffs = np.zeros((4,1)) # Assuming no lens distortion
        (success, rotation_vector, translation_vector) = cv2.solvePnP(model_points, image_points, camera_matrix, dist_coeffs, flags=cv2.SOLVEPNP_UPNP)
        (nose_end_point2D, jacobian) = cv2.projectPoints(np.array([(0.0, 0.0, 1000.0)]), rotation_vector, translation_vector, camera_matrix, dist_coeffs)

        for p in image_points:
            cv2.circle(image, (int(p[0]), int(p[1])), 3, (0,0,255), -1)

        p1 = (int(image_points[0][0]), int(image_points[0][1]))
        p2 = (int(nose_end_point2D[0][0]), int(nose_end_point2D[0][1]))
        x1, x2 = head_pose_points(image, rotation_vector, translation_vector, camera_matrix)

        try:
            m = (p2[1] - p1[1])/(p2[0] - p1[0])
            ang1 = int(math.degrees(math.atan(m)))
        except:
            ang1 = 90

        try:
            m = (x2[1] - x1[1])/(x2[0] - x1[0])
            ang2 = int(math.degrees(math.atan(-1/m)))
        except:
            ang2 = 90

        if ang1 >= 48:
            user_move1 = 2
            print("Head is down")
        elif ang1 <= -48:
            user_move1 = 1
            print("Head is up")
        else:
            user_move1 = 0

        if ang2 >= 48:
            print("Head is right")
            user_move2 = 4
        elif ang2 <= -48:
            print("Head is left")
            user_move2 = 3
        else:
            user_move2 = 0

```

(10) User movement detection ML code

```

ret, jpeg = cv2.imencode('.jpg', image)
jpg_as_text = base64.b64encode(jpeg)

gaze.refresh(image)

frame = gaze.annotated_frame()
eye_movements = ""

if gaze.is_blinking():
    eye_movements = 1
    print("Blinking")
elif gaze.is_right():
    eye_movements = 4
    print("Looking right")
elif gaze.is_left():
    eye_movements = 3
    print("Looking left")
elif gaze.is_center():
    eye_movements = 2
    print("Looking center")
else:
    eye_movements = 0
    print("Not found!")
print(eye_movements)

```

(11) Eye tracking detection ML code

```
from flask import Flask,request, render_template, flash, redirect, url_for,session, logging, send_file , jsonify
from flask.mysql import MySQL
from wtforms import Form, StringField, TextAreaField, PasswordField, validators, DateTimeField, BooleanField, IntegerField, DecimalField, HiddenField, SelectField, RadioField
from flask_wtf import FlaskForm
from flask_wtf.file import FileField, FileRequired, FileAllowed
from passlib.hash import sha256_crypt
from functools import wraps
from werkzeug.utils import secure_filename
from docx import Document
from coolname import generate_slug
from datetime import timedelta, datetime
from random import randint
import numpy as np
import base64
import cv2
from deepface import DeepFace
from flask_mail import Mail, Message
from threading import Thread
from flask import render_template_string
from itsdangerous import URLSafeTimedSerializer
from validate_email import validate_email
import random
import json
import csv
import math
import operator
import pandas as pd
from wtforms.components import TimeField , DateField
#from wtforms.fields.html5 import DateField
from wtforms.validators import ValidationError, NumberRange , InputRequired
import socket
import camera
#from emailverifier import Client
from waitress import serve
```

(12) Libraries used



# 4. Testing



## Unit Testing

- Unit testing has been used as our main testing method.
- It is a testing process that tests individual components of the application in terms of its functionality and working.
- Our application have many small units like
  - Login and registration
  - creation of exam
  - Adding - updating questions
  - monitoring the students
  - Attempting test and checking results
  - Log generation.
- Helps finding issues in individual components at early stage.



# 5. Result

- Two-factor authentication which is the combination of secret password and the face validation at the time of login for both the examiners and the students.
- Log generation of all the student's activities done in the exam.
- Easy exam generation process for the exam creation and its management.
- Can detect all the window and user movements but difficult to analyze front view of the candidate.



(a) Registered User in system



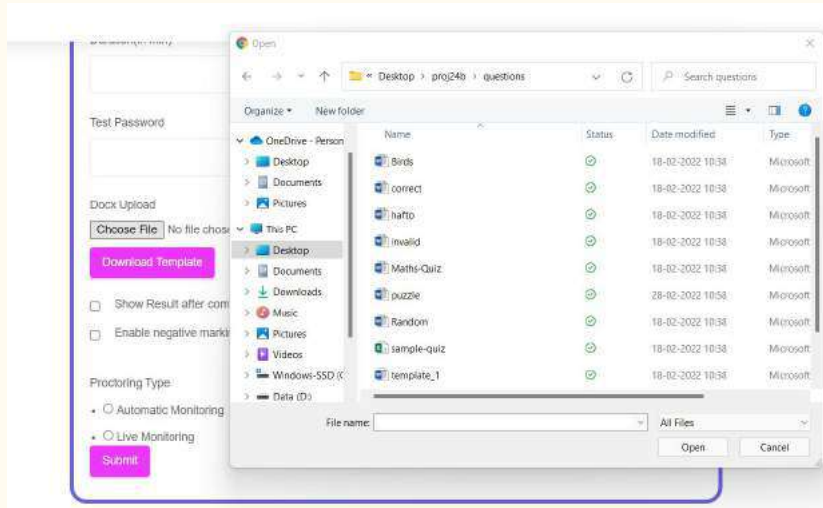
(b) Different User logging in system

Either image not verified or Invalid password

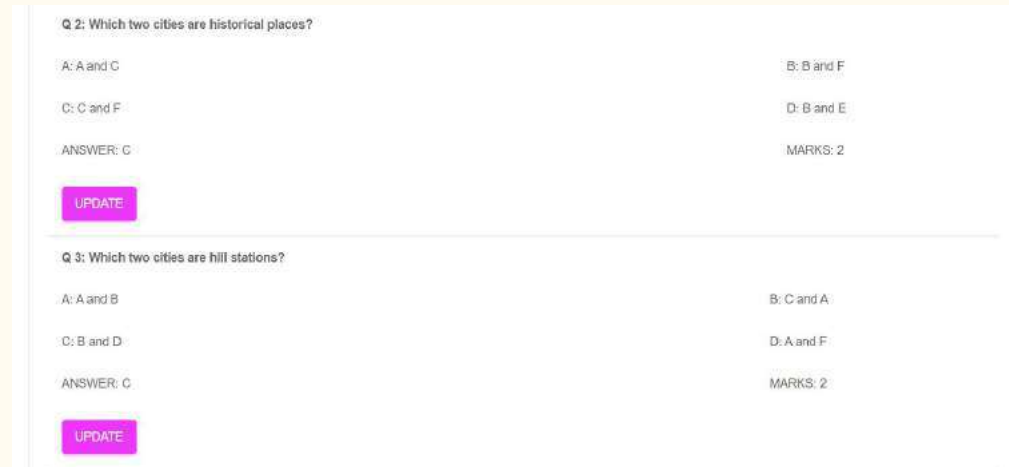
(c) Face validation error message

Fig 1. Face capture and validation process.

This shows registration process in which user in (a) captures their image. However a different user shown in (b) tries to login which result to images not getting validated. So the system displays the error to the user as image not verified.



(a) Importing questions from document file

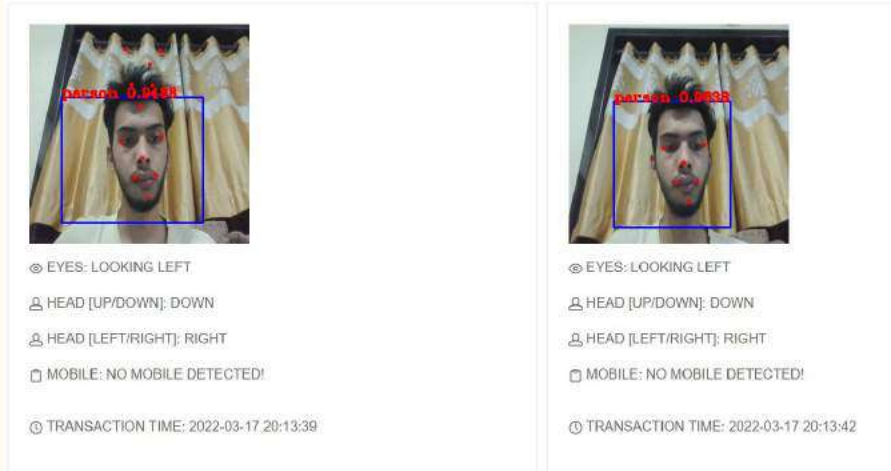


(b) Updating questions of the exam

Fig 2. Exam generation process.

This shows a document file containing questions being uploaded in (a).

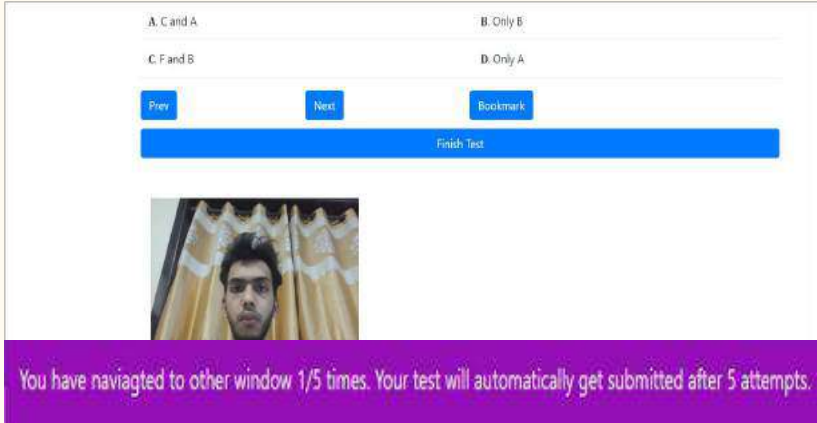
Part (b) shows teacher can later update the questions, options, marks, etc. by clicking respective question.



(a) Logs generation  
The student logs are displayed on teacher's dashboard after analyzing.



(b) Mobile detection  
Student is using mobile and it gets detected  
And shown as log



### (c) Alert on tab switching

Student performed tab navigation so system displayed alert message

WINDOW LOGS of melodic-mandrill of swapsap09		
NAME	WINDOW EVENT	TIME
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:14:23
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:14:34
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:14:50
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:01
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:08
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:16
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:18
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:27
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:28
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:30
Swapnil Sapre	WINDOW EVENT WAS OCCURRED!	2022-03-17 20:15:56

### (d) Tab switching activity

All the tab navigation attempts are shown as logs along with date and time

# 6. Conclusion and Future Scope

1. Face detection by capturing the face snapshot of the user and verification of it.
2. Able to perform student tracking throughout the exam and generating alert.
3. Able to detect the objects in the candidate's live environment and generate reports.

## Future Scope

1. To extend the examination to subjective based questions as well.
2. future scope of this work is to make the entire system a part of the e-learning framework and also provide support for practical as well as subjective-based examinations.
3. In future it aims to collect the background noise coming from the student's live environment and analyze it.

# References

- [1] Walaa M. Abd-Elhafiez, Mohamed Heshmat, and Seham Elaw. Efficient Method for Face Recognition and Its Role in Supporting E-Learning Systems. 2015. doi: 10.1109/ECONF.2015.21.
- [2] Wang Aimin and Wang Jipeng. Design and Implementation of Web-Based Intelligent Examination System. 2009. doi: 10.1109/WCSE.2009.77.
- [3] Hadian S. G. Asep and Yoanes Bandung. A Design of Continuous User Verification for Online Exam Proctoring on MLearning. 2019. doi: 10.1109/ICEEI47359.2019.8988786.
- [4] Yousef Atoum et al. Automated Online Exam Proctoring. 2017. doi: 10.1109/TMM.2017.2656064
- [5] Samuel S. Chua et al. Online Examination System with Cheating Prevention Using Question Bank Randomization and Tab Locking. 2019.
- [6] Radhika C. Damale and Bazeshree. V Pathak. Face Recognition Based Attendance System Using Machine Learning Algorithms. 2018. doi: 10.1109/ICCONS.2018.8662938.
- [7] Jegatha Deborah L et al. Secure Online Examination System for e-learning. 2019. doi: 10.1109/CCECE43985.2019. 9052408.
- [8] Ondrej Kainz et al. Enhancing Attention through the Eye Tracking. 2020. doi: 10.1109/ICETA51985.2020.9379229.
- [9] Mansi Mahendru and Sanjay Kumar Dubey. Real Time Object Detection with Audio Feedback using Yolo vs. YOLOv3. 2021. doi: 10.1109/Confluence51648.2021.9377064.
- [10] Arief Agus Sukmandhani and Indrajani Sutedja. Face Recognition Method for Online Exams. 2019. doi: 10.1109/ICIMTech.2019.8843831.

# Paper Publication

- Conference: 3rd International Conference on Deep Learning, Artificial Intelligence and Robotics (ICDLAIR) 2021
- Journal: Springer LNNS
- Published By: Swapnil Sapre , Kunal Shinde , Keval Shetta





**Thank You**

