**1. INTRODUCTION**

**1.1 Introduction**

Remote proctoring is the process of authenticating, authorizing and controlling the online examination process in a scalable manner. It is a technology that allows organizations to enable assessment anywhere and anytime, ensuring full security standards. In other words, candidates don’t need to come to a specific place as they can give examinations from their homes. In the traditional exam process, an invigilator has to be present at the exam centre to check candidates appearing for the exam. To examine 30-40 candidates, you require one invigilator. However, to conduct an exam of 1000+ candidates, you would need more than 25 invigilators controlling the exam process. Online proctoring can be conducted through the internet via the web camera of the candidate. It can record every single examination session from beginning to end, not just via video, but also captures desktop screens, chat logs and images.

**1.2 Motivation**

AI Proctoring system is designed for educational Institutions

* It can be used anywhere any time as it is a web based proctoring application
* It reduces the time of taking examinations of the students manually.
* This system will provide better security and transparency in the examination
* The system handles all the operations, and generates reports as soon as the test is finish, that include name, mark, time spent to solve exams.
* Don’t allows students to get indulge in any unfair means during the examination.

**1.3 Problem Definition**

The shift to online exams has introduced issues related to exam integrity, attendance verification, and fraud detection. Traditional methods of in-person invigilation and attendance tracking are no longer feasible, and existing online proctoring solutions often have limitations. These limitations include accuracy issues in participant identification, lack of real-time attendance monitoring, and difficulties in detecting sophisticated fraudulent activities. The proposed project seeks to overcome these challenges by developing an integrated system that employs facial recognition for participant identification, tracks attendance in real-time, and detects fraudulent behaviours during online exams.

**1.4 Objective of the Project**

To develop an AI-based exam proctoring system that leverages advanced technologies to ensure the integrity and security of online examinations. The primary goal is to create a robust and user-friendly platform that effectively monitors and detects any instances of cheating or misconduct during exams, thereby upholding the credibility and fairness of the assessment process. The system should employ cutting-edge artificial intelligence algorithms to analyse various behavioural and environmental cues, such as facial recognition, eye movement tracking, keyboard dynamics, and background noise analysis, to identify and flag suspicious activities. Additionally, the AI-based exam proctoring system aims to provide real-time alerts to administrators, offering them the ability to intervene promptly and take appropriate actions when irregularities are detected. The system should prioritize privacy and comply with relevant data protection regulations to ensure a secure and ethical examination environment for both students and institutions.

**1.5 Limitations of the Project**

**1.5.1 Technical Limitations**

Reliance on internet connectivity and technology may introduce technical issues such as network disruptions, camera malfunctions, or compatibility problems with certain devices.

Candidates from regions with poor internet connectivity may face difficulties in participating in remote proctored exams.

**1.5.2 False Positives and Negatives**

Automated proctoring systems may generate false positives (flagging non-cheating behaviour as suspicious) or false negatives (failing to detect cheating activities). Regular system updates and calibration are necessary to minimize these errors.

**1.5.3 Internet Dependence**

Online proctoring relies heavily on a stable internet connection. Any disruptions in the candidate's internet may affect the examination process.

**1.6 Organization of the Project**

**1.6.1 Research and Requirement Analysis**

* Conduct a thorough literature review on AI-based exam proctoring systems.
* Identify requirements and specifications through consultation with educators, administrators, and potential end-users.

**1.6.2 Algorithm Development and Testing**

* Develop advanced AI algorithms for behavioural analysis, voice and audio analysis, and biometric verification.
* Implement cheating detection mechanisms and integrate adaptive learning features. Conduct rigorous testing to ensure the accuracy and reliability of the algorithms.

**1.6.3 Software Development**

* Design and implement the software architecture for the proctoring system.
* Ensure compatibility with various devices and platforms.
* Integrate the system with learning management systems for seamless administration.

**1.6.4 Privacy and Ethics Implementation**

* Develop privacy-preserving features and mechanisms.
* Implement transparent monitoring and reporting to address ethical considerations.

**1.6.5 User Interface Development**

* Design an intuitive and user-friendly interface for both educators and students.
* Incorporate features for real-time monitoring and reporting.

**1.6.6 System Integration and Testing**

* Integrate all components of the system and conduct thorough testing.
* Address any issues identified during testing and optimize system performance.

**1.6.7 Deployment and Maintenance**

* Deploy the AI-based exam proctoring system in a controlled environment.
* Monitor system performance post-deployment and address any maintenance requirement

**2. SYSTEM SPECIFICATIONS**

**2.1 Software Specifications**

Software Requirement Specification is the starting point of the software developing activity. As system grew complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software is initiated by the client's needs. The SRS is the means of translating the ideas of the minds of the clients ((I/P) into a formal document (the O/P of the requirement phase).

|  |  |  |
| --- | --- | --- |
| * Programming Language | : | Python |
| * Operating System | : | Windows 8 |
| * Tool Kit | : | OpenCV, Flask |
| * IDE | : | Visual Studio Code |
| * Database | : | MySQL |

**2.2 Hardware Specifications**

The computer desktop or a laptop will be utilized to run the visual software in order to display what webcam had captured. A notebook which is a small, lightweight and inexpensive laptop computer is proposed to increase mobility.

System will be using

|  |  |
| --- | --- |
| **Name of Components** | **Specification** |
| Processor | Pentium III 630MHz, CORE i3, CORE i5 etc. |
| RAM | 128 MB, 8GB, 4GB |
| Hard Disk | 20 GB, 1 TB |
| Monitor | 15’’ color monitor |
| Keyboard | 122 keys |

**3. LITERATURE SURVEY**

**3.1 Introduction**

In recent years, the prevalence of online education has led to a surge in remote online exams. However, ensuring the integrity of these exams and tracking attendance accurately have emerged as significant challenges. Traditional methods of invigilation and attendance tracking are not as effective in the digital environment, which has prompted the need for advanced solutions that harness cutting-edge technologies. The proposed "AI-Based Exam Monitoring System for Fraud Detection and Attendance Tracking" aims to address these challenges by leveraging artificial intelligence and real-time tracking mechanisms.

The proposed integrated system represents a promising solution to the challenges posed by online exams. By leveraging facial recognition technology, real-time attendance monitoring, and advanced fraud detection techniques, this project aims to restore confidence in the integrity of online assessments. Through careful implementation and iterative refinement, educators and institutions can be better equipped to uphold academic honesty and ensure a level playing field for all participants.

**3.2 Existing System**

The current landscape of online examination systems reflects a dynamic interplay between the growing demand for remote assessment and the challenges posed by maintaining exam integrity. Traditional in-person invigilation methods have proven inadequate in the digital era, prompting the exploration of online proctoring solutions. In this section, we review the existing systems and technologies employed in online examination environments, shedding light on their functionalities, advantages, and limitations.

**3.2.1 Online Proctoring Solutions**

Online proctoring has emerged as a popular solution to address the need for secure and monitored online examinations. These systems often incorporate various features such as video monitoring, screen sharing detection, and audio analysis to track participants during exams. While online proctoring has provided a level of surveillance, it is not without its drawbacks. Issues such as accuracy in participant identification, real-time attendance tracking, and the detection of sophisticated fraudulent activities remain persistent challenges.

**ProctorU**

ProctorU is a live online proctoring service where human proctors monitor exams in real-time through webcam and audio feeds. They verify the identity of the test-taker and ensure compliance with exam rules.

**Examity**

Examity provides a range of proctoring solutions, including live proctoring and automated proctoring. Their platform uses AI to monitor for suspicious behavior and can flag issues for later review by human proctors.

**Proctorio**

Proctorio is an automated online proctoring solution that uses AI to monitor and record the exam session. It can flag potential issues such as excessive movement, background noise, or multiple faces in the camera. The recorded sessions can be reviewed later.

**Respondus LockDown Browser**

Respondus LockDown Browser is a browser lockdown tool that prevents students from accessing other websites or applications during an exam. It is often used in conjunction with webcam monitoring to enhance security.

**AIProctor**

AIProctor is an artificial intelligence-based proctoring system that uses facial recognition, eye tracking, and behavior analysis to monitor exams. It can automatically detect and flag suspicious activities.

**Safe Exam Browser**

Safe Exam Browser is not a proctoring system per se, but it is a tool that restricts the student's ability to browse the internet or access other applications during an exam. It enhances security by creating a controlled environment.

**Proctortrack**

Proctortrack is an automated proctoring solution that uses AI to monitor the exam environment and the test-taker's behaviour. It includes features such as facial recognition, room scanning, and browser lockdown.

**3.3 Disadvantages of Existing System**

While existing online examination systems have made strides in facilitating remote assessments, several limitations and challenges persist. These drawbacks underscore the need for innovative solutions that can overcome these shortcomings. The key disadvantages include

**3.3.1 Limited Accuracy in Participant Identification**

Current systems often rely on basic verification methods, such as photo matching or ID uploads, for participant identification. However, these methods may be susceptible to errors, leading to inaccuracies in the identification process. Instances of identity fraud or impersonation can compromise the integrity of exam results.

**3.3.2 Inadequate Real-time Attendance Tracking**

Many existing systems face challenges in providing robust real-time attendance tracking. This limitation can result in difficulties ensuring that participants remain actively engaged throughout the entire duration of the exam. Lack of continuous monitoring may allow for unauthorized assistance or other irregularities.

**3.3.3 Difficulty in Detecting Sophisticated Fraudulent Activities**

Current online proctoring solutions may struggle to detect advanced fraudulent activities, such as content sharing, collaboration, or the use of external aids. The limitations in fraud detection capabilities can compromise the security of the examination environment and undermine the fairness of the evaluation process.

**3.3.4 User Privacy Concerns**

Privacy concerns are a significant drawback of many online proctoring systems. The use of features like constant video monitoring and screen sharing detection raises questions about the protection of participants' privacy. Striking a balance between exam security and respecting individuals' privacy remains a challenging aspect of current systems.

**3.3.5 Dependence on Stable Internet Connection**

Online examination systems heavily depend on a stable internet connection. Participants in regions with unreliable internet connectivity may face challenges accessing and completing exams. This dependence introduces an element of inequality in the examination experience for students in different geographical locations.

**3.3.6 User Interface and Experience Issues**

Usability challenges in the user interface of existing systems can impact both administrators and participants. Complicated interfaces may lead to confusion and frustration, potentially affecting the overall user experience and system adoption.

**3.3.7 Lack of Customization and Flexibility**

Many current systems may lack the flexibility to accommodate various examination formats and may not provide customizable options for different educational institutions. A one-size-fits-all approach may not be suitable for diverse assessment needs.

**3.3.8 Vulnerability to Technological Cheating**

Some existing systems may be vulnerable to technological cheating, where participants exploit vulnerabilities in the software or employ external tools to gain an unfair advantage during the examination.

**3.4 Proposed System**

The shortcomings identified in existing online examination systems underscore the necessity for an innovative and comprehensive solution. Our proposed AI-Based Exam Monitoring System for Fraud Detection and Attendance Tracking is designed to address these challenges and elevate the standards of online assessments. This section details the key features and advantages of our proposed system.

**3.4.1 Facial Recognition for Participant Identification**

Our system incorporates advanced facial recognition technology to ensure accurate and reliable participant identification. By analyzing facial features and unique biometric patterns, the system minimizes the risk of identity fraud and impersonation, enhancing the overall integrity of the examination process.

**3.4.2 Real-Time Attendance Tracking**

One of the distinctive features of our proposed system is its robust real-time attendance tracking mechanism. Leveraging state-of-the-art algorithms, the system continuously monitors participant engagement, providing administrators with instantaneous insights into attendance patterns throughout the examination duration.

**3.4.3 Fraud Activity Detection**

Our system employs sophisticated fraud detection algorithms that go beyond traditional methods. By analyzing participant behaviour, screen activity, and other contextual cues, the system can identify and flag suspicious activities, such as content sharing, collaboration, or unauthorized aids, contributing to a more secure examination environment.

**3.4.4 Post-Exam Analysis**

After the completion of exams, our system offers a comprehensive post-exam analysis. This includes generating detailed reports on attendance patterns, highlighting any irregularities or potential fraud instances. The post-exam analysis provides valuable insights for administrators to assess the integrity of the examination process and make data-driven decisions.

**3.4.5 User-Friendly Interface**

Recognizing the importance of user experience, our system features an intuitive and user-friendly interface. Both administrators and participants can navigate the system with ease, enhancing overall usability and minimizing the learning curve associated with new technologies.

**3.4.7 Scalability**

Scalability is a key consideration in our system design. The architecture is engineered to accommodate various class sizes and institutions, ensuring optimal performance even as the user base expands.

**3.4.8 Ethical Considerations**

In addressing the ethical concerns related to privacy and data security, our system prioritizes the protection of user information. Strict adherence to relevant regulations and guidelines is a fundamental aspect of our design to ensure responsible and ethical use of technology.

**3.4.9 Future Enhancements**

Looking ahead, our system is designed to be adaptable to emerging technologies and educational trends. Future enhancements may include the integration of advanced machine learning models, additional features for customization, and compatibility with evolving educational platforms.

In summary, our proposed AI-Based Exam Monitoring System presents a holistic and cutting-edge solution to the challenges faced by current online examination systems. By combining facial recognition, real-time attendance tracking, and advanced fraud detection, our system aims to set a new standard for exam integrity and administration in the digital age.

**4. SYSTEM DESIGN**

**4.1 Data Flow Diagram**

* The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
* The data flow diagram (DFD) is one of the most important modelling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
* DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
* DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

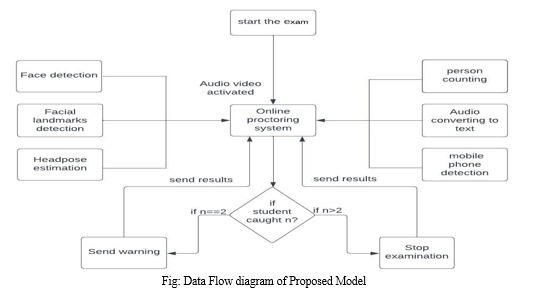
****

Fig 4.1 Data Flow Diagram

**4.2 UML Diagrams**

* UML stands for Unified Modelling Language. UML is a standardized general- purpose modelling 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. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.
* The Unified Modelling Language is a standard language for specifying. Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.
* The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.
* The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS**

The Primary goals in the design of the UML, are as follows.

* Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
* Provide extendibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development process.
* Provide a formal basis for understanding the modelling language
* Encourage the growth of OO tools market.
* Support higher level development concepts such as collaborations, frameworks, patterns

and components.

* Integrate best practices.

**4.2.1 Use Case Diagram**

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural, overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), 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.

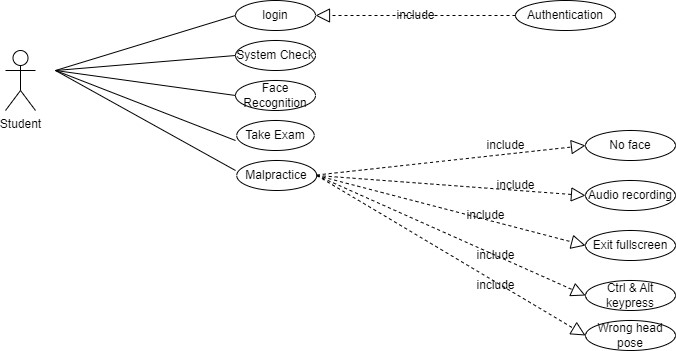


Fig 4.2 Use Case Diagram

**4.2.2 Class Diagram**

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information

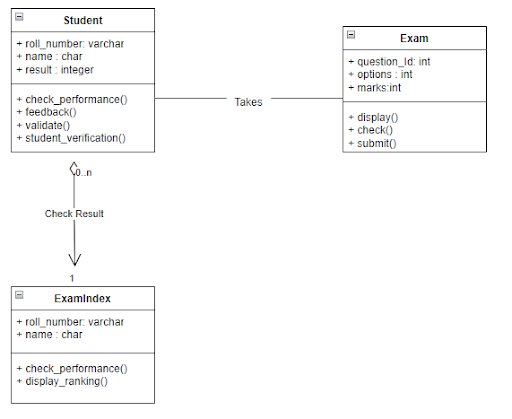


Fig 4.3 Class Diagram

**4.2.3 Sequence Diagram**

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

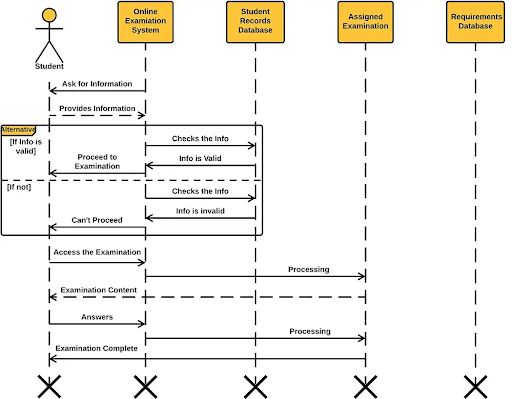
****

Fig 4.4 Sequence Diagram

**4.2.4 Activity Diagram**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow.

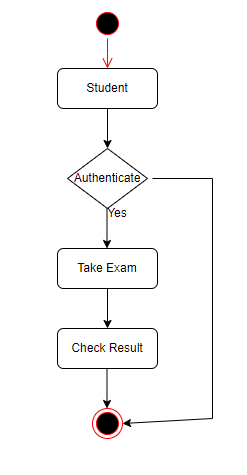
****

Fig 4.5 Activity Diagram

**5. IMPLEMENTATION**

In this section, we delve into the design and implementation details of the AI-Based Exam Monitoring System for Fraud Detection and Attendance Tracking. The system is developed using the Flask web framework for the backend and integrates computer vision (OpenCV) and audio processing (Paudie) technologies. The source code for the core functionalities is provided below.

**5.1 Web Application Initialization**

from flask import Flask, render template, Response, request

import cv2

import pyaudio

import audioop

import pymysql.cursors

# ... (omitting connection setup for brevity)

app = Flask(\_\_name\_\_)

# Initialize audio input stream

audio = pyaudio.PyAudio()

stream = audio.open(format=pyaudio.paInt16, channels=1, rate=44100, input=True, frames\_per\_buffer=1024)

# ... (omitting database setup for brevity)

@app.route('/')

def index():

return render\_template('quiz1.html')

@app.route('/video\_feed')

def video\_feed():

return Response(generate\_frames(), mimetype='multipart/x-mixed-replace;

boundary=frame')

# ... (omitting other routes for brevity)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, port=5005)

**5.2 Video Processing and Fraud Detection**

The generate\_frames function captures video frames from the user's webcam, processes facial features, and detects potential fraudulent activities:

# ... (omitting imports for brevity)

def generate\_frames():

# ... (omitting variable initializations for brevity)

while True:

ret, frame = cap.read()

if not ret:

break

audio\_data = stream.read(1024)

rms = audioop.rms(audio\_data, 2)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# ... (omitting face detection and other processing for brevity)

ret, buffer = cv2.imencode('.jpg', frame)

if not ret:

continue

frame = buffer.tobytes()

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

# ... (omitting other functions for brevity)

**5.3 Data Processing and Database Interaction**

The system interacts with a MySQL database to store student information and examination results. The following code demonstrates the handling of form data and database insertion:

# ... (omitting imports for brevity)

@app.route('/process\_data', methods=['POST', 'GET'])

def process\_data():

mb = request.form.get('nam')

count = request.form.get('scor')

roll = request.form.get('rol')

sql = '''insert into `sd`(name,roll,score)values(%s,%s,%s)'''

v = [mb, roll, count]

cursor.execute(sql, v)

connection.commit()

return render\_template("thank.html")

# ... (omitting other routes for brevity)

**5.4 Results Display**

The system fetches and displays examination results from the database:

# ... (omitting imports for brevity)

@app.route('/res', methods=['POST', 'GET'])

def res():

data = []

sql = "select \* from `sd`"

cursor.execute(sql)

for i in cursor:

data.append(list(i))

return render\_template("res.html", data=data)

This structured breakdown provides an organized and detailed overview of your code implementation within the "Design and Implementation" section of your report. Ensure that you include explanations, design considerations, and any unique features or algorithms employed in your system.

**5.5 System Enhancements**

While the current implementation provides a foundation for an AI-Based Exam Monitoring System, several enhancements can be considered to further improve its functionality and user experience. These enhancements may include:

**5.5.1 Integration of Machine Learning Models**

* Explore the integration of machine learning models for more sophisticated facial recognition and fraud detection. Train models on a diverse dataset to improve accuracy and reliability.

**5.5.2 Real-Time Notifications**

* Implement real-time notifications to alert administrators of suspicious activities, allowing for immediate intervention during online exams.

**5.5.3 Multi-Platform Compatibility**

* Enhance the system's compatibility by developing dedicated applications for various platforms, ensuring a seamless user experience across devices.

**5.5.4 User Authentication and Authorization**

* Implement a robust user authentication and authorization system to ensure that only authorized individuals can access the system, enhancing security.

**5.5.5 Continuous Monitoring**

* Consider implementing continuous monitoring features to track user activities even when exams are not in progress, providing a comprehensive solution for academic integrity.

**5.5.6 Usability Improvements**

* Conduct user testing to identify areas for improvement in the user interface and overall user experience. Implement changes based on feedback to optimize usability.

**5.5.7 Data Analytics and Reporting**

* Integrate data analytics capabilities to generate comprehensive reports on exam performance, attendance patterns, and user behaviour, providing valuable insights for administrators.

**5.5.8 Mobile Application Integration**

* Explore the development of a mobile application to provide users with a convenient and accessible platform for exam monitoring and result checking.
* These enhancements aim to elevate the system's capabilities and address potential areas for improvement, ensuring a more robust and user-friendly AI-Based Exam Monitoring System.

**5.6 Source code**

**5.6.1 Project Structure**

exam

|

+-- ImageAttendance

| |

| +-- 812201.jpg

| +-- 812202.jpg

| +-- 812203.jpg

| +-- 812204.jpg

| +-- ..........

|

+-- procted

| |

| +-- css

| | |

| | +-- style.css

| |

| +-- js

| | |

| | +--controller

| | | |

| | | +-- list.js

| | | +-- quiz.js

| | | +-- result.js

| | +--factories

| | | |

| | | +-- dataservice.js

| | | +-- quizMetrics.js

| | |

| | +-- app.js

| |

| +-- quiz.html

|

+-- templates

| |

| +--authentication.html

| +--dashboard.html

| +--home.html

| +--results.html

|

+-- attendance.csv

|

+-- procted

| |

| +-- css

| | |

| | +-- style.css

| |

| +-- js

| | |

| | +--controller

| | | |

| | | +-- list.js

| | | +-- quiz.js

| | | +-- result.js

| | +--factories

| | | |

| | | +-- dataservice.js

| | | +-- quizMetrics.js

| | |

| | +-- app.js

| |

| +-- quiz.html

|

+-- templates

| |

| +--authentication.html

| +--dashboard.html

| +--home.html

| +--results.html

|

+-- attendance.csv

**app.py**

|

+-- app.py

**app.py**

from flask import Flask, render\_template, Response,request, redirect,session, send\_from\_directory

import os

from flask import send\_from\_directory

from flask\_login import login\_required

import cv2

import pyaudio

import json

import audioop

import pymysql.cursors

import numpy as np

import face\_recognition

import os

from datetime import datetime

app = Flask(\_name\_)

app.secret\_key = 'bsddsjvGVVJ876483jVJV'

connection = pymysql.connect(host='localhost',

user='root',

password='@g209X1A05H6g@',

database = 'students'

)

cursor = connection.cursor()

audio = pyaudio.PyAudio()

stream = audio.open(format=pyaudio.paInt16, channels=1, rate=44100, input=True, frames\_per\_buffer=1024)

path = 'ImagesAttendance'

images = []

classNames = []

myList = os.listdir(path)

name\_global=''

print(myList)

for cl in myList:

curImg = cv2.imread(f'{path}/{cl}')

images.append(curImg)

classNames.append(os.path.splitext(cl)[0])

print(classNames)

def findEncodings(images):

encodeList = []

for img in images:

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

encode = face\_recognition.face\_encodings(img)[0]

encodeList.append(encode)

return encodeList

def markAttendance(name):

with open('Attendance.csv','r+') as f:

myDataList = f.readlines()

nameList = []

for line in myDataList:

entry = line.split(',')

nameList.append(entry[0])

if name not in nameList:

now = datetime.now()

dtString = now.strftime('%H:%M:%S')

f.writelines(f'\n{name},{dtString}')

encodeListKnown = findEncodings(images)

print('Encoding Complete')

def verify\_face():

cap = cv2.VideoCapture(0)

while True:

ret, img = cap.read()

if not ret:

break

imgS = cv2.resize(img,(0,0),None,0.25,0.25)

imgS = cv2.cvtColor(imgS, cv2.COLOR\_BGR2RGB)

facesCurFrame = face\_recognition.face\_locations(imgS)

encodesCurFrame = face\_recognition.face\_encodings(imgS,facesCurFrame)

for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):

matches = face\_recognition.compare\_faces(encodeListKnown,encodeFace)

faceDis = face\_recognition.face\_distance(encodeListKnown,encodeFace)

matchIndex = np.argmin(faceDis)

if matches[matchIndex]:

name = classNames[matchIndex].upper()

#print(name)

y1,x2,y2,x1 = faceLoc

y1, x2, y2, x1 = y1\*4,x2\*4,y2\*4,x1\*4

cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)

cv2.rectangle(img,(x1,y2-35),(x2,y2),(0,255,0),cv2.FILLED)

cv2.putText(img,name,(x1+6,y2- 6),

cv2.FONT\_HERSHEY\_COMPLEX,1,(255,255,255),2)

global name\_global

name\_global=name

markAttendance(name)

ret, buffer = cv2.imencode('.jpg', img)

if not ret:

continue

img = buffer.tobytes()

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + img + b'\r\n')

def generate\_frames():

max\_blink\_count = 5

blink\_counter = 0

cap = cv2.VideoCapture(0)

prev\_eye\_state = True

tab\_switch\_detected = False

noise\_detected = False

noise\_reset\_counter = 0

noise\_reset\_threshold = 40

while True:

ret, frame = cap.read()

if not ret:

break

audio\_data = stream.read(1024)

rms = audioop.rms(audio\_data, 2)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +

'haarcascade\_frontalface\_default.xml')

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,

minSize=(30, 30))

if len(faces) == 0:

cv2.putText(frame, "Face Not Detected!", (20, 50),

cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 0, 255), 2)

for (x, y, w, h) in faces:

roi\_gray = gray[y:y + h, x:x + w]

eyes = cv2.Canny(roi\_gray, 100, 200)

eye\_state = True

contours, \_ = cv2.findContours(eyes, cv2.RETR\_EXTERNAL,

cv2.CHAIN\_APPROX\_SIMPLE)

for contour in contours:

area = cv2.contourArea(contour)

if area < 1000:

eye\_state = False

if eye\_state != prev\_eye\_state:

prev\_eye\_state = eye\_state

if not eye\_state:

blink\_counter += 1

key = cv2.waitKey(1)

if key == 9:

tab\_switch\_detected = True

if rms > 300:

noise\_detected = True

noise\_reset\_counter = 0

else:

if noise\_reset\_counter < noise\_reset\_threshold:

noise\_reset\_counter += 1

else:

noise\_detected = False

if blink\_counter > max\_blink\_count or noise\_detected:

cv2.putText(frame, "Suspicious Activity Detected!", (20, 80),

cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 0, 255), 2)

ret, buffer = cv2.imencode('.jpg', frame)

if not ret:

continue

frame = buffer.tobytes()

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

@app.route('/')

def index():

return redirect('/home')

@app.route('/dashboard')

def dashboard():

if 'rollno' in session:

data=[]

sql="select \* from stu\_data where rollno="+str(session['rollno'])+";"

cursor.execute(sql)

for i in cursor:

data.append(list(i))

print(data)

return render\_template('dashboard.html',data=data)

return redirect('/home')

@app.route('/home', methods=['GET', 'POST'])

def login():

if 'rollno' in session:

return redirect('/dashboard')

if request.method == 'POST':

roll = request.form['rollno']

pwd = request.form['pwd']

data=[]

sql="select \* from stu\_details where rollno="+str(roll)+";"

cursor.execute(sql)

for i in cursor:

data.append(list(i))

print(roll,pwd,data)

if(data[0][0]==roll.strip() and data[0][1]==pwd.strip()):

session['rollno'] = roll

return redirect('/dashboard')

else :

return "<h2 style='color:red;'> Invalid Password </h2> <a href='/home'>Goback</a>"

return render\_template('home.html')

@app.route('/protected/<path:filename>')

def protected(filename):

datac=[]

sql="select \* from quiz\_results where rollno="+str(session['rollno'])+";"

cursor.execute(sql)

for i in cursor:

datac.append(list(i))

if(len(datac)>0):

return "<h2 style='color:red;'>You Have Already Taken the Test</h2> <br> <a

href='/home'>Go Back</a>"

elif 'rollno' in session and session['rollno']==name\_global:

print(name\_global)

return send\_from\_directory('protected', filename)

elif 'rollno' in session and session['rollno']!=name\_global :

return "<h2 style='color:red;'>User is Not Matched with our DataBase</h2> <a

href='/dashboard'>Goback</a>"

return redirect('/home')

@app.route('/video\_feed')

def video\_feed():

return Response(generate\_frames(), mimetype='multipart/x-mixed-replace;

boundary=frame')

@app.route('/auth\_video\_feed')

def auth\_video\_feed():

return Response(verify\_face(), mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/logout')

def logout():

session.pop('rollno', None)

return redirect('/')

@app.route('/store-quiz-result', methods=['POST'])

def result():

data = request.get\_json()

print(data['percentage'])

try:

sql='''insert into quiz\_results(rollno,marks)values(%s,%s)'''

v=[str(session.get('rollno')),str(data['percentage'])]

cursor.execute(sql,v)

connection.commit()

return f'data: Success'

except Exception as e:

print(e)

return f'data: Failed'

@app.route('/auth')

def auth():

return render\_template('auth.html')

@app.route('/results')

def res():

data=[]

sql="select \* from quiz\_results;"

cursor.execute(sql)

for i in cursor:

data.append(list(i))

if(len(data)==0):

return "<h2 style='color:red;'>Results Not Yet Available</h2> <br> <a href='/home'>Go

Back</a>"

else:

return render\_template('results.html',data=data)

@app.errorhandler(404)

def not\_found(e):

return "<h2>404 Not Found</h2> <a href='/home'>Goback</a>"

if \_name\_ == '\_main\_':

app.run(debug=True,port=5056)

**templates\home.html**

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>AI Based Exam Proctering System </title>

<style>

body {font-family: Arial, Helvetica, sans-serif;}

input[type=text], input[type=password] {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

box-sizing: border-box;

}

button {

background-color: #045aaa;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

cursor: pointer;

width: 100%;

}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

position: relative;

}

img.avatar {

width: 40%;

border-radius: 50%;

}

.container {

padding: 16px;

}

span.psw {

float: right;

padding-top: 16px;

}

.modal {

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgb(0,0,0);

background-color: rgba(0,0,0,0.4);

padding-top: 60px;

}

.modal-content {

background-color: #fefefe;

margin: 0.5% auto 15% auto;

border: 1px solid #888;

width: 40%;

}

.close {

position: absolute;

right: 25px;

top: 0;

color: #000;

font-size: 35px;

font-weight: bold;

}

.close:hover,

.close:focus {

color: red;

cursor: pointer;

}

.animate {

-webkit-animation: animatezoom 0.6s;

animation: animatezoom 0.6s

}

@-webkit-keyframes animatezoom {

from {-webkit-transform: scale(0)}

to {-webkit-transform: scale(1)}

}

@keyframes animatezoom {

from {transform: scale(0)}

to {transform: scale(1)}

}

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;

}

.cancelbtn {

width: 100%;

}

}

</style>

</head>

<body>

<div id="title">

<h1 style=" margin-left: 30%;">AI BASED EXAM PROCTERING SYSTEM</h1>

<button onclick="document.getElementById('id01').style.display='block'" style="width:auto; float: right; position: absolute; top: 5px; right: 8px;">Login</button>

</div>

<div>

<video style="margin-left: 25%;">

<source src="/static/ai.webm">

</video>

</div>

<div id="id01" class="modal">

<form class="modal-content animate" action="/home" method="post">

<div class="imgcontainer">

<span onclick="document.getElementById('id01').style.display='none'" class="close"

title="Close Modal">&times;</span>

<img src="/static/img\_avatar2.png" alt="Avatar" class="avatar">

</div>

<div class="container">

<label for="uname"><b>Roll Number</b></label>

<input type="text" placeholder="Enter Roll Number" name="rollno" required>

<label for="psw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="pwd" required>

<button type="submit">Login</button>

<label><input type="checkbox" checked="checked" name="remember"> Remember me

</label>

</div>

<div class="container" style="background-color:#f1f1f1">

<button type="button" onclick="document.getElementById('id01').style.display='none'"

class="cancelbtn">Cancel</button>

</div>

</form>

</div>

<script>

var modal = document.getElementById('id01');

window.onclick = function(event) {

if (event.target == modal) {

modal.style.display = "none";

}

}

</script>

</body>

</html>

**templates\dashboard.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Student Dashboard</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<style>

.row.content {height: 550px}

.sidenav {

background-color: #f1f1f1;

height: 100%;

}

@media screen and (max-width: 767px) {

.row.content {height: auto;}

}

</style>

</head>

<body>

<div class="container-fluid">

<div class="row content">

<div class="col-sm-3 sidenav hidden-xs">

<h2><i class="glyphicon glyphicon-user"></i> {{session['rollno']}}</h2>

<ul class="nav nav-pills nav-stacked">

<li class="active"><a href="/results">Results</a></li>

<li class="active"><a href="/auth">Start Quiz</a></li>

<li class="active"><a href="/logout">Logout</a></li>

</ul><br>

</div>

<br>

<div class="col-sm-9">

<div class="well">

<h4 style="color: blueviolet">Dashboard</h4>

<p>Roll Number &nbsp;&nbsp;&nbsp; &nbsp;&nbsp; :&nbsp;

<strong>{{data[0][0]}} </strong></p>

<p>Student Name &nbsp;&nbsp;&nbsp; :&nbsp; <strong>{{data[0][1]}}

</strong></p>

<p>Father Name &nbsp; &nbsp;&nbsp;&nbsp; :

&nbsp;<strong>{{data[0][2]}}</strong></p>

<p>Mother Name &nbsp; &nbsp;&nbsp; :&nbsp; <strong>{{data[0][3]}}</strong></p>

</div>

<div class="row">

<div class="col-sm-3">

<div class="well">

<h4>Live Quiz</h4>

<p>1</p>

</div>

</div>

<div class="col-sm-3">

<div class="well">

<h4>UpComming Quizes</h4>

<p>0</p>

</div>

</div>

<div class="col-sm-3">

<div class="well">

<h4>Total Quizes</h4>

<p>0</p>

</div>

</div>

<div class="col-sm-3">

<div class="well">

<h4 style="color: red;">Missing</h4>

<p>0</p>

</div>

</div>

</div>

<div class="row">

<div class="row">

<div class="col-sm-8">

<div class="well">

<p>Remarks</p>

</div>

</div>

<div class="col-sm-4">

<div class="well">

<p>Total Active sessions : 1</p>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

**templates\authentication.html**

<!DOCTYPE html>

<html lang="en">

<head>

<style>

.ha{

position: absolute;

top:35px;

justify-content: center;

font-size : 100px;

}

body{

height: 100vh;

background: -webkit-repeating-linear-gradient(-45deg, #71b7e6, #69a6ce, #b98acc,

#ee8176, #b98acc, #69a6ce, #9b59b6);

background-size: 400%;

}

</style>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" g

rel="stylesheet" integrity="sha384- crossorigin="anonymous">

H2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/vI1Bx"

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Authentication</title>

</head>

<body class="" >

<div class="container" style="display: flex;justify-content: center;height: 700px;align-

items: center;">

<div class="row d-flex justify-content-center align-items-center a1" id="fst"

style="height:400px;width:600px;" >

<div class="col-12">

<div class="d-flex justify-content-center" ><p class="text-light ha">?</p></div>

<div class="card shadow-lg bg-dark">

<div class="card-body bg-dark">

<div class="card-text bg-dark">

<ul class="list-group bg-dark">

<li class="list-group-item lg bg-dark ">

<div class="row bg-dark" >

<div class="col-12 d-flex justify-content-center">

<img id="video\_feed" src="/auth\_video\_feed" alt="Live Video

Feed" width="300" height="250" >

</div>

<button onclick="window.location.href='/protected/quiz.html'"

style="width: 100%;" class="btn btn-primary">Capture

Image</button><br></li>

</ul>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

**templates\results.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Results</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Results</h2>

<p>Turtle Facts Quiz</p>

<a href="/dashboard">Goback</a>

<table class="table">

<thead>

<tr>

<th>Roll Number</th>

<th>Marks</th>

</tr>

</thead>

<tbody>

<tr>

{% for i in data %}

<td>{{i[0]}}</td>

<td>{{i[1]}}</td>

{% endfor %}

</tr>

</tbody>

</table>

</div>

</body>

</html>

**6. TESTING AND VALIDATION**

**INTRODUCTION**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

Testing is a critical element that assures the quality and effectiveness of the proposed system in satisfying its objectives. Testing is done in various stages in the system designing and Implementation process with the objective of developing a transparent, flexible, and secured system. Testing is an integral part of software development. The testing process, in way, certifies, whether the product, that is developed, complies with the standards, that it was designed to. The testing process involves the building of test cases against which the product has to be tested.

**Test Objectives**

* Testing is a process of executing a program with the intent of finding an error.
* A good case is one that has a high probability of finding an undiscovered error.
* A Successful test is one that uncovers a yet undiscovered error. If the testing is conducted successfully, it will uncover errors in the software. Testing can’t show the absence of defects. It can only show that software defects are present.

**Testing Principles**

Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

**6.1 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**6.1.1 White Box Testing**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It has a purpose. It is used to test areas that cannot be reached from a black-box level.

**6.1.2 Black Box Testing**

In this testing by knowing the internal operation of a product, tests can be conducted to ensure that "all gears mesh", that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black-box test case design are

* Graph based testing methods
* Equivalence partitioning
* Boundary value analysis
* Comparison Testing

**6.2 Testing Strategies**

A software testing strategy provides a road map for the software developer. Testing is a set of activities that can be planned in advance and conducted systematically. For this reason, a template for software testing, a set of steps into which we can place specific test case design methods should be defined for the software engineering process.

Any software testing strategy should have the following characteristics

* Testing begins at the module level and works outward toward the integration of the entire computer based system.
* Different testing techniques are appropriate at different points in time.
* The developer of the software and an independent test group conducts testing.
* Testing and debugging are different activities but debugging must be accommodated in any testing strategy.

**6.3 Levels of Testing**

**6.3.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.

It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**6.3.2 Integration Testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level.

**6.4 Test Results**

All the test cases mentioned above passed successfully. No defects encountered.Functional TestingValid Input: identified classes of valid input must be accepted.Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**6.4.1 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end-user. It also ensures that the system meets the functional requirements.

**6.4.2 Test Results**

All the test cases mentioned above passed successfully. No defects encountered

**7. SCREENSHOTS**

**Home Page**

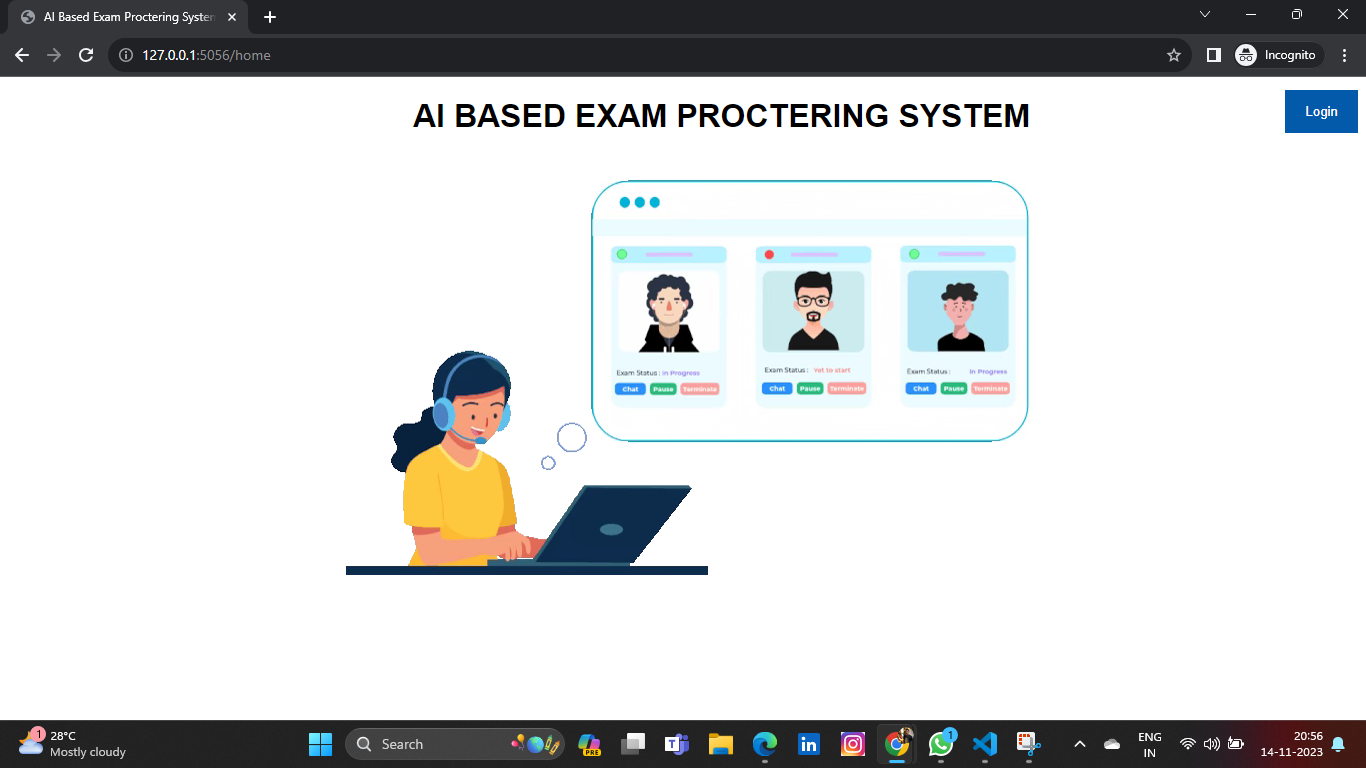
****

Fig 7.1 Home Page

**Login Page**

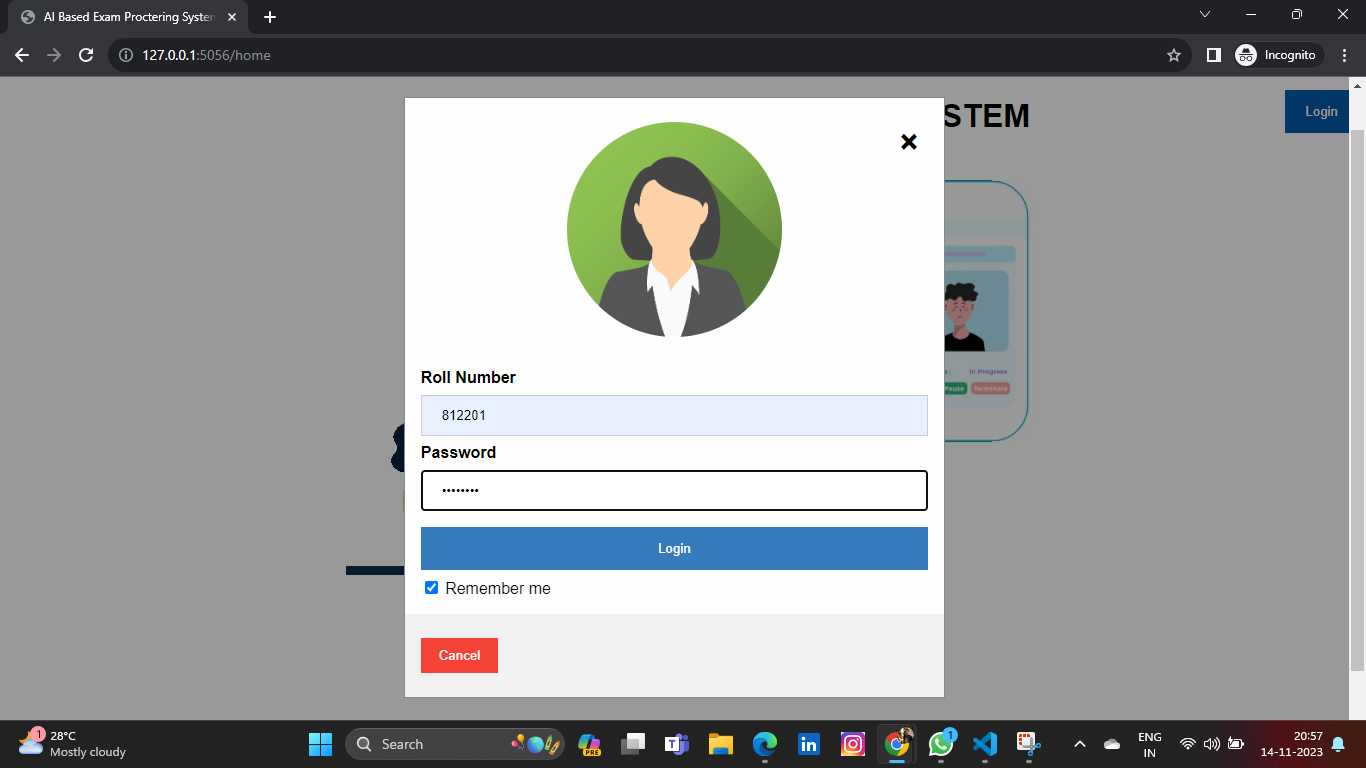
****

Fig 7.2 Login Page

**Dashboard**

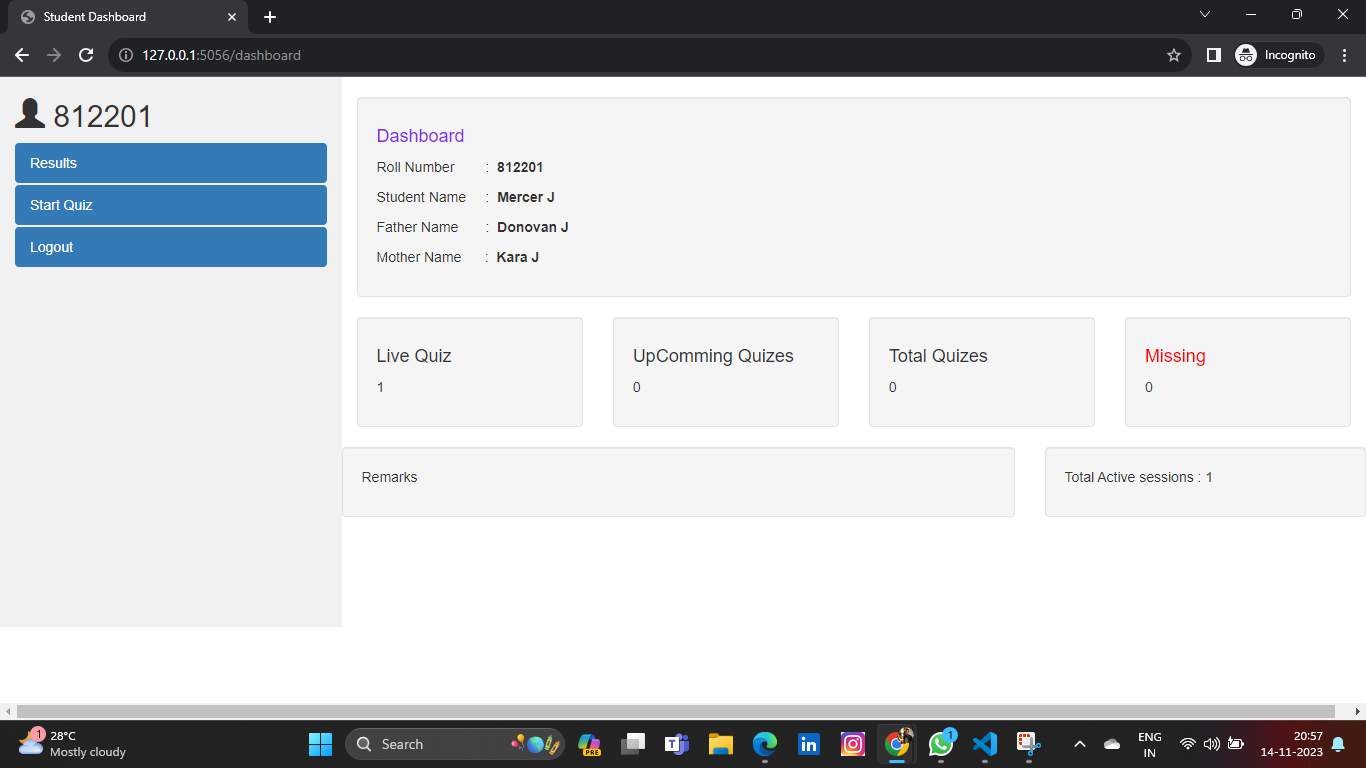
****

Fig 7.3 Dashboard

**User Authentication**

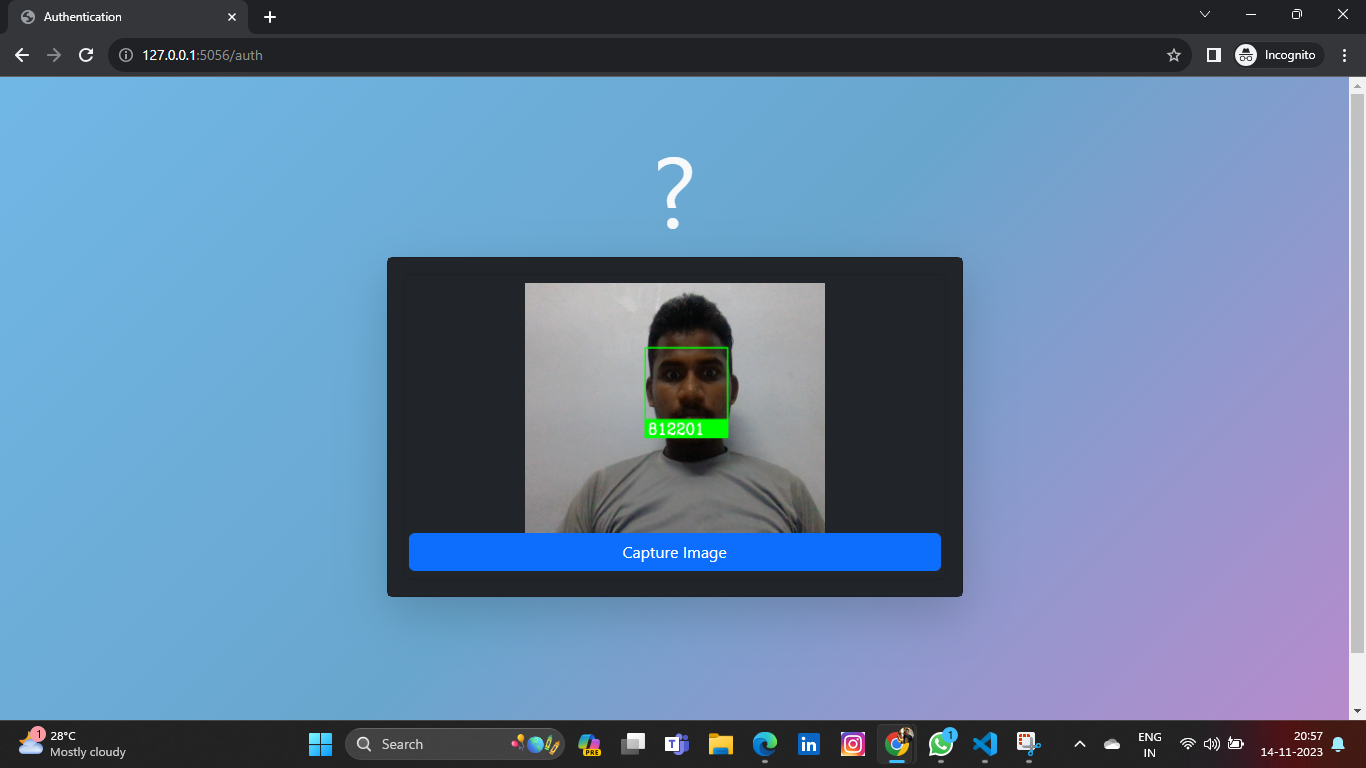
****

Fig 7.4 User Authentication

**Detecting Unidentified User**

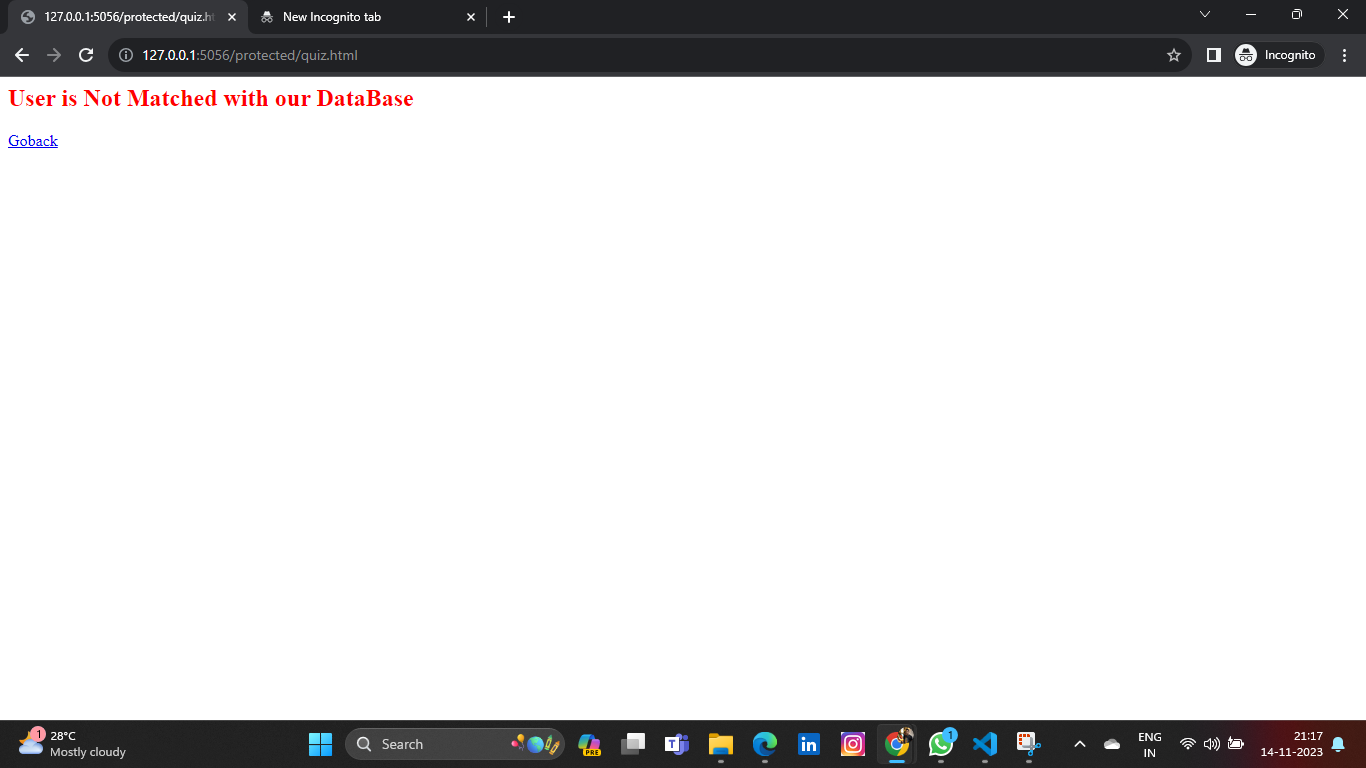
****

Fig 7.5 Detecting Unidentified User

**Entering Full Screen Mode**

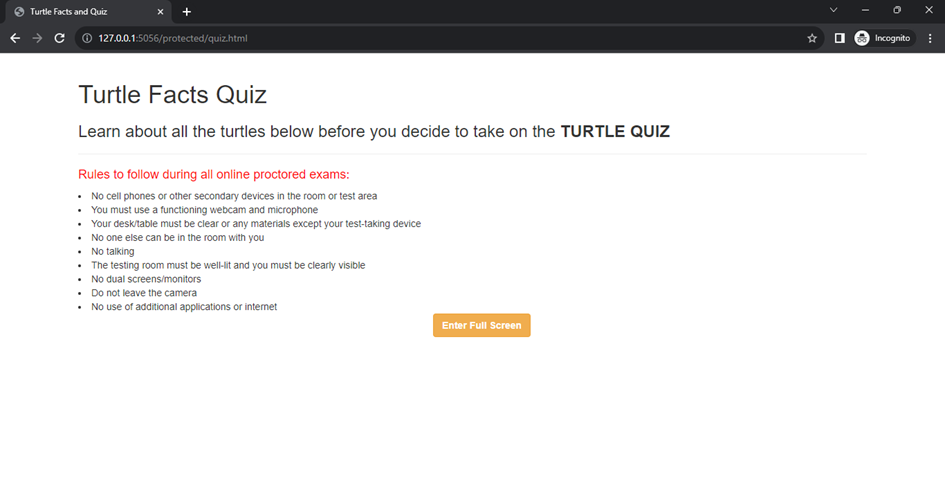
****

Fig 7.6 Entering Full Screen Mode

**Starting Quiz**

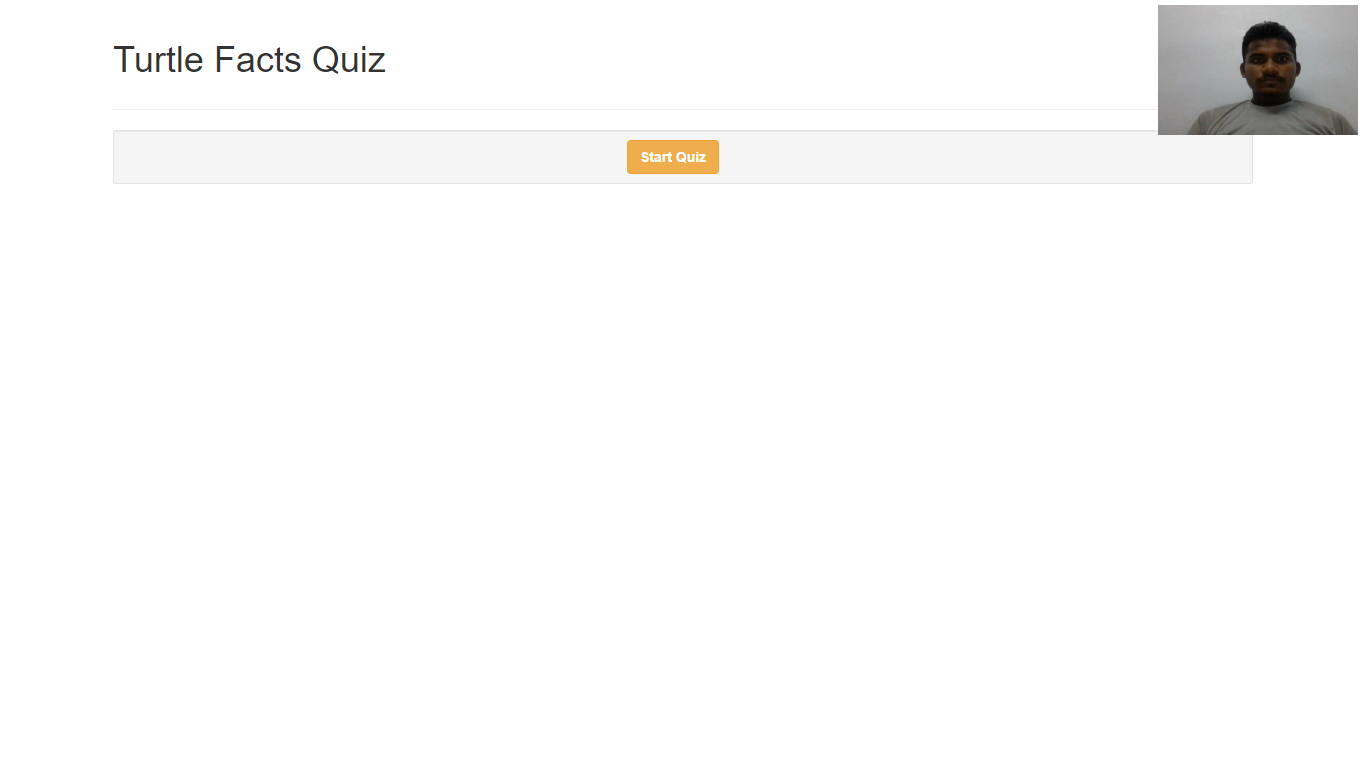
****

Fig 7.7 Starting Quiz

**Face Not Detected During Exam**

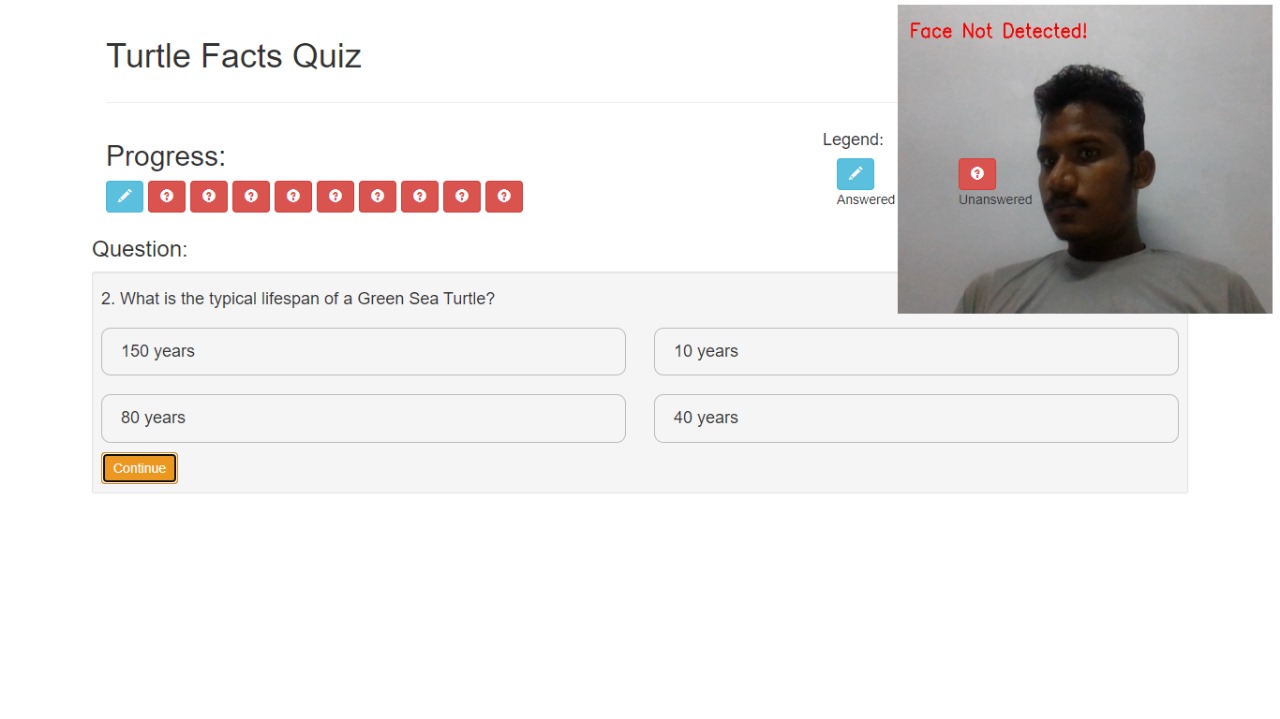


Fig 7.8 Face Not Detected During Exam

**Suspicious Activity Detection**

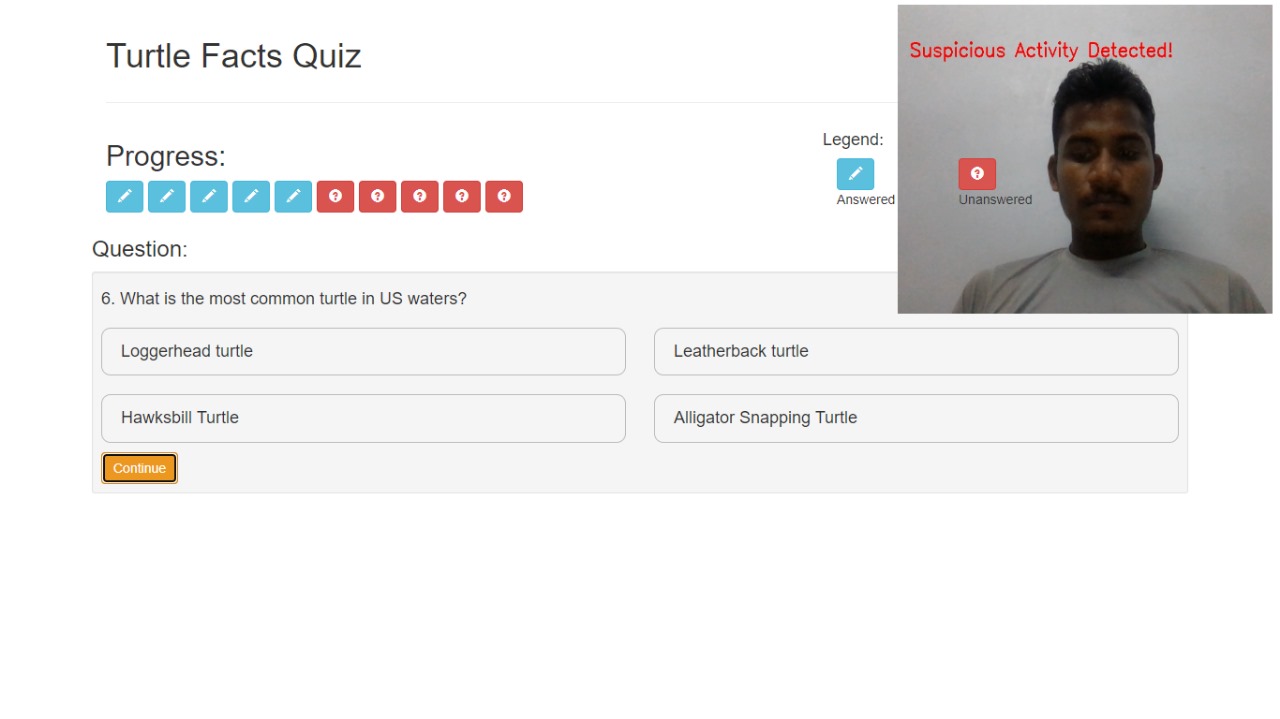


Fig 7.9 Suspicious Activity Detection

**Full Screen Exit Detection**

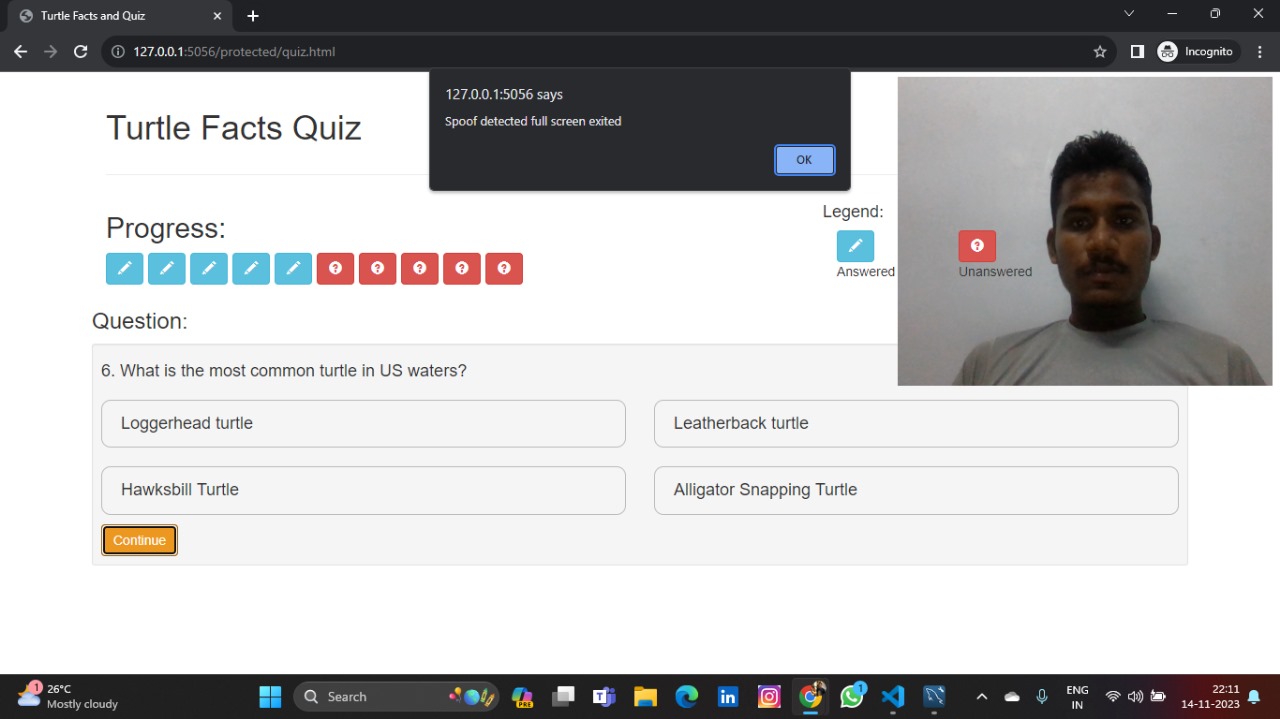


Fig 7.10 Full screen exit detection

**Detected new tab opening**

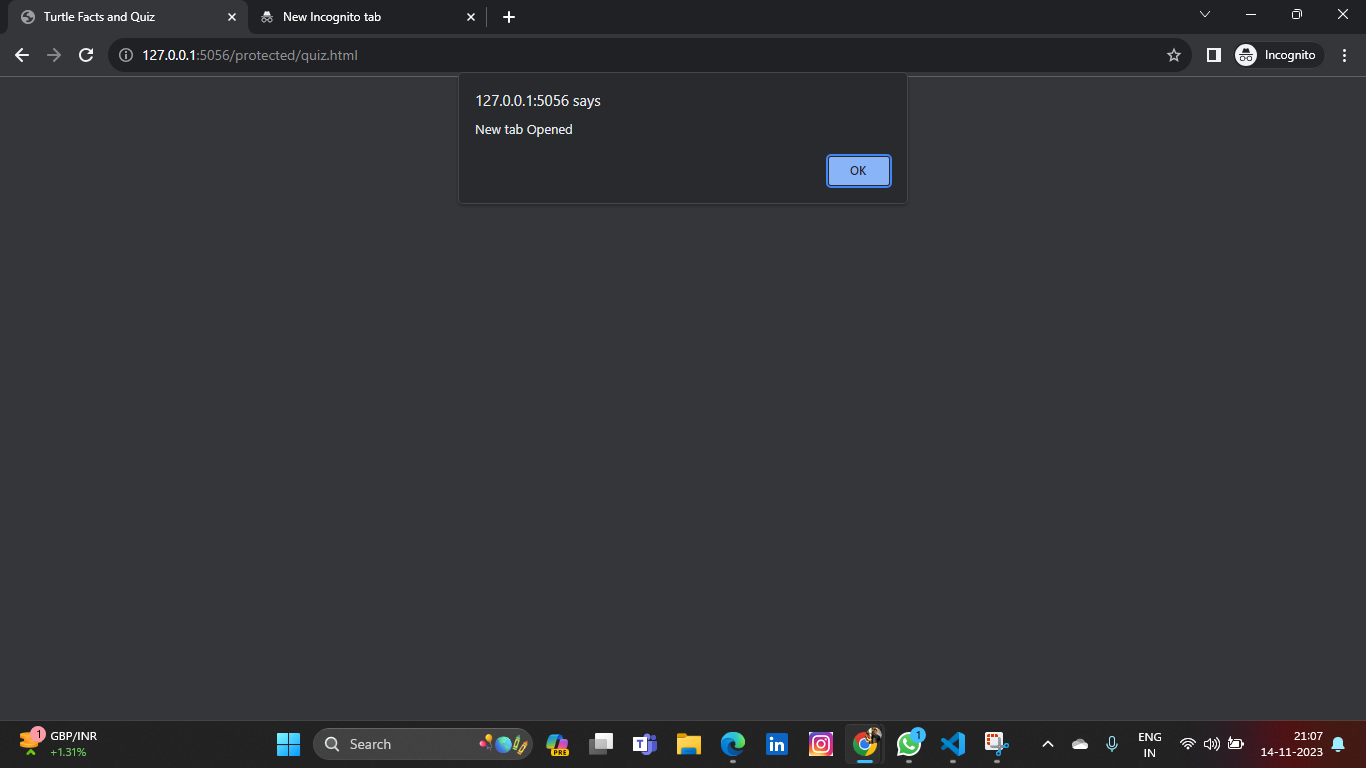
****

Fig 7.11 Detected new tab opening

**Submitting Quiz**

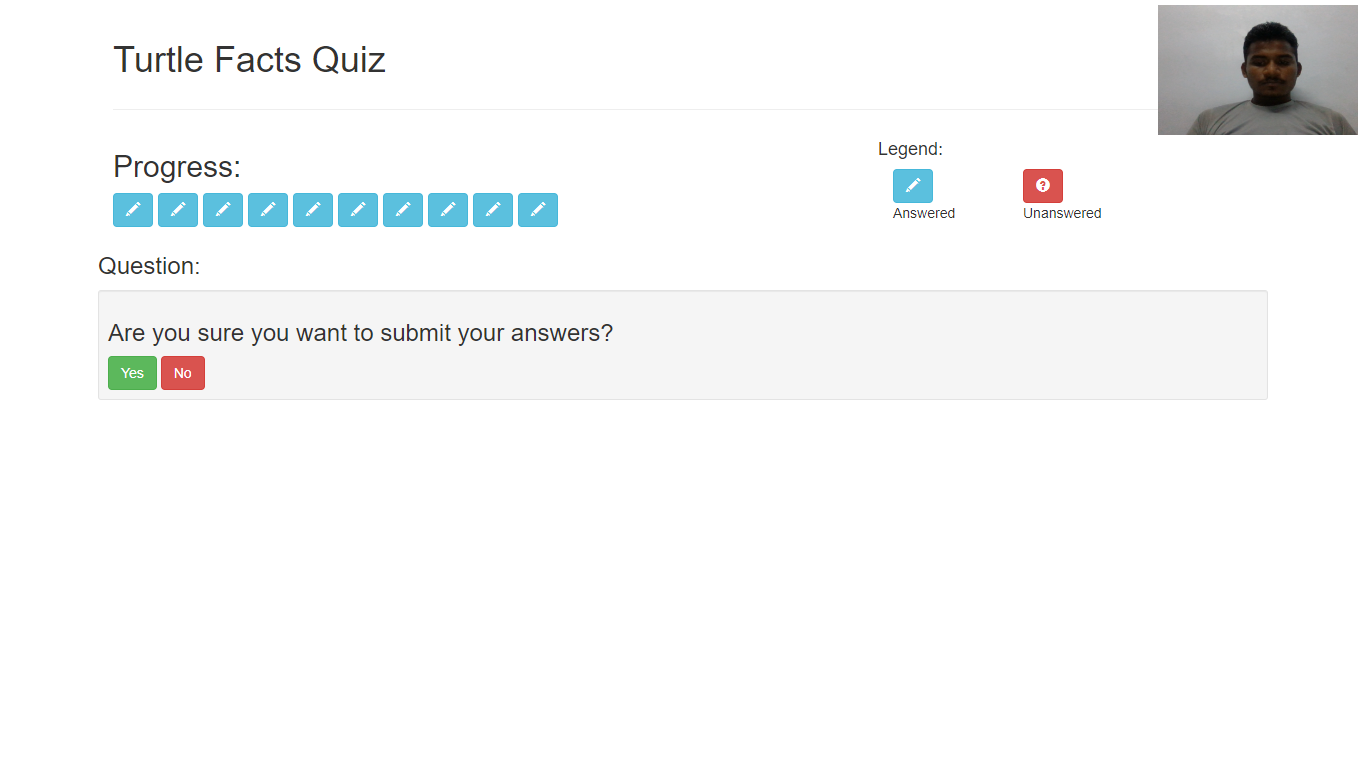
****

Fig 7.12 Submitting Quiz

**Results**

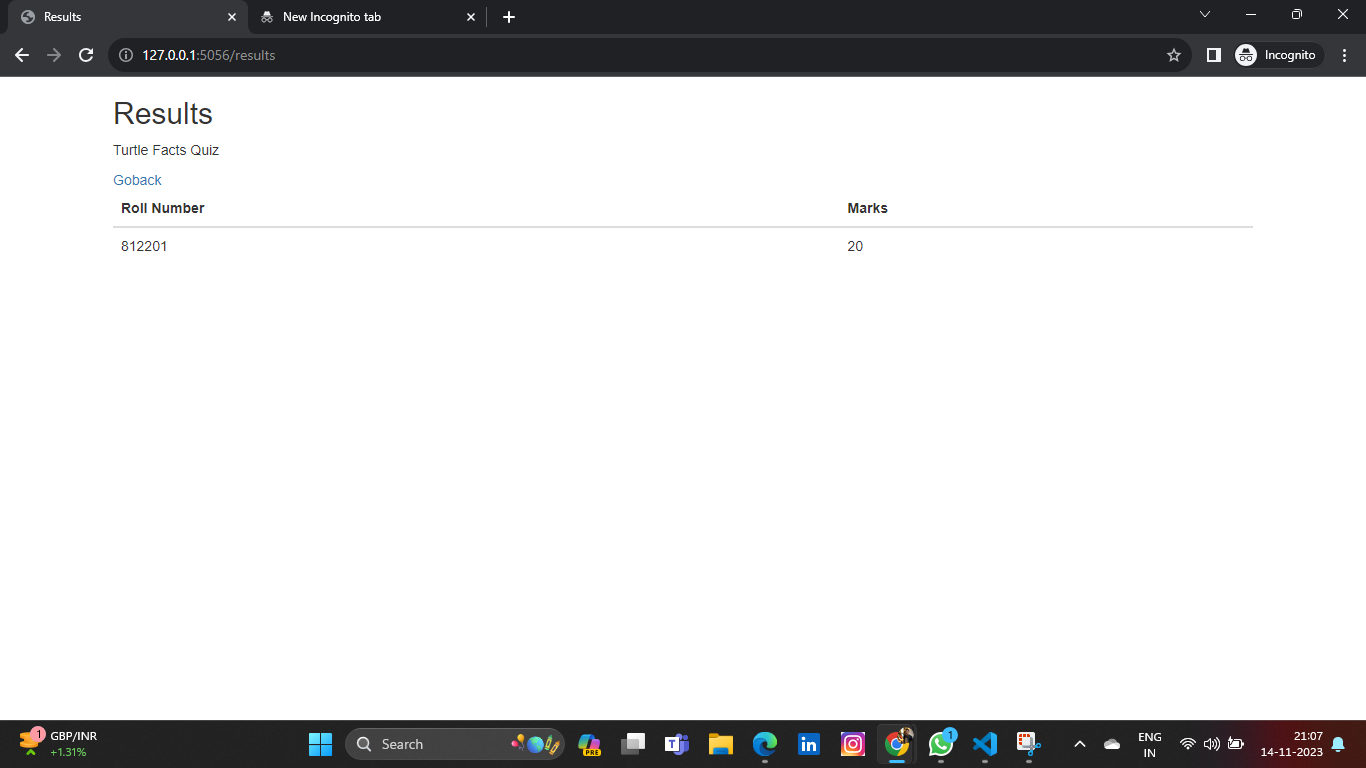
****

Fig 7.13 Results

**Invalid Password**

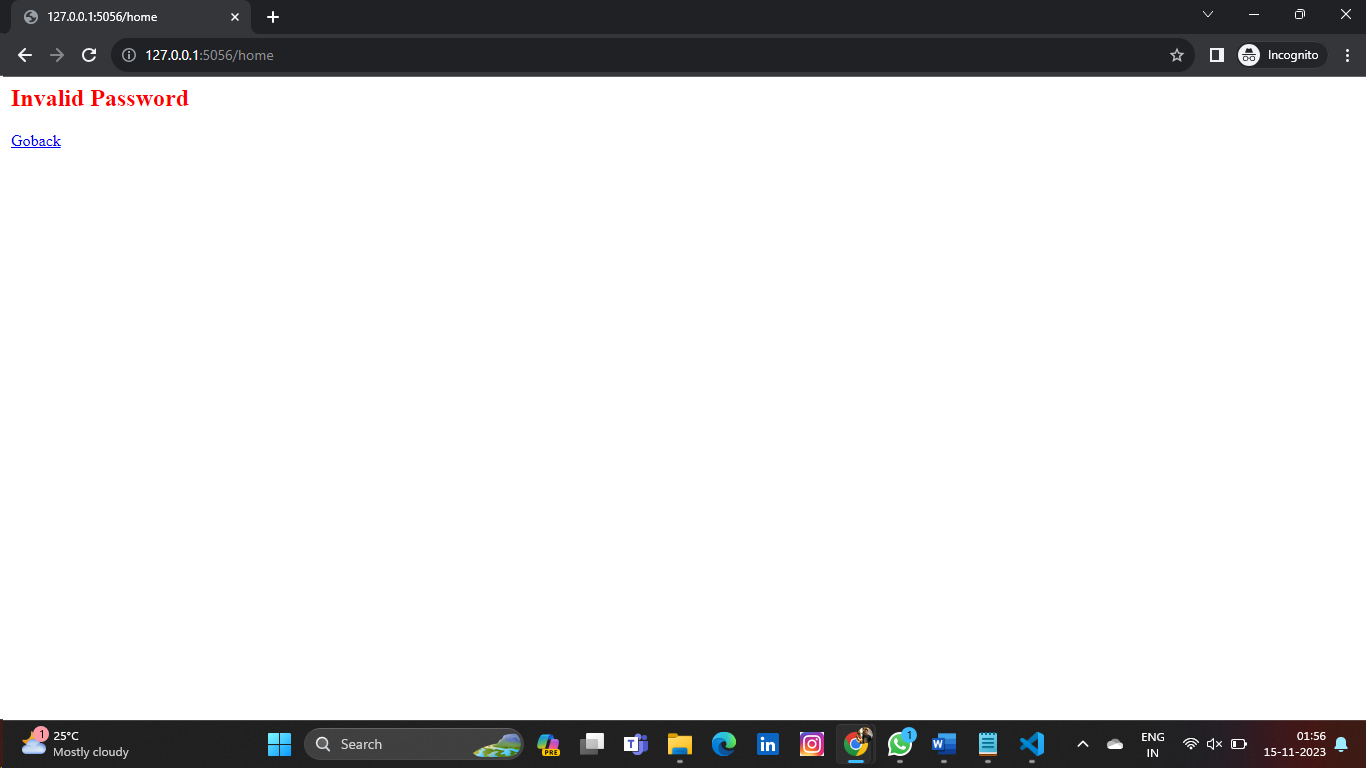


Fig 7.14 Invalid Password

**404 Not Found**

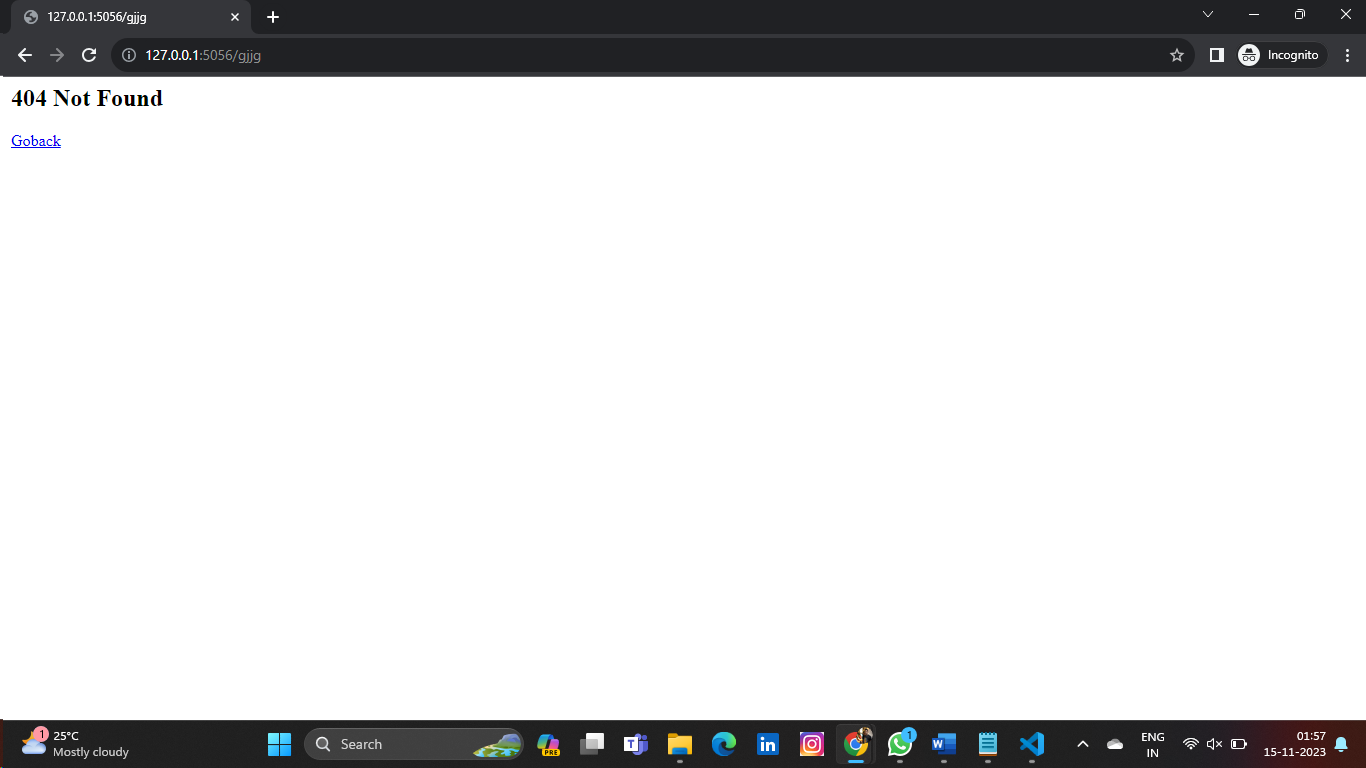


Fig 7.15 404 Not Found

**8. CONCLUSION AND FUTURE ENHANCEMENTS**

The AI-based exam proctoring system presents a transformative solution for enhancing the integrity and security of online examinations. By leveraging advanced artificial intelligence technologies, this system addresses traditional challenges associated with remote testing, ensuring a fair and reliable assessment environment. The implementation of facial recognition, gaze tracking, and other biometric features adds a layer of authentication that goes beyond conventional methods, significantly reducing the risk of academic dishonesty.

Furthermore, the system provides valuable insights into the test-taking process by monitoring behaviours. This not only fosters a secure examination environment but also contributes to the continuous improvement of assessment methodologies. While acknowledging concerns related to privacy, the design of the system prioritizes data protection and compliance with ethical standards.

In the evolving landscape of education, an AI-based exam proctoring system emerges as a vital tool for educational institutions and certification bodies seeking to embrace the flexibility of online assessments without compromising the credibility of results. As technology continues to advance, this system stands at the forefront, showcasing the potential of AI in ensuring the authenticity and reliability of examinations conducted in virtual environments**.**

**REFERENCES**

[1] Chaitanya Thombare, Kushank Sapate, Aniket Rane, Ankush Hutke, “Exam Proctoring System”, International Journal for Research in Applied Science & Engineering (IJRASET), Issue 10/2021.

[2] Kakarla S, Gangula P, Rahul M.S, Singh C.S.C, Sarma T.H, “Smart attendance management system based on face recognition using CNN”, IEEE-HYDCON, Issue 10/2020.

[3] Kohei Arai, Akifumi Yamashita, Hiroshi Okumura, “Head Position and Pose Model and Method for Head Pose Angle Estimation based on Convolution Neural Network”, International Journal of Advanced Computer Science and Applications (IJACSA), Issue 10/2021.

[4] John M, Barbara P, “Cheating on Unproctored Online Exam”, Online Learning and Academic Integrity Journal, Issue 03/2022.

[5] Aditya Nigam, Rhitvik Pasricha, Tarishi Singh, Prathamesh Churi, “A Systematic Review on AI-based Proctoring Systems: Past, Present and Future”, Education and Information Technologies, Issue 23 June 2021.

[6] Sahil Motwani,Chirag Nagpal,Manav Motwani, Nikhil Nagdev,Anjali Yeole, “AI-Based Proctoring System for Online Tests”, International Conference on Advances in Science & Technology, Issue 15 Jun 2021.