AI BASED EXAM PROCTORING SYSTEM

A Project Report submitted in partial fulfillment of requirements for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

V. PRANAY KUMAR REDDY
(209X1A05H9)

Under the esteemed guidance of
Smt. B. Swathi
Assistant Professor
Department of C.S.E.

Department of Computer Science and Engineering

G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL

(Affiliated to JNTUA, ANANTAPURAMU)

2023 - 2024

Department of Computer Science and Engineering

G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL

(Affiliated to JNTUA, ANANTAPURAMU)

CERTIFICATE

This is to certify that the Project Work entitled "AI BASED EXAM PROCTORING SYSTEM" is a bonafide record of work carried out by

V. PRANAY KUMAR REDDY
(209X1A05H9)

Under my guidance and supervision in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Smt. B. Swathi
Assistant Professor,
Department of C.S.E.,
G. Pulla Reddy Engineering College,
Kurnool.

Dr. N. Kasiviswanath
Professor & Head of the Department,
Department of C.S.E.,
G. Pulla Reddy Engineering College,
Kurnool.

DECLARATION

I hereby declare that the project titled **"AI BASED EXAM PROCTORING SYSTEM"** is an authentic work carried out by me as a student of **G. PULLA REDDY ENGINEERING COLLEGE(Autonomous) Kurnool**, during 2023-24 and has not been submitted elsewhere for the award of any degree or diploma in part or in full to any institute.

V. PRANAY KUMAR REDDY

(209X1A05H9)

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to our project guide **Smt. B. Swathi, Assistant Professor** of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for her immaculate guidance, constant encouragement and cooperation which have made possible to bring out this project work.

I am grateful to our project incharge **Sri. J. Swami Naik**, **Associate Professor** of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for helping us and giving us the required information needed for our project work.

I am thankful to our Head of the Department **Dr. N. Kasiviswanath**, for his whole hearted support and encouragement during the project sessions.

I am grateful to our respected Principal **Dr. B. Sreenivasa Reddy**, for providing requisite facilities and helping us in providing such a good environment.

I wish to convey my acknowledgements to all the staff members of the Computer Science and Engineering Department for giving the required information needed for our project work.

Finally, I wish to thank all our friends and well wishers who have helped me directly or indirectly during the course of this project work.

ABSTRACT

The "**AI-Based Exam Proctoring System**" is a cutting-edge project proposed for the Online Examination. This project aims to develop an intelligent and automated system that ensures exam integrity, tracks attendance, and detects fraudulent activities during online exams. The key features of the proposed system include:

- Face Detection for Participant Identification
- Real-Time Attendance Tracking to Excel Sheet
- Fraud Activity Detection
- Post-Exam Analysis

This project aims to create a comprehensive solution that combines facial recognition, attendance tracking, and fraud detection to address the challenges posed by remote online exams. Overall, this project serves as a robust solution to streamline the exam administration process, enhance exam security, and maintain academic integrity. By leveraging cutting-edge technologies, this project presents an innovative approach to conducting exams in a digital age. Its implementation will significantly benefit educational institutions.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

1.	RAM		Random Access Memory
2.	IDE		Integrated Development Environment
3.	UML	-	Unified Modeling Language
4.	DFD	-	Data Flow Diagram

1. INTRODUCTION

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. 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

- 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. 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** 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** 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** 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** 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.
- **AI Proctor** 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** 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.
- **Proctor track** 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.

3.4.6 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.

3.4.7 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.

3.4.8 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. DESIGN

4. 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.

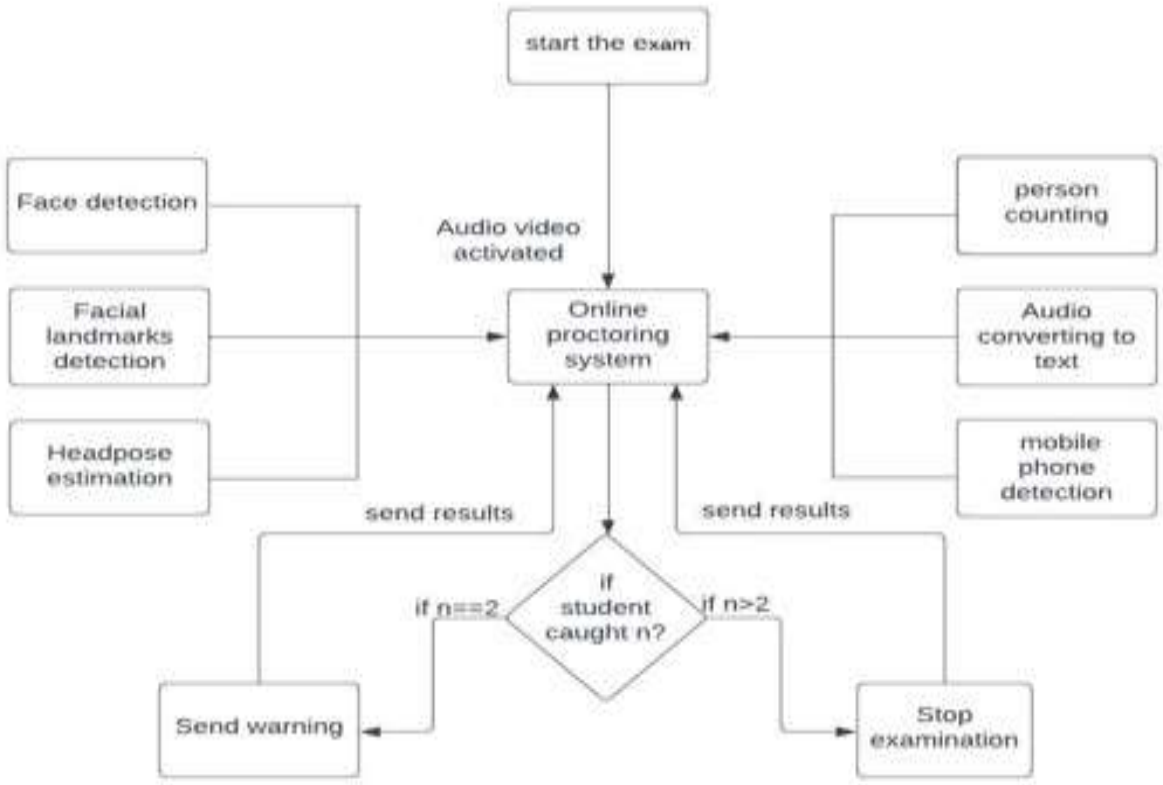

Fig 4.1 Data Flow Diagram

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.

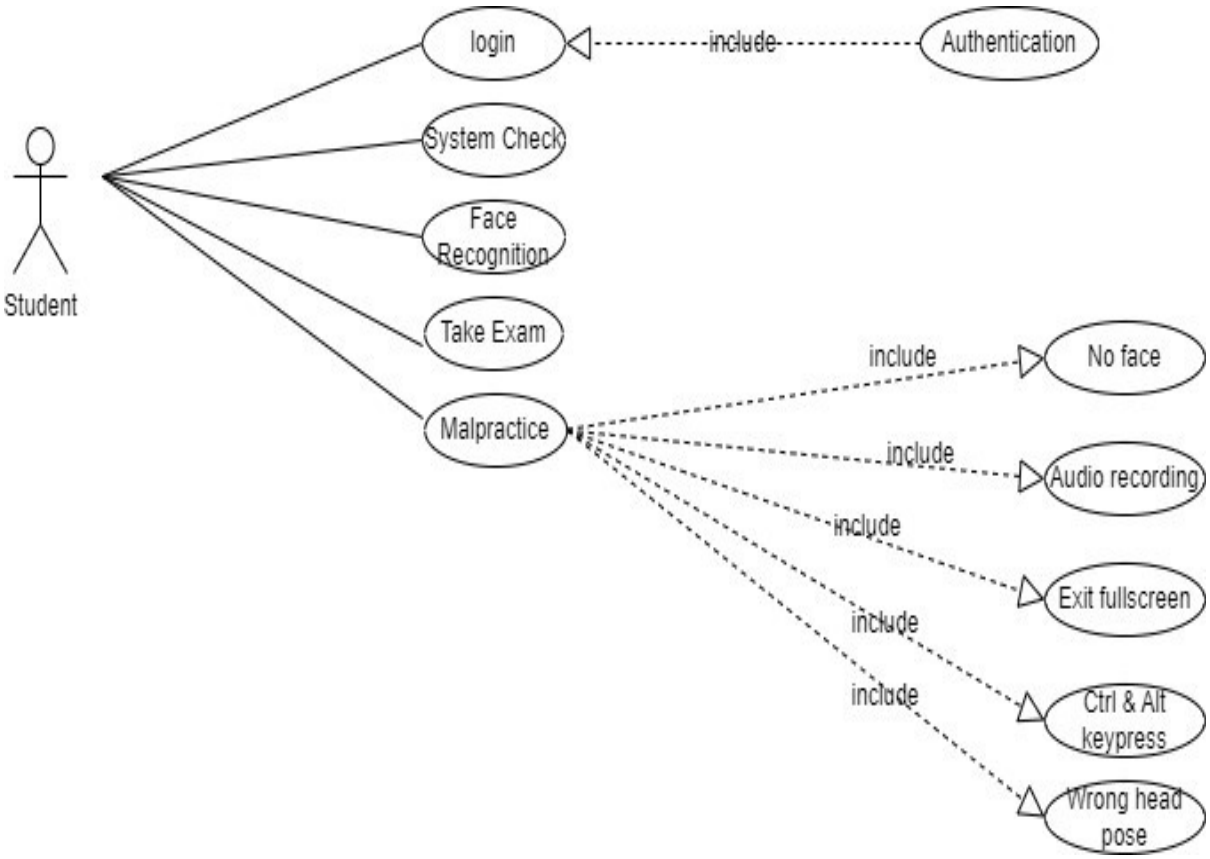

Fig 4.2 Use Case Diagram

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.

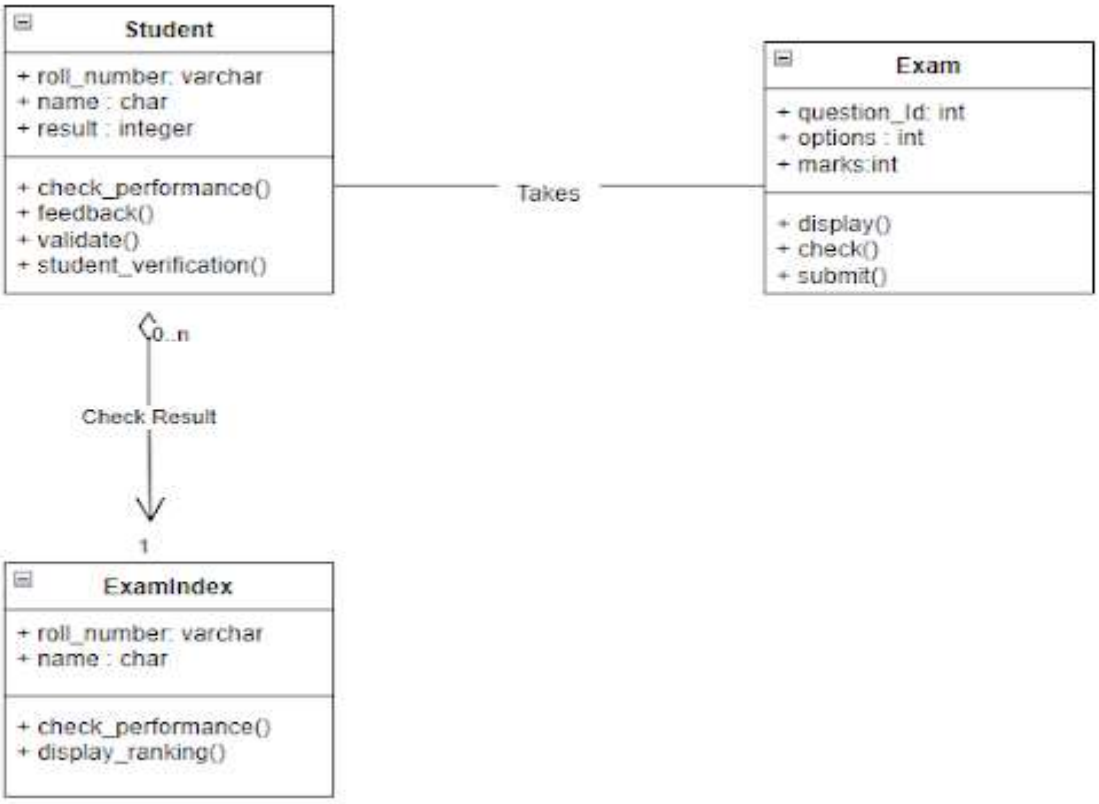

Fig 4.3 Class Diagram

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.

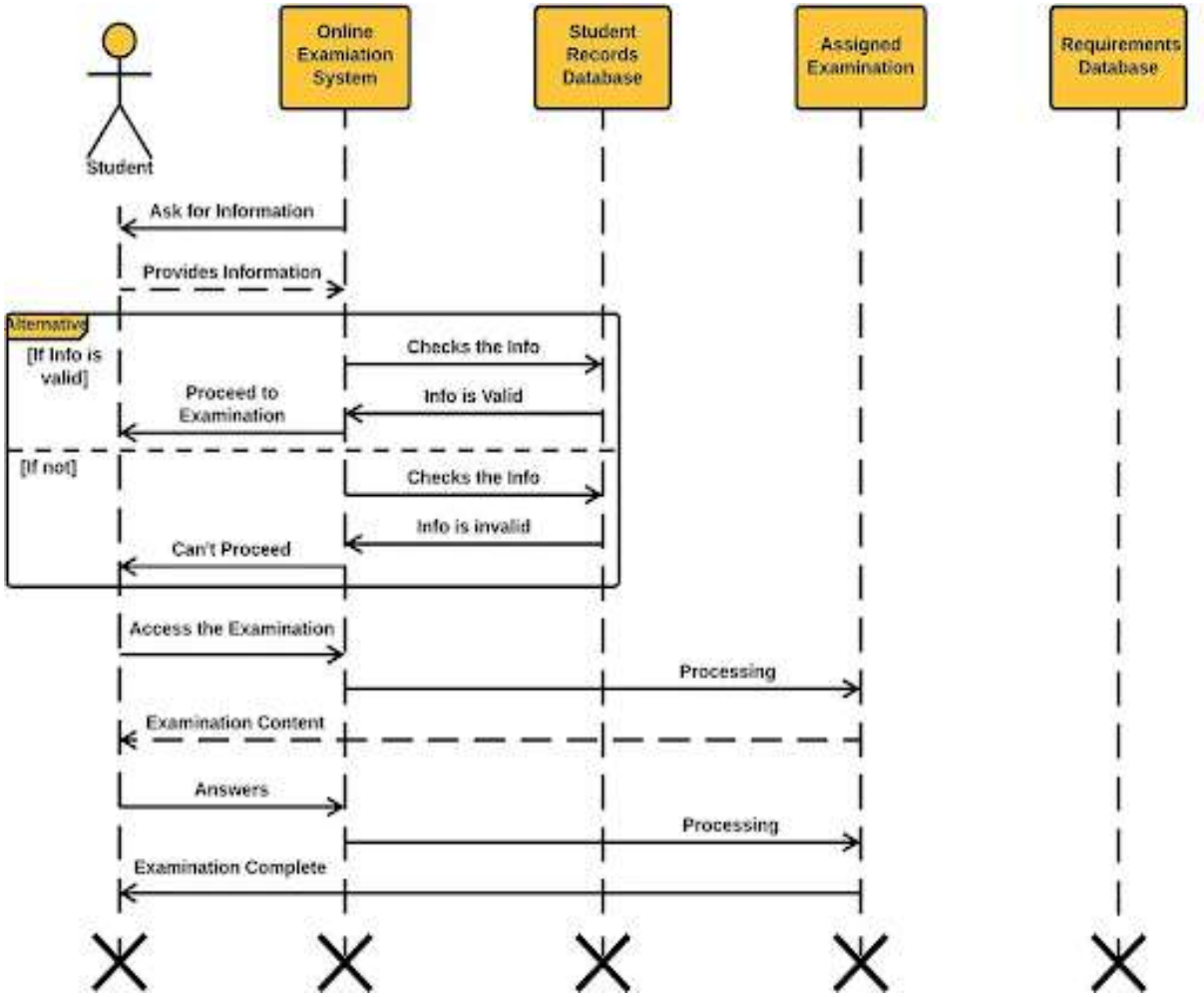

Fig 4.4 Sequence Diagram

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.

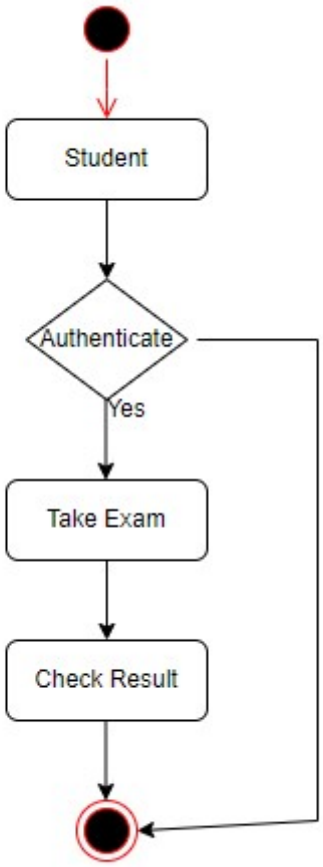

Fig 4.5 Activity Diagram

5. IMPLEMENTATION

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
```

```
|
+-- 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=''

fraud=""

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}')
def postAnalysis(name):
    with open('postanalysis.csv','r+') as f:
        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            global fraud
            f.writelines(f'\n RollNo - {name},{fraud}')
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))
global fraud
if len(faces) == 0:
    cv2.putText(frame, "Face Not Detected!", (20, 50),
    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    now = datetime.now()
    dtS = now.strftime('%H:%M:%S')
    fraud=fraud+"\nFace Not Detected! - "+dtS
    print(fraud)
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)
            now = datetime.now()
            dtS = now.strftime('%H:%M:%S')
            fraud=fraud+"\nSuspicious Activity Detected! - "+dtS
            print(fraud)
    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=[]
        cursor = connection.cursor()
        sql="select * from `stu_data` where rollno="+str(session['rollno'])+";"
        cursor.execute(sql)
        for i in cursor:
            data.append(list(i))
        cursor.close()
        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=[]
```

```
        cursor = connection.cursor()
        sql="select * from `stu_details` where rollno="+str(roll)+";"
        cursor.execute(sql)
        for i in cursor:
            data.append(list(i))
        cursor.close()
        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):
    if filename=='quiz.html':
        datac=[]
        cursor = connection.cursor()
        sql="select * from `quiz_results` where rollno="+str(session['rollno'])+";"
        cursor.execute(sql)
        for i in cursor:
            datac.append(list(i))
        cursor.close()
        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;'>We couldn't find a match for your information in our
            database.</h2> <a href='/dashboard'>Goback</a>"
    else :
        return send_from_directory('protected', filename)
@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'])
    postAnalysis(session.get('rollno'))
    try:
        cursor = connection.cursor()
        sql="'insert into `quiz_results`(rollno,marks)values(%s,%s)'"
        v=[str(session.get('rollno')),str(data['percentage'])]
        cursor.execute(sql,v)
        connection.commit()
        cursor.close()
        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=[]
    cursor = connection.cursor()
    sql="select * from `quiz_results`;"
    cursor.execute(sql)
    for i in cursor:
        data.append(list(i))
```

```
    cursor.close()
    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.route('/stu-data')
def stu_details():
    data=[]
    cursor = connection.cursor()
    sql="select * from `stu_data`;"
    cursor.execute(sql)
    for i in cursor:
        data.append(list(i))
    cursor.close()
    return render_template('student_data.html',data=data)
@app.route("/ImagesAttendance/<path:filename>")
def profilephoto(filename):
    return send_from_directory('ImagesAttendance', filename)
@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          : 
        <strong>{{data[0][0]}} </strong></p>
        <p>Student Name       :  <strong>{{data[0][1]}}
        </strong></p>
        <p>Father Name       :
          <strong>{{data[0][2]}}</strong></p>
        <p>Mother Name      :  <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>
```

5.7 TESTING AND VALIDATION

5.7.1 Verification of Facial Recognition and Identification Accuracy

Our primary goal was to verify the system's accuracy in recognizing and identifying test-takers through facial recognition technology. By subjecting the system to rigorous testing scenarios, we ensured that it reliably identified individuals with precision and consistency.

5.7.2 Assessment of Behaviour Monitoring Algorithms Effectiveness

We placed significant emphasis on evaluating the effectiveness of behavior monitoring algorithms in detecting suspicious activities such as eye movement, head orientation, and keyboard activity. Through comprehensive testing, we validated that the system effectively flagged and responded to irregular behaviors indicative of potential misconduct.

5.7.3 Evaluation of System Performance under Various Network Conditions

Understanding the importance of system performance under different network conditions, we conducted thorough testing to assess the system's resilience and reliability. By simulating various network environments, including high-speed, low-speed, and intermittent connectivity, we ensured that the system functioned optimally across diverse networking scenarios.

5.7.4 Ensuring Compatibility with Different Devices and Browsers

Compatibility across devices and browsers is paramount for seamless user experience. To guarantee compatibility, we rigorously tested the system on a multitude of devices (PCs, laptops, tablets, smartphones) and web browsers (Chrome, Firefox, Safari, etc.). This ensured that users could access the system effortlessly regardless of their device or browser preference.

5.7.5 Validation of Cheating Detection Mechanisms Reliability

The reliability of cheating detection mechanisms is crucial for maintaining exam integrity. Through comprehensive testing, we validated the effectiveness of anti-cheating measures such as screen monitoring, browser lockdown, and plagiarism detection. This ensured that the system robustly identified and prevented any attempts at academic dishonesty.

5.8 TESTING METHODOLOGIES

5.8.1 Functional Testing

- **Facial Recognition:** Our testing verified the system's ability to accurately recognize and identify test-takers using facial recognition technology.
- **Behavior Monitoring:** We assessed the effectiveness of behavior monitoring algorithms in detecting suspicious activities.
- **Browser Compatibility:** The system was extensively tested across different web browsers to ensure seamless operation.
- **Device Compatibility:** Performance on various devices was evaluated to ensure compatibility and optimal user experience.
- **Network Performance:** Testing under different network conditions ensured the system's reliability and resilience.

5.8.2 Security Testing

- **Data Encryption:** We verified data encryption protocols to prevent unauthorized access to transmitted data.
- **Access Control:** Testing access control mechanisms ensured only authorized individuals had access to exam data and monitoring features.
- **Data Privacy:** Compliance with data privacy regulations was ensured through secure handling and storage of sensitive information.
- **Anti-Cheating Measures:** The effectiveness of anti-cheating measures such as screen monitoring and plagiarism detection was rigorously evaluated.

5.8.3 Performance Testing

- **Load Testing:** The system's performance under heavy loads was assessed to ensure scalability and stability.
- **Response Time:** We measured the system's response time for various actions to optimize user experience.
- **Stress Testing:** Extreme conditions were simulated to identify potential vulnerabilities and ensure system robustness.

5.9 TESTING PRINCIPLES

Throughout the testing phase, we meticulously adhered to the fundamental principle that all tests must be traceable to customer requirements. This ensured that every aspect of the system's functionality was thoroughly evaluated against specified criteria, guaranteeing alignment with user expectations.

5.9.1 System Testing

In our testing endeavors, system testing served as a pivotal stage where the entire integrated software system was rigorously evaluated to ensure adherence to requirements. By conducting configuration-oriented system integration tests, we verified that the system consistently delivered known and predictable results, validating its reliability under diverse scenarios.

5.9.2 White Box and Black Box Testing

Employing both white box and black box testing methodologies, we scrutinized the system's inner workings and functionality. White box testing enabled us to delve deep into the system's architecture, ensuring that internal processes functioned as intended. Meanwhile, black box testing focused on functional requirements, guaranteeing that the system's external behavior aligned with specified criteria.

5.9.3 Acceptance Testing

User Acceptance Testing, a critical phase of the project, was conducted with significant participation from end-users. This phase ensured that the system met functional requirements and garnered user approval.

In conclusion, the completion of thorough testing activities has fortified the AI-based Exam Proctoring System reliability, functionality, and alignment with customer requirements.

SCREENSHOTS

Login Page

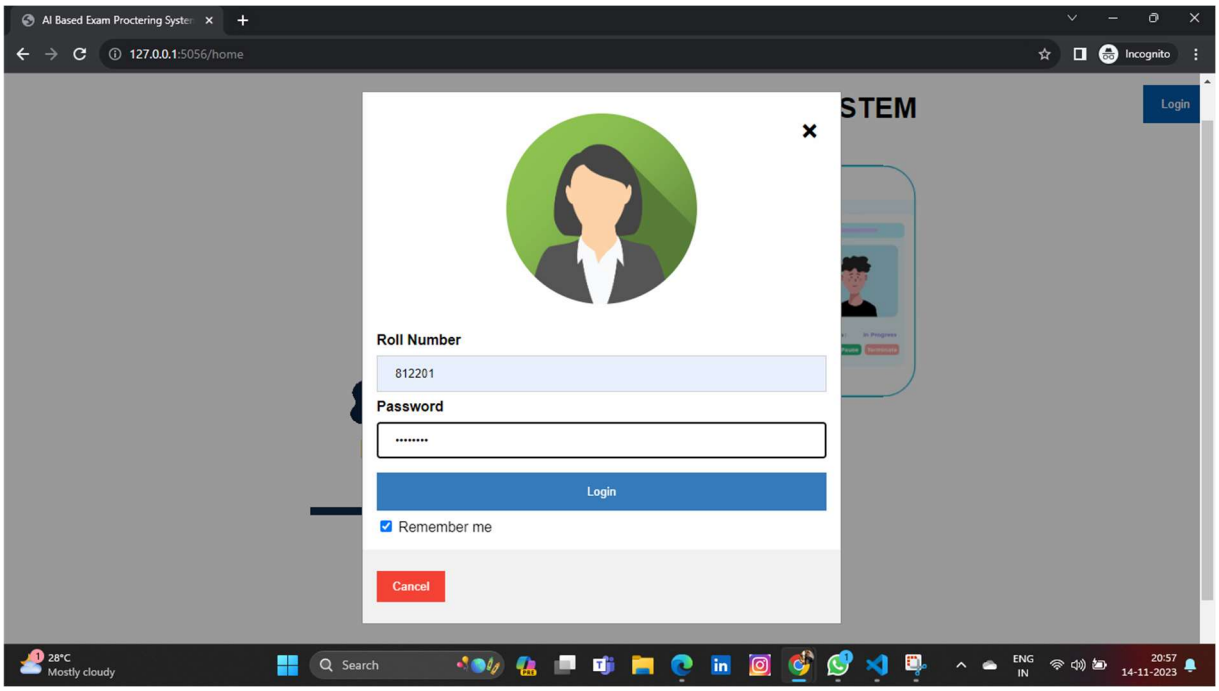

Fig 5.1 Login Page

User Authentication

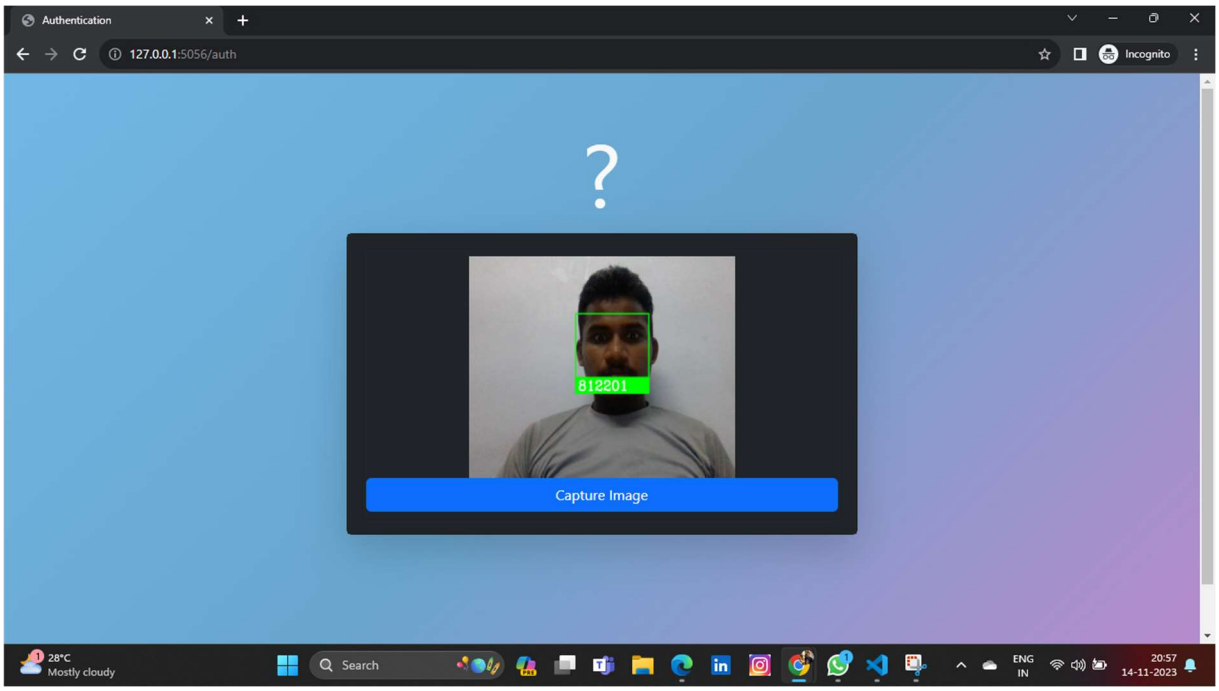

Fig 5.2 User Authentication

Face Not Detected During Exam

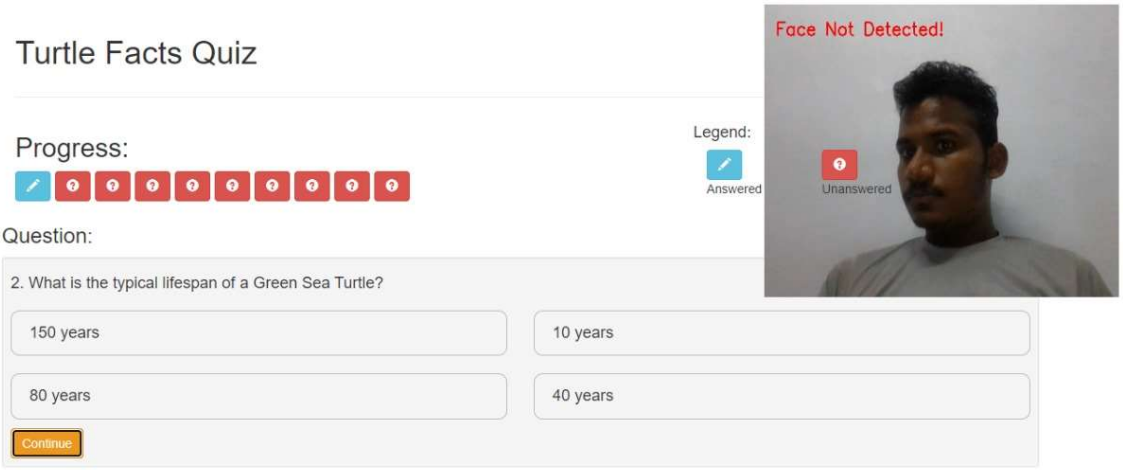

Fig 5.3 Face Not Detected During Exam

Full Screen Exit Detection

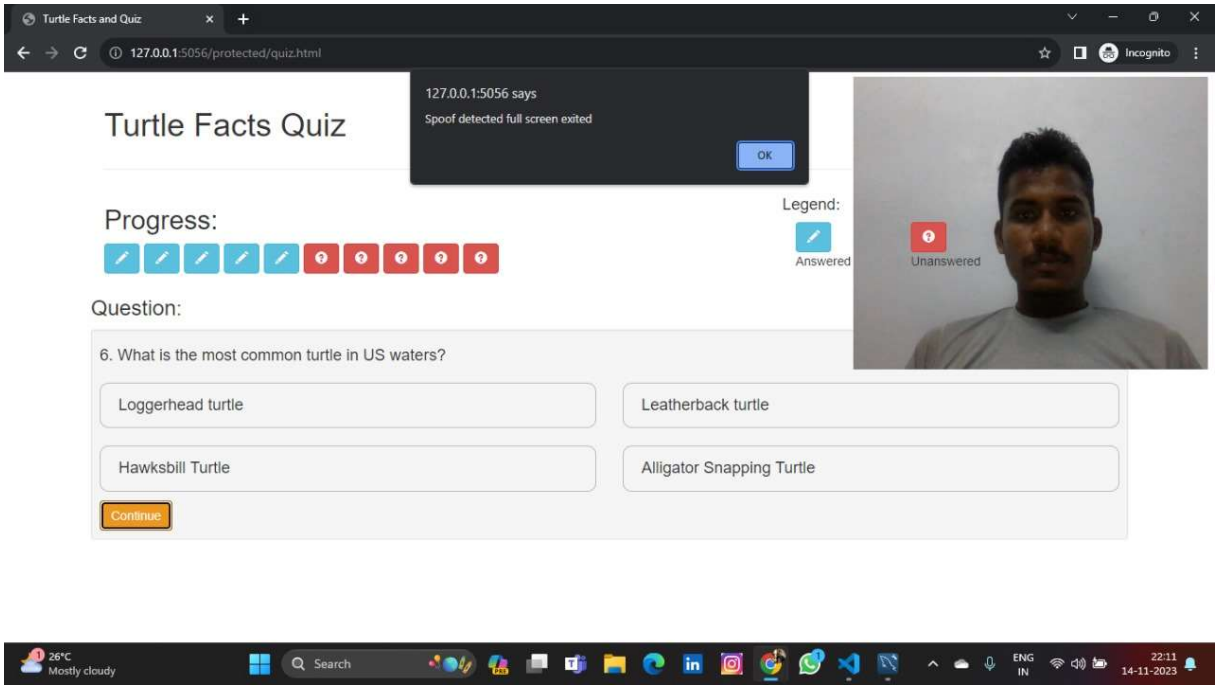

Fig 5.4 Full screen exit detection

6. CONCLUSION AND FUTURE ENHANCEMENTS

6. 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.

[7] Vidhya SG, Hema GA, Jeevitha MG, Nischitha KB, Vandana, "Ai-Based Proctoring System For Online Tests", International Research Journal of Modernization in Engineering Technology and Science, Issue July 2022.

[8] Anjali Rajendra Y, Reena Vishwas P, Diksha Anil K, Pallavi Pravin R, "Online Exam Proctoring System", International Journal of Creative Research Thoughts (IJCRT), Issue 1 January 2022.

[9] Al-Hakeem M.S., Abdulrahman M.S. Developing a new e-exam platform to enhance the university academic examinations, International Journal of Modern Education and Computer Science, Issue 2017.