**ACKNOWLEDGEMENT**

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

We are grateful to our project incharge Dr./Sri./Smt. **PROJECT INCHARGE NAME**, **Designation** 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.

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

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

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

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

**DECLARATION**

I hereby declare that the project titled “ **AI Based Exam Proctoring System** ” is an authentic work carried out by me as the 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.

**Student Name**

**(Student Roll Number)**

**Introduction**

Remote proctoring is the process of authenticating, authorizing and controlling the online examinationprocess 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 center 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 recordevery single examination session from beginning to end, not just via video, but also captures desktopscreens, chat logs and images

**Scope**

AI Proctoring system is designed for educational Institutions (like schools,colleges,universities,training 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.

1.4 Objectives

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 analyze various behavioral 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.

2.2 Software and Hardware Requirements

Software and Hardware Requirements are used to describe the minimum hardware and software requirements to run the Software. These requirements are described below.

Software Requirements :

Open CV Library:-

OpenCV are also included in the making of this program. OpenCV (Open Source Computer Vision) is a library of programming functions for real time computer vision.OpenCV have the utility that can read image pixels value, it also have the ability to create real time eye tracking and blink detection.

Python:

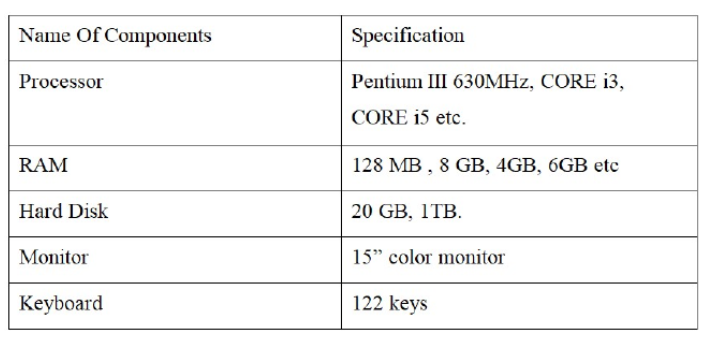
Python is an interpreted high-level general-purpose programming language.

Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object- oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming.

2.2.1 Hardware Requirements

Computer Desktop or Laptop:- 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



Webcam:-

Webcam is utilized for image processing, the webcam will continuously taking image in order for the program to process the image and find pixel position

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.

False Positives and Negatives:

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

Internet Dependence:

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

#### Limitations of Current Solutions

A critical examination of current online proctoring solutions reveals certain limitations, including accuracy issues in participant identification, challenges in real-time monitoring, and difficulties in detecting advanced fraudulent activities. Recognizing these limitations underscores the need for a more comprehensive and innovative approach, motivating the development of our proposed system.

In summary, this introduction to the Literature Survey aims to contextualize our project within the broader landscape of online exam monitoring and attendance tracking, outlining the key challenges addressed and the existing knowledge upon which our work builds. The subsequent sections will delve into specific studies, methodologies, and technologies identified in the literature survey, providing a foundation for the development and implementation of our AI-Based Exam Monitoring System.

—------------------------------------------------

The Literature Survey is a critical component of this report, aiming to explore and analyze existing research, technologies, and solutions in the realm of online exam monitoring, attendance tracking, and fraud detection. This section sets the stage for the reader by outlining the context within which our project has been developed.

3.1.1 Online Examination Challenges

The transition to online examinations has brought about a paradigm shift in educational assessment, accompanied by a myriad of challenges. Traditional methods of in-person invigilation and attendance tracking are no longer viable in the digital landscape. Recognizing these challenges is fundamental to understanding the necessity and relevance of our proposed solution.

3.1.2 Facial Recognition Technologies

Facial recognition has emerged as a key technology in the field of participant identification. A review of the literature reveals various methodologies and algorithms employed in facial recognition systems for exam settings. Notable advancements in this area contribute to our understanding of the capabilities and limitations of facial recognition technologies.

3.1.3 Real-Time Attendance Tracking

Real-time attendance tracking is a crucial aspect of online exam monitoring. Existing literature discusses different approaches to monitoring participant attendance during online exams, including the use of biometric data and automated systems. This section explores the effectiveness and challenges associated with real-time attendance tracking solutions.

3.1.4 Fraud Detection Techniques

Fraud detection in online exams demands sophisticated techniques to identify and prevent various forms of academic dishonesty. The literature survey delves into existing methodologies for fraud detection, encompassing behavioral analysis, anomaly detection, and machine learning algorithms. Understanding the state-of-the-art in fraud detection informs our project's approach to mitigating such challenges.

Disadvantages of Existing Systems

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.

—----------------------------------------------

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 behavior, 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.

Design and 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 (PyAudio) technologies. The source code for the core functionalities is provided below:

#### 4.1 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)

#### 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)

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.

#### 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:

##### **4.5.1 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.

##### **4.5.2 Real-Time Notifications**

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

##### **4.5.3 Multi-Platform Compatibility**

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

##### **4.5.4 User Authentication and Authorization**

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

##### **4.5.5 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.

##### **4.5.6 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.

##### **4.5.7 Data Analytics and Reporting**

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

##### **4.5.8 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.

Source code:

from flask import Flask, render\_template, Response,request

import cv2

import pyaudio

import json

import audioop

import pymysql.cursors

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

user='root',

password='@g209X1A05H6g@',

database = 'students'

)

cursor = connection.cursor()

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)

def generate\_frames():

# Initialize video capture

# phone\_cascade= cv2.CascadeClassifier('C:\\Users\\sai\\OneDrive\\Desktop\\FYP\\haarcascade\_phone.xml') # Replace with the correct path to your XML file

max\_blink\_count = 5 # Maximum allowed blinks

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 = 50

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]

# Use Haar Cascade to detect eyes

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

# Check for eye state (open or closed)

eye\_state = True # Default to open

contours, \_ = cv2.findContours(eyes, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

for contour in contours:

area = cv2.contourArea(contour)

if area < 1000: # Threshold for closed eyes

eye\_state = False # Eyes are closed

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

prev\_eye\_state = eye\_state

if not eye\_state:

blink\_counter += 1

# Check for tab switching (use a more sophisticated method for actual implementation)

key = cv2.waitKey(1)

if key == 9: # Tab key

tab\_switch\_detected = True

# Check for excessive noise

if rms > 1000: # Adjust the noise threshold as needed

noise\_detected = True

noise\_reset\_counter = 0

else:

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

noise\_reset\_counter += 1

else:

noise\_detected = False

# Display warning if suspicious activity is detected

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 render\_template('quiz1.html')

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

def video\_feed():

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

@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")

connection.commit()

@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)

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

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

var count = 0;

var elements = ['aa', 'bb', 'cc', 'dd', 'ee', 'ff', 'gg', 'hh', 'ii', 'jj', 'ww', 'A12', 'B12', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'AA', 'BB', 'CC', 'DD', 'EE', 'FF', 'GG', 'HH', 'II', 'JJ', 'KK', 'LL', 'inp', 'cn', 'name'];

var current = 0;

function setColor(id, color) {

var element = document.getElementById(id);

element.style.backgroundColor = color;

}

function clearAndShowNext(nextId, currentId, displayFunction) {

setColor(currentId, 'black');

setTimeout(() => {

setColor(currentId, 'white');

setTimeout(displayFunction, 700);

}, 200);

}

function displayAndIncrement(nextIndex, displayFunction) {

clearAndShowNext(elements[nextIndex], elements[current], displayFunction);

current = nextIndex;

}

function one() {

displayAndIncrement(1, one);

}

function two() {

displayAndIncrement(2, two);

}

function three() {

displayAndIncrement(3, three);

}

function four() {

displayAndIncrement(4, four);

}

function five() {

displayAndIncrement(5, five);

}

function six() {

displayAndIncrement(6, six);

}

function seven() {

displayAndIncrement(7, seven);

}

function eight() {

displayAndIncrement(8, eight);

}

function nine() {

displayAndIncrement(9, nine);

}

function ten() {

displayAndIncrement(10, ten);

}

function startGame() {

document.getElementById('strt').style.display = 'block';

document.getElementById('roll').style.display = 'block';

}

function checkName() {

var bn = document.getElementById('sid');

if (bn.value.length > 0) {

startGame();

}

}