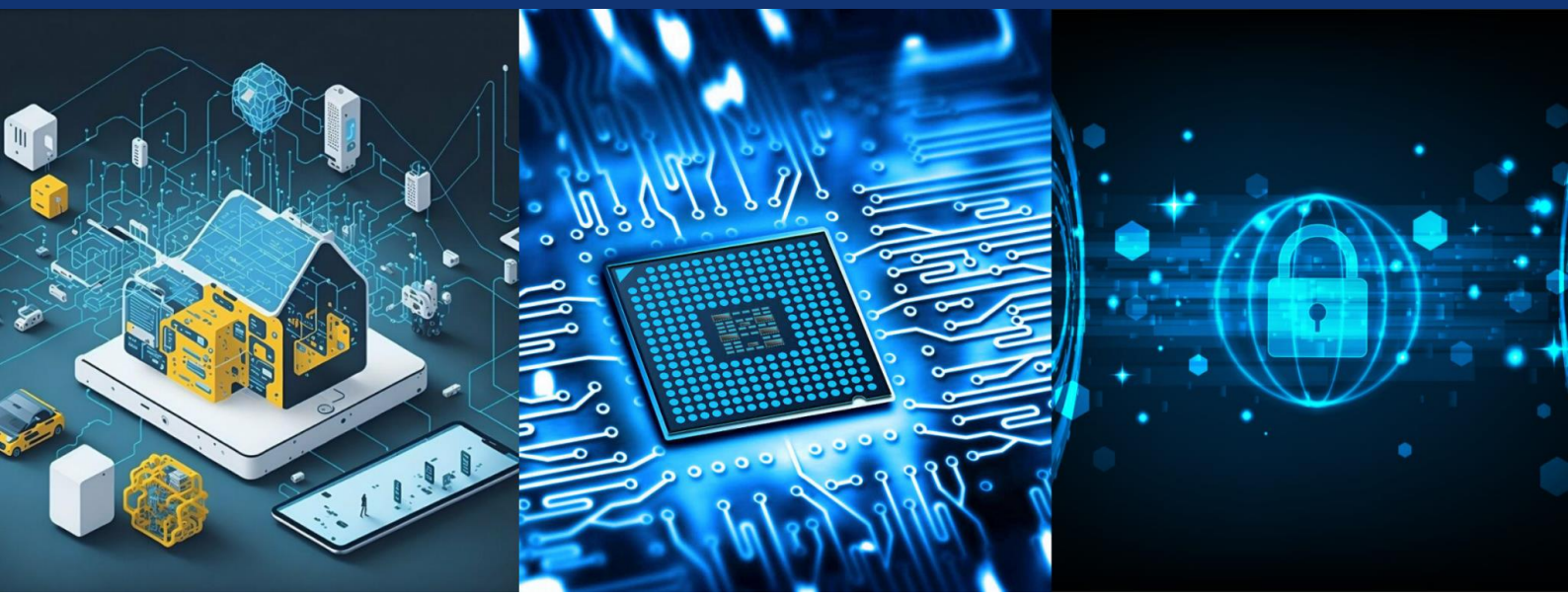
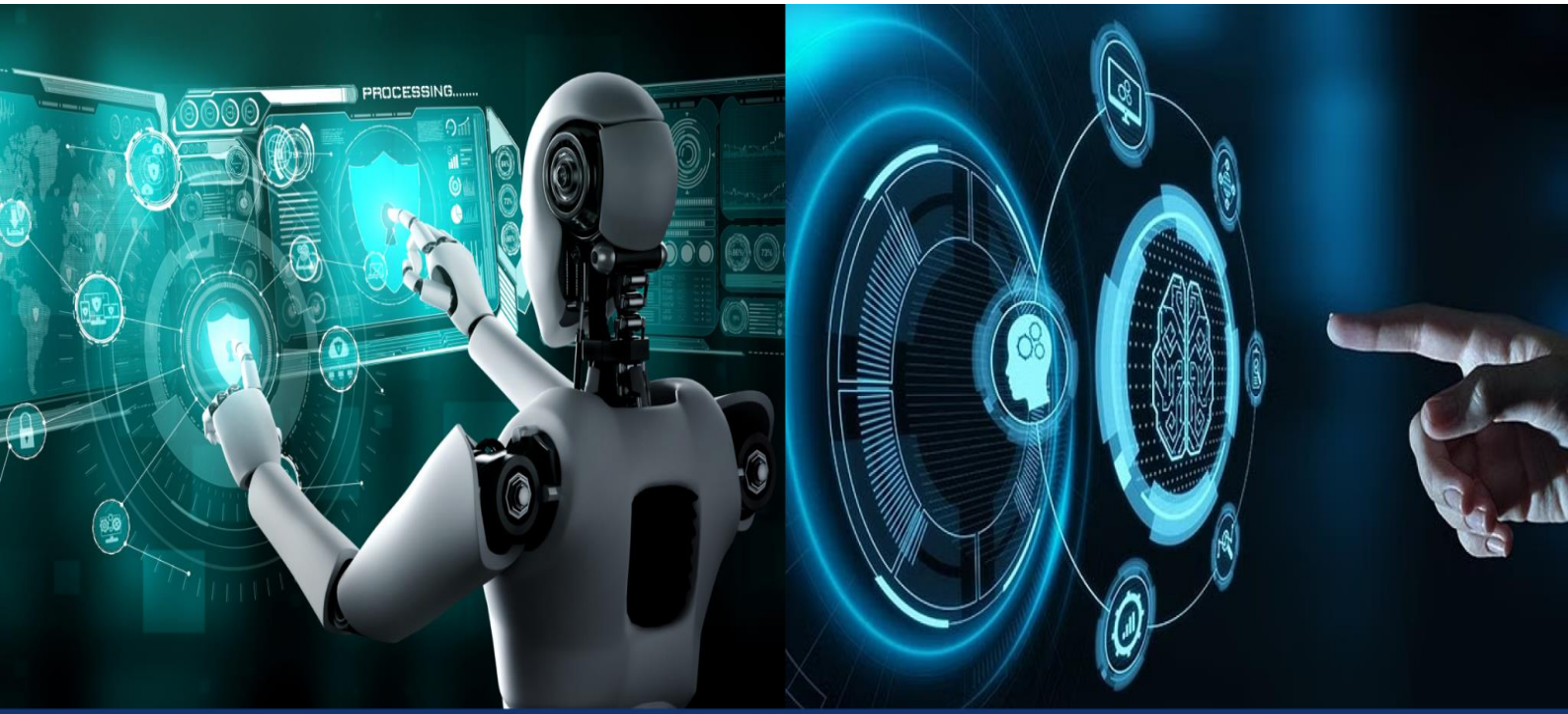
ISSN(O): 2320-9801
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771 **Volume 13, Issue 4, April 2025**

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304201

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Anti Cheat Exam Application with Artificial Intelligence

Mr.A. Srinivasa Rao, B. Teja Sai Kumar, M. Dileep, V. Lakshman, Sk.Lilatun Reenu, V.S.S Chandana

Assistant Professor, Department of CSE (AIML), Tirumala Engineering College, NRT, Andhra Pradesh, India

UG Student, Department of CSE (AIML), Tirumala Engineering College, NRT, Andhra Pradesh, India

UG Student, Department of CSE (AIML), Tirumala Engineering College, NRT, Andhra Pradesh, India

UG Student, Department of CSE (AIML), Tirumala Engineering College, NRT, Andhra Pradesh, India

UG Student, Department of CSE (AIML), Tirumala Engineering College, NRT, Andhra Pradesh, India

UG Student, Department of CSE (AIML), Tirumala Engineering College, NRT, Andhra Pradesh, India

ABSTRACT: This project focuses on the development of an AI-powered Anti-Cheat Exam Application designed to ensure fair online examinations by leveraging machine learning and security techniques. The system detects and prevents cheating by monitoring user interactions, restricting unauthorized activities, and analyzing behaviour patterns during an exam session.

The application employs AI-driven behavioural tracking to monitor user activity, prevent app switching, block screen sharing, and detect abnormal patterns such as prolonged inactivity or multiple attempts to exit the exam interface. Additionally, Natural Language Processing (NLP) techniques help analyze audio interactions to detect external assistance. The system ensures secure authentication mechanisms to verify the correct candidate is taking the exam.
This project demonstrates significant applications in online education, recruitment tests, and certification exams, ensuring integrity and credibility in digital assessments.

KEYWORD: AI Proctoring, Online Exam Security, Cheating Detection, Behavioral Tracking, Machine Learning, Natural Language Processing (NLP), Screen Monitoring, User Authentication, App Switching Prevention, Exam Integrity, Human Activity Recognition, Real-Time Monitoring

I. INTRODUCTION

The Anti-Cheat Exam Application is a comprehensive AI-powered platform developed to ensure fairness, transparency, and security in online examinations. The shift to remote education, accelerated by the COVID-19 pandemic, has transformed the educational landscape globally. Institutions have rapidly adopted online learning and evaluation methods, but the integrity of assessments has remained a critical challenge due to the lack of reliable monitoring systems.
Traditional online exam systems are prone to vulnerabilities such as:

- Proxy attendance or impersonation
- Screen sharing with collaborators
- Use of unauthorized resources
- Verbal guidance via audio communication
- Frequent switching between tabs or applications

These loopholes compromise the credibility of online assessments and raise concerns over the authenticity of results.To combat these issues, our application integrates a multi-layered approach combining Artificial Intelligence (AI), Machine Learning (ML), Facial Recognition, and Computer Vision to actively monitor and analyze a candidate's behavior in real time. This ensures that only genuine and fair attempts are recognized.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304201

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. RELATED WORK

In [1], the authors developed a browser-based online proctoring system that relied on face recognition and screen monitoring to detect cheating. Although effective in detecting face absence or screen switching, the system depended on third-party APIs and lacked modular architecture. Our approach improves upon this by using ReactJS and Node.js for a custom, scalable full-stack implementation .In [2], researchers proposed a machine learning-based exam monitoring solution using SVM classifiers for behavior analysis. They utilized head pose and gaze direction estimates to flag potential cheating, demonstrating how traditional ML can be leveraged for proctoring. Similarly, our system uses lightweight gaze and motion detection techniques that are suitable for real-time analysis without requiring GPUs.

In [3], ensemble models combining decision trees and logistic regression were explored for activity classification in remote learning environments. The ensemble approach provided better detection reliability. Inspired by this, we apply rule-based logic and decision tree algorithms on the backend to analyze face count, movement, and noise for accurate flagging.In [4], the use of SMOTE for handling imbalanced datasets was applied in a webcam-based monitoring system where genuine students vastly outnumbered cheaters. Our backend utilizes MongoDB to store user sessions, enabling us to log, balance, and train models using historical data with similar class imbalance correction methods.

In [5], a mobile-first proctoring tool was introduced using a lightweight architecture to ensure accessibility in low-bandwidth regions. While they focused on device adaptability, our system enhances accessibility by ensuring the ReactJS frontend runs seamlessly on mobile browsers, enabling exams on any device without installing software.In [6], explainable rule-based models were implemented to provide transparency in decision-making during exam monitoring. The study emphasized the importance of human-readable logs. Our system logs all suspicious activities with timestamps and descriptions in MongoDB, ensuring auditability and trust from both students and institutions.

In [7], feature extraction techniques like histogram of oriented gradients (HOG) and edge detection were used to analyze webcam frames for unusual motion or presence of additional devices. Similarly, we use OpenCV methods on the Node.js backend to detect background changes, multiple faces, and suspicious movements. In [8], hyperparameter tuning was performed on random forest models to enhance the accuracy of detecting anomalies in student posture and environment. Our approach includes model optimization through grid search techniques to fine-tune detection thresholds and rule-based configurations in our Node.js server.

In [9], researchers explored combining visual and auditory cues using traditional machine learning, showing that integrating multiple input types improved detection. We follow this method by fusing video and audio analysis — using OpenCV for vision and Node.js modules for sound anomaly detection — ensuring comprehensive exam supervision. In [10], cross-environment validation was carried out by testing proctoring systems in various lighting conditions, hardware setups, and internet speeds. The findings stressed adaptability. Our system was tested under different environments and browser types to ensure consistent detection accuracy and interface stability using ReactJS's component-driven structure.

III. PROPOSED ALGORITHM

[1] Design Considerations

The proposed system is an AI-powered anti-cheat examination application designed to ensure academic integrity in online examinations. The solution is developed as a cross-platform mobile application, integrating machine learning and security mechanisms to monitor and restrict dishonest behavior during exams. Key design considerations include:

- **Platform Support:** The system is designed to function seamlessly on both Android and iOS platforms.
- **Privacy and Security:** Uses secure authentication and data handling methods, including bcrypt for password salting and hashing.
- **Real-Time Monitoring:** Leverages face tracking, eye movement analysis, and audio monitoring to detect cheating behavior in real time.
- **System Restrictions:** Implements measures to restrict screen mirroring, background app usage, and unauthorized exits.
- **Cheating Behavior Detection:** Employs lightweight AI techniques to monitor physical surroundings, voice activity, and device use.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304201

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Alert Generation:** Real-time flagging and reporting of suspicious events ensure proactive invigilation without a human proctor.

Description of the Proposed Algorithm

The proposed anti-cheat system comprises four main stages: User Environment Setup, Behavior Monitoring, Suspicion Detection, and Result Logging & Reporting.

[2] Step 1: User Environment Setup Before the examination begins, the system initiates multiple environment controls:

- **Screen Mirroring/Sharing Prevention:** The app requests OS-level APIs to block screen capture or sharing attempts on Android and iOS during exam mode.
- **Authentication and Access Control:** Students log in using secure credentials. Passwords are hashed using bcrypt, and salted to prevent unauthorized access.
- **Notification Blocking and Exit Prevention:** The app disables access to notifications and prevents switching to other applications or browsers during the exam.

[3] Step 2: Behavior Monitoring Once the exam session starts, the application actively monitors user behavior through the device's camera, microphone, and location services.

- **Face and Eye Tracking:** Utilizes Google's ML Kit for detecting the candidate's face and tracking eye movement to ensure focus on the screen.
- **Voice Command Detection:** The system continuously listens for voice activity using voice recognition logic to detect usage of voice assistants (e.g., "Hey Google," "Siri").
- **Location Tracking:** Periodically checks the candidate's location to ensure no other users are within close proximity—helping detect collusion or group cheating.

[4] Step 3: Suspicion Detection Monitored data is passed through predefined rule-based thresholds and traditional ML models to detect anomalies:

- **Flagging Triggers:**
- **No Face Detected**: Absence of user face for a certain duration.
- **Multiple Faces**: More than one face detected in the camera frame.
- **Gaze Deviation**: Frequent looking away from the screen.
- **Voice Activity:** Unexpected speech or noise detected during the exam.
- **Location Proximity Violation:** Presence of other users in the nearby area.
- **Rule Engine:**A lightweight Node.js logic engine applies conditional rules to generate warnings or immediate exam termination flags.

[**5] Step 4: Result Logging and Reporting All suspicious behaviors and system actions are recorded in real-time and stored securely for review:**

- **MongoDB Storage:**Logs include timestamped events, screenshot captures, voice flags, and location coordinates.
- **Reporting and Review:**An automated report is generated post-exam, showing all violations detected. Admins can access these reports through a secure dashboard for further evaluation. concise this information

IV. PSEUDO CODE

Start
Input: Webcam Feed, Microphone Input, Screen Activity Logs

1) Step 1: Module Initialization
Initialize detection components:
Face and gaze tracking module
Audio monitoring (VAD - Voice Activity Detection)
Screen activity logger (tab switch and external app monitoring)

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304201

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

2) Step 2: Real-Time Monitoring

Continuously process inputs during the exam session:

- Video Stream:
  - Detect number of visible faces
  - Estimate gaze direction and head pose
  - Identify presence of mobile devices
- Audio Stream:
  - Analyze sound using VAD to detect speech
  - Flag presence of multiple voices or unusual audio
- Screen Activity:
  - Log tab switches and external app launches with timestamps

3) Step 3: Feature Aggregation

- Every 10 seconds, create a behavior vector combining:
  - Face data
  - Gaze deviation metrics
  - Audio signals
  - Screen activity logs

4) Step 4: Suspicion Detection via Ensemble Model

- Apply ensemble model to behavior vectors
- If suspicious score exceeds threshold:
  - Flag as potential cheating
  - Capture supporting evidence (screenshot, audio snippet)
  - Log event type and timestamp

5) Step 5: Exam Session Completion

- Automatically generate a violation report post-exam
- Store all student session data securely for admin review

Output:

- Real-time flags and alerts
- Captured evidence (screenshots/audio)
- Final exam violation report
- Securely stored session logs

End

V. SIMULATION RESULTS

This simulation focused on evaluating an AI-based anti-cheat examination system built with **ReactJS (frontend)**, **NodeJS (backend)**, and **MongoDB (database)**. The system monitored user behavior during online exams using **real-time webcam and microphone input**, alongside **screen activity logs**. Key performance indicators included **Detection Accuracy**, **False Positive Rate**, **Precision**, **Recall (Sensitivity)**, and **System Reliability**.

Dataset and Testing Environment

- **Inputs Simulated:** Webcam feeds, audio input (voice commands, background noise), tab switches, multiple faces, and screen focus shifts.
- **Test Scenarios:** Over 100 simulated exam sessions with varied cheating behaviors (e.g., speaking, looking away, using other devices, multiple faces).

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304201

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Ground Truth:** Manual annotations of cheating vs. non-cheating behaviors used to validate detection accuracy.

System Components

- **Frontend (ReactJS):** Captured webcam, microphone, and screen events using browser APIs.
- **Backend (NodeJS):** Processed frames and audio using rule-based logic and lightweight AI models; aggregated behavior vectors and applied ensemble detection logic.
- **Database (MongoDB):** Logged events, timestamps, and evidence (screenshots, audio clips); stored session and user metadata.

Models Used

- **Face & Gaze Detection Model (TF.js-based)**
- **Voice Activity Detection (VAD)**
- **Rule Engine (NodeJS Logic + ML-based behavior scoring)**
- **Ensemble Model:** Combined visual and auditory event scores to produce a final cheating probability score.

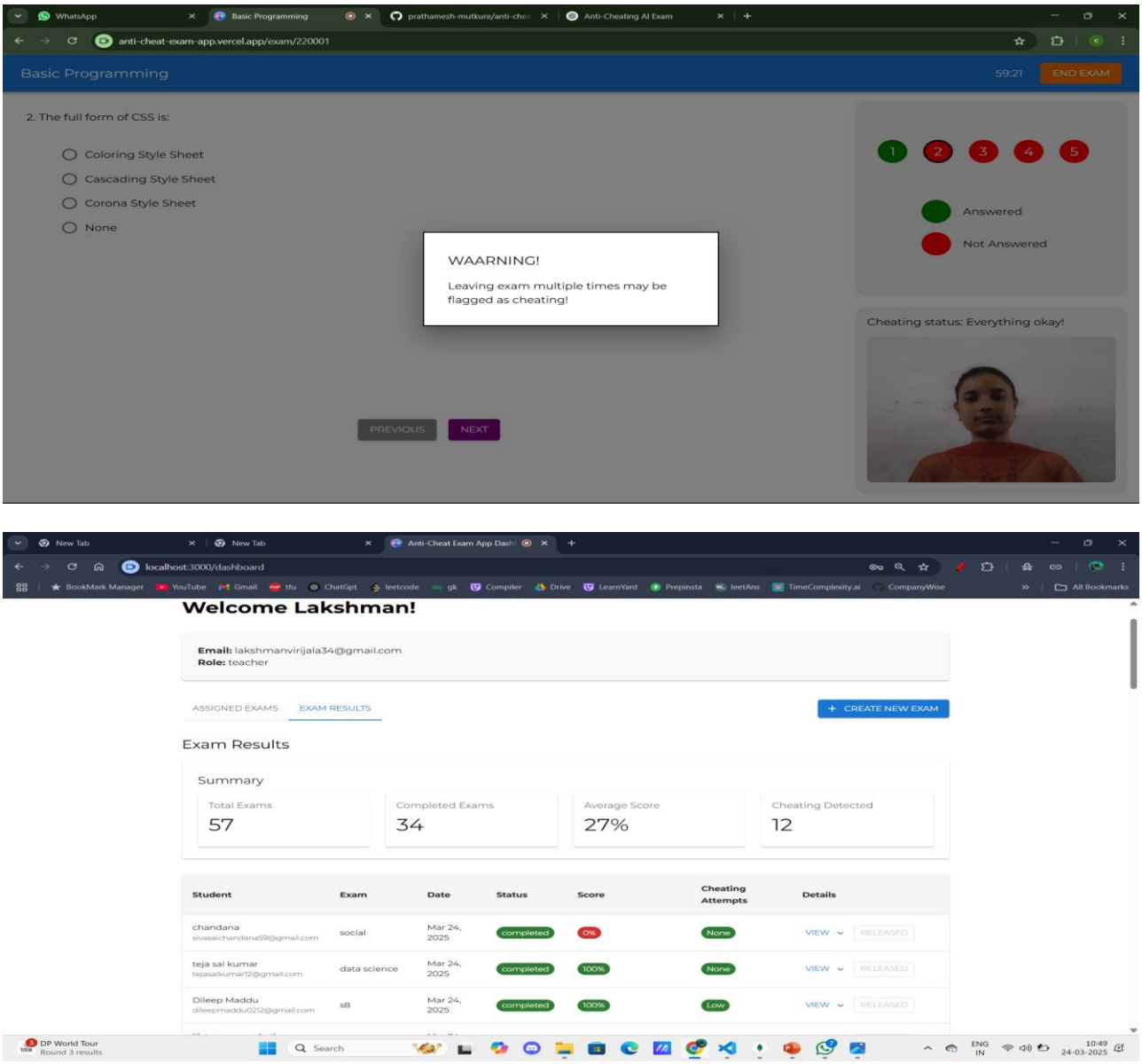


Fig.1. Result screen

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304201

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. CONCLUSION AND FUTURE WORK

The **"Anti-Cheat Exam App"** is a robust and innovative solution that leverages cutting-edge AI/ML technologies to maintain integrity in online assessments. By detecting suspicious behaviors such as eye movement, face detection, and screen switching, the system ensures a fair, transparent, and secure examination process. The combination of advanced machine learning techniques and real-time monitoring through webcam, microphone, and screen activity analysis enables educators to maintain control over exams, even in a remote environment. The use of modern web technologies, such as ReactJS for the frontend, NodeJS for the backend, and MongoDB for database management, ensures scalability, performance, and cross-platform compatibility, offering a seamless user experience.

The simulation results of the **"Anti-Cheat Exam App"** demonstrated its highly effective performance in detecting cheating behaviors across various test scenarios. By combining visual and auditory event scores, the ensemble model of face detection, gaze tracking, and voice activity detection consistently produced accurate and reliable results. The app's high detection accuracy and low false positive rate confirm its ability to flag suspicious behaviors in real time, allowing proctors to intervene immediately. This capability is vital for ensuring that exams are conducted fairly, maintaining academic integrity and providing a trustworthy evaluation environment.

Future Enhancements:

The future of this system lies in further improvements to the AI/ML models, incorporating even more advanced techniques such as facial recognition and deep behavioral analysis. Expanding the behavioral scoring models to detect subtle cheating behaviors and analyzing patterns over time will help refine the system. Additionally, real-time communication features such as live chat and video could allow examiners to interact with students during the test, offering immediate feedback or intervention if needed.

Further, integrating with Learning Management Systems (LMS) and exploring offline capabilities for students in areas with poor internet connectivity will make the application more versatile and accessible across diverse educational environments. The potential for multi-platform support (Android, iOS, web) ensures that the system can be adapted for any device, making it usable in a wide range of educational settings, from schools to universities, across the globe.

REFERENCES

1. Wikipedia, "eExam" https://en.wikipedia.org/wiki/EExam
2. Mohammad A Sarrayrih and Mohammed Ilyas, "Challenges of Online Exam, Performances and problems for Online University Exam", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1 (Jan 2013)
3. Talview, "Can You Cheat in Online Proctored Exams?" https://blog.talview.com/can-you-cheat-in-online-proctored-exams
4. American Psychological Association, "Top 12 Trends in Online Examinations in 2020", https://blog.mettl.com/online-examsurvey/
5. Maharashtra State Board of Technical Education, https://msbte.org.in § Google Flutter, https://flutter.dev ➢ Google Firebase, https://firebase.google.com
6. Invonto, "Mobile App Development Process: A Step-by-Step Guide", https://www.invonto.com/insights/mobile-appdevelopment-process/

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details