iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** XI **Month of publication:** November 2026

DOI: https://doi.org/10.22214/ijraset.2026.85947

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

Proctor-AI: An Online Examination System with Real Time Malpractice Detection

Vishal[1], Punit Reddy[2], Rajesh Biradar[3], Mutturaj[4], Sangameshwar K[5]
Department of Information Science and Engineering, Guru Nanak Dev Engineering College, Bidar, Visvesvaraya Technological University, Belagavi, Karnataka, India

Abstract: ***Online examinations are widely used in colleges today, but maintaining fairness during these exams is still a difficult task. Students writing exams from home can easily take help from others, use mobile phones, or look at notes. Most existing platforms only show questions and collect answers. They do not give the examiner clear proof of what happened during the exam session. In this project we have developed an AI-based online examination system that monitors the student in real time. The system continuously checks the webcam. If more than one face appears, the exam is terminated immediately and a warning is shown. If any electronic gadget such as a mobile phone or earphones is detected, a snapshot is taken and stored as evidence. The system also watches the student's head movement. If the student keeps looking away from the screen, a notification is displayed. Common keyboard shortcuts like Copy, Paste and Escape are completely blocked during the exam. The platform has three types of users – Student, Examiner and Admin. Students can register and log in directly. Examiners can also register, but their account is activated only after the Admin approves it. This is done to prevent misuse of the platform. The Admin has full authority over examiner approvals. Examiners can create questions with four options and mark the correct answer while creating the exam. They can also set a fixed start and end time for the exam. Students can write the exam only within that time window. After the exam ends, students get their results immediately. The examiner can later review all the snapshots and generate a PDF report of the evidence. The system is developed using Python and Flask on the backend and normal web technologies for the frontend. The main aim of this work is to help examiners by collecting clear evidence instead of forcing them to watch long video recordings of every student.***

Keywords: Online Examination, AI Proctoring, Malpractice Detection, Face Detection, Object Detection, Admin Approval, Timed Exam, Flask.

I. INTRODUCTION

In the last few years online examinations have become very common in colleges and universities. Students can write exams from their homes, which is convenient for those who stay far from campus. But this convenience also creates serious problems related to academic honesty. In a normal classroom, teachers can walk around and keep an eye on students. In an online exam, the same kind of supervision is not possible. A student writing an online exam can easily open another tab, search for answers, use a mobile phone, wear earphones to get help, or even allow another person to sit nearby. Most of the exam platforms that are currently used only focus on showing questions and collecting answers. They do not record proper evidence of these activities. Some platforms record the full video of the student, but checking long videos of every student takes a lot of time for teachers, especially when the number of students is large.

Keeping these problems in mind, we decided to develop a practical system that can monitor the student in real time using the webcam and also control the exam process properly. Our system detects multiple faces, electronic gadgets and unusual head movements. When any of these activities are found, the system takes a snapshot and stores it as evidence. In the case of multiple faces, the exam is terminated at once. The examiner can later look at all the saved snapshots and decide the action to be taken.

We have also added proper user management. Students can register and log in directly. Examiners can register, but their registration first goes to the Admin for approval. Only after the Admin approves the examiner, the examiner can log in, create questions and publish exams. This is done so that random people cannot create examiner accounts and misuse the platform. The Admin registration itself is kept hidden for privacy and security reasons. Examiners can also set a fixed start time and end time for every exam. Students can write the exam only between those two times. After the end time the system automatically closes the exam and shows a message that the session is over. While creating questions, the examiner enters the question, four options and also marks which option is correct. Because of this, students get their results immediately after submitting the exam. The examiner can also generate PDF report of all the evidence for any student. The whole system is built using Python and Flask for the backend and standard web technologies for the frontend.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue XI Nov 2026- Available at www.ijraset.com

II. RELATED WORK

Many researchers have worked on online exam proctoring. Some systems only use browser events such as tab switching and loss of focus. These methods are simple but they cannot detect physical objects or the presence of another person. Other systems use continuous video recording. While this gives complete visual information, reviewing the videos of a large number of students is time consuming for teachers.

Computer vision techniques have also been used for face detection and object detection. Libraries such as OpenCV and MediaPipe are commonly used for detecting faces in real time.

YOLO-based models are popular for detecting objects like mobile phones. Some research papers have also explored head pose estimation to check whether the student is looking at the screen or looking away.

From the study of existing work, we observed that most systems either focus only on browser events or only on camera monitoring. Very few systems combine both and also provide clear evidence snapshots, proper role-based access control and timed exams in one platform. Our project tries to bring all these features together in a practical way that can be used in colleges.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

The main problem we are trying to solve is the lack of proper real-time monitoring and clear evidence in online examinations. Existing platforms do not give the examiner enough information to decide whether a student has committed malpractice. Watching full video recordings is not practical for large classes. Also, if anyone can freely create an examiner account, the platform can be easily misused. Therefore, a system is needed that can automatically detect common malpractice activities, capture evidence, control user access properly, and present everything in an organized way for review.

B. Objectives

The main objectives of this project are:

1) To develop a web-based online examination platform with separate registration and login for students and examiners.
2) To keep examiner registration under Admin approval so that only authorized persons can create and publish exams.
3) To allow examiners to create questions with four options and mark the correct answer while creating the exam.
4) To allow examiners to set a fixed start time and end time for every exam so that students can write only within that window.
5) To detect multiple faces in the webcam feed and terminate the exam immediately if more than one face is found.
6) To detect electronic gadgets such as mobile phones and earphones and take snapshots as evidence.
7) To monitor the student's head movement and show notifications when the student looks away from the screen.
8) To completely block Copy, Paste and Escape keys during the examination.
9) To store all malpractice evidence (snapshots and logs) so that the examiner can review them later.
10) To allow the examiner to generate a PDF report of the evidence after reviewing the snapshots.

IV. SYSTEM ARCHITECTURE

The system follows a simple client-server architecture. There are three main types of users – Student, Examiner and Admin. On the student side there is a web interface that displays the exam questions and also runs the monitoring scripts. The webcam feed is processed using computer vision techniques to detect faces, objects and head pose. Browser events and keyboard restrictions are handled using JavaScript. All the data is sent to a Python Flask backend which stores the information in a database.

The examiner side allows creation of questions, setting of exam time window, viewing of results and generation of PDF reports. The Admin side is used only for approving or rejecting examiner registration requests. The Admin registration page is kept hidden so that only authorized persons can create the Admin account. Whenever a malpractice event is detected, a snapshot is taken from the webcam and saved along with the event type, student ID, exam ID and timestamp. This evidence can later be opened by the examiner from the staff dashboard.

©IJRASET: All Rights are Reserved

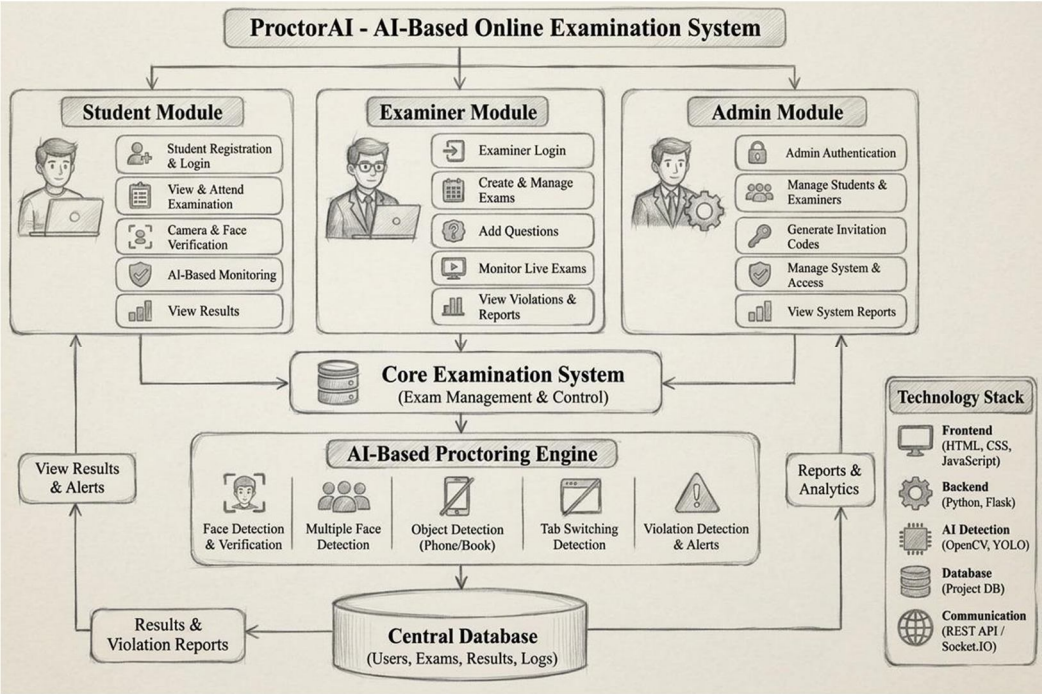


Fig 1: High-level architecture of the proposed system showing Student, Examiner and Admin modules

V. MAIN FEATURES OF THE SYSTEM

A. User Registration and Role-Based Access

The system has three roles – Student, Examiner and Admin. Students can register and log in directly. Examiners can also fill the registration form, but their account is not activated immediately. The registration details first go to the Admin for approval. Only after the Admin checks the information and approves the request, the examiner can log in and start creating exams. This is done to prevent anyone from creating an examiner account and misusing the platform by uploading wrong questions or creating fake exams. The Admin registration itself is kept hidden for privacy and security. The Admin has complete authority over the approval of examiners.

B. Question Creation and Timed Examinations

After getting approval, the examiner can create questions. For every question the examiner enters the question text, four options and also marks which option is the correct answer. Because the correct answer is stored at the time of creating the exam, the system can calculate the result immediately after the student submits the paper. The examiner also sets a start time and an end time for the exam. For example, if the examiner sets the exam from 12:00 to 1:00, students can write the exam only between these two times. If a student tries to open the exam after 1:00, the system shows a message that the session has been completed and does not allow the student to continue. This helps in conducting the exam in a controlled time window.

C. Multiple Face Detection and Exam Termination

One of the strongest features of our system is continuous multiple face detection. The webcam feed is checked regularly. If the system finds more than one face in the frame, it immediately considers this as a serious malpractice. The exam is terminated at that moment and a clear notification is shown to the student saying that the rules have been violated and the exam has been stopped. A snapshot of the frame is also saved as evidence for the examiner.

D. Detection of Electronic Gadgets

Students sometimes keep a mobile phone near the laptop or use earphones to get help during the exam. Our system tries to detect common electronic gadgets such as mobile phones and earphones using object detection. When any such object is found in the camera view, the system takes a snapshot and stores it. The event is marked as a possible malpractice case so that the examiner can check the image later.

©IJRASET: All Rights are Reserved

E. Head Pose / Looking Away Detection

If a student keeps looking left, right or above the screen for a long time, it may indicate that the student is reading notes or looking at another device. The system monitors the head orientation of the student. When it detects that the student is looking away from the screen, a notification is displayed on the exam interface warning the student. Repeated looking-away events are also logged for later review.

F. Keyboard Shortcut Blocking

During the examination, common keyboard shortcuts that are often used for copying and pasting are completely disabled. Ctrl+C (Copy), Ctrl+V (Paste) and the Escape key are blocked using JavaScript. This reduces the chance of students copying answers from other sources or trying to leave the exam page using keyboard shortcuts.

G. Evidence Collection, Review and PDF Report

All the malpractice events (multiple faces, gadget detection, looking away) are stored along with snapshots. The examiner can open any student's exam session from the dashboard and view the list of events with the corresponding images. After reviewing the evidence, the examiner can also generate a PDF report that contains the summary of the session and the captured snapshots. This report can be downloaded and kept for official records.

VI. IMPLEMENTATION

A. Technology Stack

The main technologies used in the project are listed below:

Component	Purpose
HTML, CSS, JavaScript	Student, Examiner and Admin web interfaces, keyboard blocking, browser event handling
Python	Backend logic and computer vision processing
Flask	Web framework for routes, authentication, role management and API endpoints
OpenCV / MediaPipe	Face detection, multiple face checking and head pose estimation
Object Detection Model	Detection of mobile phones, earphones and other gadgets
Database	Storage of student data, examiner approval status, exam details, violation logs and evidence images

Table I – Technology Stack

B. Working of the System

A typical flow of the system is as follows:

1) Student registers and logs into the system.
2) Examiner registers → request goes to Admin → Admin approves → Examiner can log in.
3) Examiner creates questions (with 4 options + correct answer) and sets start time & end time for the exam.
4) Student opens the exam only within the allowed time window.
5) System starts webcam monitoring (face, objects, head pose) and blocks keyboard shortcuts.
6) If multiple faces are detected → exam is terminated and evidence is saved.
7) If a gadget is detected or the student looks away → snapshot is taken and a notification is shown.
8) After submission, student gets the result immediately.
9) Examiner reviews the evidence and can download a PDF report.

VII. RESULTS AND SCREENSHOTS

The prototype has been developed and the main features have been tested functionally. Student registration and login are working. Examiner registration goes for Admin approval and only after approval the examiner can log in. Question creation with four options and correct answer marking is working. The timed exam feature correctly stops students from writing the exam after the end time and shows a session completed message. Multiple face detection is working and the exam is terminated when more than one face appears.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Snapshots are successfully captured when electronic gadgets are detected. Notifications appear when the student looks away from the screen. Keyboard shortcuts are blocked during the exam. All the evidence is stored and can be viewed from the examiner side. PDF report generation is also available.

It is important to note that the accuracy of face and object detection depends on lighting conditions, camera quality and the position of the student. In some cases the system may raise a false alert if the lighting is poor or if another person accidentally appears in the background. But still our Platform has 95 percent accuracy in both object detection and Multiple face detection. Therefore the final decision is always left to the examiner after reviewing the evidence.

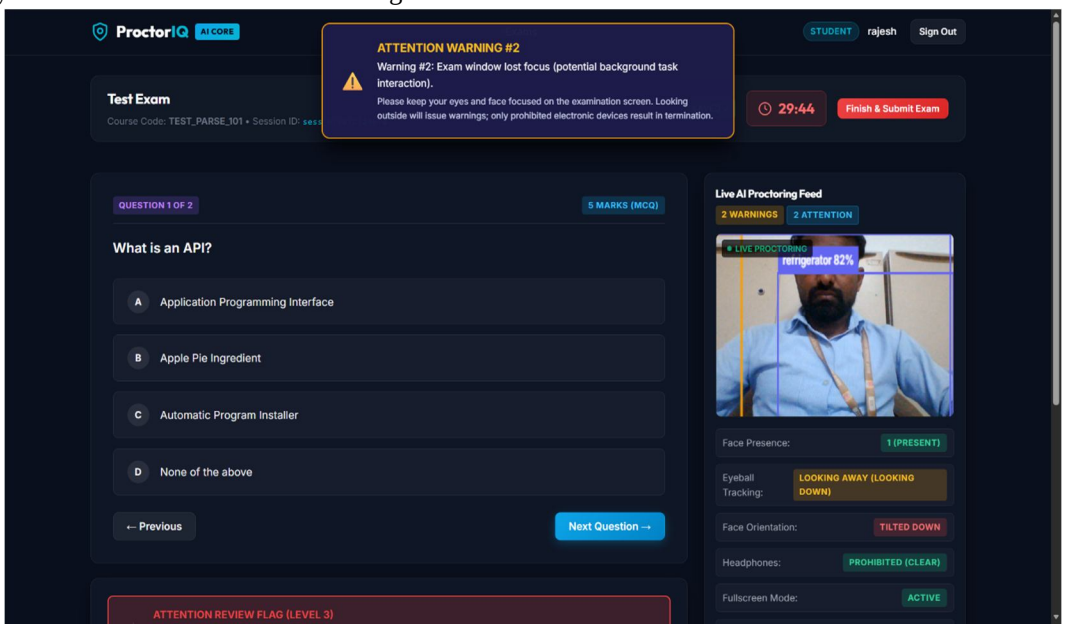


Fig 2: Student examination interface with live camera panel

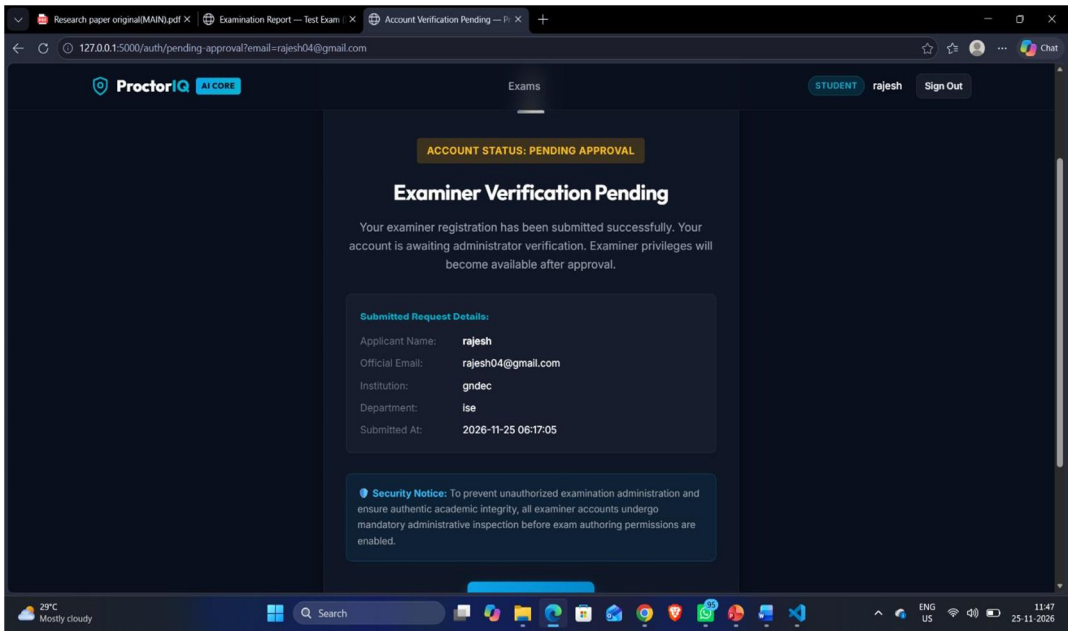


Fig 3: Examiner panel requesting for approval from admin

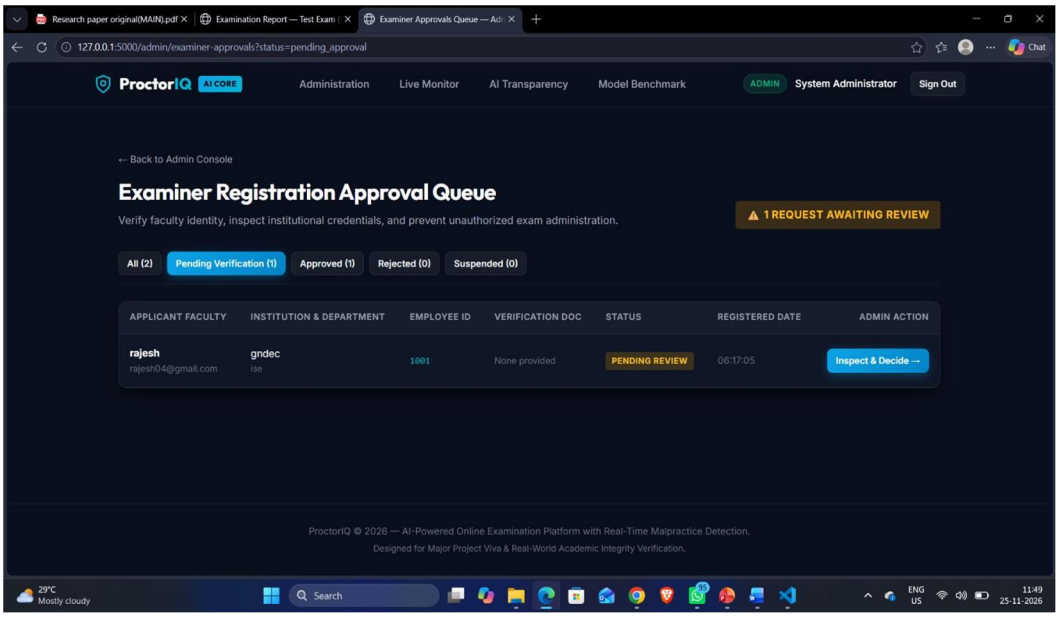


Fig 4: Admin panel for approving examiner registration requests

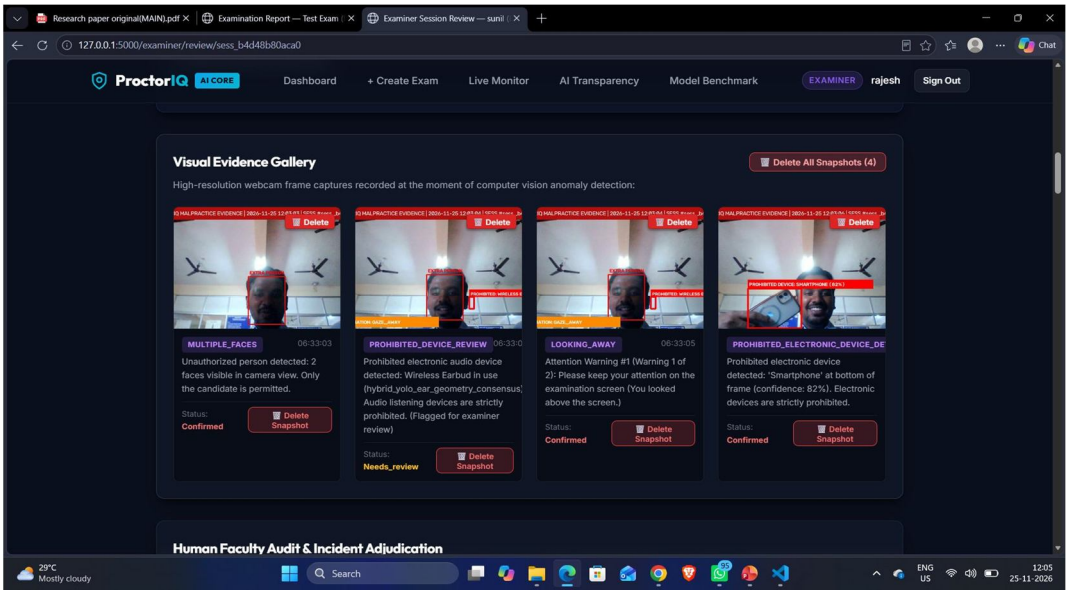


Fig 5: Evidence pictures in examiner panel for checking status of student exam

VIII. LIMITATIONS

The current version of the system has some limitations that we would like to mention honestly:

1) The accuracy of object detection and face detection can vary with lighting and camera quality.
2) The system cannot detect activities that happen completely outside the camera view.
3) The object detection model needs further fine-tuning for better reliability in real college environments.

IX. CONCLUSION AND FUTURE WORK

In this project we have developed an AI-based online examination system that focuses on real-time malpractice detection along with proper user management. The system supports student registration and login, examiner registration with Admin approval, timed examinations, question creation with correct answer marking, immediate result generation, multiple face detection with exam termination, gadget detection with snapshots, looking-away notifications, keyboard shortcut blocking, evidence review and PDF report generation.

The system is built with Python and Flask and uses computer vision techniques for monitoring. While the current version is functional, there is still scope for improvement. In future we plan to improve the accuracy of object detection, make the monitoring more robust under different lighting conditions, and test the system with a larger number of students. We also plan to add more detailed analytics for the examiners and improve the Admin dashboard.

Overall, this project shows that a practical and useful proctoring system can be built by combining face detection, object detection, head pose estimation, browser monitoring and proper role-based access control in one platform.

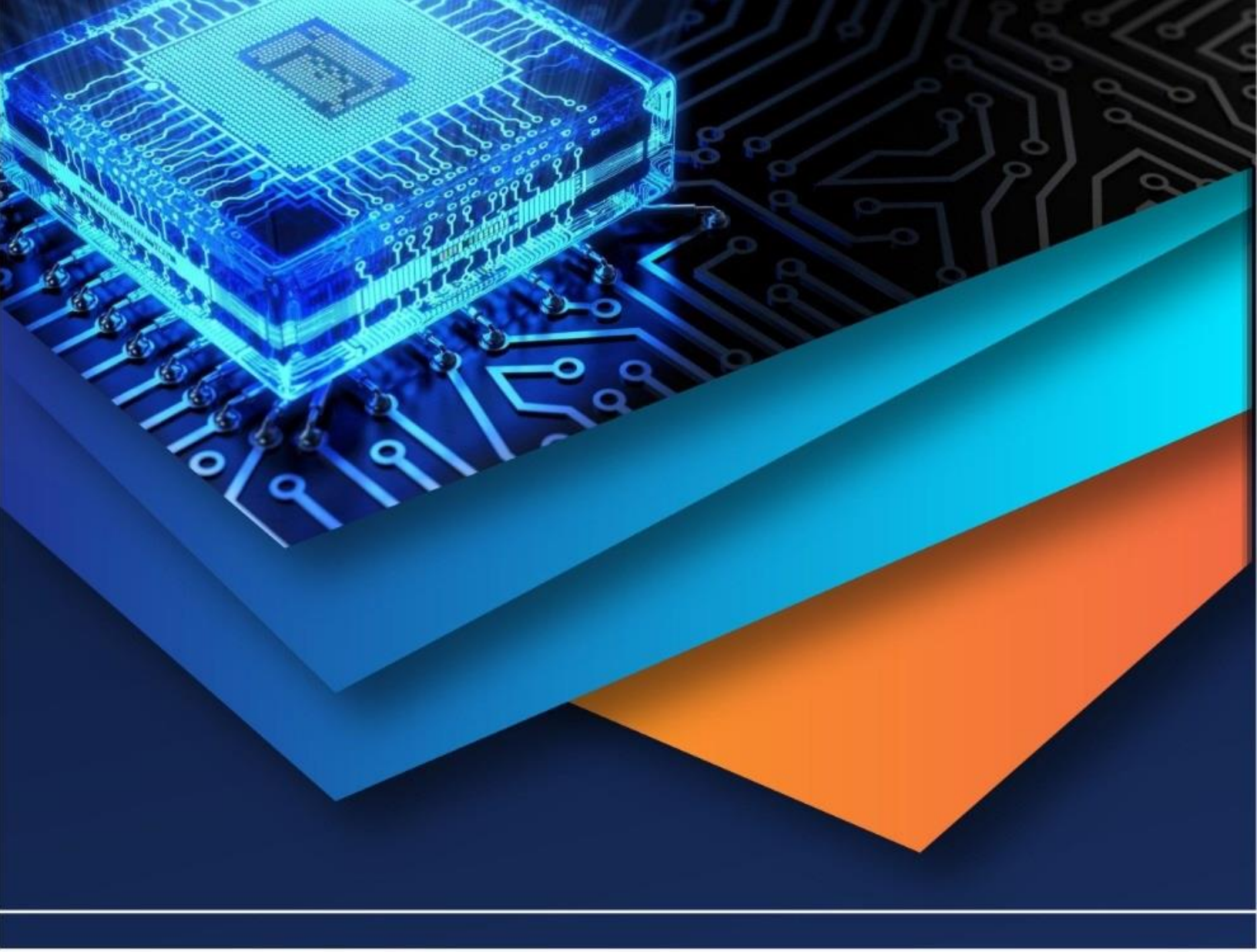
X. ACKNOWLEDGMENT

We express our sincere gratitude to Mr. Sangameshwar K, Department of Information Science and Engineering, Guru Nanak Dev Engineering College, Bidar, Karnataka, India, for his valuable guidance and continuous support throughout this research work. His suggestions, knowledge, and encouragement helped us improve our ideas and complete this study.

REFERENCES

[1] Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, “Automated online exam proctoring,” IEEE Transactions on Multimedia, vol. 19, no. 7, pp. 1609–1624, Jul. 2017, doi: 10.1109/TMM.2017.2656064.
[2] S. Prathish, A. Narayanan S, and K. Bijlani, “An intelligent system for online exam monitoring,” in Proc. Int. Conf. Information Science (ICIS), Kochi, India, 2016.
[3] C. S. Indi, V. Pritham, V. Acharya, and K. Prakasha, “Detection of malpractice in e-exams by head pose and gaze estimation,” International Journal of Emerging Technologies in Learning (iJET), vol. 16, no. 8, pp. 47–60, 2021, doi: 10.3991/ijet.v16i08.15995.
[4] S. Kaddoura and A. Gumaei, “Towards effective and efficient online exam systems using deep learning-based cheating detection approach,” Intelligent Systems with Applications, vol. 16, Art. no. 200153, Nov. 2022.
[5] A. Nigam, R. Pasricha, T. Singh, and P. Churi, “A systematic review on AI-based proctoring systems: Past, present and future,” Education and Information Technologies, vol. 26, pp. 6421–6445, 2021, doi: 10.1007/s10639-021-10597-x.
[6] S. Coghlan, T. Miller, and J. Paterson, “Good proctor or ‘Big Brother’? Ethics of online exam supervision technologies,” Philosophy & Technology, vol. 34, pp. 1581–1606, 2021.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
[8] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2001, pp. I-511–I-518.
[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. European Conf. Computer Vision (ECCV), 2014, pp. 740–755.
[10] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
[11] G. Bradski, “The OpenCV library,” Dr. Dobb’s Journal of Software Tools, 2000.
[12] M. Swain and D. Ballard, “Color indexing,” International Journal of Computer Vision, vol. 7, no. 1, pp. 11–32, 1991.
[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934.
[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, K. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117–2125.
[15] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2018.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)