WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Web-Based Smart Examination System with Face Verification and AI-Assisted Proctoring

Kanyashree C J[a], Manya P R[a], Monisha N A[a], Rakshitha D[a],Deepika G L[a,*]

[a]Department of Computer Science and Engineering, Kalpataru Institute of Technology, Tiptur 572201, Karnataka, India
Affiliated to Visvesvaraya Technological University,Belgavi,Karnataka,India

A B S T R A C T :

Online examinations reduce logistical overhead but make invigilation difficult, exposing assessments to impersonation, tab switching, unauthorised devices and outside assistance. This paper presents a web-based examination platform that combines a conventional examination workflow with AI-assisted proctoring and administrator supervision. The system is implemented with Python Flask, HTML, CSS, JavaScript and SQLite. Student identity is verified through webcam-based face recognition before an examination begins. During the examination, OpenCV and MediaPipe process webcam frames to flag missing, multiple or unknown faces and head, eye and mouth movement, while browser events record tab switching and the microphone level is monitored for sustained audio activity. A mobile-phone detection component and a voice assistant that issues spoken warnings complement these signals. Detected events are stored with timestamp, severity and image evidence; administrators can follow camera and screen views, review the activity log and reject an attempt when a violation is confirmed, so the final decision remains with a human reviewer. Multiple-choice answers are evaluated automatically and a downloadable PDF report combines marks with proctoring information. The prototype was demonstrated end to end with a 20-question, 30-minute mock examination, and unit and integration test cases were specified. No quantitative accuracy measurements are reported; they are identified as the main direction for future work. The study shows that a lightweight, open-source stack can integrate identity verification, multi-signal monitoring and human-controlled decisions in one examination platform.

Keywords: Online examination; AI proctoring; Face recognition; Computer vision; MediaPipe; Flask

Introduction

Online examinations have become a routine part of educational delivery because they reduce the need for physical examination centres, shorten the time to results and lower the administrative effort of paper-based assessment. When candidates are not physically supervised, however, the integrity of the assessment is exposed to impersonation, the use of mobile phones or other unauthorised devices, switching to other browser tabs and assistance from other persons. Continuous manual observation of every candidate through a video link is tiring and does not scale as the number of simultaneous candidates grows [1], [4].

AI-assisted proctoring addresses this problem by analysing webcam, microphone and browser signals automatically and flagging suspicious events for review [1]. Automated flags are nevertheless probabilistic: camera quality, lighting and ordinary behaviour can trigger false alerts, and the flags alone do not establish misconduct. A practical system therefore needs the artificial intelligence to produce evidence and the administrator to take the decision.

This paper reports the design, implementation and demonstration of a web-based smart examination system that follows this principle. The system lets students register, verify their identity by face recognition, attend timed multiple-choice examinations and receive results, while administrators create examinations, supervise attempts, review recorded events and reject an attempt when a serious violation is confirmed. The techniques used (face detection and recognition, landmark-based head and eye analysis, object detection for mobile phones) are established; the contribution of this work lies in their integration into one deployable platform, and not in a new detection algorithm. The main contributions are:

- a layered architecture that separates the presentation, application, AI-monitoring, database and result-generation functions of an online examination system;
- a multi-signal monitoring module that records missing, multiple and unknown faces, head, eye and mouth movement, audio activity, tab switching and mobile-phone presence, with timestamp, severity and image evidence;
- a voice assistant that gives spoken instructions and warnings, and an administrator console with camera and screen views, an activity log and an explicit reject-attempt action;
- automatic evaluation of multiple-choice answers and a downloadable PDF report that combines marks with proctoring information; and
- a working prototype, demonstrated with a mock examination, together with a unit and integration test specification.

The remainder of the paper is organised as follows. Section 2 reviews related work, Section 3 states the problem and objectives, Section 4 describes the methodology, Section 5 presents the system design and implementation, Section 6 reports results and discussion, Section 7 lists limitations, and Section 8 concludes with future work.

Related work

1.1. Online proctoring approaches

Nigam et al. reviewed 43 papers published between 2015 and 2021 on AI-based and non-AI online proctoring, covering system architectures, design parameters, technical and human issues, trends and future directions [1]. Atoum et al. proposed an automated proctoring framework that combines several visual and audio cues verify the exact sensing modalities against the original paper [2]. Kaddoura and Gumaei proposed a real-time cheating-detection approach based on deep learning, with front-camera, back-camera and speech modules, and evaluated it on a public audio and video dataset [3]. Hussein et al. evaluated commercially available online proctoring tools verify the scope of the evaluation [4].

Beyond these studies, the project report on which this paper is based groups existing approaches into five categories: AI-based proctoring, face-recognition-based examination, automated online examination management, computer-vision-based activity detection and automated warning systems. Table 1 summarises their typical features and limitations as described in the report. The categories are generalisations and are not attributed to specific published systems.

Table 1 - Categories of existing approaches and their limitations

Approach	Main features	Limitation
AI-based proctoring	Webcam monitoring, suspicious-activity detection	False alerts; cannot always identify malpractice
Face recognition	Identity verification before access	Sensitive to lighting, camera quality and appearance change
Online exam management	Exam delivery and automatic results	Limited or no monitoring
Computer-vision monitoring	Face, movement and phone detection	Accuracy depends on camera, lighting, visibility and model
Automated warnings	Visual or audio alerts to candidates	Does not cover all misconduct; human review still needed

1.2. Research gap

Basic examination platforms often authenticate only with a username and password and provide little or no monitoring, whereas monitoring-centred systems may lack spoken candidate warnings or direct administrator control over an ongoing attempt. The gap addressed here is the integration, within a lightweight open-source stack, of face-based identity verification, multi-signal monitoring, audible warnings and administrator-controlled rejection, with all recorded evidence kept available for human review. This draft does not include a systematic comparison with specific published systems; the literature table should be extended to 15–25 verified papers before submission.

Problem statement and objectives

Problem statement. Online examination systems that rely on credential-based login and manual observation cannot reliably detect impersonation, unauthorised devices, tab switching or unauthorised assistance, especially when many candidates are examined at once. The problem addressed is to build an online examination platform that verifies candidate identity, monitors the candidate automatically during the attempt, warns the candidate, and presents evidence to an administrator who can reject the attempt, without requiring continuous manual supervision.

Objectives. The objectives of the work are:

- O1: to develop a web-based platform for registration, secure login and timed multiple-choice examinations;
- O2: to verify student identity by face recognition before the examination;
- O3: to detect missing, multiple and unknown faces, head, eye and mouth movement, audio activity, tab switching and visible mobile phones;
- O4: to give spoken instructions and warnings through a voice assistant;
- O5: to let administrators view camera and screen information, review recorded events and reject an attempt when a serious violation is confirmed; and
- O6: to evaluate answers automatically, generate results and downloadable PDF reports, and store all records in a database.

Scope and significance. The system targets multiple-choice examinations taken on a desktop browser with a webcam, a microphone and screen sharing. The report identifies schools, colleges, universities, training institutes, recruitment organisations and distance education as possible users. By turning invigilation into reviewable, time-stamped evidence, the system aims to lower supervision effort while keeping accountability with a human administrator.

Proposed methodology

1.3. Overall workflow

The methodology follows the workflow in Fig. 1. A student logs in and completes face verification through the webcam; on a successful match the student reaches the dashboard, reads the examination instructions and starts the attempt. The instructions state that the timer begins after face verification, that camera, microphone and entire-screen sharing are required, that head and eye signals need administrator review and do not automatically reject the attempt, and that leaving the examination tab or using copy and paste is recorded as a violation. During the attempt the monitoring module analyses the camera, microphone and browser signals, stores events, and triggers a spoken warning when configured. Answers are saved automatically and the attempt is submitted when the timer reaches zero or the student submits. The system then evaluates the answers and generates the result and the proctoring report. In parallel, the administrator may open the monitoring view, review the activity log and reject the attempt.

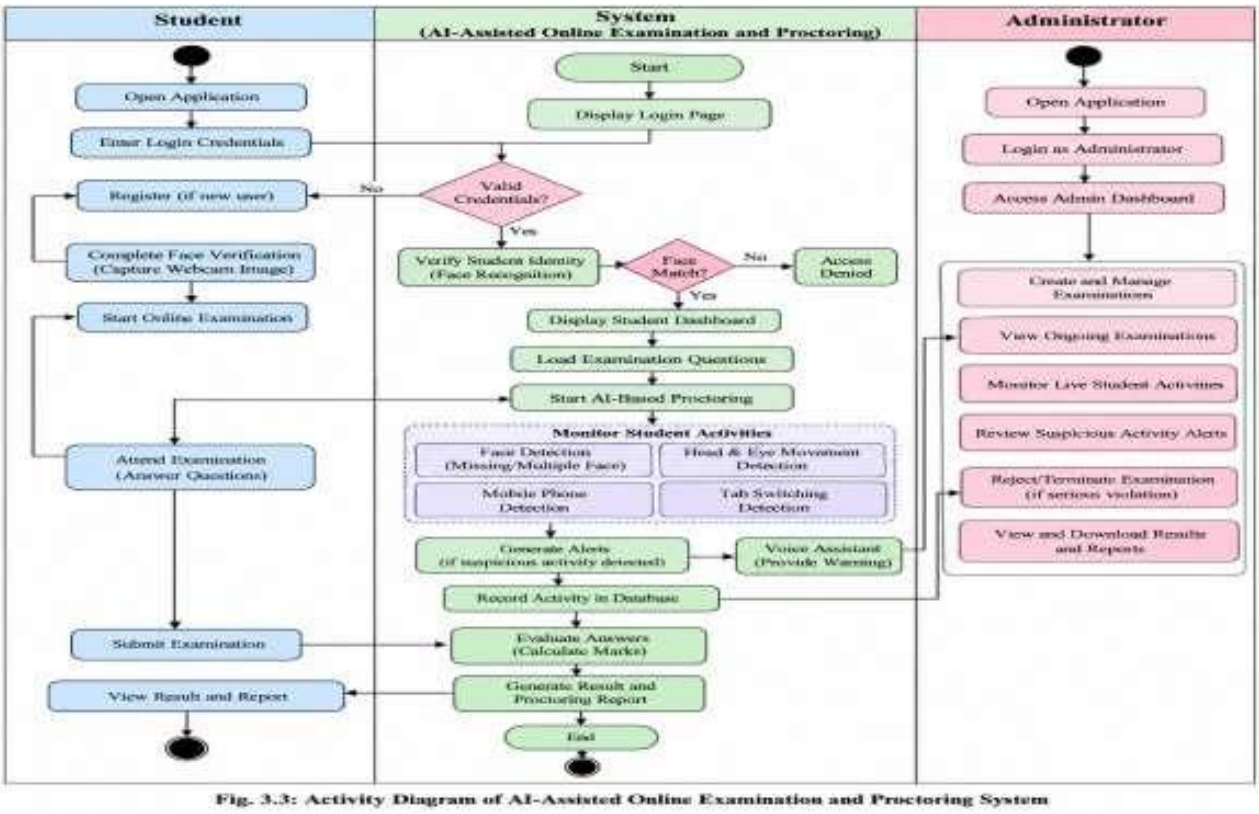


Fig. 3.3: Activity Diagram of AI-Assisted Online Examination and Proctoring System

Fig. 1 - Activity diagram of the proposed examination and proctoring workflow (student, system and administrator lanes). Source: project report, Fig. 3.3.

1.4. Identity verification

Before the examination, the system captures a webcam image, detects the face and compares it with the facial data registered for the student; a mismatch denies access. The student login page also offers face-based login, in which the camera opens automatically and the student is identified from the captured face. **The report does not state the face-recognition library or model, the representation used for matching, or the decision threshold; these must be added.**

1.5. Monitoring signals and event recording

Table 2 lists the signals handled by the monitoring module, as evidenced by the report's module descriptions, the monitoring interface and the generated report. Each recorded event stores the activity type, time, severity, a description and, for camera-based events, an image as evidence. The activity log shows that camera-based events are recorded only when a condition is sustained across at least three frames, which limits alerts from transient movement; audio events likewise require a sustained elevated microphone level.

Table 2 - Monitored signals

Signal	Source and mechanism (as described in the report)
Face missing, multiple faces, unknown face	Webcam frames; face detection and recognition (OpenCV, MediaPipe)
Head movement	Head orientation from facial landmarks (MediaPipe); status such as facing forward / centred
Eye movement	Gaze estimation (MediaPipe); reported as unavailable when tracking fails
Mouth movement	Facial landmarks; sustained signal over at least 3 frames
Audio activity	Microphone level; sustained elevated level
Tab switch	Browser visibility event: exam tab hidden (tab switch, minimise or another application)
Mobile phone	Object-detection model on webcam frames model not specified

The monitoring loop can be summarised as in Algorithm 1, where N is the number of consecutive frames required before an event is recorded. The value $N = 3$ is inferred from the log messages, and the audio threshold θ is not given in the report; both should be confirmed from the source code.

Algorithm 1: Monitoring loop (reconstructed)

```
for each captured frame f:
  s <- extract_signals(f)  # faces, head, eye, mouth, phone
  for each signal type k in s:
    c[k] <- c[k] + 1 if cond_k(s) else 0
    if c[k] >= N:
      record_event(k, time, severity, evidence(f))
      voice_warning(k); c[k] <- 0
on exam tab hidden: record_event(tab_switch)
on mic level > theta for sustained period:
  record_event(audio_activity)
```

1.6. Administrator supervision

The administrator console shows, for each ongoing attempt, the student's exam screen and camera image, head and eye status, face count, the number of answers saved and the number of recorded violations, with links to the report and a reject-attempt action. According to the instructions shown to students, the camera and screen images are periodically refreshed, and the interface marks an image as stale with its age. When an attempt is rejected, the backend records the status and reason and blocks further participation.

1.7. Evaluation and scoring

Answers are evaluated against the stored correct options. For n questions with marks m_q, student answer a_q and correct answer c_q, the obtained marks S and the percentage P are

$$S = \Sigma\, m_q \cdot \delta(a_q, c_q),\quad q = 1 \ldots n \qquad (1)$$

$$P = (S / T) \times 100 \qquad (2)$$

where δ is 1 if the two answers are equal and 0 otherwise, and T is the total marks of the examination. In the demonstration examination $T = 20$, and one correct answer out of 20 questions gives $S = 1.0$ and $P = 5.0$ %, as shown in the generated report (Fig. 6).

System design and implementation

1.8. System architecture

The system follows a five-layer web architecture (Fig. 2). The **presentation layer** (HTML, CSS, JavaScript) provides registration and login pages, dashboards, the examination interface with timer, and the camera and screen-sharing views. The **application layer** (Python Flask) handles authentication, examination and question management, answer processing, timers, rejection requests and communication with the database. The **AI-monitoring layer** (OpenCV, MediaPipe and face recognition) performs identity verification and monitoring and passes detection results and alerts to the application layer. The **database layer** (SQLite) stores users, examinations, answers, results, suspicious-activity records and rejection records. The **result-generation layer** evaluates answers, computes marks and percentages and produces PDF reports with the ReportLab library.

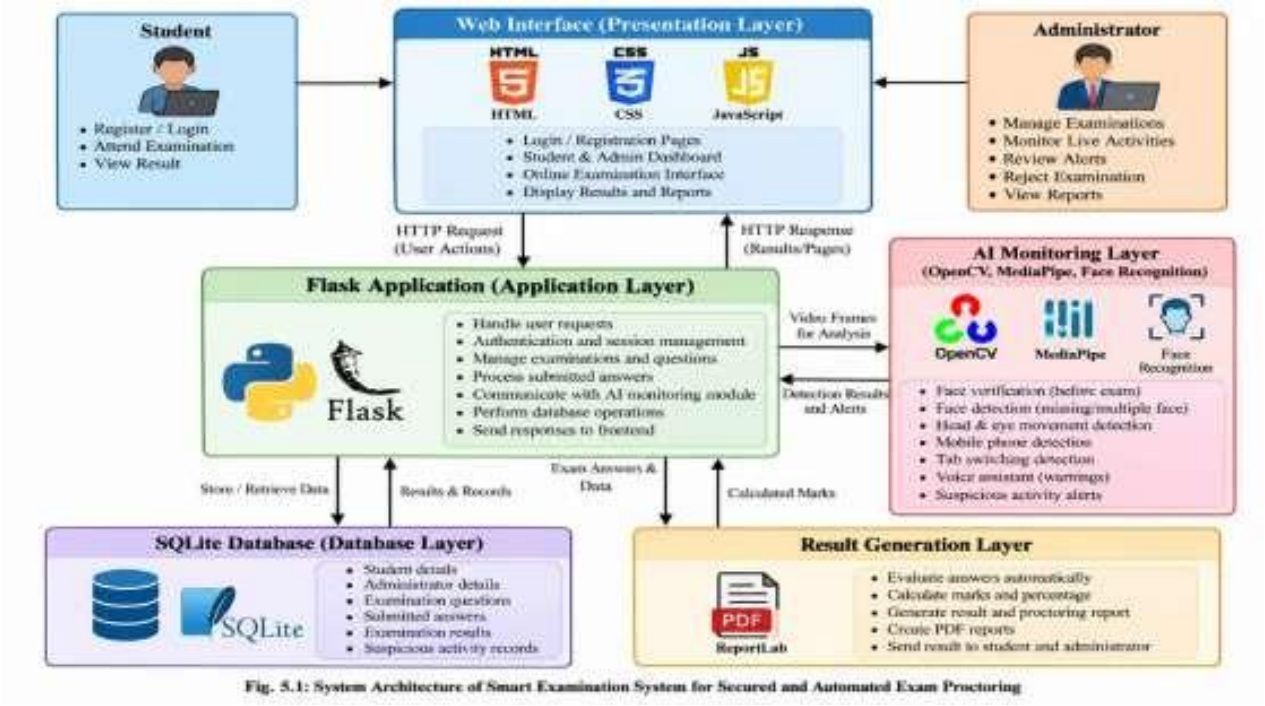


Fig. 5.1: System Architecture of Smart Examination System for Secured and Automated Exam Proctoring

Fig. 2 - System architecture of the proposed system. Source: project report, Fig. 5.1.

Table 3 - Role of each technology

Technology	Role in the system
Python, Flask	Backend logic, routing, authentication, session and request handling [7]
HTML, CSS, JavaScript	User interface, timer, camera, microphone and screen-sharing access, tab-visibility events
SQLite	Persistent storage of users, exams, answers, results and events [8]
OpenCV	Image capture and processing for face detection and recognition [6]
MediaPipe	Landmark-based head, eye and face analysis in the live-monitoring module [5]
Voice assistant	Spoken instructions and warnings; can be muted by the student implementation to be specified
ReportLab	Generation of downloadable PDF result and proctoring reports [9]

1.9. Modules

The system is decomposed into ten modules: administrator, student management, face recognition, online examination, AI proctoring, mobile-phone detection, voice assistant, administrator exam rejection, result and report generation, and database management. The administrator module covers login, creation of examinations, question management, viewing students, monitoring, alert review, rejection and results. The online-examination module displays instructions, loads questions, shows options and the timer, saves answers and submits on completion or timeout.

1.10. Database design

Table 4 summarises the relational schema. Primary and foreign keys relate students, examinations, questions, results and events, and a separate table records rejections. **The schema in the report omits some fields that the interface displays (for example register number, mobile number, guardian names and date of birth for students, and a fourth answer option); please update the table to match the implemented schema.**

Table 4 - Database tables (from the report)

Table	Main fields
administrator	admin_id, username, password_hash, email
student	student_id, student_name, email, password_hash
examination	exam_id, exam_name, duration, total_marks
question	question_id, exam_id, question_text, options, correct_answer

Table	Main fields
result	result_id, student_id, score, percentage
suspicious_activity	activity_id, student_id, exam_id, activity_type, event_time, severity, description
examination_rejection	rejection_id, student_id, exam_id, admin_id, rejection_reason, rejected_at

1.11. Implementation steps

Implementation proceeded as follows. (1) The development environment was set up with Python, Flask and Visual Studio Code, and the application was tested in Google Chrome. (2) The SQLite schema was created for users, examinations, questions, results, suspicious activities and rejections. (3) Flask routes were written for registration, login, examination listing, attempt start, answer saving, submission, administrator actions and report download. (4) Ten user interfaces were built: registration, login, student dashboard, administrator dashboard, face verification, examination instructions, online examination, AI monitoring, administrator live monitoring and result pages. (5) Face verification was integrated before the examination starts. (6) The monitoring module was connected to the examination page, showing live status (face count, head position, eye status, microphone and phone-detection status) and storing events**. State whether MediaPipe inference runs in the browser or on the server, since this affects the architecture description and the privacy discussion.** (7) The voice assistant, tab-switch detection and the administrator rejection action were added. (8) Automatic evaluation and PDF report generation were implemented. (9) Modules were integrated and tested with unit and integration test cases. Passwords are stored as hashes according to the database schema.

Results and discussion

1.12. Demonstration setting

The prototype was demonstrated with a mock examination titled “UPSC General Studies Mock Examination” of 20 multiple-choice questions, 30 minutes duration and 20 marks, using a desktop browser with camera, microphone and screen sharing. The administrator dashboard in the demonstration showed one registered student and one examination. **Please add the hardware, operating system, browser, camera model, lighting conditions and number of test participants.**

1.13. Functional results

Figs. 3–6 show the main outputs. During the attempt (Fig. 3) the candidate sees the timer, the examination question and a live-monitoring panel showing face count, head position, microphone status and phone-detection status; a notice states that the administrator can see the camera and shared screen. The administrator view (Fig. 4) shows the candidate's screen and camera, head and eye status, the number of answers saved (3 of 20 in the example) and recorded violations (3), with a reject action. The activity log (Fig. 5) lists time-stamped events with severity, description and image evidence. The result page (Fig. 6) reports marks, attempted and unanswered questions, correct and wrong counts, percentage and a proctoring status.

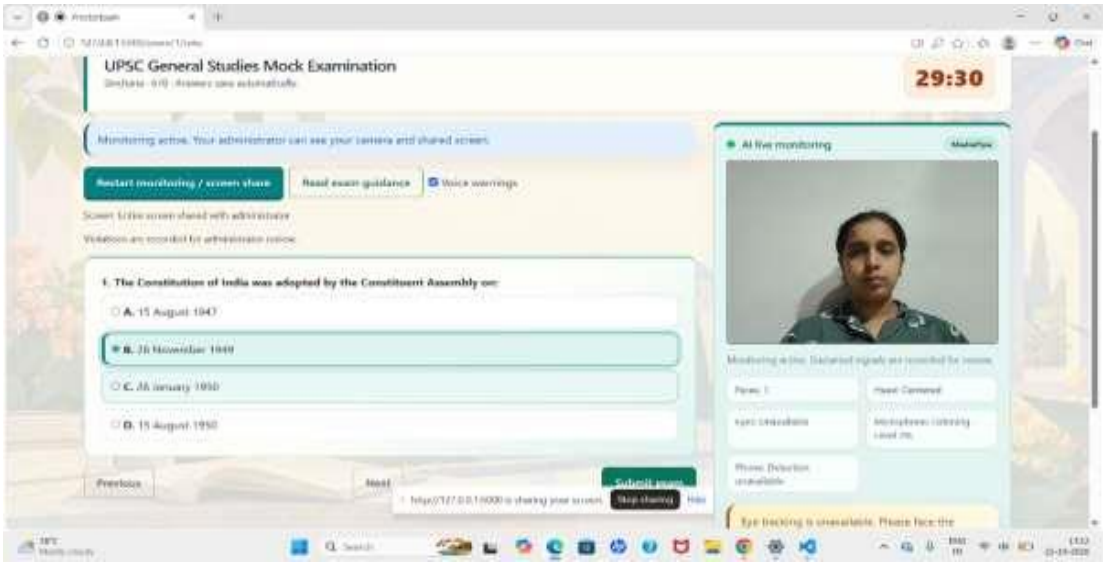


Fig. 3 - Examination page with AI live monitoring.

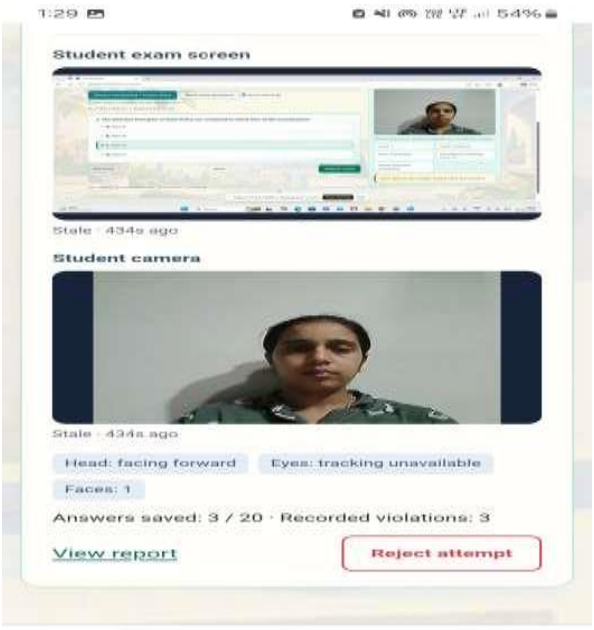


Fig. 4 - Administrator view of an ongoing examination.

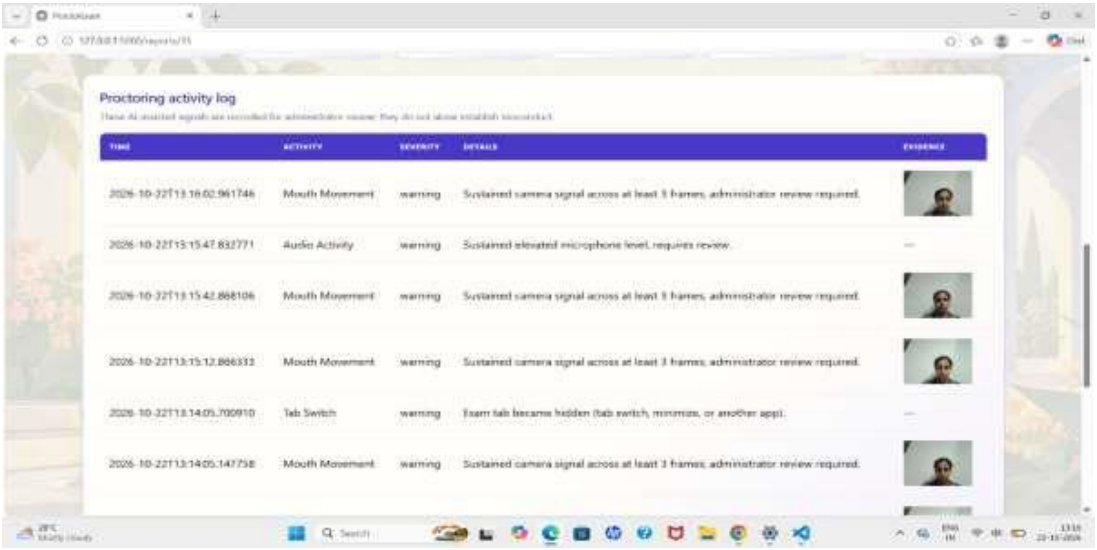


Fig. 5 - Proctoring activity log with evidence images.

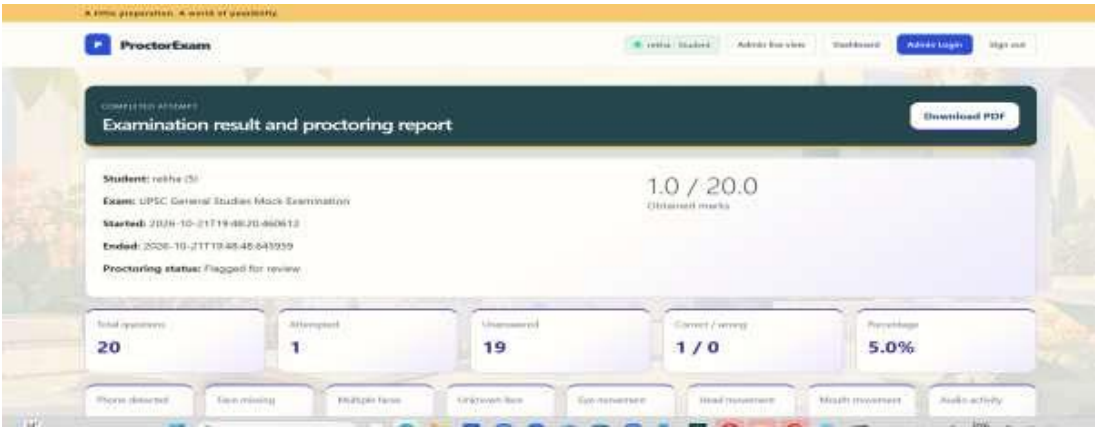


Fig. 6 - Examination result and proctoring report (top of page).

Table 5 - Values observed in the demonstration

Item	Observed value
Examination	UPSC General Studies Mock; 20 questions; 30 min; 20.0 marks
Sample attempt	Attempted 1; unanswered 19; correct/wrong 1/0
Marks and percentage	1.0 / 20.0; 5.0 %
Proctoring status	Flagged for review
Admin dashboard	Students 1; exams 1; active attempts 1; completed 2; flagged 12
Log entries visible in Fig. 5	Mouth movement ×4, audio activity ×1, tab switch ×1; all severity "warning"
Live view (Fig. 4)	Head: facing forward; eyes: tracking unavailable; faces: 1; violations: 3

1.14. Testing

The report specifies 15 unit test cases (UT01–UT15) and 13 integration test cases (IT01–IT13) with expected results. Unit cases cover valid and invalid login, face verification of a registered and an unknown face, starting an examination, answer submission, no-face and multiple-face events, mobile-phone detection, tab switching, voice warning, administrator rejection, evaluation, report generation and logout. Integration cases cover login with dashboard, face verification with examination access, question loading, answer storage, monitoring and alert storage, phone alerts to the administrator, voice alerts, live monitoring, rejection status update, evaluation and report generation, and session handling**. The report lists expected results only; the actual pass/fail outcomes, any defects found and the number of runs must be added before this section can be presented as a result.**

1.15. Discussion

The demonstration shows that the platform covers the complete cycle from identity verification to a proctoring-annotated report. Three observations stand out. First, the design keeps the human in the loop: the activity log states that signals are recorded for administrator review and do not alone establish misconduct, and the student instructions say that head and eye signals do not automatically reject an attempt. This responds directly to the false-alert limitation of AI-based proctoring (Table 1). Second, the sustained-frame rule for camera events trades detection latency for fewer alerts from momentary movement. Third, in the captured sessions eye tracking and phone detection were reported as unavailable, and the interface asked the candidate to face the camera; the system therefore reports degraded sensing explicitly, but it also shows that these two signals depended on conditions that were not met in the demonstration.

No comparison with existing systems in terms of accuracy or speed is made, because the report contains no benchmark dataset, accuracy, precision, recall or false-alert figures, and no existing system was evaluated under the same conditions. The results are therefore functional demonstrations and should not be read as performance claims.

Limitations

- No quantitative evaluation: face-recognition accuracy, event-detection precision and recall, false-alert rate and latency were not measured.
- The demonstration used one mock examination and, according to the dashboard, a single registered student; scalability with many simultaneous candidates was not tested.
- Eye tracking and phone detection were unavailable in the captured sessions, and the phone-detection model is not specified.
- Face recognition and landmark analysis are sensitive to lighting, camera quality and appearance changes, and no liveness or anti-spoofing check is described.
- Administrator views are periodically refreshed images and not continuous video, so brief events may be missed in live supervision.
- The system stores facial images and screen and audio-derived events; consent, data protection, retention and fairness issues are not analysed. SQLite may limit concurrent write load in large deployments.
- Only multiple-choice questions are supported.

Conclusion and future work

This paper presented a web-based smart examination system that integrates face-based identity verification, multi-signal AI-assisted monitoring, spoken warnings, administrator-controlled rejection, automatic evaluation and PDF reporting on a Flask, SQLite, OpenCV and MediaPipe stack. A mock-examination demonstration showed the complete student and administrator workflows, including recorded events with image evidence and a proctoring-annotated result report. The contribution is the integration of established techniques into a reviewable, human-controlled platform. Because no accuracy measurements were made, the work supports claims about functionality and design, not about detection performance.

Future work should (i) evaluate face recognition and each monitoring signal on a labelled dataset with participants under varied lighting and cameras, reporting precision, recall and false-alert rates; (ii) calibrate thresholds such as the sustained-frame count and audio level; (iii) evaluate the mobile-phone detector and make eye tracking robust; (iv) add liveness detection against photo or video spoofing; (v) load-test the platform and consider

a server database for concurrent use; (vi) add privacy safeguards, including consent, retention limits and encrypted evidence storage; and (vii) extend the question types beyond multiple choice.

Acknowledgements

The authors thank [guide / institution / participants] for their support**. Complete or remove.**

REFERENCES :

[1] A. Nigam, R. Pasricha, T. Singh, and P. Churi, "A systematic review on AI-based proctoring systems: Past, present and future," *Educ. Inf. Technol.*, vol. 26, pp. 6421–6445, 2021, doi: 10.1007/s10639-021-10597-x. VERIFY: full author list

[2] Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1609–1624, Jul. 2017. VERIFY: volume, pages, DOI

[3] S. Kaddoura and A. Gumaei, "Towards effective and efficient online exam systems using deep learning-based cheating detection approach," *Intell. Syst. Appl.*, vol. 16, Art. no. 200153, Nov. 2022. VERIFY: initials, volume, article number

[4] M. J. Hussein, J. Yusuf, A. S. Deb, L. Fong, and S. Naidu, "An evaluation of online proctoring tools," *Open Praxis*, vol. 12, no. 4, pp. 509–525, 2020. VERIFY: volume, pages

[5] C. Lugaresi *et al.*, "MediaPipe: A framework for building perception pipelines," arXiv:1906.08172, 2019.

[6] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000. VERIFY

[7] Pallets Projects, "Flask documentation." [Online]. Available: https://flask.palletsprojects.com/ (listed in the project report)

[8] SQLite Development Team, "SQLite documentation." [Online]. Available: https://www.sqlite.org/docs.html (listed in the project report)

[9] ReportLab, "ReportLab PDF toolkit documentation." [Online]. VERIFY: URL and access date