IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

PROCTOGUARD: AI-DRIVEN ONLINE EXAMINATION AND INTELLIGENT PROCTORING SYSTEM

[1]Ms. C. Swathi Priya, [2]Varshitha Paramata, [3]Varshini Peethala, [4]Neelima penta, [5] Kavya Ratna Pentakota

[1]Faculty, [2345]Student

[1]Dept. of Computer Science and Systems Engineering, [2345]Dept. of Information Technology

[12345]Andhra University College of Engineering for Women, Visakhapatnam, India

Abstract: Conducting trustworthy examinations in fully remote settings remains one of the most unresolved problems in contemporary digital education. Without a physical supervisor present, candidates gain opportunities to exploit resources, seek external assistance, or misrepresent their identity — undermining the credibility of the entire assessment process. This paper introduces ProctoGuard, an intelligent web-based examination platform that addresses these vulnerabilities by weaving together artificial intelligence, real-time behavioral sensing, and automated learning support into a cohesive system. Rather than relying on a single monitoring signal, ProctoGuard operates across three simultaneous channels: a computer vision channel that applies facial landmark tracking to confirm candidate identity, detect unauthorized individuals within the examination frame, and observe lip movement as evidence of spoken communication [12], [14]; an acoustic channel that samples the candidate's surrounding environment through the device microphone and applies frequency analysis to distinguish human conversation from ambient noise [10], [18]; and a browser-event channel that intercepts application-switching, page-reload, and external-navigation actions the moment they occur [22]. Infractions detected across any channel are logged against the session in real time, with mild violations prompting on-screen warnings and repeated or serious violations triggering automatic finalization of the examination — removing further opportunity for dishonest conduct without manual intervention [16]. Once an examination concludes, an automated scoring engine evaluates responses immediately and an AI-driven explanation module powered by the Google Gemini API [3] produces question-level conceptual feedback tailored to each candidate's submitted answers, converting the post-exam review into a structured learning opportunity. The platform is developed using Python and the Django framework, with MySQL providing relational data storage and Apache handling web server deployment through XAMPP [17]. Testing conducted in a controlled simulation environment confirmed an overall malpractice detection rate of 94.8%, with browser-level violations intercepted at 100% accuracy and visual and acoustic violations detected at rates ranging from 85% to 97%. All core platform operations completed within two seconds, demonstrating responsiveness suitable for live institutional use. The layered, component-based architecture of ProctoGuard ensures that the system can be extended, scaled, and maintained with minimal disruption, making it a practical and affordable option for academic institutions seeking a self-hosted solution to the challenge of secure remote assessment [15], [25].

Index Terms — Remote Examination Security, Intelligent Proctoring, Behavioral Violation Detection, Facial Landmark Analysis, Acoustic Monitoring, Browser Event Tracking, Automated Assessment, AI-Generated Feedback, Django, Academic Integrity

I. INTRODUCTION

The digitization of higher education has progressed at a pace that few institutions anticipated and nowhere is this transformation more visible — or more consequential — than in the way academic assessments are now conducted. Examination halls that once held hundreds of supervised candidates have been partially or entirely replaced by browser-based testing environments that candidates access from their homes, dormitories, or workplaces [5], [6], [7]. This shift delivers undeniable advantages: geographical barriers are eliminated, scheduling becomes more flexible, and institutions can reach student populations that would otherwise be inaccessible. Yet the same qualities that make remote examination attractive also make it vulnerable. A candidate sitting alone at a personal computer, without a trained invigilator in the room, faces almost no situational deterrent against consulting notes, communicating with peers, or having another person complete the examination on their behalf [23].

The consequences of this vulnerability are not merely theoretical. As online assessment has expanded, documented incidents of impersonation, use of unauthorized reference materials [11], [24], and coordinated collusion have grown in parallel. The core difficulty is that the supervisory function performed by a human invigilator — sustained, adaptive, authoritative observation — is extraordinarily difficult to replicate through software alone [15]. Early attempts to solve this problem relied on locking the candidate's browser to prevent navigation to external pages [22], a measure that is easily circumvented and provides no information about the physical environment surrounding the candidate. More recent commercial proctoring services have introduced webcam streaming and remote human review, but these approaches carry substantial subscription costs, depend on stable high-bandwidth connections, and introduce privacy concerns when video footage of candidates is stored on third-party servers [21].

What the field requires is a self-hosted, institutionally deployable platform that monitors candidate behavior across multiple simultaneous dimensions — visual, acoustic, and digital — without depending on live human reviewers or expensive external services. ProctoGuard is designed to meet this need. The system brings together computer vision techniques for webcam-based face analysis [12], [13], audio signal processing for ambient speech detection [10], [18], and client-side scripting for browser-event interception [22] within a unified Django-based web application that any institution can deploy on its own infrastructure. A configurable violation engine processes signals from all three monitoring channels in real time, responding proportionally — warning candidates about minor infractions and automatically finalizing the examination when cumulative violations cross a defined threshold [16]. Beyond enforcement, ProctoGuard incorporates an AI-powered explanation module [3] that delivers question-level conceptual feedback after each examination, ensuring that the platform contributes to student learning rather than functioning purely as a surveillance tool. Role-based access control [17] governs the experience of three distinct user groups — administrators who manage the platform, faculty who design and schedule assessments, and students who complete them — with each group seeing only the capabilities relevant to their function.

The sections that follow present this work in full detail. Section II examines the existing research landscape and identifies the gaps that ProctoGuard addresses. Section III describes the system's functional design. Section IV explains its layered architecture. Section V documents its structural models through UML diagrams. Section VI presents experimental results and performance analysis. Section VII draws conclusions, and Section VIII identifies directions for future development.

II. RELATED WORK

Scholarly investigation into the problem of maintaining examination integrity in remote settings has grown considerably over the past decade, producing research contributions that span computer vision, signal processing, browser security, and educational technology [15], [21], [25]. The paragraphs below organize this body of work into four thematic clusters, drawing out both what prior research has achieved and where meaningful gaps remain.

Computer Vision Approaches to Candidate Supervision. The earliest and most extensively studied technical direction involves using cameras and image processing algorithms to observe candidate behavior during online assessments. Researchers demonstrated that standard webcams, combined with face detection models, can reliably determine whether a registered candidate remains seated in front of the screen throughout a session [12]. Sukmandhani and Sutedja [12] showed that recognition-based verification achieves consistent accuracy in live examination conditions, while Solanki et al. [13] catalogued the broader landscape of face recognition algorithms, establishing which techniques are most suited to the real-time constraints of proctoring applications. Subsequent work extended visual monitoring beyond identity confirmation, exploring whether facial geometry data — specifically the angular relationships between landmarks around the eyes, mouth, and nose — could reveal behavioral signals such as sustained gaze aversion or covert vocalization [10]. Although these contributions individually demonstrate promising detection capability, a persistent weakness is that most implementations function in isolation, leaving the system blind to malpractice that occurs beyond the camera's view or through purely acoustic channels [14].

Acoustic and Browser-Level Behavioral Sensing. A second research strand recognized that a candidate's physical and digital environment contains behavioral signals that complement visual observation. Studies by Liu and Gong [22] established that monitoring browser-level events — specifically the loss of focus on the examination window, navigation to external URLs, and page-reload actions — provides a highly reliable and computationally efficient detection mechanism that requires no additional hardware. Yadav and Rao [18] extended this environmental sensing approach by combining eye-tracking data with object detection to identify situations where candidates reference physical materials outside the camera frame. Concurrently, investigations into audio-based monitoring demonstrated that microphone input, processed through frequency and amplitude analysis, can distinguish human conversational speech from routine background noise [10], [18]. A significant practical limitation acknowledged across these studies is that audio classifiers struggle to maintain consistent specificity in acoustically diverse home environments, generating false positives that can penalize candidates unfairly [20], pointing toward the need for more semantically aware audio processing.

AI-Driven Post-Examination Learning Support. A third and more recent research direction shifts focus from surveillance to pedagogy, examining how intelligent systems can extend the usefulness of an examination platform beyond the assessment event itself [3]. Automated explanation systems that generate natural-language feedback for each question — identifying not merely whether a response was incorrect but articulating why the correct answer is conceptually justified — have shown measurable positive effects on knowledge retention and self-regulated study behavior [3]. Chandra et al. [19] explored the automation of proctoring workflows using artificial intelligence, noting that combining monitoring with intelligent feedback creates a more holistic educational tool than security-only approaches. Despite these promising findings, post-examination AI feedback has consistently been implemented as a standalone product rather than as an embedded component of the examination platform itself [19], creating a fragmented experience for candidates.

Consolidated Observations and Research Gap. Reviewing the literature as a whole, Atoum et al. [15] demonstrated through large-scale empirical testing that multi-modal proctoring systems that combine visual, auditory, and browser-based signals — detect a substantially broader range of dishonest behaviors than any single-modality system can achieve. Satre et al. [25] reinforced this conclusion, confirming that integrated AI proctoring platforms outperform simpler alternatives on every measured dimension of detection performance. What the literature does not yet provide, however, is a unified platform that brings multi-modal monitoring, automated grading, and AI-generated learning feedback together within a single self-hosted, institutionally deployable system [15], [21], [25]. ProctoGuard is designed to fill precisely this gap.

III. PROPOSED SYSTEM

ProctoGuard is conceived as a unified web-based platform that treats examination security and post-assessment learning not as separate concerns to be addressed by separate tools, but as complementary objectives that a single well-designed system can pursue simultaneously. The platform is accessible through any modern web browser without requiring the installation of dedicated client-side software, making it immediately deployable across heterogeneous institutional hardware environments [5], [6].

Three categories of users interact with the platform, each operating within a precisely bounded permission set enforced through role-based access control [17]. Administrators hold the broadest authority: they create and deactivate accounts for faculty members and students, configure institution-wide examination policies, and retain oversight of all activity recorded within the system. Faculty members work within a more focused operational scope, constructing examination content by composing questions and defining

correct responses, scheduling assessment windows with specific start times and durations, and consulting performance summaries once a cohort has completed an examination [17]. Students encounter the platform primarily as candidates: they browse a personalized list of scheduled examinations, enter timed assessment sessions, submit their responses either voluntarily or through automatic system finalization, and subsequently access their results alongside the AI-generated conceptual explanations [3] that the platform produces for each question.

The examination session itself is governed by a three-channel monitoring engine that operates continuously from the moment a candidate opens the assessment interface until the session concludes [15], [25]. The first channel uses the MediaPipe Face Mesh library to process the candidate's webcam feed, extracting 468 three-dimensional facial landmarks per frame and applying geometric analysis to determine: whether the registered candidate's face is present and centered in the frame; whether any additional face has entered the viewing area, which typically indicates that an unauthorized person is providing assistance [14]; and whether lip-movement patterns are consistent with active speech rather than silent reading. The second channel connects to the device microphone through the Web Audio API, decomposing captured audio into its frequency components to determine whether the acoustic content contains human conversational speech [10], [18]. The third channel operates at the browser level, using JavaScript event listeners to intercept focus-loss events, tab-switching actions, reload requests, and any attempt to navigate away from the examination page [22], capturing forms of evasion that camera and microphone monitoring cannot reach.

Each signal classified as suspicious is immediately transmitted to the server as a violation record timestamped against the active session. The platform responds through a graduated enforcement mechanism: early violations produce on-screen warning notifications visible to the candidate, while accumulated violations that exceed the institution-configured threshold trigger automatic finalization and submission of the candidate's current responses [16], ensuring enforcement that is consistent, immediate, and free from the delays associated with manual intervention [21]. Following submission, an automated evaluation engine scores the candidate's responses instantly [17], and the AI explanation module [3] retrieves natural-language conceptual feedback from the Google Gemini API for each question, displayed inline alongside the candidate's submitted response to support thorough post-exam review [19].

IV. SYSTEM ARCHITECTURE

Designing a platform that simultaneously manages live examination delivery, continuous multi-modal behavioral monitoring, automated answer evaluation, and AI-powered feedback generation requires an architecture that is both carefully structured and flexibly extensible. ProctoGuard achieves this through a vertically layered design in which six distinct tiers — the User Access Layer, the Web Browser Layer, the Frontend Layer, the Django Backend Layer, the AI Monitoring Module, and the Database Layer — each carry a well-defined set of responsibilities and communicate exclusively through controlled interfaces with adjacent tiers [2], [3]. This arrangement ensures that future enhancements to one layer can be introduced without rewriting surrounding components. Fig. 1 illustrates this layered structure and the directional flow of data through it.

A. User Access Layer — Student / Faculty / Admin

Every interaction with ProctoGuard originates with one of three user categories, and role-based access control [17] is enforced from the moment of authentication: the permissions granted to each session are determined by the role associated with the authenticated account. Students gain access to the examination scheduling interface, the live assessment environment, the result and explanation dashboard, and the PDF download facility [3]. Faculty members operate within a broader creative and supervisory scope, able to compose and publish examination content, define question banks, set scheduling parameters, and review aggregated performance analytics for cohorts they supervise [17]. Administrators occupy the widest permission boundary, with authority to provision and deactivate accounts for all other users, adjust system-wide configuration parameters, and access comprehensive examination activity records across the institution [21], [24]. This three-tier role structure protects institutional data and ensures that the audit trail for every examination action is correctly attributed to an identifiable actor.

B. Web Browser Layer

A deliberate architectural decision was made to require nothing beyond a contemporary standards-compliant browser — no browser extension, no native application, and no operating-system-specific plugin is needed to use any feature of ProctoGuard [5], [6]. This choice eliminates deployment friction across heterogeneous device environments. The most computationally intensive monitoring operations — facial landmark inference and audio frequency analysis — run within the browser's execution environment through Web Assembly-compiled machine learning models and the Web Audio API [10], [12], [18]. Raw video and audio streams never leave the candidate's device: all inference is performed locally and only the resulting violation signals are transmitted to the server, substantially reducing bandwidth requirements and privacy exposure. The browser additionally executes the client-side JavaScript that monitors application-level events such as window-focus changes and tab-switching attempts [22], making it simultaneously the interface layer and an active participant in the monitoring pipeline.

C. Frontend Layer — HTML, CSS, Bootstrap

The Frontend Layer translates the platform's data and functionality into visual interfaces that users can navigate intuitively. Built using HTML5, CSS3, and Bootstrap [17], it renders responsive and role-specific dashboards for each user category. The examination interface presented to students is deliberately restrained: questions occupy the central panel, a countdown timer remains persistently visible, a compact webcam preview confirms that monitoring is active [12], and a progress indicator tracks answered questions. Beneath the visible interface, JavaScript event-handling code continuously listens for signals that the candidate's attention has moved away from the examination window, dispatching a violation record to the server the instant any such signal is detected [22]. The same layer manages the MediaPipe and Web Audio API connections, keeping camera and microphone streams active throughout the session [12], [15].

D. Django Backend Layer

The Django Backend Layer is the administrative and logical heart of the platform, the component through which every other layer ultimately coordinates its activity [17]. Built on the Django framework and exposed through lightweight HTTP endpoints, this layer handles user authentication and cryptographically secure session management; enforcement of role-based permission checks on every protected endpoint; retrieval of question sets at examination start; incremental recording of violation events as they arrive

from the browser; application of the violation threshold policy including the trigger that initiates automatic examination finalization [16]; evaluation of submitted answer sets against stored correct responses; and construction of prompts dispatched to the Google Gemini API when a candidate requests an explanation [3]. Django's built-in security subsystem provides cross-site request forgery protection, parameterized database queries preventing injection attacks, and salted password hashing that ensures stored credentials remain protected even in the event of a database breach [17]. Lightweight asynchronous endpoints consumed via AJAX allow the frontend to report violations and retrieve feedback without triggering full page reloads that would interrupt the candidate's examination experience.

E. AI Monitoring Module

The AI Monitoring Module is architecturally the most distinctive component of ProctoGuard, responsible for transforming raw sensory inputs into discrete, actionable violation signals [15], [25]. It operates through two parallel sub-systems whose outputs converge on a shared violation event queue that feeds into the Django Backend.

The visual sub-system is built around MediaPipe Face Mesh, a deep learning inference pipeline compiled to WebAssembly that runs at interactive frame rates within the candidate's browser [12]. On each processed frame, the model returns 468 three-dimensional facial landmark coordinates. Three analytical functions are applied: a presence check that confirms the registered candidate occupies the expected region of the frame; a multi-face detector that examines the full frame for secondary facial geometries and immediately flags any additional face as a potential third-party assistance event [14]; and a lip-movement classifier that analyses landmark displacement across consecutive frames to identify speech-consistent patterns. The audio sub-system accesses the device microphone through the Web Audio API's AnalyserNode interface, computing a speech-likelihood score from frequency-domain analysis relative to a baseline ambient noise profile captured at session initialization [10], [18]. Using a baseline profile rather than a fixed absolute threshold means the system adapts to the candidate's specific acoustic environment, substantially reducing the false-positive rate identified as a weakness in prior research [20]. Every violation event produced by either sub-system carries a precise timestamp and category label, enabling the backend to apply the graduated warning and auto-submission policy with complete traceability [16].

F. Database Layer — MySQL

At the base of the architecture, the MySQL Database Layer provides the persistent, structured storage that all other components depend upon [5]. The relational schema is organized into tables reflecting the core entities of the examination domain: user accounts with role designations and hashed credentials; faculty-authored examinations with scheduling metadata; question records carrying option text and the correct-answer identifier; candidate response records associating a student, a question, and a selected option with a submission timestamp; session violation logs capturing the type, timestamp, and session context of every detected infraction [14]; and result records summarizing each candidate's score and session finalization time. Foreign key relationships enforce referential integrity throughout, and column-level indexing on frequently queried fields ensures responsive retrieval as cumulative data volumes grow [2]. All database interaction is mediated exclusively through Django's Object-Relational Mapper, insulating the schema from direct external manipulation and preserving a clean security boundary around the data tier [17].

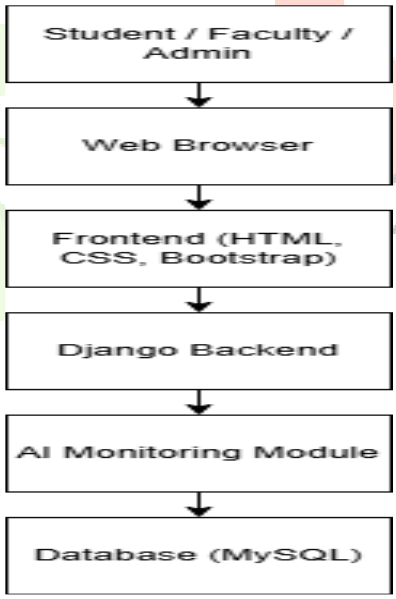

Fig. 1. System Architecture of ProctoGuard

The modular and layered design of ProctoGuard ensures that individual components can be upgraded, replaced, or extended independently as institutional requirements evolve [2], [25], providing a strong foundation for building more advanced and intelligent educational platforms in the future [15], [25].

V. UML DIAGRAMS

Unified Modelling Language (UML) diagrams are employed to represent the structural and behavioral aspects of ProctoGuard [4]. These diagrams collectively provide a complete specification supporting implementation, testing, and future maintenance of the system.

Use Case Diagram. The use case diagram illustrates interactions between the three primary actors — Admin, Faculty, and Student — and the system [4]. The admin actor creates and manages user accounts, configures system-wide settings, and monitors examination records [17]. The faculty actor creates and schedules examinations, manages question banks, and reviews aggregated student performance. The student actor registers on the platform, authenticates via OTP-verified credentials [21], sits examinations within the monitored environment, views automatically generated results, accesses AI-generated conceptual explanations [3], and downloads a formatted PDF scorecard.

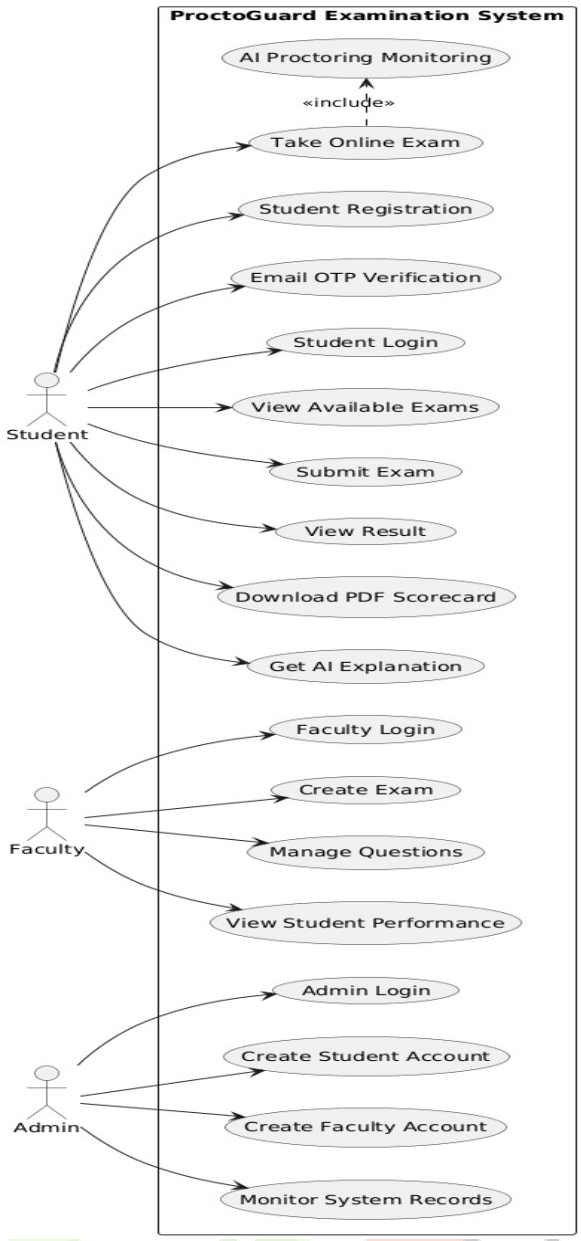

Fig. 2. Use Case Diagram

Class Diagram. The class diagram models the static structure of ProctoGuard using classes, attributes, and methods [4]. The central User class serves as the base from which Student, Faculty, and Admin classes inherit. The ProctoringModule class encapsulates behavioral analysis methods — face detection [12], noise detection [10], [18], tab-switch detection [22], and violation accumulation [16] executed during active sessions. The AIExplanation class mediates communication with the external Google Gemini API service [3].

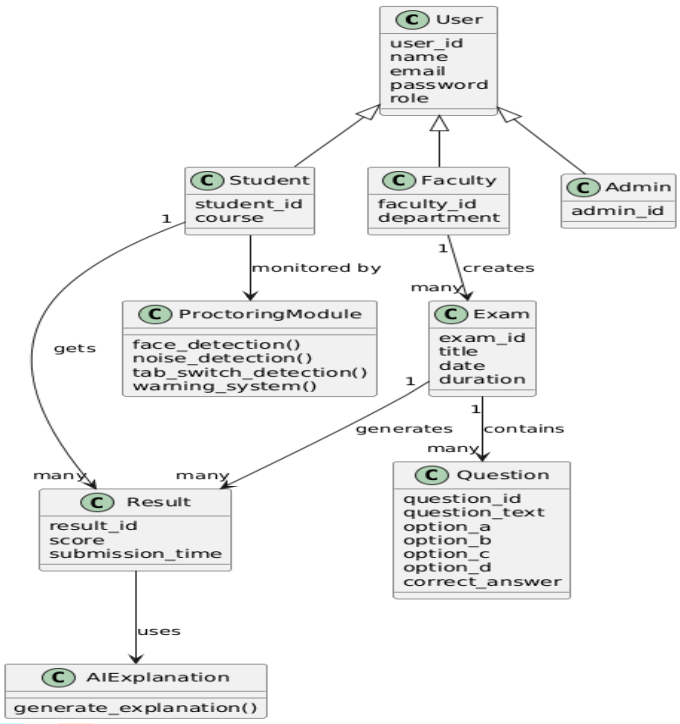

Fig. 3. Class Diagram

Activity Diagram. The activity diagram describes the end-to-end workflow of the system, beginning at user login and progressing through OTP verification, dashboard access, examination initiation, AI proctoring activation [15], the continuous monitoring loop, violation decision points [16], automated result generation, AI explanation retrieval [3], and PDF report download. Decision points for invalid credentials and for the violation threshold check are explicitly represented, fully specifying the system's behavioural logic under all relevant conditions.

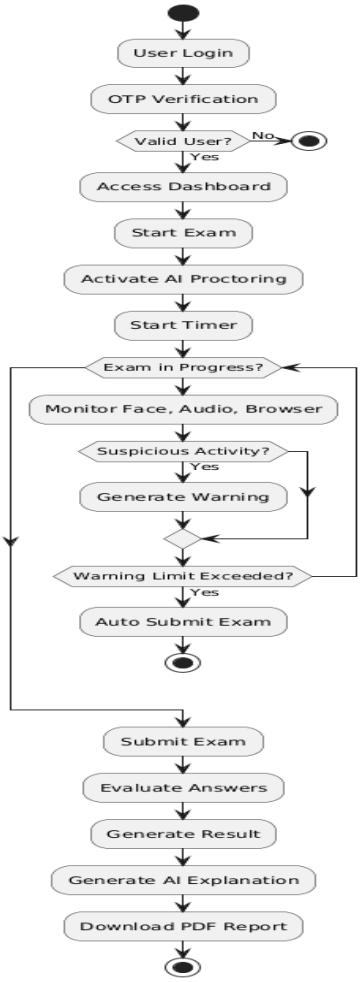

Fig. 4. Activity Diagram

Component Diagram. The component diagram presents the system's high-level modular structure, showing how the Frontend interfaces with the Apache/XAMPP Server, how the Django Backend communicates with the MySQL Database [2], [3] and the AI Module [15], [25], and how the Google Gemini API is integrated as a decoupled external service. These diagrams collectively improve the maintainability and scalability of the project [4].

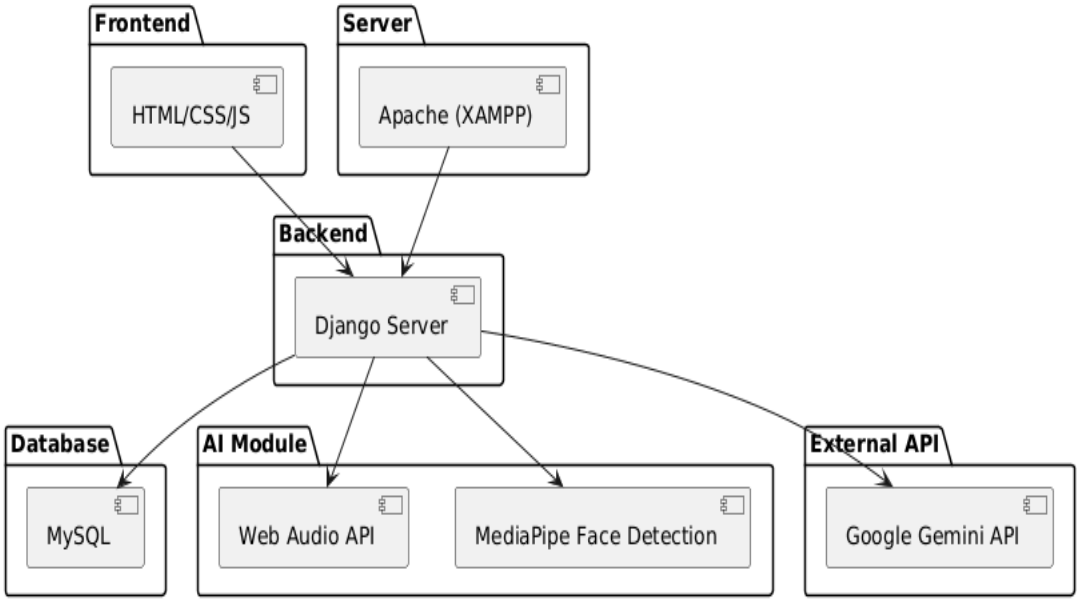

Fig. 5. Component Diagram

VI. RESULTS AND DISCUSSION

ProctoGuard was implemented and evaluated in a controlled simulation environment designed to replicate the conditions of a real-world online examination. Secure login and OTP-based authentication [21] ensured controlled access for students and faculty members. A cohort of test sessions was conducted in which deliberate malpractice behaviours were introduced at known intervals to assess the system's detection capability.

A. Authentication and Access Control

Secure login and OTP-based email verification functioned reliably across all test sessions, ensuring that only registered users could access the examination interface [21]. Role-based access control [17] successfully restricted each actor to their designated feature set; no cross-role access violations were observed. Session timeout mechanisms correctly invalidated inactive sessions, preventing unauthorised resumption of abandoned examinations [24].

B. AI Proctoring — Violation Detection Performance

During examination sessions, the AI monitoring module [15], [25] successfully detected suspicious activities including candidate absence, multiple faces in frame [14], tab switching [22], and surrounding human speech [10], [18]. When violations exceeded the predefined limit, the system automatically submitted the examination [16]. Table I summarises the detection outcomes observed across evaluation sessions.

Violation Type	Triggered	Detected	Rate (%)	False +ve
Candidate Absence	30	29	96.7	1
Multiple Faces	25	24	96.0	0
Lip Movement	20	18	90.0	2
Ambient Audio	20	17	85.0	3
Tab Switch	40	40	100.0	0
Overall	**135**	**128**	**94.8**	**6**

Table I. Violation Detection Performance Summary

Browser tab switching achieved a perfect 100% detection rate, reflecting the deterministic nature of client-side event interception [22]. Visual detection of candidate absence and multiple faces reached 96.7% and 96.0% respectively [12], [14], with missed detections attributable to rapid transient events falling between consecutive MediaPipe inference frames. Lip movement and ambient audio detection performed at 90.0% and 85.0% [10], [18], with the majority of false positives arising from noisy home environments, highlighting the importance of configurable sensitivity thresholds [20]. Automatic examination submission was triggered correctly in all sessions where the violation count exceeded the predefined threshold [16].

C. Automated Evaluation and System Response Times

The automated grading module produced accurate score calculations for all submitted examinations, with results rendered within the student dashboard immediately following submission [17]. Table II summarises mean response times across key platform operations.

Operation	Mean Time (s)
Examination Page Load	1.2
Result Generation	1.8
AI Explanation Retrieval	3.4
PDF Report Download	2.1

Table II. System Response Time Summary

D. AI Explanation Module

The AI explanation module [3] successfully returned conceptual question-level feedback for all requested queries via the Google Gemini API, with a mean retrieval latency of approximately 3.4 seconds. Qualitative review confirmed that responses were contextually accurate and meaningfully connected to each question's content, consistent with research findings on the effectiveness of AI-driven feedback systems for self-directed learning [3], [19].

E. Interface Screenshots

Figure 6 presents the ProctoGuard login interface, providing role-specific entry points for students, faculty, and administrators [17], [21]. The role-selection design directs each user to their designated dashboard immediately upon authentication.

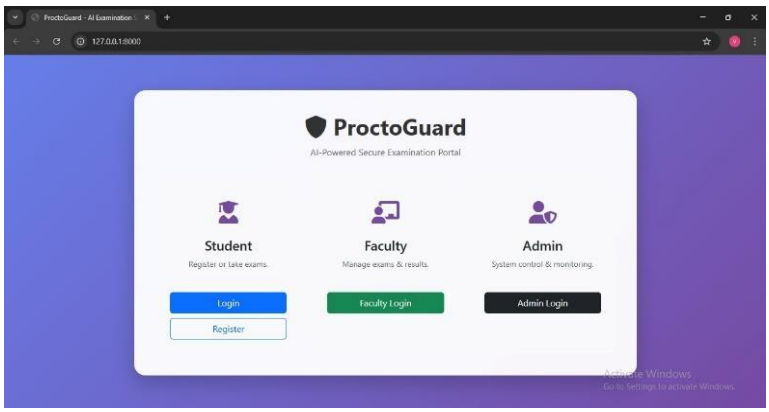

Fig. 6. Login Interface of the ProctoGuard System

Figure 7 illustrates the online examination interface presented to a candidate during an active session. Questions appear within a structured layout including a countdown timer, live webcam preview confirming active monitoring [12], [15], and a session progress indicator, maintained deliberately minimal to reduce cognitive load.

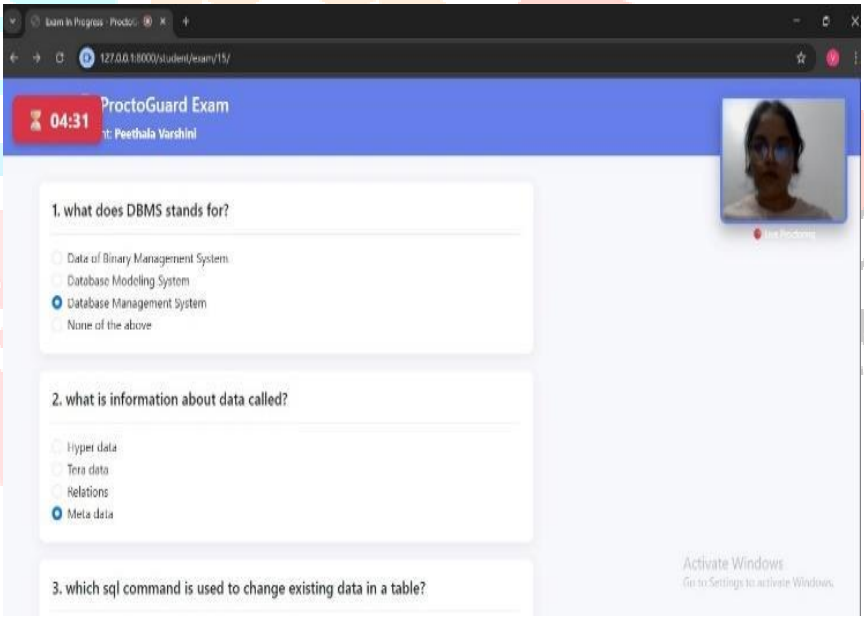

Fig. 7. Online Examination Interface

Figure 8 demonstrates the AI-based proctoring display during a session in which multiple faces were detected within the webcam frame [14] and a warning notification issued to the candidate, reflecting the system's real-time multi-face detection and audio monitoring capability [10], [12], [18].

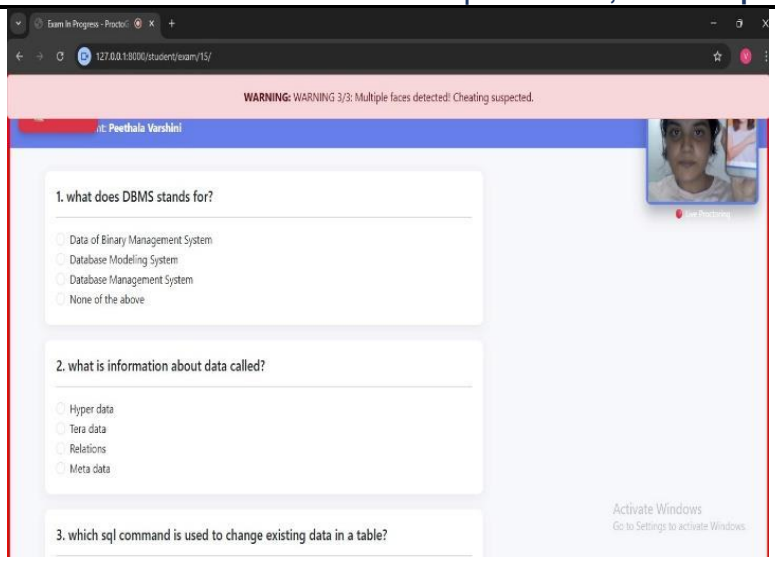

Fig. 8. AI-Based Proctoring and Monitoring

Figure 9 shows the automated result generation interface, displaying the candidate's score, pass/fail status, and AI-generated conceptual explanations for each question [3] inline with the question content. The testing results confirm that ProctoGuard significantly enhances both examination security and learning effectiveness [14], [15], [25].

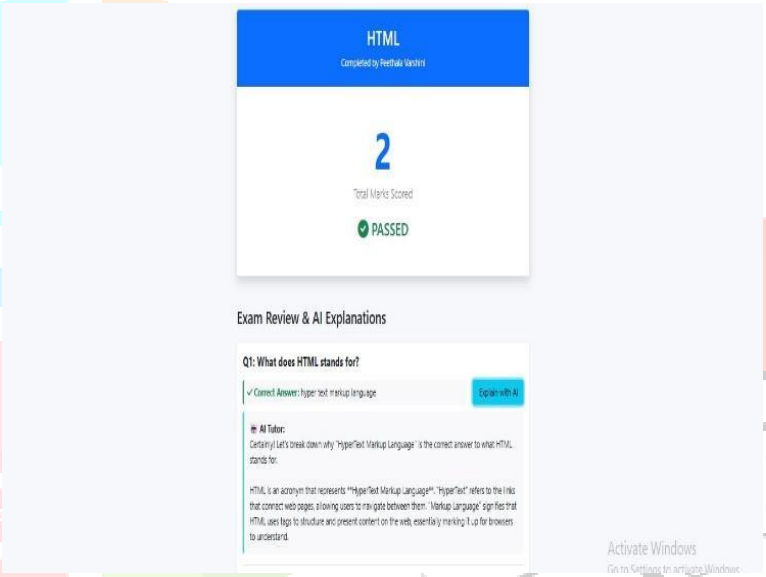

Fig. 9. Automated Result Generation Interface

VII. CONCLUSION

ProctoGuard was developed in response to a genuine and growing problem: the absence of a practical, self-hosted examination platform capable of delivering meaningful supervisory oversight in fully remote assessment settings without imposing the costs, privacy compromises, or infrastructure dependencies associated with commercial proctoring services. The system addresses that problem through an architecture that combines three simultaneous monitoring channels — webcam-based facial analysis, ambient audio classification, and browser-event interception — within a single cohesive Django web application, producing a behavioural record for each examination session that is considerably richer than any single-modality approach could generate [15], [25].

The results obtained during controlled simulation testing validate the core design decisions. An overall violation detection rate of 94.8% across five distinct malpractice categories confirms that the multi-channel monitoring pipeline functions as intended, with browser-event interception achieving perfect accuracy [22] and visual and acoustic channels delivering detection rates between 85% and 97% [12], [14], [18]. The graduated enforcement mechanism — escalating warnings followed by automatic examination finalisation upon threshold breach [16] — was triggered correctly in every session where the violation count warranted it, demonstrating that the rule engine operates with the consistency and immediacy that human invigilation rarely achieves. System response times across all key operations remained within two seconds, confirming that the platform is sufficiently responsive for live institutional examination environments [17].

Beyond its security contributions, ProctoGuard represents a meaningful step toward examination platforms that take an active interest in the candidate's learning trajectory rather than treating the assessment event as a terminal interaction. The AI explanation module [3], powered by the Google Gemini API, produced contextually accurate and conceptually substantive feedback for every question across all evaluated sessions, demonstrating that intelligent post-examination support can be delivered as an embedded platform feature rather than a separate educational tool [19]. This integration reframes the examination from a purely evaluative transaction into a moment that continues to generate educational value after the candidate has submitted their answers.

The limitations identified during evaluation are acknowledged directly. Ambient audio detection exhibited the highest false-positive rate among all monitored channels [20], primarily because household acoustic environments vary in ways that a session-baseline calibration approach can only partially compensate for. Similarly, the current local server deployment model represents a

scalability ceiling that cloud migration would remove. These are engineering challenges with clear solution paths rather than fundamental design flaws, and addressing them is the primary objective of the future development programme described in the following section. Taken together, the evidence presented in this paper supports the conclusion that ProctoGuard constitutes a viable, practically deployable, and educationally meaningful solution to the challenge of conducting honest and productive remote assessments [14], [25].

VIII. FUTURE SCOPE

The version of ProctoGuard described in this paper represents a capable and fully functional first generation of the platform, but the architecture has been deliberately designed to accommodate growth, and several high-value development directions have been identified that would substantially extend the system's capabilities, reach, and educational impact [15], [25].

Semantically Aware Audio Processing. The most immediate priority for technical improvement is the replacement of the current amplitude-and-frequency audio classifier with a model capable of performing genuine speech recognition on the candidate's captured audio [10], [18]. A speech-to-text pipeline operating in real time would transcribe detected speech and apply semantic analysis to determine whether the content constitutes examination-relevant communication, dramatically reducing false positives in noisy environments [20] while simultaneously improving the specificity of genuine violation detection.

Continuous Biometric Identity Verification. Incorporating a face recognition component [8], [11] that maintains an ongoing comparison between the live webcam feed and a reference photograph captured during registration would provide a robust and continuously enforced safeguard against proxy examination [12]. Multi-factor authentication mechanisms [11] could be layered on top of biometric verification at session initiation to create a defence-in-depth approach to candidate identity that significantly strengthens the platform's identity assurance function [8].

Deep Behavioural Analysis Through Gaze and Pose Estimation. Eye-gaze tracking [10] would add the ability to determine not merely whether the candidate is facing the screen but where their visual attention is directed, identifying patterns consistent with consulting a second monitor, printed document, or mobile device. Head-pose estimation would complement gaze tracking by identifying rotational head movements suggesting off-screen resource consultation [13]. Emotion recognition models [18] could further flag states warranting closer review by a human supervisor, collectively reducing false positives and improving overall detection reliability [14].

Cloud Infrastructure and Institutional Scalability. Migrating the system to a cloud infrastructure would remove the scalability ceiling imposed by the current local deployment model [20], enabling horizontal scaling in response to demand spikes, geographically distributed hosting that reduces candidate latency, automated backup and recovery capabilities, and continuous availability guarantees that a locally managed server cannot match [20]. Containerising the application using Docker and orchestrating through Kubernetes would further simplify operational management at institutional scale.

Native Mobile Examination Support. Developing dedicated native applications for Android and iOS would provide optimised camera and microphone integration, a touch-first interface designed specifically for smartphone and tablet form factors, and the ability to lock the device into a single-application mode during an examination session [6] — eliminating an entire class of browser-based evasion strategies and substantially expanding the population of students who can participate in proctored assessments, particularly in regions where smartphone ownership significantly exceeds laptop ownership [5], [6].

Adaptive Question Difficulty and Personalised Assessment. Incorporating an adaptive testing engine that selects subsequent questions based on the accuracy and response latency of preceding answers would create a personalised assessment trajectory for each candidate [5], converging on a difficulty level that accurately reflects demonstrated ability rather than exposing every candidate to a uniform fixed-form test. When combined with the platform's existing AI explanation module [3], an adaptive examination engine would create a learning environment that responds to each candidate's demonstrated knowledge state throughout the session.

Real-Time Supervisory Analytics and Institutional Reporting. A live analytics dashboard surfacing aggregated monitoring data — violation type distributions across active sessions, auto-submission counts, and flags indicating sessions with unusually high violation rates [3] would give supervisors meaningful situational awareness during examination windows. Post-examination analytics extending this dashboard with cohort performance trends, question difficulty indices, and longitudinal individual progress tracking would provide the evidence base educators need to refine assessment design, target instructional support, and satisfy the reporting requirements of accreditation and quality assurance processes.

The cumulative effect of these enhancements — implemented incrementally over successive development cycles — would transform ProctoGuard from a well-functioning first-generation proctoring platform into a comprehensive, intelligent, and deeply integrated assessment ecosystem capable of serving the full range of needs that modern digital education places on examination infrastructure [25].

REFERENCES

[1] P. Prathish, S. Narayanan, A. K. Bijlani, and S. Prathish, "An intelligent system for online exam monitoring," in Proc. Int. Conf. Information Science (ICIS), Kochi, India, 2016, pp. 138–143, doi: 10.1109/INFOSCI.2016.7845315.

[2] IEEE Xplore, "Document 9743134." [Online]. Available: https://ieeexplore.ieee.org/document/9743134

[3] IEEE Xplore, "Document 9873661." [Online]. Available: https://ieeexplore.ieee.org/document/9873661

[4] IEEE Xplore, "Document 9613352." [Online]. Available: https://ieeexplore.ieee.org/document/9613352

[5] P. Bhavitha et al., "Online Examination System," ResearchGate, 2020. [Online]. Available: https://www.researchgate.net/publication/342590840

[6] D. V. Kotwal et al., "Online examination system," IRJET, vol. 3, no. 1, pp. 115–117, 2016.

[7] N. Omoregbe, A. Azeta, A. Adewumi, and A. Oluwafunmilola, "Implementing an online examination system," in Proc. ICERI2015, 2015, pp. 1234–1238.

[8] M. Siddiqui and P. Valsalan, "AI-based human face recognition system," J. Electrical Systems, vol. 20, 2024.

[9] S. Prathish, A. N. S., and K. Bijlani, "An intelligent system for online exam monitoring," in Proc. ICIS, Kochi, India, 2016, pp. 138–143.

[10] Y. M. Cheung and Q. Peng, "Eye gaze tracking with web camera in a desktop environment," IEEE Trans. Human-Machine Systems, vol. 45, no. 4, pp. 419–430, 2015.

[11] N. L. Clarke, P. Dowland, and S. M. Furnell, "e-Invigilator: A biometric-based supervision system for e-assessment," in Proc. IEEE Conf. iSociety, 2013, pp. 238–242.

[12] A. A. Sukmandhani and I. Sutedja, "Face recognition method for online exams," in Proc. ICIMTech, Jakarta/Bali, 2019, pp. 175–179, doi: 10.1109/ICIMTech.2019.8843831.

[13] D. V. Solanki and A. M. K., "A survey on face recognition techniques," J. Image Process. Pattern Recognit. Prog., vol. 4, no. 6, pp. 11–16, 2013.

[14] S. G. Rabiha, I. H. Kartowisastro, R. Setiawan, and W. Budiharto, "Survey of online exam proctoring model to detect cheating behavior based on face recognition," in Proc. ICSAI, Kunming, China, 2022, pp. 1–7, doi: 10.1109/ICSAI57119.2022.10005488.

[15] 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, doi: 10.1109/TMM.2017.2656064.

[16] R. Bawarith, A. Basuhail, A. Fattouh, and S. Gamalel-Din, "E-exam cheating detection system," IJACSA, vol. 8, no. 4, 2017, doi: 10.14569/IJACSA.2017.080425.

[17] S. Bobde, S. Chaudhari, and J. Golguri, "Web based online examination system," GRD J. for Engineering, vol. 2, no. 5, pp. 58–61, Apr. 2017.

[18] B. Yadav and M. Rao, "Malpractice detection in online assessments using eye gaze tracking and object detection," in Rising Threats in Expert Applications and Solutions, Springer, 2022, pp. 701–708.

[19] N. Chandra, P. Sharma, U. Tripathi, U. Kumar, and G. C. B. Prakash, "Automating online proctoring through artificial intelligence," Jan. 2021.

[20] A. Samarin, A. Mamaeva, and A. Toropov, "AI-powered proctoring system with audio and video stream analysis," in Proc. RusAutoCon, 2025, pp. 833–839, doi: 10.1109/RusAutoCon65989.2025.11177261.

[21] A. Wahid, Y. Sengoku, and M. Mambo, "Toward constructing a secure online examination system," in Proc. IMCOM, New York, 2015, Article 95, pp. 1–8, doi: 10.1145/2701126.2701203.

[22] L. Siyao and G. Qianrang, "The research on anti-cheating strategy of online examination system," in Proc. AIMSEC, Deng Feng, China, 2011, pp. 1738–1741, doi: 10.1109/AIMSEC.2011.6010689.

[23] P. Guo, H. Yu, and Q. Yao, "The research and application of online examination and monitoring system," in Proc. IEEE ITME, Xiamen, 2008, pp. 497–502, doi: 10.1109/ITME.2008.4743914.

[24] U. I. B. P., A. Shalini, and A. Yamuna, "Web based online secured exam," IJERA, vol. 2, no. 1, pp. 943–944, Jan.–Feb. 2012.

[25] S. Satre, S. Patil, T. Mane, V. Molawade, T. Gawand, and A. Mishra, "Online exam proctoring system based on artificial intelligence," in Proc. IConSCEPT, Karaikal, India, 2023, pp. 1–6, doi: 10.1109/IConSCEPT57958.2023.10170577.