Gromage Innovation Research Journal (GIRJ)
WWW.GIRJ.ORG I GIRJ : 2950 - 9269
An International Access , Peer – Reviewed , Referred Journal
WWW.GIRJ.ORG

Online Exam System With Real Time Proctoring

Guardian Name – Prof. P.G.Patil

Rohit Sadashiv Pondkule, Anurag Satish Khilare, Onkar Dinkar Hake,

Shivraj Dynaneshwar Kale, Soham Samadhan Pawar,

U.G.Student, SMSMPITR Institute of Technology, Akluj, India
Assistant Professor, SMSMPITR Institute of Technology, Akluj, India

Abstract : In the wake of increasing demand for remote education and assessments, ensuring the integrity of online examinations has become a critical concern. This paper presents the design and implementation of an Online Exam System integrated with Real-Time Proctoring capabilities. The system is built using a modular architecture comprising three core components: the Exam Management Module, the Student Interface, and the Proctoring Module. The platform facilitates seamless exam creation, scheduling, and automated result processing.

INTRODUCTION

With the rapid growth of online learning, conducting secure remote examinations has become a major concern. Traditional exams ensure fairness through physical supervision, but online exams often lack such control. This leads to increased risks of cheating, impersonation, and dishonest practices. To solve this problem, we propose an Online Exam System with Real-Time Proctoring. The system allows students to take exams from home while being monitored throughout the session. It includes features like student verification, randomized questions, and live video and screen monitoring. Real-time proctoring helps identify suspicious behavior in real-time. This enhances transparency and accountability during exams. Our system aims to maintain integrity in remote assessments. It offers a fair secure, and reliable way to conduct online exams.

The shift to digital assessments has created both opportunities and challenges for educators and institutions.
Ensuring exam integrity without physical invigilation is a complex task in virtual environments.
Real-time proctoring provides a practical solution by allowing live supervision through webcams and screen tracking.
This approach minimizes the chances of malpractice and helps in identifying violations during the exam.
In addition to live monitoring, the system logs user activity for further analysis and verification.
It is designed to be user-friendly for both students and administrators, with clear workflows and minimal setup.
The system supports different types of exams, including multiple-choice, descriptive, and time-based questions.
Authentication and monitoring help ensure that only the registered candidate attends the exam.

NEED OF THE STUDY.

The integration of online examination systems with real-time proctoring has become a necessity in the current educational environment. This section explores the detailed need for such a system under various thematic categories.

2.1 Transition to Online Learning

The global shift toward digital learning has increased the adoption of online education platforms. COVID-19 significantly accelerated the move from traditional classrooms to remote learning, exposing the limitations of manual exam monitoring. Institutions faced challenges in assessing students fairly and accurately outside physical settings.

2.2 Limitations of Traditional Online Exams

Existing online exams often lack control mechanisms to ensure academic honesty. Common problems include: Impersonation by unauthorized individuals. Use of unauthorized materials (phones, books, websites). Communication with others during the exam. There is no real-time way to intervene or flag suspicious behavior without supervision.

2.3 Importance of Real-Time Proctoring

Real-time proctoring replicates the invigilator's role in a virtual setting using webcams, microphones, and screen sharing. It enables: 1) Continuous candidate monitoring. 2) Detection of violations during the exam. 3) Immediate alerts for abnormal activities like face absence, background movement, or multiple persons.

2.4 Supporting Academic Integrity

The core of any assessment system is fairness and trust. Academic dishonesty harms institutional reputation and devalues qualifications. Real-time proctoring builds confidence among: Students (that they are fairly assessed). Educators (that learning outcomes are accurately measured). Employers (that certified skills reflect real knowledge).

2.5 Accessibility and Inclusivity

Many students reside in remote or rural areas with limited access to physical test centers. Online exams with remote monitoring: Reduce travel and logistical burdens. Allow students with disabilities or health constraints to participate equally. Promote inclusivity in education systems across regions.

2.6 Preparation for Future Technologies

Real-time proctoring creates a foundation for integrating: Artificial Intelligence (AI) to analyze facial expressions and behaviors. Machine Learning (ML) models to detect patterns of malpractice. Automated proctoring systems to reduce human oversight requirements.

RESEARCH METHODOLOGY

The methodology adopted for this research focuses on the systematic design, development, implementation, and evaluation of an Online Exam System integrated with real-time proctoring capabilities. Given the objective of ensuring academic integrity in online assessments, the methodology is structured around a practical software engineering approach combined with user-centered design principles. This section outlines the methods and strategies used throughout the research, including requirement analysis, system design, development technologies, testing protocols, and evaluation mechanisms.

Understanding the Problem and Defining Requirements

The first step in the research methodology involved identifying the core problem: the lack of integrity in remote assessments due to the absence of supervision. This challenge was amplified during the COVID-19 pandemic when educational institutions worldwide had to adopt online learning. While online platforms facilitated remote learning, the transition exposed vulnerabilities in the assessment process. To address this, it was imperative to develop a solution that not only allows students to take exams online but also ensures they are monitored throughout the process in a way that discourages cheating and upholds fairness.

To gather requirements, structured interviews and questionnaires were conducted with stakeholders including students, teachers, examination coordinators, and IT professionals. These interactions helped uncover practical issues faced in previous online assessments, such as identity fraud, unauthorized collaboration, and system failures. Based on this analysis, key system requirements were documented: secure authentication, real-time webcam and screen monitoring, live flagging of suspicious behavior, automatic result calculation, a responsive and intuitive interface, and scalability to support concurrent users.

Design and Planning

Following the requirement analysis, the system architecture was planned using modern software engineering models. The system was divided into several key components: the user interface, backend services, database, and proctoring engine. UML diagrams including use case, sequence, and activity diagrams were created to visualize the workflow of the application. The Model-View-Controller (MVC) pattern was selected to ensure a clean separation of concerns, facilitating easier debugging, testing, and future upgrades.

Wireframes were designed using tools such as Figma to prototype the dashboards and interfaces for each user role—Administrator, Faculty, Student, and Proctor. Special emphasis was placed on accessibility and usability, ensuring that even users with limited technical knowledge could navigate the system efficiently. Real-time features like live webcam feed, screen sharing, and alert notifications were embedded in the design to simulate a supervised physical exam hall.

Technology Stack Selection

Technology selection was driven by criteria including performance, compatibility, security, ease of development, and community support. For frontend development, React.js was chosen due to its dynamic component rendering, efficient state management, and compatibility with WebRTC APIs required for real-time proctoring functionalities. React's reusable component architecture made it suitable for developing modular, scalable dashboards for different user roles.

The backend was developed using ASP.NET Core, a high-performance framework known for its robust security features and seamless integration with Microsoft SQL Server. ASP.NET Core facilitated the development of secure REST APIs for user management, exam operations, and result processing.

The proctoring module was developed in Python, leveraging libraries such as OpenCV and MediaPipe to enable facial recognition, motion tracking, and object detection. These libraries allowed the system to monitor and log user behavior, such as multiple faces in the frame, frequent gaze diversion, or absence from the camera view. SQL Server was used as the primary database system to handle user data, exam content, and logs, offering strong data integrity and relational capabilities.

Development Process

The system was developed using an Agile approach, with iterative development cycles and continuous feedback from users. Each sprint focused on delivering a functional module, which was then tested and refined before moving on to the next phase. The development cycle included code implementation, version control using Git, peer reviews, and integration testing.

Particular focus was placed on the proctoring component, which involved enabling the system to access a user's webcam and screen with their consent and display both feeds to a remote proctor. The system also included functionality for automatic flagging, where specific behaviors (e.g., face not visible, person absent, background movement) were logged and displayed for manual review.

Testing and Evaluation

Once development reached a stable version, rigorous testing was conducted to ensure functionality, reliability, and security. Unit testing was applied to individual components to validate their correctness. Integration testing ensured that all components interacted seamlessly. User acceptance testing (UAT) was conducted with a sample group of 20 students and 5 faculty members who used the system in a simulated exam environment.

Performance testing was carried out by simulating multiple concurrent exam sessions to assess system stability and resource utilization. Security testing included testing authentication mechanisms, preventing unauthorized access, and verifying data encryption for webcam and screen transmissions.

Following testing, data was collected through feedback forms and usage logs. Feedback was overwhelmingly positive in terms of ease of use, effectiveness of monitoring, and result accuracy. However, some users highlighted issues related to low webcam quality or unstable internet connections, which were noted for future improvement.

Ethical Considerations

Ethical implications were thoroughly considered in the development of this system. All participants involved in testing were informed about data collection, webcam and screen access, and privacy policies. Explicit consent was taken before accessing any personal device features. Recorded media and behavioral logs were stored securely and deleted after analysis. The system design ensured that proctoring was active only during exams and not before or after, respecting user privacy.

The research methodology outlines the approach taken to develop, test, and evaluate the Online Exam System with Real-Time Proctoring. It covers the phases of system development, data collection, technology selection, testing procedures, and evaluation criteria. The detailsare as follows;

3.1 Research Design

The study follows an applied research approach aimed at solving real-world problems in online education. It employs a design and development methodology, combining software engineering practices with user-centered design. Both qualitative and quantitative methods were used to evaluate system performance and usability.

3.2 Objectives of the Methodology

To design a secure, scalable online examination system for remote learners. To integrate real-time proctoring features (webcam, screen sharing, and monitoring). To ensure a user-friendly interface for students, faculty, and administrators. To test the system for reliability, usability, and academic integrity support To analyze user feedback and identify areas for improvement.

3.3 Phases of Methodology

Phase 1: Requirement Analysis

- Conducted discussions with:
  - Faculty members from academic institutions.
  - Technical experts in web development and cybersecurity.
  - Students who have experience with online exams.
- Identified key requirements: 1) Exam scheduling 2) Secure login and authentication 3) Real-time candidate monitoring 4) Result calculation 5)Admin control panel

Phase 2: System Design

- Created flowcharts, wireframes, and UML diagrams to represent system behavior.
- Architecture designed using Modular MVC Pattern for maintainability.
- Developed separate modules for: 1) User management (registration, login) 2) Exam management (question bank, schedule, timer) 3) Proctoring (video feed, screen sharing, flag detection) 4) Result analysis and reporting

Phase 4: System Development

- Used Agile methodology with iterative sprints for rapid prototyping.
- Major tasks included: 1) Building frontend with role-based dashboards (Admin, Student, Faculty). 2) Implementing secured REST APIs for all backend services. 3)Integrating webcam and screen access using browser permissions. 3)Developing a monitoring system to capture and log suspicious behavior.

Phase 6: Deployment and Evaluation

- Deployed on a secure server and conducted live exam sessions.
- Logged metrics such as: 1) Number of users 2)System response time 3)Incidents flagged by the proctoring module
- Collected feedback through surveys and recorded: 1) User satisfaction 2) Suggestions for improvement 3) Issues faced during the exam

3.4 Data Collection Methods

- **Primary Data**: User feedback through Google Forms and in-app surveys. Real-time monitoring logs (video, screenshots, flags).

- **Secondary Data**: Literature review on existing exam systems and security models. Technical documentation of tools used (e.g., OpenCV, React).

3.5 Tools and Platforms Used

Development Tools: Visual Studio, VS Code, SQL Server Management Studio

Testing Tools: Postman, JMeter, Selenium

Design Tools: Figma, Draw.io for diagrams

Monitoring Tools: Custom-built video and screen capture scripts using Python

I. ACKNOWLEDGMENT

Thepreferredspellingoftheword “acknowledgment” inAmericaiswithoutan “e” afterthe “g”.Avoidthestiltedexpression, “Oneofus(R.B.G.)thanks...”
Instead,try“R.B.G.thanks”.Putapplicablesponsoracknowledgmentshere;DONOTplacethemonthefirstpageofyourpaperorasafootnote.

REFERENCES

[1] "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma et al. This book is highly useful for designing the backend architecture of an online examination system using object-oriented principles.

[2] "Design and Implementation of an Online Examination System" by S. K. Sharma. This paper will discuss how online exams are structured, how data is securely transferred between client and server, and how users (students and instructors) interact with the system. Key concepts such as user authentication, data validation, and exam scheduling will be addressed.

[3] "Database Systems: Design, Implementation, and Management" by Carlos Coronel Steven Morris
Provides essential knowledge on database design, which is crucial for building systems to store and manage exam data.

[4] "A Study on Online Examination System: Design and Analysis" – A journal or conference paper that explores challenges and methodologies in creating effective online exam systems. On Google Scholar or through academic databases.