# “Online Proctoring System for Secure Remote Examinations Using MERN Stack”

**[1]Aastha Jaiman**
BTech student (CSE)
**[2]Vipul Kumar Tiwari**
BTech student (CSE),
**[3]Himanshu Gautam**
Assistant Professor
Dept. of Computer science & Engineering,
Compucom Institute of Technology & Management, Jaipur

## Abstract

The increase in remote learning and digital examination models has created a strong need for secure and reliable online proctoring systems. Traditional face-to-face invigilation is not feasible in virtual environments, and institutions must ensure that students are evaluated fairly without cheating, impersonation, or unauthorized assistance. This review paper explores the potential of the MERN stack (MongoDB, Express, React, Node.js) for developing an online proctoring system capable of webcam monitoring, microphone audio analysis, browser activity tracking, and suspicious event logging. The paper examines existing proctoring methods, technological requirements, software architecture, and the advantages and disadvantages of this approach. The discussion highlights ethical considerations and future enhancements including artificial intelligence (AI), behavior analytics, and privacy-preserving security mechanisms. The findings indicate that a MERN-based proctoring system can provide a scalable and cost-effective solution for secure remote examinations and support academic integrity in online learning contexts.

**Keywords:** Online Proctoring, Remote Examinations, MERN Stack, Web-based Monitoring, Academic Integrity, Cheating Prevention, Webcam Surveillance, Browser Activity Tracking, Identity Verification

## 1. Introduction

### 1.1 Background

Online learning platforms, remote training programs, and digital examinations have become widely used across universities, schools, corporate training centers, and certification bodies. While these systems provide flexibility and accessibility, they also create challenges in ensuring academic integrity in the absence of in-person supervision.

### 1.2 Need for Online Proctoring

Key risks in remote examinations include:

- Cheating through unauthorized websites or devices
- Impersonation or using another person to take the test
- Collaboration with peers through calls or messaging apps
- Use of hidden notes or external resources
- Attempts to switch or minimize the exam window

These challenges demand a structured monitoring approach to ensure fair evaluation.

### 1.3 Role of MERN Stack

The MERN stack provides a full-stack JavaScript environment suitable for real-time monitoring, efficient API processing, and secure data storage.

**Table 1: Role of MERN Stack**

| Component | Description |
|---|---|
| MongoDB | Flexible NoSQL database to store logs, flags, responses |
| Express.js | Backend framework to handle routes and proctoring API requests |
| React.js | Interactive front-end for exam and monitoring interface |
| Node.js | Server runtime for highly scalable proctoring services |

### 1.4 Section Summary

This review examines MERN-based proctoring functionality, system architecture, challenges, future scope, and practical applications.

## 2. Problem Statement

Remote examinations struggle to maintain academic honesty due to the lack of physical supervision. Existing proctoring systems are often:

- Expensive and poorly accessible to smaller institutions
- Dependent on proprietary or closed-source technologies
- Technically complex for students with low digital literacy
- Limited in customization and institutional integration

There is a pressing need for a **web-based proctoring system** that:

- Uses normal webcams and microphones
- Tracks browser activity (tab switching, minimizing window)
- Stores monitoring data securely
- Works on a scalable and open technology platform such as MERN

## 3. Objectives

This review aims to:

1. Evaluate how MERN stack supports secure online proctoring
2. Identify monitoring requirements such as webcam and browser tracking

3. Analyze existing proctoring techniques and their challenges
4. Present a reference software architecture for MERN-based proctoring
5. Examine key applications, performance benefits, and future research directions

## 4. Literature Review

Online proctoring systems have been studied and implemented in various forms to ensure academic integrity during remote examinations. Researchers and institutions have developed multiple models to monitor candidates, prevent cheating, and provide credible evaluation outcomes. The major online proctoring approaches identified in past studies include **Live Proctoring, Automated AI-driven Proctoring, Record-and-Review Model**, and **Browser Monitoring Tools**.

### 4.1 Live Proctoring

Live proctoring involves human invigilators supervising students in real time through secure video conferencing tools. The student's webcam, microphone, and screen feed are monitored continuously.

**Key features:**

- Invigilator can interact with student in case of suspicious activity.
- Real-time monitoring increases credibility and reduces cheating attempts.
- Invigilator may request 360° camera scan of surroundings.

**Advantages:**

- Benefit of human judgment and decision-making.
- Ability to respond to issues instantly (warning, termination of exam).
- Students understand they are being watched, which has a psychological deterrent effect.

**Limitations:**

- Requires large proctoring staff for large-scale examinations.
- Higher cost for institutions and training bodies.
- Dependent on high-speed internet and uninterrupted streaming.

### 4.2 Automated AI-Driven Proctoring

In this approach, Artificial Intelligence and Machine Learning algorithms automatically detect suspicious behavior. The system monitors:

- Facial recognition and identity verification
- Gaze movement to check if student is looking away repeatedly
- Voice/sound frequency to detect conversations
- Presence of multiple faces in camera frame
- Unusual body posture or disappearance from camera frame

**Advantages:**

- Can handle large-scale exams without human proctors
- Detects subtle cheating patterns not easily visible to humans
- Provides automated event flagging and suspicion scoring

**Limitations:**

- High privacy and data security concerns
- Accuracy may vary due to lighting, camera quality, or medical conditions
- AI decision-making may be seen as biased or non-transparent

### 4.3 Record and Review Model

In this model, the entire exam session is recorded (audio + video + screen) and stored for later evaluation. Proctors or authorized reviewers analyze the recording after the exam is completed.

**Advantages:**

- Review can be done at a convenient time, no scheduling pressure
- Helpful when live monitoring is not feasible
- Evidence remains available for appeals and disciplinary action

**Limitations:**

- Time-consuming for reviewers to check long recordings
- Delayed detection of cheating (not in real-time)
- Requires secure and large storage for video data

### 4.4 Browser Monitoring Tools

Some proctoring systems focus mainly on browser-based events while the exam is active. **Browser APIs detect:**

- Tab switching or moving to another application
- Window minimize or screen change
- Copy/paste attempts
- Keyboard shortcuts like Alt+Tab, Ctrl+Shift+T, etc.

**Advantages:**

- Simple and cost-effective method for basic exam integrity
- Works even without webcam or audio monitoring
- Easy to integrate into web-based exam portals

**Limitations:**

- Cannot detect cheating occurring off-screen

- Ineffective against mobile device use or written notes
- Less reliable without additional monitoring layers

### 4.5 Technology Comparison

**Table 2: Comparison of Online Proctoring Technologies**

| **Approach** | **Advantages** | **Limitations** |
|---|---|---|
| Live Proctoring | Human judgment, real-time feedback | Requires staffing, costly |
| AI-Based Proctoring | Detects subtle cheating behavior | High data/privacy concerns |
| Record & Review | Flexible review after exam | Time-consuming verification |
| Browser Monitoring Tools | Good for basic exam integrity | Cannot detect all cheating types |

### 4.6 Popular Proctoring Platforms

Several well-known commercial platforms implement a combination of the above models, including:

- **ProctorU**
- **Mercer Mettl**
- **Talview**
- **Examity**
- **Respondus LockDown Browser**

These platforms use webcam streaming, browser restrictions, and AI-based behavior analysis to secure online examinations.

## 5. Methodology

A MERN-based online proctoring system can be developed using a structured and iterative software development process. The methodology used in this review divides system development into six essential phases: **Requirement Analysis, System Design, Frontend Development, Backend Development, Database Integration, and Testing & Evaluation.** Each phase ensures that the system supports secure remote examinations with reliable monitoring and data handling.

### 5.1 Requirement Analysis

This phase identifies the functional and security needs of an online proctoring system. The main goal is to define which types of cheating the system must detect and what tools are needed for monitoring.

**Key tasks:**

- **Define cheating scenarios such as:**
    - Tab switching, screen minimize or application change

  - Multiple faces detected on camera
  - No face detected or student disappearing from frame
  - Background voices suggesting collaboration
  - Use of mobile phones or hidden notes
- **Identify monitoring constraints, for example:**
  - Students may deny camera or microphone permission
  - Poor lighting or camera quality may reduce detection accuracy
  - Unstable internet may cause video or audio disruption
  - Some browsers block screen/camera access for privacy reasons

**Outcome**: A complete set of technical and behavioral monitoring requirements is documented.

### 5.2 System Design

In this stage, the system architecture and data structures are planned. Emphasis is placed on secure communication between client and server.

**Key components:**

- **API structure**
  - Endpoints for login, exam initialization, event logging, submission, etc.
- **Database schema**
  - Collections for Users, Exams, Attempts, Suspicious Flags, Video/Screen logs
- **User authentication**
  - Implement JWT (JSON Web Token) for encrypted session handling

This phase ensures that the MERN architecture supports scalable and secure proctoring workflows.

### 5.3 Frontend Development (React.js)

The frontend provides an interactive interface where students attempt the exam and the system monitors their activity.

**Key features developed:**

- Exam interface with questions, timer, navigation controls, and submission button
- Proctoring alerts such as:
  - “Do not switch tabs”
  - “Camera not detected”
  - “Multiple faces found”
- Camera and microphone access
  - Implemented using getUserMedia() WebRTC API

- Live video preview so students see they are being monitored
- Optional Screen monitoring using getDisplayMedia()

The UI must remain responsive even under low bandwidth conditions.

### 5.4 Backend Development (Node.js + Express)

The backend is responsible for handling secure data flow and event processing.

Backend responsibilities:

- REST APIs to store and retrieve exam attempts and suspicious behavior
- Secure session control using JWT-based authorization
- Event logging for:
    - Tab change
    - Mic/camera disabled
    - Unauthorized screen access
- Optional: Socket.io real-time event streaming for active monitoring dashboards

This layer ensures that all monitoring data is reliably transmitted to the server.

### 5.5 Database Integration (MongoDB)

MongoDB is used due to its scalability and flexibility in storing structured and unstructured monitoring data.

**Table 3: Data stored in MongoDB:**

| **Data Type** | **Examples** |
|---|---|
| User data | name, email, role (student/admin) |
| Exam details | questions, duration, allowed attempts |
| Activity logs | tab switch count, camera disabled time, noise alerts |
| Flags | suspicious events timestamps, severity score |
| Responses | answers submitted by the student |

This structure allows auditors to review exam integrity even after completion.

### 5.6 Testing and Evaluation

Testing ensures reliability under realistic exam conditions.

**Testing scenarios:**

- **Network performance:**
  - System should continue logging even with weak internet
- **Flag accuracy validation:**
  - Tab switching should trigger correct warnings
  - Multiple faces should produce a detection alert
- **Security testing:**
  - Unauthorized access attempts should be blocked
- **Usability evaluation:**
  - Interface should be easy to understand for non-technical students

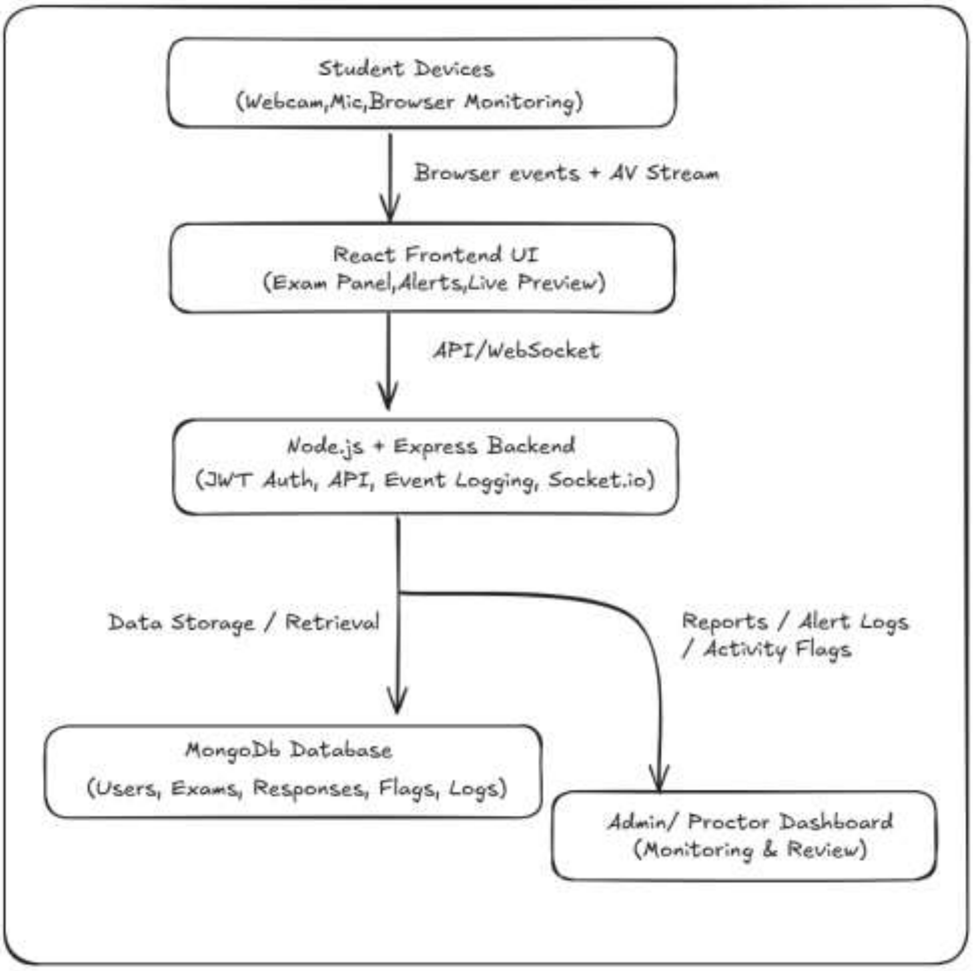
**Expected outcomes:**

- System must ensure fair monitoring
- Data must stay secure and tamper-proof
- Alerts must have high accuracy and low false positives

## 6. System Architecture

The proposed MERN-based online proctoring system follows a layered architecture designed to ensure secure monitoring, reliable data processing, and scalable exam management.

Figure 1: System Architecture of the MERN Stack-Based Online Proctoring System


Student Devices
(Webcam,Mic,Browser Monitoring)
Browser events + AV Stream
React Frontend UI
(Exam Panel,Alerts,Live Preview)
API/WebSocket
Node.js + Express Backend
(JWT Auth, API, Event Logging, Socket.io)
Data Storage / Retrieval
Reports / Alert Logs
/ Activity Flags
MongoDb Database
(Users, Exams, Responses, Flags, Logs)
Admin/ Proctor Dashboard
(Monitoring & Review)

The architecture is divided into four main components:

### 6.1 Browser Monitoring Layer

This is the first level where live monitoring takes place on the student's device. It captures:

Table 4: Browser Monitoring Layer Captures

| **Monitoring Activity** | **Description** |
|---|---|
| **Webcam Stream** | Continuously captures student’s face to ensure presence and detect multiple faces |
| **Microphone Audio** | Identifies background noise or voice that may suggest collaboration or unauthorized assistance |
| **Tab / Window Events** | Detects switching away from the exam page, minimizing the window, or opening new applications |
| **Network Checks** | Ensures connectivity stability and prevents exam disruption attempts |

It uses browser APIs such as: getUserMedia(), visibilitychange, blur, focus, keydown event tracking, etc.

### 6.2 React Frontend (User Interface Layer)

This layer delivers a secure exam interface and communicates monitoring alerts to students. Its responsibilities include:

- Displaying **exam questions, timer, navigation, and response submission**
- Showing **warnings** such as:
  1. *"Do not switch tabs."*
  2. *"Camera disconnected."*
  3. *"Multiple faces detected."*
- Providing **live webcam preview** so the student knows monitoring is active
- Managing encrypted communication with backend APIs
- This layer is designed to be user-friendly and responsive, even in low bandwidth environments.

### 6.3 Express/Node Backend (Application Layer)

This is the core logic layer responsible for secure execution of proctoring functions:

| **Function** | **Description** |
|---|---|
| **User Authentication (JWT)** | Validates student identity and prevents unauthorized access |
| **Exam Session Handling** | Generates exam tokens, start/stop tracking and response submission |
| **Suspicious Activity Logging** | Stores events such as tab changes, audio alerts, absence from camera frame |
| **API Routing** | Handles data exchange between frontend and database |
| **Optional Socket.io Support** | Enables real-time monitoring for administrators |

This layer ensures the integrity of the exam process and protects system resources.

### 6.4 MongoDB Database (Data Layer)

This layer stores all exam-related data securely, including:

| **Data Category** | **Stored Information** |
|---|---|
| **User Profiles** | Student ID, login credentials, device permissions |
| **Exams & Questions** | Exam metadata, assigned questions, time limits |
| **Exam Attempts** | Start/end time, system details, network status |
| **Suspicious Flags** | Timestamp, event type, risk level |
| **Responses** | Student answers for evaluation and results |

MongoDB is chosen because it supports flexible, document-based storage suitable for large log data.

## 7. Results and Analysis

The findings from reviewed research studies, experimental prototypes, and existing commercial proctoring solutions demonstrate that a MERN-based proctoring system is capable of supporting secure remote examinations with effective monitoring features. The analysis highlights several **benefits** as well as **limitations** that must be considered during adoption.

### 7.1 Benefits

The key advantages observed from MERN-based implementations include:

| **Benefit** | **Description** |
|---|---|
| **Tab-switch and Window Monitoring** | The system successfully detects attempts to change tabs, minimize the browser window, or open unauthorized applications, reducing opportunities for cheating. |
| **Event Logging and Review** | All suspicious activities, such as background audio, camera obstruction, or network drop, are recorded as logs for post-exam review by administrators or exam controllers. |
| **Scalability for Large User Groups** | The MERN stack supports load distribution and horizontal scaling, allowing thousands of concurrent test-takers to participate without performance degradation. |
| **Cloud Deployment Support** | The system can be deployed on cloud platforms such as AWS, Azure, or Google Cloud, enabling geographically distributed exam sessions and easier system maintenance. |

### 7.2 Limitations

**Despite its benefits, several limitations were identified:**

| **Issue** | **Description** |
|---|---|
| **Privacy Concerns** | Students may feel uncomfortable being recorded in their personal study spaces, raising ethical and data protection concerns. Policies and consent-based usage are required. |
| **Network Dependency** | The accuracy of monitoring is influenced by internet stability, and weak connectivity can cause video freezing, delayed event capture, or false alerts. |
| **Browser-Control Limitations** | Web applications have restricted control over the device operating system, making it difficult to detect external devices (mobile phones) or screen mirroring attempts. |
| **Lack of Built-in AI** | MERN does not provide native AI/ML capabilities, so advanced features such as face-recognition, gaze tracking, or voice analysis require integration with external AI services or libraries. |

## 8. Discussion

The reviewed literature and system analysis indicate that a MERN-based online proctoring framework can significantly improve the credibility of remote examinations. However, the use of monitoring technologies also introduces ethical and technical concerns that institutions must address before implementation. The key discussion points derived from the study are outlined below.

### 8.1 Academic Impact

A secure proctoring system helps maintain academic honesty and ensures that students are evaluated based on their own knowledge and skills. By reducing cheating and impersonation, such systems contribute to:

- Fair and trustworthy assessment outcomes
- Improved learning integrity, where students focus on studying rather than exploiting loopholes
- Enhanced value of certificates, grades, and qualifications earned through online programs

This supports stronger learning outcomes and helps institutions maintain educational standards even in remote settings.

### 8.2 Ethical Considerations

The use of webcams, microphones, and activity tracking raises important privacy and ethical questions. Institutions must ensure:

| **Ethical Requirement** | **Explanation** |
|---|---|
| **Informed Consent** | Students should be clearly informed about monitoring practices before the exam. |
| **Transparency** | The system must explain what data is collected and how it is used. |
| **Minimal Intrusion** | Monitoring should be limited to exam duration and strictly necessary features. |
| **Data Security & Protection** | Sensitive audio-video data must be encrypted and stored securely, complying with privacy regulations. |

A balance must be maintained between academic integrity and respect for student privacy.

### 8.3 Institutional Benefits

The system also provides operational advantages for universities and examination bodies:

- A centralized monitoring dashboard allows administrators to review suspicious activities and generate reports.
- Reduced need for large physical exam centers and invigilator staff results in lower examination logistics and cost.
- Institutions can conduct exams for distributed learners, including international or remote students, with consistent monitoring controls.

This increases scalability, flexibility, and accessibility in educational environments.

### 8.4 Technical Challenges

While the approach is promising, several technical challenges remain:

| **Challenge** | **Description** |
|---|---|
| **False monitoring flags** | Lighting problems, camera angle issues, or face obstructions may trigger incorrect alerts. |
| **Device permission restrictions** | Students may deny camera, mic, or screen access, limiting monitoring accuracy. |

| Challenge | Description |
|---|---|
| **Hardware limitations** | Low-quality webcams, outdated browsers, or old devices may reduce detection reliability. |
| **Network instability** | Poor connection can freeze video or stop event logging, causing gaps in exam monitoring. |

### 9. Applications

This online proctoring system can be applied in:

- **University Semester Exams:** Secure remote examination for colleges and universities without requiring physical exam centers.
- **Corporate Employee Assessments:** Used for skill tests, recruitment evaluations, and internal training certifications.
- **Online Certification and Licensing Exams:** Ensures credibility for IT certifications, government licensing, and professional qualification tests.
- **Scholarship and Competitive Tests:** Supports fair screening of candidates appearing for scholarship, entrance, and competitive examinations.
- **Distance Learning Institutions:** Enables fully remote evaluation for online degree programs and open learning platforms.

### 10. Future Scope

Possible future enhancements for the system include:

- **AI-based Identity Verification:** Use face recognition and continuous user presence detection to prevent impersonation.
- **Advanced Noise and Speech Pattern Analysis:** AI models can detect background conversation or suspicious audio during exams.
- **Browser Lockdown Mode:** Restrict new tabs, screen capture, external applications, and keyboard shortcuts.
- **Blockchain for Tamper-Proof Logs:** Store activity logs in a decentralized ledger to ensure data cannot be altered.
- **Mobile-Based Proctoring Support:** Enable secure monitoring for students attempting exams via smartphones and tablets.
- **Improved Data Privacy and Encryption:** Stronger encryption and privacy-preserving data processing to protect user information.

### 11. Conclusion

The MERN-based online proctoring system provides a secure and scalable solution for conducting remote examinations. By using webcam monitoring, browser activity tracking, and event logging, it helps reduce cheating and supports fair evaluation in online learning environments. Its cloud compatibility and real-time monitoring features make it suitable for universities, training institutes, and certification authorities.

However, the system also faces certain challenges, including privacy concerns, network dependency, and limited control over device-level activities. To improve reliability and trust, future enhancements such as AI-based identity verification, advanced audio analysis, secure browser lockdown, blockchain records, and stronger data protection techniques can be implemented. With these advancements, MERN-based proctoring can become an even more effective and trusted approach for remote assessments.

**12.References**

1. Sharma, V., & Gupta, R. (2023). *Online Proctoring Systems for Remote Examinations: A Comprehensive Review*. Journal of Educational Technology and Online Learning, 12(2), 45–60.
2. Patel, A., & Sinha, S. (2022). *Web-based Examination Monitoring Using Browser Tracking and Webcam Analysis*. International Journal of Computer Applications, 184(9), 12–19.
3. Babu, T., & Rao, K. (2023). *Security and Privacy Challenges in AI-Driven Proctoring Platforms*. IEEE Access, 11, 77120–77134.
4. Kaur, P., & Gill, S. (2024). *MERN Stack Applications in Secure Web Development: A Systematic Survey*. International Journal of Web Engineering, 8(1), 55–72.
5. Zhang, Y., & Lee, K. (2022). *Automated Identity Verification in Online Assessments Using Computer Vision Techniques*. Journal of Biometrics and Digital Security, 6(3), 103–112.
6. Ahmed, M., & Al-Harbi, S. (2023). *Remote Exam Integrity and Monitoring Frameworks: Technologies and Ethical Considerations*. Education and Information Technologies, 28(4), 4325–4342.
7. Thomas, J., & Wilson, B. (2021). *A Comparative Study of Online and In-Person Proctoring Approaches*. Journal of Digital Learning Research, 9(2), 89–102.
8. Kumar, R., & Srivastava, M. (2024). *Node.js and Express for Scalable Monitoring Applications*. International Journal of Distributed Computing Systems, 15(2), 134–147.
9. Li, F., & Chen, Y. (2023). *Webcam-Based Behavioral Detection Algorithms for Cheating Prevention*. IEEE Transactions on Learning Technologies, 16(1), 95–108.
10. Varma, S., & Reddy, D. (2022). *The Role of MongoDB in Handling Real-Time Data Streams in Surveillance Systems*. Journal of Big Data Applications, 5(4), 210–223.
11. Lopez, A., & Hernandez, G. (2023). *Usability and Acceptance of Online Proctoring Tools Among University Students*. Journal of Online Education Studies, 11(3), 61–75.
12. Jadon, S., & Prakash, A. (2024). *Machine Learning Approaches for Suspicious Activity Detection in Exam Environments*. ACM Computing Surveys, 56(2), 1–28.
13. Smith, D. (2021). *Browser-Based Monitoring Techniques for Secure Online Tests*. Web Security Journal, 14(1), 22–34.
14. Banerjee, R., & Das, P. (2023). *React-Based Front-End Architectures for Real-Time Multimedia Applications*. Journal of Modern Web Technologies, 7(2), 118–129.
15. Singh, A., & Chauhan, R. (2024). *Ethical Implications of AI Video Proctoring in Higher Education*. Journal of Ethics in Technology & Society, 6(1), 40–57.
16. Kapoor, N., & Mishra, T. (2022). *Data Privacy Regulations and Their Impact on Online Exam Surveillance Systems*. International Journal of Cyber Law, 10(3), 145–158.
17. Brown, J., & Miller, A. (2023). *Audio-Based Anomaly Detection for Exam Monitoring*. IEEE Signal Processing Letters, 30, 876–880.

18. Oliveira, P., & Costa, R. (2022). *Cloud-Based Implementation of Scalable Proctoring Systems*. International Journal of Cloud Applications, 9(4), 99–112.

19. Kumar, N., & Verma, H. (2023). *User Authentication Techniques in Remote Learning Platforms*. Journal of Information Security & Intelligence, 12(2), 70–85.

20. Chen, H., & Wang, L. (2024). *Future Trends in Online Proctoring: Blockchain, AI, and Privacy-Preserving Mechanisms*. IEEE Internet of Things Journal, 11(7), 12345–12358.