

Online Proctoring System - README

Project Overview

Our **Online Proctoring System** is designed to **enhance exam integrity** by tracking student behavior **without intrusive webcam monitoring**. The system **monitors and assigns a risk score** based on predefined suspicious activities, allowing teachers to identify potential cheating.

Features

- **Real-time student activity tracking**
- **Event-based risk calculation** using predefined scoring
- **Instructor dashboard** with dynamic risk meters
- **Customizable risk thresholds**

Risk Calculation Process

The system calculates a **composite risk score** based on predefined suspicious actions in `monitoringService.js`:

Event	Risk Points
Tab switching	25 pts
Fullscreen exit	20 pts
Copy/paste attempts	25 pts
Suspicious keyboard shortcuts	15 pts

Formula Example:

- Student **switches tabs (25 pts), exits fullscreen (20 pts), and uses a shortcut (15 pts)**:
 - **Total Risk Score = 25 + 20 + 15 = 60 (High Risk)**

Technology Stack

- **Frontend**: React.js
- **Backend**: Node.js + Express.js
- **Risk Calculation Logic**: JavaScript (`monitoringService.js`)
- **Database**: MongoDB / PostgreSQL

System Architecture & Workflow

1. **Student starts the exam** → The system **monitors user actions** in the background.
2. **System captures suspicious events** → If the student **switches tabs, exits fullscreen, or uses shortcuts**, these events are logged.
3. **Risk score is calculated in `monitoringService.js`** → The system assigns predefined **risk values** to each detected event.

4. **Risk score is stored in the database** → All events and calculated scores are saved for review.
5. **Instructor dashboard flags high-risk students** → Teachers **review flagged students** and decide on necessary actions.

Installation & Setup

1. Clone the Repository

```
git clone [repository_url]
cd online-proctoring-system
```

2. Install Dependencies

```
npm install # For Backend
cd frontend && npm install # For Frontend
```

3. Start the Application

Backend (Node.js Server)

```
npm start
```

Frontend (React.js UI)

```
cd frontend
npm start
```

Future Enhancements

- AI-based risk prediction and anomaly detection
- Advanced analytics and reports for student behavior
- Integration with LMS platforms (Moodle, Blackboard, etc.)

Conclusion

This system provides a **non-intrusive, scalable, and effective exam monitoring solution**, ensuring fairness and integrity.