QuizProctor:

# A Secure Online MCQ Test Platform

## 1. Project Overview

QuizProctor is a web-based platform designed to facilitate secure MCQ (Multiple Choice Question) tests. The platform includes two primary roles: Admin and Student. Admins can create tests, generate shareable test links, and monitor results, while students take the tests with enhanced proctoring features to prevent cheating.

## 2. Features and Roles

### Admin Features

- Create and manage MCQ tests.

- Generate and share secure test links with students.

- Monitor student submissions and view test results.

### Student Features

- Take MCQ tests within a secure environment.

- Real-time tab-switch detection to prevent cheating.

- Auto-submission if students navigate away from the test.

## 3. Implementation Details

### Frontend

The frontend will be built using React.js and styled with Tailwind CSS for a responsive user interface. Proctoring features like tab-switch detection will be implemented using JavaScript.

### Backend

The backend will be built using Node.js and Express.js to handle API requests. It will manage user authentication, test creation, and submission handling.

### Database

MongoDB will be used to store user details, tests, and student submissions.

## 4. Site Structure and User Flow

### Admin Flow

1. Login or register as an Admin.

2. Create a new test with title, questions, and options.

3. Generate and share the test link with students.

4. Monitor test submissions and download results.

### Student Flow

1. Access the test link shared by the Admin.

2. Login or register as a Student.

3. Take the test while being monitored for tab-switching.

4. Submit the test and view the results.

## 5. Wireframes

\*\*Admin Dashboard\*\*: Displays options to create tests, view results, and manage tests.

\*\*Student Test Page\*\*: Displays the test with questions, options, and a timer.

## 6. Technology Stack and Code Examples

Frontend: React.js, Tailwind CSS

Backend: Node.js, Express.js

Database: MongoDB

Authentication: JWT (JSON Web Tokens)

### Example Code: Tab-Switch Detection

document.addEventListener("visibilitychange", () => {  
 if (document.hidden) {  
 alert("You switched tabs! The test will be auto-submitted.");  
 // Auto-submit test logic here  
 }  
});

## 7. Future Enhancements

- Implement video proctoring using WebRTC.

- Add AI-based cheating detection.

- Provide detailed analytics on student performance.