

**Government College of Engineering Kannur**  
**Department of Computer Science and Engineering**  
**CSD334: Mini-Project**

**Project Proposal**

Proposal Prepared By			
Sl No.	KTU Reg No.	Name of Student	Signature of Student
1	KNR23CS005	Abhiram K	
2	KNR23CS041	Mohammed Shahmel A N P	
3	KNR23CS043	Muhammad Rashad Hashim	
4	KNR23CS046	Najih V	

1. Project Title : **FocusFlow: AI-Proctored Exam & Learning Management System**

2. Problem Statement

Existing online examination and learning management systems (LMS) suffer from a significant lack of integrity and flexibility. Traditional platforms often fail to provide adequate safeguards against academic dishonesty during remote assessments, relying heavily on intrusive, high-bandwidth human proctoring or disparate third-party tools. Furthermore, current systems lack integrated support for technical assessments, forcing institutions to use separate platforms for multiple-choice questions (MCQs) and practical coding tests.

Our project aims to bridge this gap by developing a robust, unified web application that combines a secure exam environment with lightweight, client-side AI proctoring and an integrated coding sandbox.

3. Solution approach

Our proposed solution is a feature-rich web application designed to maintain exam integrity while providing a modern interface for both students and instructors. Below are the key features and functionalities.

(a) **User Registration and Role-Based Access**

- Secure authentication using Sessions and Cookies (Passport.js).
- Distinct roles for Students (exam taking) and Instructors (exam creation).

(b) **Exam Builder Dashboard**

- Instructors can create dynamic exams with support for rich text and images.
- A bulk-upload feature allows for efficient management of large question banks.

(c) **Integrated Coding Sandbox**

- A built-in code editor (Monaco Editor) allows students to write and compile code within the browser.
- Supports real-time syntax highlighting and output generation for technical assessments.

(d) **AI-Based Proctoring**

- A client-side ML module monitors user behavior (e.g., looking away, leaving the tab) via webcam.
- Generates “Flag Events” to notify instructors of suspicious activity without storing video streams.

(e) **Student Analytics**

- Post-exam visualization provides students with performance insights.
- Comparative graphs display individual scores against class averages.

(f) **Database Component**

- A centralized NoSQL database stores user profiles, exam papers, question banks, and submission logs.
- Efficiently handles hierarchical data structures like nested questions and code submissions.

4. Does any other solution already exist for the stated problem? If yes give details.
- Yes, there are existing solutions for online assessment, but they face distinct limitations.
    - (a) Generic Form Builders (e.g., Google Forms)
      - Limitations
        - \* Lack of proctoring, no integrated coding environment, and limited security features suitable for formal grading.
    - (b) Enterprise Proctoring Services
      - Limitations
        - \* Extremely high cost, privacy concerns regarding data storage, and heavy bandwidth requirements for live video streaming.

A tailored MERN-stack application can address these cost and privacy issues effectively.

5. In what way is your proposed solution better than the existing solution(s)?

Our proposed solution offers several strategic advantages:

(a) **Unified Ecosystem**

- Unlike fragmented tools where coding happens in one tab and submission in another, Focus Flow integrates MCQs and Coding in a single interface.

(b) **Privacy-First Proctoring**

- By running the ML model (MediaPipe) on the client side, we ensure that video feeds are processed locally and never sent to a server, addressing major privacy concerns.

(c) **Accessibility & Performance**

- Built as a responsive web application, it ensures fast load times and a seamless user experience across devices.

(d) **Detailed Analytics**

- Provides granular insights into student strengths and weaknesses, going beyond simple score reporting.

6. List out the stakeholders of the proposed project.

(a) **Students**

- The primary users who take exams and review their learning progress.

(b) **Instructors/Faculty**

- Responsible for designing assessments, managing question banks, and reviewing integrity reports.

(c) **Educational Institutions**

- The organizations deploying the system to maintain academic standards during remote learning.

7. What will be the impact of this project work on society directly or indirectly?

The impact of a secure and integrated learning platform is multifaceted.

(a) **Direct Impact**

- **Academic Integrity:** Ensures fair assessment by deterring malpractice through intelligent monitoring.
- **Skill Development:** The coding sandbox encourages practical programming practice rather than rote memorization.
- **Efficiency:** Automates grading and result generation, saving valuable faculty time.

(b) **Indirect Impact**

- **Digital Literacy:** Familiarizes students with modern, computer-based testing environments used in the industry.

- **Resource Conservation:** Reduces paper usage and logistical overhead associated with physical examinations.

8. What technologies do you plan to use for the front-end and back-end?

(a) **Front-end (Client-side) :**

- **React.js:** A library for building dynamic user interfaces. It manages the complex state of the exam timer and interactive question components.
- **Bootstrap 5:** A CSS framework used to ensure a responsive, mobile-first design and consistent layout using the grid system.

(b) **Back-end (Server-side) :**

- **Node.js:** The runtime environment enabling JavaScript execution on the server, handling high-concurrency requests efficiently.
- **Express.js:** A framework for building the RESTful API, managing routes, and handling middleware for authentication (Passport.js).

(c) **Database :**

- **MongoDB Atlas:** A cloud-based NoSQL database used to store flexible documents for Exams, Users, and Results. It aligns perfectly with the JSON data structure used in the application.

(d) **Machine Learning (Client-side) :**

- **MediaPipe:** A lightweight, pre-trained framework utilized for real-time, privacy-preserving face detection and gaze tracking. It runs efficiently in the browser, eliminating the need to train or deploy complex TensorFlow models manually.

Together, these technologies form the MERN stack, augmented by lightweight client-side AI, providing a comprehensive solution for modern web-based assessments.

Proposal Approved By		
Name of Guide	Signature of Guide	Date of Approval
Dr. Bindu P V		