# **System Features**

1. Face Authentication: To verify their identity and improve security by preventing unwanted access, users submit to facial recognition during login.
2. Constant Face and Voice Monitoring: To ensure the integrity of the test, the system keeps a close eye on users' faces and voices throughout. This helps to identify any questionable behavior.
3. Admin Dashboard: Exams can be easily created, edited, and managed by administrators through a centralized dashboard.
4. Student Interface: A user-friendly interface makes it easy for students to access examinations, read questions, answer them, and submit their responses. This expedites the exam-taking process.
5. Intuitive Design: An easy-to-use interface makes navigating and interacting with the system effortless and fun for both students and administrators.
6. Accessibility: The system is made to be usable on a variety of screens and devices, meeting the needs of a wide range of users.
7. Adjustable Exam Settings: Administrators possess the ability to personalize exam parameters, including time restrictions, question kinds, and scoring standards, enabling customized evaluations to fulfill certain learning goals.
8. Detailed Reports: Admins can generate comprehensive reports on exam results, student performance, and system usage, facilitating data-driven decision-making and evaluation.
9. Feedback Mechanisms: Students receive feedback on their exam performance, including scores, correct/incorrect answers, and areas for improvement, promoting self-assessment and learning.

## **Hardware Requirements**

Laptops that allow voice and face authentication with a webcam and microphone.

Sufficient processing power and memory to manage the computational demands of algorithms for speech and facial recognition.

## **Software Requirements**

Frontend: Web browsers that support contemporary web technologies and are compatible (Chrome).

- Frontend development frameworks and libraries for HTML, CSS, and JavaScript (React.js).

Backend: Python is a server-side programming language used for developing backend applications.

- Web frameworks (Flask) for routing and managing HTTP requests.

- MySQL database management system for storing and retrieving data.

## **Libraries for authentication and monitoring**

- OpenCV.js, a face recognition library, which allows for continuous facial surveillance during tests and face authentication during login.

- WebRTC, a voice recognition library, for voice recording during tests

.