

# **INTERNSHIP PROJECT-11 DOCUMENT**

## **ON “Online Proctoring System(JAVA)”**

**Submitted by:**

**Navpreet Singh (INTERN)**

**Submitted To:-**

**Kanduri Abhinay (FOUNDER)**

**RITHIN VERMA (CTO)**

### **INDEX**

- 1. INTRODUCTION**
- 2. SOFTWARE REQUIREMENTS**
- 3. DESIGN**
- 4. INPUT**
- 5. IMPLEMENTATION**
- 6. TESTING**
- 7. ADVANTAGE**
- 8. CONCLUSION**
- 9. REFERENCES**

## INTRODUCTION:-

The Online Exam Proctoring System is a web-based solution that enables secure, remote assessment of students. It offers user authentication (signup/login), monitors exams for suspicious activity, and generates cheating alerts in real time. Designed to support modern education, the system leverages webcams and (future) machine learning models to enhance exam integrity, aiming to minimize cheating and ensure fair evaluations.

## SOFTWARE REQUIREMENTS

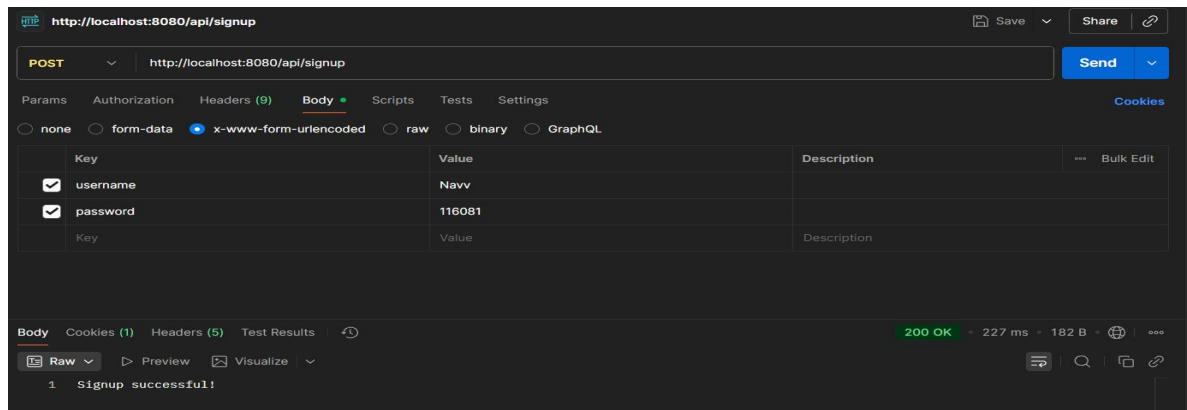
- **Operating System:** Windows, Linux, or macOS
- **Java Version:** Java 17 or newer
- **IDE:** IntelliJ IDEA, Eclipse, or VS Code
- **Framework:** Spring Boot
- **Build Tool:** Maven
- **API Testing Tool:** Postman
- **Libraries:** spring-boot-starter-web, webcam-capture (for desktop image capture)
- **Web Browser:** For accessing web endpoints (for frontend extension)
- **Hardware:** Standard webcam for proctoring, 4GB+ RAM, dual-core CPU

## DESIGN:-

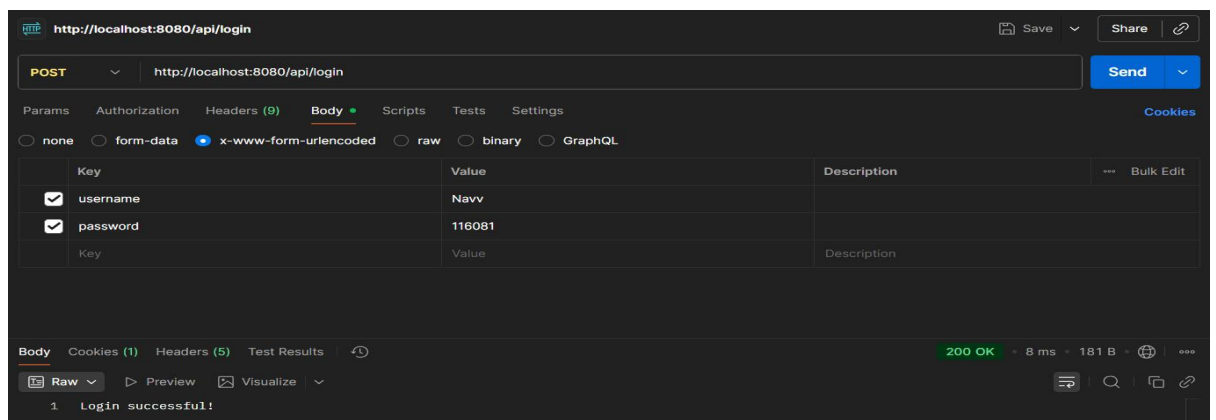
The system uses a modular design, featuring RESTful web services for login, signup, and cheating alert endpoints. A simple in-memory user and alert store is managed in the backend. For webcam integration, a desktop Java application captures images; these can later be sent to a future ML API for automated cheat detection. The code structure maintains separation between models and controllers to improve maintainability and future expansion.

## INPUT

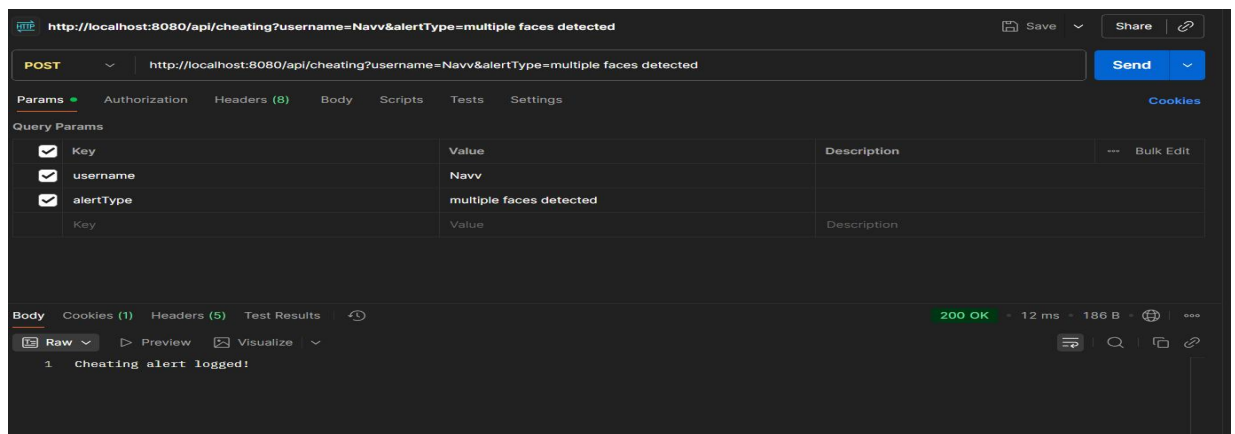
### 1. SIGNUP:-



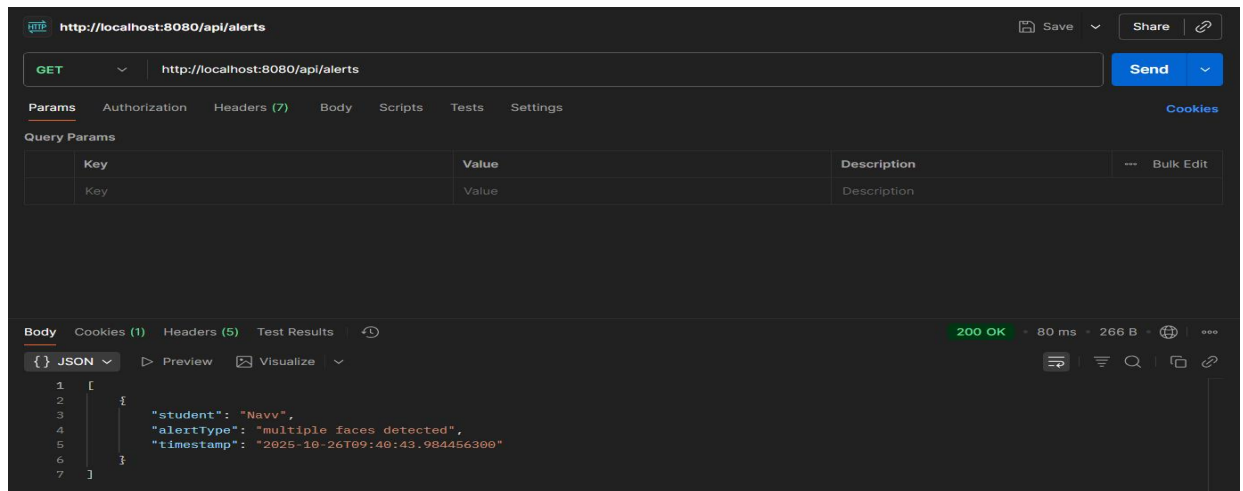
## 2. LOGIN:-



## 3. CheatingAlert:-



## 4. ALERTS:-



## IMPLEMENTATION:-

- **Backend:** Spring Boot REST API handles signup, login, and alert reporting. User data and alerts are stored in memory (simple HashMaps and Lists).
- **Endpoints:** /api/signup, /api/login, /api/cheating, /api/alerts
- **Testing:** API calls are executed using Postman, verifying correct flows for registration, login, reporting, and retrieval of alerts.
- **Webcam Module:** A standalone Java class (using webcam-capture) demonstrates image capture, with future support to send images to ML models via REST.

## TESTING:-

- **Unit Testing:** REST endpoints checked via Postman for all expected flows.
- **Use Cases:** Signup with new username; reject duplicate; successful/failed login; cheating alert logging and retrieval.
- **Negative Testing:** Invalid credentials checked for rejection; missing parameters handled with expected error messages.
- **Integration Testing:** Webcam capture demonstrated on supported desktop hardware.

## ADVANTAGES:-

- Supports remote exams without requiring human proctors.
- Real-time detection and reporting of suspicious activities.
- Simple, scalable backend easily extended with databases or advanced ML.

- Desktop webcam integration enables automated monitoring.
- No sensitive data persisted—safe for demo or learning use.

## **CONCLUSION:-**

The Online Exam Proctoring System is a practical, lightweight solution to monitor remote exams. It proves the core concepts of a proctoring workflow: secure login, cheating alert generation, and webcam capture. With future enhancements, such as real-time ML-driven detection and persistent databases, this project can evolve into a robust professional product.

## **REFERENCES:-**

- Spring Boot Documentation: <https://spring.io/projects/spring-boot>
- Webcam Capture Java Library: <https://github.com/sarxos/webcam-capture>
- "Online Examination Proctoring System Using Artificial Intelligence" Project Report
- Postman API Platform: <https://www.postman.com/>
- Maven: <https://maven.apache.org/>
- Java SE Docs: <https://docs.oracle.com/en/java/>