Software Requirements Specification (SRS): Exam Security System

1. Introduction

1.1 Purpose

This document will aim at defining the functional and non-functional requirements of the Exam Security System. It is aimed at checking student identities, seating compliance, and recording the violations related to exams.

1.2 Project Scope

The system offers access to web-based platform that allows Exam Coordinators and Proctors to control exam day security. It incorporates the basic machine learning functionality of face verification to automate identity verification.

2. User Roles (Actors)

The system supports three core actors:

- **Exam Coordinator (Admin):** Responsible for creating exams, defining rooms, and importing student rosters.
- **Proctor (Invigilator):** Responsible for the check-in workflow, verifying student identities, and logging violations.
- **Student:** The subject of the verification process and seating assignment.

3. Functional Requirements (FR)

Based on the project's core goals, the following functions are required:

3.1 Authentication & Management

- **Role-Based Access Control:** The system must restrict access based on roles (Admin vs. Proctor).
- **Exam Creation:** Admins must be able to create exam sessions including date, time, and room assignments.
- **Roster Management:** Admins must be able to import or enter student rosters for specific exams.
- **Seating Plan:** The system must allow the creation of seating plans using rows, columns, or specific seat codes.

3.2 Verification & Check-in Workflow

- **Image Capture/Upload:** Proctors must be able to capture or upload a live photo of the student.
- **ML Identity Verification:** The system must use a library-based image recognition component to compare the live photo against a registered ID photo.
- **Seating Compliance:** The system must automatically check if the student is sitting in the correct assigned seat according to the plan.
- **Check-in Logging:** The system must record the result of the verification and a timestamp for every check-in attempt.

3.3 Violations & Reporting

- **Violation Logging:** Proctors must be able to log incidents including reason, notes, and optional evidence images.
- **Basic Reporting:** The system must generate reports for check-ins, identity mismatches, and recorded violations.

4. Non-Functional Requirements (NFR)

- **Reliability:** The system must ensure data integrity for check-in timestamps and violation records.
- **Usability:** The web interface must be intuitive for proctors to use quickly during high-pressure exam starts.
- **ML Integration:** The machine learning component must be implemented as an integrated service/wrapper.

---

5. Business Rules & Validations

The following rules govern the system logic:

- **(Duplicate Check-in):** A student cannot be checked into the same exam more than once.
- **(Missing Data):** Mandatory fields (e.g., student ID, seat code, violation reason) must be validated before submission.
- **(ML Accuracy):** While the ML model is not tested for accuracy, the system must handle "No Match" or "Multiple Faces" scenarios as edge cases.
- **(Permissions):** Only users with the "Proctor" or "Admin" role can access the check-in and violation screens.