**Functional Requirements**

**1. User Roles**

• FR1: The system shall support two roles:

o Admin/Invigilator: Create/manage exam sessions, validate student enrolments, view reports.

o Student: Pre-enroll with documents/biometrics, authenticate before exams.

• FR13: Admins shall upload class lists (for each course) containing:

- student ids (matriculation numbers ) and names

- CA marks.

• FR14: The system shall link CA marks to student biometric enrollments (via student ids-matriculation number).

**2. Pre-Enrollment**

• FR2: Students shall upload scanned copies of:

o Form B (registered courses).

o ID card (school/NIC).

o Fee receipt (proof of payment).

• FR3: Students shall enroll their biometric data (modality-agnostic) via a physical kiosk.

• FR4: Admins shall manually approve/reject enrollments after verifying document validity.

**3. Exam Session Management**

• FR5: Admins shall create authentication sessions for:

o Course (e.g., "Computer Engineering Level 500: Database Systems").

o Date/time window (e.g., 8:00 AM – 10:00 AM, 20 Nov 2024).

o Venue (e.g., "NAHPI Hall A").

• FR6: The system shall auto-generate a list of students enrolled in the course (based on Form B data).

**4. Authentication Workflow**

• FR7: During an exam session, students shall authenticate using:

o Biometric data (modality-agnostic, e.g., scan/template match).

o Optional ID number (fallback if biometric fails).

• FR8: The system shall validate in real-time:

o Biometric match against enrolled templates.

o Document validity (fee paid, course registration).

• FR9: The system shall:

o Mark students as "Authenticated" if valid.

o Flag as "Question Marked" if invalid (with reasons: e.g., "Fee unpaid", "Biometric mismatch").

**5. Reporting**

• FR10: Admins shall view/download:

o Real-time list of authenticated students.

o "Question Marked" students with reasons.

o Final attendance report (PDF/Excel).

6. Data Management

• FR11: The system shall operate offline (no dependency on central university DB).

• FR12: Store data locally (students, enrollments, sessions, logs).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Non-Functional Requirements (MVP Constraints)**

1. Performance

• NFR1: Authentication response time ≤ 3 seconds per student.

• NFR2: Support 50+ concurrent authentications (scalable to 500+ with optimizations).

2. Security

• NFR3: Biometric templates (not raw data) stored with encryption.

• NFR4: Document storage secured with role-based access (admins only).

3. Usability

• NFR5: Admin interface requires ≤ 30 mins of training (intuitive workflows).

• NFR6: Authentication kiosk has a 3-step max interface (e.g., "Scan → Verify → Result").

4. Scalability

• NFR7: Modular architecture (e.g., plug-and-play biometric modules).

• NFR8: Support 1 department (e.g., NAHPI Computer Engineering) in MVP, with design for 10k+ students later.

5. Reliability

• NFR9: System uptime ≥ 95% during exam sessions.

• NFR10: Daily backups of critical data (enrollments, logs).

6. Cost

• NFR11: Total hardware/software cost ≤ $200 (MVP phase).

7. Interoperability

• NFR12: Support common document formats (PDF, JPG, PNG).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Out-of-Scope (For Now)

1. Automated document validation (admins will manually approve).

2. Integration with university’s central systems.

3. Multi-modal biometrics (stick to one modality for MVP).

4. Mobile app support.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Key Assumptions

1. Biometric hardware (sensor) is available for enrollment/authentication.

2. Exam venues have basic computing infrastructure (e.g., laptop + sensor).

3. Admins can manually cross-check documents during pre-enrollment.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MVP Workflow (Modality-Agnostic)

1. Pre-Enrollment Phase

o Students upload documents via a web portal.

o Visit a kiosk to enroll biometric data (e.g., sensor scan).

o Admins manually approve/reject enrollments.

2. Exam Session Setup

o Admin creates a session for a course/venue/time.

o System auto-generates the student list for that course.

3. Authentication Phase

o Student approaches kiosk, provides biometric sample.

o System checks:

♣ Biometric match.

♣ Valid documents (approved Form B, fee receipt).

o Result: Authenticated (grant access) or Question Marked (deny + flag).

4. Reporting

o Real-time dashboard for invigilators.

o Post-session report for record-keeping.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prioritized Implementation Steps

1. Build Document Upload & Approval System

o Use FastAPI (Python) for a simple admin/student portal.

o Database: PostgreSQL (free, scalable).

2. Biometric Module (Generic)

o Choose a sensor (e.g., low-cost fingerprint scanner or webcam for facial).

o Use a Python library (e.g., Python-Biometrics for modality-agnostic handling).

3. Authentication Kiosk

o Develop a simple Python app for sensor interaction + validation logic.

4. Testing

o Pilot with 20–50 students in one department (NAHPI Computer Engineering).