Software Requirements Specification

for

AI-Based Automated Answer Sheet Evaluation System

Version 1.0 approved

Prepared by

Vaibhav Sutar

KIT’s College Of Engineering Kolhapur **28-11-2025**

Table of Contents

1. Introduction

1.1 Purpose

This SRS defines the functional and non-functional requirements for the **AI-Based Automated Answer Sheet Evaluation System**, a web-based and AI-powered platform to automate handwritten answer sheet evaluation using **OCR**, **semantic similarity**, **keyword-based scoring**, and **paper similarity analysis**.
It serves as a blueprint for developers, testers, faculty, exam cell staff, and stakeholders.

1.2 Document Conventions

- *Bold → Important terminology*
- *Italic → External resources*
- *Requirements follow prefix REQ- and are sequentially numbered*
- *Architecture follows IEEE SRS format as referenced*

1.3 Intended Audience and Reading Suggestions

- *Developers: Section 2, 3, 4*
- *Testers: Section 3 & 4*
- *Project Leads / Mentors: Section 1 & 2*
- *Faculty & Exam Cell: Section 4 summaries*

1.4 Product Scope

The system automates the examination evaluation workflow including:

- *QR-based answer sheet identification*
- *OCR for handwritten text extraction*
- *AI-based semantic and keyword scoring*
- *Faculty-provided model answer processing*
- *Question paper similarity checking*
- *Student result dashboards*

1.5 References

- *IEEE Std 830-1998*
- *FastAPI, Node.js, MongoDB, MySQL documentation*
- *OCR frameworks: PaddleOCR, EasyOCR, Tesseract*

2. Overall Description

2.1 Product Perspective

This is a new standalone AI-powered multi-service system consisting of:

- Frontend (Vite + React)
- Node.js Backend (Primary Business Logic Service)
- FastAPI Backend (AI Microservice)
- Databases: MySQL + MongoDB
- Redis Queue for background AI jobs

2.2 Product Functions

Core Functions

- Map scanned sheets using QR
- Extract text via OCR
- Evaluate answers using AI models
- Compare question papers for similarity
- Generate feedback and marks
- Provide dashboards for faculty, exam cell, students

System Functions

- Store OCR data, AI outputs in MongoDB
- Store users, marks & metadata in MySQL
- Queue heavy AI jobs through Redis
- Export results as PDF

2.3 User Classes and Characteristics

- **Students:** Non-technical, view results only
- **Faculty:** Technical/academic users providing model answers, keywords, papers
- **Exam Cell:** Semi-technical staff handling uploads & result publication
- **Admin:** Technical users managing system controls.

2.4 Operating Environment

- ***Frontend:*** *Vite + React*
- ***Backend:*** *Node.js + Express*
- ***AI Engine:*** *Python + FastAPI*
- ***Databases:*** *MySQL & MongoDB*
- ***Hosting:*** *Local server / Cloud*
- ***Devices:*** *Mobile or PC browser*

2.5 Design & Implementation Constraints

- OCR engine accuracy depends on scan quality
- Handwriting may vary (noise, skew, blur)
- MySQL must maintain relational exam data
- MongoDB must support large AI document storage
- QR code format depends on college-provided format

2.6 User Documentation

- User manual for Exam Cell

- Faculty guide for model answer upload
- Student result portal guide
- API documentation (Swagger for FastAPI)

2.7 Assumptions & Dependencies

- All scans uploaded by exam cell are clear and valid
- QR codes follow consistent college format
- AI evaluation depends on faculty-provided model answers
- Internet access is available

3. External Interface Requirements

3.1 User Interfaces

- Login interface (Student, Faculty, Exam Cell)
- Exam Cell dashboard (upload sheets, publish results)
- Faculty dashboard (upload model answers, papers, keywords)
- Student dashboard (view evaluated answer sheet)

3.2 Hardware Interfaces

- Scanner or mobile camera
- Standard keyboard/mouse

3.3 Software Interfaces

- Node.js API
- FastAPI AI endpoints
- MySQL ORM (Prisma/Sequelize)
- MongoDB ODM (Mongoose/PyMongo)

3.4 Communication Interfaces

- REST API
- JSON payloads
- Secure JWT authentication

4. System Features

4.1 QR-Based Answer Sheet Mapping

REQ-001

System shall automatically extract student & exam information from college-provided QR codes.

REQ-002

System shall group all pages belonging to the same student automatically.

REQ-003

System shall detect:

- Missing pages
- Duplicate pages
- Wrong-student pages

4.2 OCR & Text Extraction

REQ-004

System shall extract handwritten text using PaddleOCR + EasyOCR.

REQ-005

System shall preprocess pages (de-skew, noise removal, contrast normalization).

REQ-006

System shall segment answers based on visual bounding boxes.

4.3 AI-Based Answer Evaluation

REQ-007

System shall compute semantic similarity using SBERT embeddings.

REQ-008

System shall compute keyword-based similarity using weighted keywords.

REQ-009

System shall generate feedback for missing points, factual errors, and incomplete answers.

REQ-010

System shall combine rubric, semantic, and keyword scores to compute final marks.

4.4 Paper Similarity Analysis (Faculty Requirement)

REQ-011

System shall accept multiple question papers (Set A/B/C).

REQ-012

System shall compute similarity across sets based on:

- Semantic similarity
- Structural similarity
- Repeated question detection

REQ-013

System shall generate a similarity matrix and recommend the most unique paper.

4.5 Exam Cell Panel

REQ-014

Exam cell shall upload scanned answer sheets.

REQ-015

Exam cell shall verify mapping before evaluation.

REQ-016

Exam cell shall publish final results.

4.6 Faculty Panel

REQ-017

Faculty shall upload model answers and keywords.

REQ-018

Faculty shall override AI marks if required.

REQ-019

Faculty shall view similarity report of question papers.

4.7 Student Panel

REQ-020

Students shall view AI-evaluated answer sheets.

REQ-021

Students shall download evaluation report PDF.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- OCR processing: ≤2 seconds per page
- AI evaluation: ≤1 second per answer
- Handle up to 1,000 answer sheets in a batch

5.2 Safety Requirements

- Sensitive student data must be protected
- Corrupted scans should not be evaluated

5.3 Security Requirements

- Password hashing (bcrypt)
- JWT authentication
- Role-based access control
- HTTPS in production

5.4 Software Quality Attributes

- Usability: Clean user experience
- Maintainability: Modular microservices
- Reliability: ≥ 99% uptime
- Scalability: Horizontal scaling

5.5 Business Rules

- One account per student/faculty
- AI evaluation depends strictly on faculty-provided model answers
- Exam cell has final authority to approve marks

6. Appendix A: Glossary

- **OCR:** Converts handwritten text to digital text
- **AI Evaluation:** Automated answer scoring
- **QR Mapping:** Detecting pages through QR codes
- **Paper Sets:** Multiple question paper versions (A/B/C)

7. Appendix B: Analysis Models

- UML diagrams
- Sequence diagrams
- DFD diagrams
- Architecture diagram

8. Appendix C: To Be Determined

- Offline evaluation support
- Multiple regional languages
- LMS integration
- Handwriting verification model