Software Requirements Specification (SRS)

# Sprint 1 – LMS Application Prototype: Student Result Management

## 1. Introduction

### 1.1 Product Scope

In Sprint 1, we aim to build the foundational LMS features focused on student result management. This prototype enables the Vitara Learning Admin to upload and store test marks into the database, and allows students to securely view their own results.

### 1.2 Product Value

This MVP (Minimum Viable Product) establishes a centralized and digitized result management system, streamlining the communication of test performance between admin and students.

### 1.3 Intended Audience

- Vitara Learning Admin (for result upload)  
- Students (for result viewing)

### 1.4 Intended Use

Admins will upload test results after correcting OMR sheets.  
Students will log in and view their respective test scores.

### 1.5 Definitions and Acronyms

LMS: Learning Management System  
OMR: Optical Mark Recognition  
MERN: MongoDB, Express.js, React.js, Node.js

## 2. Functional Requirements – Sprint 1

|  |  |  |
| --- | --- | --- |
| ID | Feature | Description |
| FR1 | User Authentication | Users (Admin, Students) can log in securely using email and password. |
| FR2 | Role-based Access | Only Admin can upload results. Students can only view their own results. |
| FR3 | Result Upload | Admin can upload test marks post-OMR correction into the database. |
| FR4 | Result Viewing | Students can view their marks test-wise after login. |
| FR5 | Database Integration | All uploaded results are stored securely and associated with student records. |

## 3. External Interface Requirements

### 3.1 User Interface

Admin dashboard with result upload form.  
Student dashboard to view results.  
Responsive design for mobile and desktop.

### 3.2 Hardware Interfaces

Accessible via standard devices: mobile phones, tablets, laptops.

### 3.3 Software Interfaces

Backend API using Express.js  
MongoDB database connection  
Frontend using React.js

### 3.4 Communication Interfaces

RESTful API endpoints for secure data transactions.  
Use of HTTPS and JWT-based authentication.

## 4. Non-Functional Requirements (NFRs)

|  |  |
| --- | --- |
| Category | Requirement |
| Security | Encrypted credentials, JWT tokens, and role-based data access |
| Performance | Quick result fetch and upload with optimized DB queries |
| Usability | Clean and intuitive UI for both Admin and Students |
| Scalability | Built on MERN stack to support future feature sprints |
| Maintainability | Modular codebase for easier updates and debugging |
| Reliability | Basic error handling and fallback for failed uploads or DB errors |

## 5. Sprint Focus and Limitations

This document covers only Sprint 1. Future sprints will introduce additional modules such as:  
- Course management  
- Quiz and assignment systems  
- Notifications and analytics  
- Admin reporting dashboards

## 6. Conclusion

This SRS defines the scope of Sprint 1 for the LMS application focused on result management. It establishes the baseline architecture and functionality to support a working MVP that is secure, usable, and scalable.