

Software Requirements Specification (SRS)

Project Title: Strategic Web-Based Dashboard

Version: 1.0

Date: 19-06-2025

Prepared by: ScriptsCode Consultants LLP

Table of Contents

1. [Introduction](#)
 - 1.1 [Purpose](#)
 - 1.2 [Document Conventions](#)
 - 1.3 [Intended Audience and Reading Suggestions](#)
 - 1.4 [Product Scope](#)
 - 1.5 [References](#)
 2. [Overall Description](#)
 - 2.1 [Product Perspective](#)
 - 2.2 [Product Functions](#)
 - 2.3 [User Classes and Characteristics](#)
 - 2.4 [Operating Environment](#)
 - 2.5 [Design and Implementation Constraints](#)
 - 2.6 [Assumptions and Dependencies](#)
 3. [System Features and Requirements](#)
 - 3.1 [Functional Requirements](#)
 - 3.2 [External Interface Requirements](#)
 - 3.3 [Non-Functional Requirements](#)
 4. [Appendices](#)
-

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) outlines the requirements for the development of a strategic web-based dashboard that enables data storage, retrieval, visualization, and analysis for faculty & administrators

1.2 Document Conventions

- Bold headings denote section titles
- Requirements are labeled as FR (Functional Requirement), NFR (Non-Functional Requirement), or UI (User Interface)

1.3 Intended Audience and Reading Suggestions

- **Project Stakeholders:** Understand functionality and scope
- **Developers:** Refer to functional and interface specifications

- **Testers:** Refer to the requirements to prepare test cases

1.4 Product Scope

The product is a data visualization dashboard with role-based access. It enables uploading, managing, analyzing, and visualizing data using various chart types and allows filtered views with export capabilities.

1.5 References

- ISO/IEC 29148:2018 Software Requirements
 - Laravel 10.x Documentation
 - MySQL 8.0 Documentation
 - Chart.js and other visualization libraries
-

2. Overall Description

2.1 Product Perspective

This is a standalone web application with a client-server architecture:

- **Frontend:** HTML5, CSS, JavaScript, jQuery, AJAX
- **Backend:** PHP (Laravel + Livewire)
- **Database:** MySQL

2.2 Product Functions

- User authentication and role-based access
- Upload data (CSV/Excel) and manual entry
- Data visualization: Pie, Donut, Histogram, Sine Wave
- Filters and dropdowns for customized views
- Export to PDF and Excel

2.3 User Classes and Characteristics

User Role Permissions		Technical Level
Admin	Full access including settings, user management	Advanced
Faculty	View and analyze data, limited upload access	Intermediate

2.4 Operating Environment

- Web server with PHP 8.x
- Laravel framework
- MySQL database

- Works in latest browsers: Chrome, Firefox, Edge

2.5 Design and Implementation Constraints

- Must use Laravel and Livewire
- Charts via open-source JS libraries (e.g., Chart.js)
- Deployment on Linux server with Apache/Nginx

2.6 Assumptions and Dependencies

- Internet connectivity is required for access
 - Users have appropriate permissions to perform actions
 - Chart libraries are compatible with frontend tech stack
-

3. System Features and Requirements

3.1 Functional Requirements

FR1. User Management

- FR1.1: System shall allow user login/logout
- FR1.2: System shall support role-based permissions (Admin, Faculty)

FR2. Data Upload

- FR2.1: Admin/Faculty shall upload data via CSV/Excel
- FR2.2: System shall validate data before storage
- FR2.3: Manual data entry through form interface

FR3. Data Visualization

- FR3.1: Display data in table and graphical formats
- FR3.2: Charts supported: Pie, Donut, Histogram, Sine Wave
- FR3.3: Charts update dynamically based on filters

FR4. Filtering & Dropdowns

- FR4.1: System shall provide filters (date, category, etc.)
- FR4.2: Dropdowns shall dynamically load values from the database

FR5. Reports

- FR5.1: Users shall export reports as PDF/Excel
- FR5.2: Reports include charts and tabular data

3.2 External Interface Requirements

3.2.1 User Interfaces

- Clean and responsive UI
- Navigation menu for dashboard sections
- Modal dialogs for upload and entry
- Pagination in tables

3.2.2 Hardware Interfaces

- None specifically required (web-based only)

3.2.3 Software Interfaces

- Laravel for backend logic
- Livewire for dynamic UI interactions
- Chart.js for graphs
- MySQL for persistent storage

3.2.4 Communication Interfaces

- AJAX for asynchronous updates
- HTTPS for secure communication

3.3 Non-Functional Requirements

NFR1. Performance

- NFR1.1: Dashboard must load under 6 seconds on broadband
- NFR1.2: Visualizations must render under 3 second for datasets < 1MB

NFR2. Scalability

- Should support 200+ concurrent users

NFR3. Security

- NFR3.1: Role-based access control must be enforced
- NFR3.2: Input validation and sanitization
- NFR3.3: CSRF protection enabled

NFR4. Usability

- NFR4.1: UI/UX must follow intuitive design principles
- NFR4.2: Provide tooltips, icons, and guidance prompts

NFR5. Maintainability

- Code must be documented and modular
- Git for version control

NFR6. Portability

- Compatible across Chrome, Firefox, Edge, Safari (latest versions)
-

4. Appendices

A. Glossary

- **Livewire:** Laravel framework for dynamic front-end without JS
- **Donut Chart:** Pie chart with a hole in the center
- **Histogram:** Bar chart representing data distribution

B. Time & Resource Estimation

Estimated Duration: 60 working days

Team Composition:

- 1 UI/UX Designer
- 1 Frontend Developer
- 1 Backend Developer

C. Future Enhancements

- AI-based analytics
- API support for mobile app integration
- Notification system for reports