STEM Projects Showcase

Software Requirement Specifications

Bachelor of Science in Software Engineering

By

	Name	Registration #	Mobile #	E-Mail
1	Abdul-Rehman	SP2021-BSSE-008	0309-0353418	a.rehman360916@gmail.com
2	Hafiz Haroon Mujtaba	SP2021-BSSE-015	0307-1691492	haroonmujtaba12345@gmail.com

Supervised by

_____

Teacher Name

Co-Supervised by

_____

Teacher Name

Department of Software Engineering
Lahore Garrison University, Pakistan

Table of Contents

1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 2
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 3
2.3 User Classes and Characteristics 4
2.4 Operating Environment 5
2.5 Design and Implementation Constraints 5
2.6 User Documentation 6
2.7 Assumptions and Dependencies 7
3. External Interface Requirements 8
3.1 User Interfaces 8
3.2 Hardware Interfaces 10
3.3 Software Interfaces 12
3.4 Communications Interfaces 13
4. System Features 15
4.1 User Management 15
4.2 Project Management 16
4.3 Communication and Collaboration 16
4.4 Dashboard and Reporting 16
4.5 Resource and Document Management 17
4.6 Security and Compliance 17
4.1.1 System Feature 1: User Management 17
4.1.2 Description and Priority 18
4.1.3 Stimulus/Response Sequences 18
4.1.4 Functional Requirements 19
4.2.1 System Feature 2: Project Management 21
4.2.2 Stimulus/Response Sequences 21
4.2.3 Functional Requirements 22
4.3.1 System Feature 3: Communication and Collaboration 23
4.3.2 Stimulus/Response Sequences 23
4.3.4 Functional Requirements 24

4.4.1 System Feature 4: Dashboard and Reporting 24
4.4.2 Stimulus/Response Sequences 24
4.4.3 Functional Requirements 24
4.5.1 System Feature 5: Resource and Document Management 24
4.5.2 Stimulus/Response Sequences 25
4.5.3 Functional Requirements 25
4.6.1 System Feature 6: Security and Compliance 25
4.6.2 Stimulus/Response Sequences 25
4.6.3 Functional Requirements 26
5. Other Nonfunctional Requirements 26
5.1 Performance Requirements 26
5.2 Safety Requirements 27
5.3 Security Requirements 28
5.4 Software Quality Attributes 29
5.5 Business Rules 30
6. Resources 31

1. Introduction

1.1 Purpose

The product identified in this Software Requirements Specification (SRS) is the **STEM Project Showcase Platform**. This SRS document outlines the requirements for the development of a digital platform dedicated to showcasing and promoting student projects within STEM disciplines at our university. The scope of this SRS includes all primary features of the platform, covering both user-facing modules such as user registration, project submission, and interactive project pages and administrative functionalities, including content management and moderation. This SRS encompasses the full system requirements for the platform's front-end and back-end, ensuring a comprehensive solution for project presentation and user engagement.

1.2 Document Conventions

This SRS follows a structured format for clarity and consistency:

Heading Levels: Section headings use bold and larger fonts to distinguish primary sections and subsections.

Requirement Priorities: Requirements are categorized by priority levels **High**, **Medium**, and **Low** with each priority level noted at the beginning of each requirement statement.

Typography:

- **Bold** text highlights section headers and important notes.
- *Italicized* text is used for examples and supplementary details.
- Monospaced font is applied to denote code elements, database fields, and technical keywords.

Numbering System: All requirements are numbered hierarchically (e.g., 1.0, 1.1, 1.1.1) to indicate their position within the document structure.

Glossary Terms: Key terms are italicized and referenced in the glossary for clarity.

1.3 Intended Audience and Reading Suggestions

This SRS is designed for a diverse group of stakeholders involved in the STEM Project Showcase development process:

Developers: For understanding detailed functional and non-functional requirements essential for implementation.

Project Managers: For tracking project scope, deliverables, and priorities to manage timelines and resources.

Testers: For gaining insights into requirements that will form the basis for test cases and quality assurance processes.

Marketing Staff: To understand the platform's key features and value propositions for effective positioning and promotion.

End Users (Students and Faculty): To gain a clear view of platform functionalities, allowing them to provide relevant feedback during development phases.

Suggested Reading Sequence:

Overview Sections (Purpose, Scope, Product Perspective) for all readers to gain a foundational understanding.

General Description and Product Functions for developers, project managers, and testers to understand the intended functionalities.

Detailed Requirements sections, organized by priority, to guide developers in implementation and testers in validation.

Appendices and Glossary for all readers needing clarification on technical terms or additional reference material.

1.4 Product Scope

The STEM Project Showcase is a web-based platform aimed at centralizing the presentation and promotion of student projects within STEM disciplines at our university. This software provides a structured and interactive environment for students, faculty, and industry professionals to discover, view, and engage with student work. Key objectives include enhancing project visibility, encouraging interdisciplinary collaboration, and bridging the gap between academic innovation and industry relevance. By supporting knowledge sharing and networking, the platform aligns with the university's goal to foster innovation and strengthen its reputation in STEM fields.

2. Overall Description

2.1 Product Perspective

The STEM Project Showcase platform is a new, standalone product developed to provide a digital space for showcasing student STEM projects. Unlike traditional project exhibitions or departmental websites, this platform offers a centralized, interactive solution for digital project display and engagement. It is not an extension or replacement of existing university systems but aims to complement them by facilitating a modern, web-based showcase accessible to students, faculty, and industry stakeholders. The system will integrate with university authentication services for secure access and leverage multimedia and feedback functionalities to enrich project presentations.

Here's a simplified system structure:

User Authentication System: Interfaces with university login for secure access.

Project Management: Core subsystem handling project submission, multimedia support, and feedback.

User Dashboard: Customized for students, faculty, and administrators, featuring content and access appropriate to each role.

Administrative Controls: Dashboard for content moderation, role assignment, and system monitoring.

2.2 Product Functions

User Registration and Authentication

Allow students, industrialists, and admin users to create accounts and securely log in.Enable role-based access, granting different permissions based on user type (e.g., student, industry professional, admin).

Project Showcase and Management

Provide students with a platform to showcase their projects, including details such as title, description, technology stack, and media attachments. Enable functionality for updating and managing individual project entries.

Project Submission and Approval Workflow

Allow students to submit projects for industrialist review. Enable admin users to approve or reject projects, maintaining quality and relevance for industry partners.

Communication and Engagement

Allow industrialists to contact students regarding specific projects through in-platform messaging or provided contact links. Enable comments and feedback on projects to facilitate engagement and mentoring.

Industry Partnership Management

Provide a dedicated portal for industrialists, allowing them to browse projects, connect with students, and view student profiles. Enable admins to manage and organize industry partnerships and showcase featured partnerships on the platform.

Notifications and Alerts

Send automated notifications to students and industrialists for key actions (e.g., project approvals, contact requests). Provide students and industrialists with alert options to stay updated on platform activity.

Search and Filter

Allow users to search and filter projects by technology stack, industry relevance, or keyword. Enable students and industrialists to discover projects and collaborators based on specific areas of interest

User Profile Management

Allow students to create detailed profiles, including skills, academic background, and contact information.Enable industrialists to create profiles showcasing their organization, areas of interest, and contact details.

2.3 User Classes and Characteristics

Students

Description: Primary users who submit and manage their STEM projects.

Characteristics: Typically tech-savvy but may have varying experience with online platforms. Need easy access to project submission, editing, and feedback features.

Frequency of Use: Frequent during project submission periods; occasional for viewing and engagement.

Faculty Members

Description: Responsible for evaluating and providing feedback on student projects.

Characteristics: Intermediate tech expertise with an academic background. Require features to review, evaluate, and offer feedback on projects.

Frequency of Use: Regularly, particularly during review periods.

Administrators

Description: Manage the platform, control content, and oversee system functionality.

Characteristics: Higher technical expertise; require access to all system modules, including user management, content moderation, and analytics.

Frequency of Use: Frequent, especially for maintaining platform integrity and performance.

Industry Professionals & External Evaluators

Description: Invited users who view and provide feedback on projects, helping bridge academia and industry.

Characteristics: Limited technical interaction; require intuitive navigation and the ability to view projects and leave feedback.

Frequency of Use: Occasional, often during showcase events or evaluations.

Guest Viewers (Public Access)

Description: Individuals interested in exploring STEM projects, such as prospective students or parents.

Characteristics: Minimal technical expertise; need easy access to view projects without requiring login.

Frequency of Use: Infrequent, as needed to explore project showcases.

2.4 Operating Environment

Hardware Platform:

The software will operate on standard desktop, laptop, and mobile devices with internet connectivity, ensuring accessibility across diverse devices.

Operating System:

Desktop/Laptop: Compatible with Windows 10+, macOS.

Mobile: Compatible with iOS and Android devices for optimal mobile responsiveness.

Browser Compatibility:

Supports modern web browsers including Chrome (version 80+), Firefox (version 75+), Safari (version 13+), and Edge (version 85+).

Backend Environment:

- Server: Node.js (version 14+)
- Database: PostgreSQL (version 12+)

Frontend Environment:

- Framework: React.js
- Styling: Tailwind CSS

2.5 Design and Implementation Constraints

Corporate Policies: Adherence to university policies regarding data privacy and student information protection will limit how user data is handled and stored. Compliance with the Family Educational Rights and Privacy Act (FERPA) is mandatory.

Regulatory Compliance: The software must comply with local and international regulations regarding accessibility (e.g., WCAG 2.1) to ensure usability for all users, including those with disabilities.

Hardware Limitations:

The application must perform optimally on devices with at least 4GB of RAM and a dual-core processor to ensure smooth user experience.Any performance constraints from hosting environments, such as latency and response times, will impact user interaction.

Technological Stack: The project will utilize the MERN stack (MongoDB, Express.js, React.js, Node.js) with PostgreSQL as the database, which imposes restrictions on technology choices and integration possibilities.

Interfaces with Other Applications: The software must integrate with existing university systems (e.g., Student Portal) for user authentication and data synchronization, necessitating adherence to their APIs and data formats.

Security Considerations: Implementation of robust security measures, such as HTTPS, data encryption, and secure coding practices, is required to protect sensitive user information and comply with best practices.

Design Conventions: The application must follow established design conventions and user experience guidelines to maintain consistency and usability across the platform. Any changes to these standards may require additional development time.

Language Requirements: The software will be developed in English, with potential future support for additional languages, contingent on the user base's needs.

2.6 User Documentation

The following user documentation components will be delivered alongside the STEM Project Showcase software to facilitate user understanding and engagement:

User Manual: A comprehensive guide detailing all features, functionalities, and navigation within the platform. This manual will be available in both printed and PDF formats for easy access.

Online Help: Context-sensitive help integrated directly into the application. Users will have access to tooltips, FAQs, and help sections that provide immediate assistance on specific features.

Tutorials: Step-by-step video tutorials and written guides that cover key functionalities such as user registration, project submission, and utilizing the administrative dashboard. These will be hosted on the platform and shared via the university's learning management system.

Quick Start Guide: A condensed guide for new users, providing essential information to get started quickly with the platform, including basic navigation tips and common tasks.

Release Notes: Documentation outlining updates, new features, bug fixes, and changes made in each version of the software. This will be provided in a changelog format accompanying major releases.

Accessibility Documentation: Guidelines and resources that explain how to use the platform with assistive technologies, ensuring compliance with accessibility standards.

Delivery Formats

PDF: For manuals and guides, ensuring compatibility across devices.

HTML: Online help and tutorials will be accessible through the application interface.

Video: Tutorials will be available in standard video formats (e.g., MP4) and hosted on platforms like YouTube or the university's internal video hosting service.

2.7 Assumptions and Dependencies

Assumptions

User Technical Proficiency: It is assumed that users (students, faculty, and industry professionals) will possess a basic level of technical proficiency, enabling them to navigate and utilize the STEM Project Showcase platform effectively.

Internet Connectivity: The platform is expected to operate in environments with reliable internet access. Users may face challenges in areas with poor connectivity, affecting their ability to access and submit projects.

Compatibility with Modern Browsers: The software is assumed to be compatible with the latest versions of modern web browsers (e.g., Chrome, Firefox, Safari, and Edge). This assumption may impact user experience if users rely on outdated browsers.

Availability of Support Resources: It is assumed that adequate technical support and resources will be available for users to address any issues they encounter while using the platform.

Adoption by Stakeholders: It is assumed that students, faculty, and industry professionals will actively participate in the platform and utilize its features for project showcasing and collaboration.

Dependencies

Third-Party Libraries and Frameworks: The project will depend on various third-party libraries (e.g., React, Tailwind CSS) and frameworks for frontend development. Any updates or changes in these libraries could impact the functionality and performance of the platform.

PostgreSQL Database: The application relies on PostgreSQL for backend data management. Any limitations or issues with the database could affect data storage, retrieval, and overall system performance.

Hosting Environment: The platform's deployment will depend on a reliable hosting service to ensure accessibility and performance. Any downtime or issues with the hosting provider may affect user access.

Compliance with Institutional Policies: The project is dependent on the university's policies regarding data privacy, user data management, and platform usage. Changes in these policies may require modifications to the software.

User Feedback: The effectiveness and evolution of the platform will depend on user feedback. Continuous improvement will rely on insights gained from user interactions and suggestions.

3. External Interface Requirements

3.1 User Interfaces

The user interface (UI) of the software product will prioritize usability, accessibility, and aesthetic appeal, ensuring a seamless experience for all users. Below are the logical characteristics of each interface:

General UI Standards

Consistent Design: All screens will follow a consistent layout and design, adhering to a style guide that includes a cohesive color palette, typography, and button styles.

Responsive Design: The UI will be fully responsive, ensuring optimal viewing across devices (desktops, tablets, and mobile phones).

Accessibility Standards: UI components will comply with WCAG (Web Content Accessibility Guidelines) to ensure accessibility for users with disabilities.

Common UI Components

Navigation Bar:

Location: Top of each screen.

Elements: Links to Home, Projects, Submit Project, My Account, Industrialists, Admin (only visible to authenticated users).

Functionality: The navigation bar will collapse into a hamburger menu on mobile devices.

Buttons:

Standard Appearance: Rounded corners, clear labeling (e.g., "Submit," "Login"), and color-coded based on function (e.g., green for actions, red for deletions).

Hover Effects: Buttons will change color on hover to provide visual feedback.

Error Messages:

Display Location: Inline next to the input field or as a banner at the top of the form.

Styling: Messages will be styled in red with a bold font for visibility.

Format: Clear and concise language (e.g., "Please enter a valid email address.").

Help Functionality:

Location: Available as a question mark icon on each screen.

Functionality: On-click, a modal will provide users with context-sensitive help relevant to the current page.

User-Specific Interfaces

Student Interface:

Dashboard: Display a summary of submitted projects, pending approvals, and recent notifications.

Project Submission Form: User-friendly form with validation, including fields for project title, description, and file uploads.

Industrialist Interface:

Dashboard: View a list of projects, with the ability to filter and search.

Contact Modal: A popup window that appears when contacting a student, displaying relevant information (LinkedIn, email, phone number).

Admin Interface:

Dashboard: Overview of all users and projects, with options to approve/reject submissions.

User Management Page: Interface for managing user roles and statuses.

Sample Screen Images

While specific screen images cannot be provided here, mockups should follow these guidelines:

Header: Consistent across all screens with the logo on the left and navigation links on the right.

Main Content Area: Divided into sections with clear headings and a balanced layout.

Footer: Contains links to terms of service, privacy policy, and contact information.

Keyboard Shortcuts

Global Shortcuts: Common actions such as "Ctrl + S" for saving, "Ctrl + Z" for undoing, and "Esc" for closing modals will be implemented.

Navigation Shortcuts: Allow quick access to major sections using "Alt + H" for Home, "Alt + P" for Projects, etc.

User Interface Documentation

Detailed specifications of the user interface design, including wireframes, mockups, and specific style guides, will be documented in a separate User Interface Specification (UIS). This will include:

- Detailed layouts for each screen.
- Style guides for typography, color schemes, and iconography.
- Interaction flows to demonstrate how users navigate through the application.

3.2 Hardware Interfaces

The hardware interfaces section outlines the logical and physical characteristics of each interface between the software product and the hardware components of the system. This includes the supported device types, data interactions, and communication protocols.

Supported Device Types

Desktop Computers:

Operating Systems: Windows, macOS.

Hardware Requirements: Minimum of 4GB RAM, 2.0 GHz processor, and 500MB free disk space.

Laptops:

Operating Systems: Windows, macOS.

Hardware Requirements: Minimum of 4GB RAM, 2.0 GHz processor, and 500MB free disk space.

Tablets and Mobile Devices:

Operating Systems: iOS, Android.

Hardware Requirements: Minimum of 2GB RAM, 1.5 GHz processor, and 300MB free storage.

Servers:

Operating Systems: Windows Server, Linux.

Hardware Requirements: Minimum of 8GB RAM, 2.5 GHz multi-core processor, and sufficient disk space to support the database and application files.

Data and Control Interactions

User Input:

Input Devices:

Keyboard: For text input across forms and data fields.

Mouse/Touchpad: For navigation, clicking buttons, and selecting options.

Touchscreen (for tablets/mobile devices): Allows direct interaction with UI elements.

Output Display:

Display Devices: Monitors for desktops/laptops and screens for mobile devices, providing visual feedback on user interactions.

Resolution Support: The application will support standard resolutions including 1920x1080 (Full HD) and 1280x720 (HD) for desktops/laptops, and common mobile resolutions.

Peripheral Devices:

Printers: For users who may want to print project submissions or reports.

Scanners: For digitizing physical documents submitted by users.

Communication Protocols

HTTP/HTTPS:

The software product will utilize HTTP/HTTPS protocols for communication between the client and server. This ensures secure data transmission and supports RESTful API architecture for backend interactions.

WebSocket:

Used for real-time communication features, such as notifications for users when a project is approved or messages from industrialists.

Local Network Communication:

For server environments, protocols like TCP/IP will be used to facilitate communication between server and client machines in a local area network (LAN).

Security Considerations

Data Encryption:

Communication between the client and server will be encrypted using TLS/SSL protocols to protect sensitive user data.

Access Control:

Hardware interfaces will support user authentication mechanisms (e.g., username/password, multi-factor authentication) to ensure that only authorized users access the system.

Physical Characteristics

System Configuration:

Hardware configurations will vary depending on user roles (e.g., students, industrialists, and admins). The performance will be monitored to optimize application responsiveness.

Device Compatibility:

The software will be tested across different devices to ensure compatibility and responsiveness on varying screen sizes and resolutions.

3.3 Software Interfaces

The software interfaces section describes the connections between the software product and other specific software components, including databases, operating systems, tools, libraries, and integrated commercial components. This section outlines the data items or messages that enter and exit the system, the purpose of each, and the necessary services and communication methods.

Software Components and Connections

Operating Systems:

Windows: Version 10 and above.

macOS: Version Mojave (10.14) and above.

Databases:

Database Management System:

PostgreSQL: Version 13 and above (for relational data storage and complex queries).

Backend Framework:

Node.js: Version 14.x or higher (for the server-side application).

Express.js: Version 4.x (for building the web server and API).

Frontend Framework:

React: Version 17.x or higher (for building the user interface).

Vite: Version 2.x (for frontend build tool).

Data Items and Messages

Incoming Data:

User Input Data:

- **Registration Data:** Full name, email, password, and role (user type) during sign-up.
- **Login Data:** Username/email and password during login.
- **Project Submission Data:** Details about projects submitted by users.

Outgoing Data:

User Notifications:

Email notifications for password resets, project approvals, or feedback.

SMS notifications for important alerts (if integrated with Twilio).

Project Data:

Data sent to users regarding the status of their project submissions (approved, rejected).

Services and Communication Nature

RESTful API Services:

The product will expose RESTful APIs for CRUD operations (Create, Read, Update, Delete) for user accounts, projects, and notifications. Each API will return JSON responses to allow easy integration with frontend components.

WebSocket Services:

Implement WebSocket connections for real-time updates and notifications to users when events occur (e.g., project status changes).

Data Sharing Mechanism

Data Sharing:

Shared data between components includes user profiles, project data, and notifications. The data will be stored in a central database (MongoDB/PostgreSQL) accessible by both the backend and frontend components.

Implementation Constraints

API Rate Limiting:

Implement API rate limiting to manage the number of requests a user can make to prevent abuse and ensure service availability.

Database Schema Design:

Follow a predefined database schema for user accounts, project submissions, and notifications to maintain data integrity and ease of use.

Error Handling Standards:

Standardize error messages across APIs and the frontend to provide consistent feedback to users.

3.4 Communications Interfaces

The communications interfaces section outlines the requirements for communication functions associated with the software product, detailing various protocols, message formats, and security measures.

Communication Functions

Email Communication:

Functionality:

Sending email notifications for account activities (e.g., registration confirmation, password reset links, project submission updates).

Email Protocols:

SMTP (Simple Mail Transfer Protocol): Used for sending emails.

IMAP (Internet Message Access Protocol): Used for receiving emails (if applicable).

Message Formatting:

Use standard MIME types for formatting email content (HTML for rich text and plain text as fallback).

Web Browser Communication:

Protocol:

HTTP/HTTPS: All communication between the client and server will use HTTPS for secure transmission of data.

Message Formatting:

REST API Responses: JSON format for all API responses, including error messages and data retrieval.

Network Server Communications:

Communication Protocols:

HTTP/HTTPS: For standard API requests and responses.

Data Transfer Rates:

Optimize network calls to ensure efficient data transfer rates, aiming for response times under 300 ms for API requests.

Communication Standards

Data Exchange Standards:

JSON (JavaScript Object Notation): Used for all API data interchange to ensure compatibility and ease of integration.

XML (eXtensible Markup Language): May be used for specific cases where required by external APIs or integrations.

File Transfer Protocols:

FTP/SFTP (Secure File Transfer Protocol): For any file uploads (if applicable), using SFTP to ensure secure file transfer.

Security and Encryption

Data Security Measures:

Encryption Standards:

TLS (Transport Layer Security): Implemented for HTTPS to secure data in transit between clients and servers.

AES (Advanced Encryption Standard): Use for encrypting sensitive data at rest, such as user passwords and personal information.

Authentication Protocols:

OAuth 2.0: For secure user authentication, especially if integrating third-party services.

JWT (JSON Web Tokens): For maintaining session security and managing user authentication.

Data Transfer Rates and Synchronization

Data Transfer Rates:

Aim for optimized performance with minimal latency for user actions, targeting load times under 2 seconds for the main application interfaces. Monitor server performance and implement caching strategies to enhance data retrieval speeds.

Synchronization Mechanisms:

Implement WebSocket connections for real-time data updates to ensure synchronization between the client and server, especially for notifications. Use polling as a fallback method to check for updates at defined intervals when WebSocket connections are not feasible.

4. System Features

This section organizes the functional requirements of the product into major features and services provided by the system. Each feature is described with its purpose, functionalities, and user interactions, ensuring a clear understanding of the product's capabilities.

4.1 User Management

Purpose: To manage user accounts and authentication processes.

Features:

User Registration: Users can create an account by providing personal information, including name, email, and password.

Validation: Ensure email format is correct and passwords meet security standards.

Login and Authentication:

Users can log in using their credentials. Implement secure authentication using JWT or OAuth 2.0.

Password Recovery:

Users can request a password reset link via email. Verification of email addresses to ensure security.

User Role Management:

Admins can manage user roles and permissions, such as admin or regular user.

4.2 Project Management

Purpose: To allow users to create, manage, and track projects.

Features:

Project Creation: Users can create new projects, specifying titles, descriptions, and relevant tags.

Project Submission: Users can submit their projects for review and approval.

Project Status Tracking: Users can view the status of their submitted projects (e.g., pending, approved, rejected).

Commenting and Feedback: Users can provide feedback on projects and engage in discussions.

4.3 Communication and Collaboration

Purpose: To facilitate communication between users and industry partners.

Features:

Email Notifications: Automatic notifications for important actions, such as project approvals or comments.

Real-time Notifications: Users receive real-time alerts for new messages or project updates through WebSocket connections.

4.4 Dashboard and Reporting

Purpose: To provide users and admins with an overview of activities and metrics.

Features:

User Dashboard: A personalized dashboard displaying user-specific information, such as submitted projects, notifications, and recent activity.

Admin Dashboard: Admins can view statistics on user registrations, project submissions, and approval rates.

Reporting Tools: Generate reports on user engagement, project success rates, and feedback trends.

4.5 Resource and Document Management

Purpose: To manage resources and documents related to projects.

Features:

File Uploads: Users can upload relevant documents, images, or presentations related to their projects.

Document Sharing: Share documents with other users or industry partners for collaboration.

Version Control: Keep track of different versions of documents and allow users to revert to previous versions if needed.

4.6 Security and Compliance

Purpose: To ensure the security and compliance of user data and interactions.

Features:

Data Encryption: Implement AES encryption for sensitive data storage and TLS for data in transit.

User Privacy Settings: Allow users to manage their privacy settings and control data sharing.

Compliance Monitoring: Ensure the application meets relevant regulations and standards (e.g., GDPR).

4.1.1 System Feature 1: **User Management**

Purpose: To manage user accounts and facilitate secure authentication.

Features:

User Registration: Users can create accounts by providing their personal information, including name, email, and password. Validation ensures correct email format and secure password strength.

Login and Authentication: Secure login using credentials with options for multi-factor authentication (MFA).

Password Recovery: Users can request a password reset link sent to their registered email address.

User Role Management: Admins can assign and modify user roles and permissions (e.g., admin, regular user).

4.1.2 Description and Priority

Description

The User Management feature enables users to create, authenticate, and manage their accounts within the system. It ensures secure access through user registration, login, password recovery, and role management, thus enhancing the overall user experience and security of the application.

Priority: High

Priority Component Ratings:

Benefit: 9

A robust user management system enhances user satisfaction and security, leading to higher retention and engagement.

Penalty: 8

Failure to implement this feature can lead to unauthorized access, data breaches, and significant user frustration.

Cost: 5

Development and maintenance costs are moderate but necessary for a secure application.

Risk: 7

Risks associated with security vulnerabilities and user data protection require careful implementation and testing.

4.1.3 Stimulus/Response Sequences

User Registration

Stimulus: User navigates to the "Sign Up" page and submits their registration details (e.g., username, email, and password).

Response: The system validates the input, creates a new user account, and displays a confirmation message or error message if the input is invalid (e.g., email already exists).

User Login

Stimulus: User enters their username/email and password on the "Login" page and clicks the "Login" button.

Response: The system verifies the credentials. If valid, the user is directed to the dashboard; if invalid, an error message is displayed.

Password Recovery

Stimulus: User clicks on the "Forgot Password?" link, enters their registered email, and submits the request.

Response: The system sends a password reset link to the provided email address and displays a confirmation message.

Account Verification

Stimulus: User receives an email with a verification link after registration and clicks on it.

Response: The system verifies the link, activates the user account, and redirects the user to the login page with a confirmation message.

Profile Update

Stimulus: User navigates to the "My Account" section, updates their profile information (e.g., email, password), and submits the changes.

Response: The system validates the input, updates the user profile in the database, and displays a success message or error message if the input is invalid.

Logout

Stimulus: User clicks the "Logout" button from any page.

Response: The system terminates the user session and redirects the user to the login page, displaying a logout confirmation message.

Role Management (Admin)

Stimulus: Admin user navigates to the "User Management" section, selects a user, and changes their role (e.g., from "User" to "Admin").

Response: The system updates the user role in the database and displays a confirmation message of the changes made.

4.1.4 Functional Requirements

FR-001: User Registration

Description: The system shall allow users to register by providing a username, email, and password.

Validation: Username must be unique. Email must be in a valid format and not already registered. Password must meet security criteria (minimum 8 characters, including uppercase, lowercase, and a special character).

Error Handling: If any validation fails, the system shall display appropriate error messages.

FR-002: User Login

Description: The system shall allow registered users to log in using their username or email and password.

Validation: Credentials must be verified against the database.

Error Handling: If credentials are invalid, the system shall display an "Invalid username or password" message.

FR-003: Password Recovery

Description: The system shall provide a password recovery mechanism by allowing users to request a password reset link via email.

Validation: Email must exist in the database.

Error Handling: If the email does not exist, the system shall display an appropriate message (e.g., "Email not found").

FR-004: Account Verification

Description: The system shall verify user accounts via a unique verification link sent to the user's email upon registration.

Functionality: The link must activate the user's account upon successful verification.

Error Handling: If the link is expired or invalid, the system shall inform the user accordingly.

FR-005: Profile Update

Description: The system shall allow users to update their profile information, including email and password.

Validation: Email must be in a valid format and unique (if changed).Password must meet security criteria.

Error Handling: The system shall display appropriate error messages for invalid inputs.

FR-006: Logout

Description: The system shall allow users to log out of their accounts.

Functionality: Upon logout, the user's session shall be terminated.

Error Handling: The system shall ensure that the user is redirected to the login page without any errors.

FR-007: Role Management (Admin)

Description: The system shall allow admin users to change user roles (e.g., user to admin).

Functionality: Admin must be able to select a user and update their role.

Error Handling: If the role update fails, the system shall display an error message.

FR-008: User Input Validation

Description: The system shall validate all user inputs across forms.

Functionality: Implement client-side and server-side validation for all input fields.

Error Handling: The system shall prevent submission of forms with invalid data and display relevant error messages.

FR-009: Data Security

Description: The system shall ensure the security of user data.

Functionality: All passwords must be encrypted before storage. Sensitive information must be transmitted securely (e.g., using HTTPS).

FR-010: Session Management

Description: The system shall manage user sessions effectively.

Functionality: Implement session timeouts after a period of inactivity.

Error Handling: The system shall prompt users to log in again if their session expires.

4.2.1 System Feature 2: Project Management
Description and Priority

Description: This feature enables users (primarily students and mentors) to create, manage, and track projects within the system, including the ability to add project details, update project statuses, and view project progress.

Priority: High

Benefit: 9

Penalty: 3

Cost: 4

Risk: 5

4.2.2 Stimulus/Response Sequences

Create Project

User Action: User clicks on "Create Project" button.

System Response: Displays a form to input project details (title, description, technology stack, etc.).

Update Project

User Action: User selects a project from the list and clicks "Edit."

System Response: Displays the project's current information in editable fields.

Change Project Status

User Action: User selects a project and updates its status (e.g., In Progress, Completed).

System Response: The system updates the project status and confirms the change.

View Project Details

User Action: User clicks on a project title.

System Response: Displays detailed information about the project, including team members, timelines, and milestones.

Delete Project

User Action: User selects a project and clicks "Delete."

System Response: Prompts a confirmation dialog and removes the project if confirmed.

4.2.3 Functional Requirements

FR-011: Create New Project

Description: The system shall allow users to create new projects by filling out a project information form.

Validation: All required fields must be completed (title, description, technology stack, etc.).

Error Handling: If the input is invalid, the system shall display relevant error messages.

FR-012: Update Existing Project

Description: The system shall enable users to update project details.

Validation: Changes must be validated against existing data.

Error Handling: The system shall inform the user of successful or failed updates.

FR-013: Change Project Status

Description: The system shall allow users to change the status of projects.

Functionality: Users can select from predefined status options (e.g., "In Progress," "Completed").

Error Handling: The system shall confirm the status change and notify if it fails.

FR-014: View Project List

Description: The system shall display a list of all projects with essential details.

Functionality: Users can filter and search through the project list.

FR-015: View Project Details

Description: The system shall provide detailed information about a selected project.

Functionality: Includes team members, timelines, milestones, and project documentation.

FR-016: Delete Project

Description: The system shall allow users to delete projects.

Functionality: Projects must be permanently removed from the database upon confirmation.

Error Handling: The system shall display an error if the deletion fails.

FR-017: Assign Team Members

Description: The system shall allow users to assign team members to projects.

Functionality: Users can select from a list of available team members.

Error Handling: If team member assignment fails, an error message shall be shown.

4.3.1 System Feature 3: Communication and Collaboration
Description and Priority

Description: Purpose: To facilitate communication between users and industry partners.

Priority: High

Benefit: 9

Penalty: 4

Cost: 5

Risk: 6

4.3.2 Stimulus/Response Sequences

Comment on Project

User Action: User clicks on a project and give stars.

System Response: The comment is saved and displayed under the project.

4.3.4 Functional Requirements

FR-018: Project Comments

The system shall allow users to comment on projects.

Validation: Comments must not exceed a certain character limit.

4.4.1 System Feature 4: Dashboard and Reporting
Description and Priority

Description: This feature provides users with a visual overview of project statuses, team performance, and key metrics through dashboards and customizable reports.

Priority: High

Benefit: 8

Penalty: 4

Cost: 5

Risk: 5

4.4.2 Stimulus/Response Sequences

View Dashboard

User Action: User accesses the dashboard from the main menu.

System Response: Displays a summary of project statuses and metrics.

4.4.3 Functional Requirements

FR-019: Dashboard Overview

The system shall provide a customizable dashboard displaying key project metrics.

Functionality: Users can select which metrics to display.

4.5.1 System Feature 5: Resource and Document Management
Description and Priority

Description: This feature allows users to manage project resources and documents effectively, including uploading, organizing, and sharing files and resources.

Priority: High

Benefit: 8

Penalty: 3

Cost: 4

Risk: 4

4.5.2 Stimulus/Response Sequences

Upload Document

User Action: User clicks "Upload" and selects a document.

System Response: The document is uploaded and stored in the system.

Organize Resources

User Action: User categorizes resources by project or type.

System Response: The system updates the resource organization.

4.5.3 Functional Requirements

FR-020: Document Uploading

The system shall allow users to upload documents and files.

Error Handling: Display an error if the upload fails.

FR-021: Document Organization

The system shall enable users to categorize and tag documents for easier retrieval.

Functionality: Users can filter documents by categories.

4.6.1 System Feature 6: Security and Compliance

Description and Priority

Description: This feature ensures that user data and project information are secure and comply with relevant regulations through authentication, authorization, and data encryption.

Priority: High

Benefit: 9

Penalty: 5

Cost: 6

Risk: 7

4.6.2 Stimulus/Response Sequences

User Authentication

User Action: User enters credentials and clicks "Login."

System Response: The system verifies credentials and grants access.

4.6.3 Functional Requirements

FR-022: User Authentication

The system shall require users to authenticate before accessing sensitive information.

Error Handling: Display an error if authentication fails.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance requirements outline the expected operational capabilities of the system under various conditions. These specifications ensure that the software performs efficiently, effectively, and reliably, meeting user expectations. Below are the key performance requirements for the system, along with their rationale.

Response Time

Requirement: The system shall respond to user inputs (e.g., login, form submissions) within **seconds** under normal operating conditions.

Rationale: Quick response times enhance user experience and satisfaction, reducing frustration and promoting efficient workflow. Users expect a seamless experience, especially in high-demand applications.

Data Handling

Requirement: The system shall be capable of processing **large datasets** (up to **1 GB** of data) during file uploads and downloads within **5 seconds**.

Rationale: With many projects generating substantial documentation, efficient data handling and processing are crucial. This requirement helps ensure that users can upload or retrieve documents without significant delays.

Scalability

Requirement: The system shall be able to scale horizontally to support an increase in users up to **10,000 concurrent users** without performance degradation.

Rationale: As the user base grows, the system must be able to accommodate additional users. This scalability ensures longevity and adaptability of the software to future demands.

Availability

Requirement: The system shall maintain an availability rate of **99.9%**, excluding scheduled maintenance.

Rationale: High availability is crucial for user trust and continuous access to the application. This ensures that users can rely on the system for their needs without interruption.

Load Handling

Requirement: The system shall remain fully functional under a simulated load of **150% of maximum expected usage** for a duration of **2 hours**.

Rationale: This load handling capability ensures the system can cope with unexpected spikes in user activity, ensuring reliability during critical usage periods.

Security Performance

Requirement: The system shall perform security checks (e.g., data encryption, user authentication) within **1 second** for each action requiring security verification.

Rationale: Security measures must be efficient and not disrupt user experience. Fast security performance is essential to maintain user trust while ensuring data protection.

Data Backup and Recovery

Requirement: The system shall perform data backup operations within **15 minutes** and allow for data recovery in less than **30 minutes** after a failure.

Rationale: Quick backup and recovery times are crucial for protecting user data and minimizing downtime, ensuring business continuity.

5.2 Safety Requirements

The safety requirements address potential risks associated with the use of the product, ensuring that it operates securely and protects users from harm, loss, or damage. These requirements focus on identifying hazards, implementing safeguards, and complying with relevant regulations and certifications.

Data Protection and Privacy

Requirement: The system shall encrypt all sensitive user data (e.g., personal information, financial data) using industry-standard encryption algorithms (e.g., AES-256).

Rationale: To protect user data from unauthorized access and breaches, ensuring compliance with data protection regulations such as GDPR and HIPAA.

User Authentication and Access Control

Requirement: The system shall implement multi-factor authentication (MFA) for all user accounts, particularly for administrative access.

Rationale: To prevent unauthorized access to sensitive areas of the application and protect user accounts from potential breaches.

Error Handling and Notifications

Requirement: The system shall display clear and actionable error messages to users, indicating the nature of the problem and recommended next steps without exposing sensitive information.

Rationale: Proper error handling helps users understand issues without compromising security, minimizing frustration and potential misuse.

Secure Software Development Practices

Requirement: The development process shall adhere to secure coding standards and undergo regular security audits to identify and mitigate vulnerabilities.

Rationale: To prevent software vulnerabilities that could be exploited by malicious actors, ensuring the integrity and safety of the product.

5.3 Security Requirements

The security requirements outline the necessary measures to protect the product, its data, and its users from unauthorized access, data breaches, and other security threats. These requirements focus on user authentication, data protection, compliance with regulations, and adherence to best practices in security.

User Authentication

Requirement: The system shall implement robust user authentication mechanisms, including support for multi-factor authentication (MFA) for all user accounts, especially for administrative and sensitive operations.

Rationale: MFA enhances security by requiring users to provide two or more verification factors, reducing the risk of unauthorized access.

Data Encryption

Requirement: The system shall encrypt sensitive data at rest and in transit using industry-standard encryption protocols (e.g., AES-256 for data at rest, TLS 1.2 or higher for data in transit).

Rationale: Encryption protects sensitive information from unauthorized access and ensures that data is secure during transmission and storage.

Access Control

Requirement: The system shall implement role-based access control (RBAC) to ensure that users have access only to the data and functionalities necessary for their roles.

Rationale: RBAC minimizes the risk of unauthorized access to sensitive data and ensures that users can only perform actions relevant to their responsibilities.

Data Integrity

Requirement: The system shall utilize checksums and cryptographic hashes to verify the integrity of data being stored and transmitted.

Rationale: Ensuring data integrity prevents unauthorized alterations and helps maintain trust in the data being used by the system.

Security Auditing and Logging

Requirement: The system shall maintain comprehensive logging of user actions and security events, with the ability to generate audit reports for review.

Rationale: Logging is essential for identifying potential security incidents, tracking user activities, and ensuring accountability.

Regular Security Assessments

Requirement: The organization shall conduct regular security assessments, including vulnerability scans and penetration testing, to identify and remediate security weaknesses.

Rationale: Proactive identification of vulnerabilities helps in strengthening the system's security posture and mitigating potential threats.

Incident Response Plan

Requirement: The organization shall establish an incident response plan that outlines procedures for identifying, managing, and recovering from security incidents.

Rationale: A well-defined incident response plan enables quick action to mitigate the impact of security breaches and protects user data.

5.4 Software Quality Attributes

Usability: The platform should offer a user-friendly interface that enables users to navigate easily and accomplish tasks without extensive training. Aiming for a System Usability Scale (SUS) score of 80 or above will indicate high usability.

Reliability: The system must ensure high availability, targeting an uptime of 99.9% to minimize disruptions for users accessing the platform. This includes robust error handling and fallback mechanisms to maintain functionality.

Maintainability: The software should be structured to allow for easy updates and modifications, aiming for a Mean Time to Repair (MTTR) of less than 2 hours for any critical issues that arise post-deployment.

Performance: The platform should load pages within 2 seconds under normal conditions, ensuring a responsive user experience. Additionally, it should support a minimum of 500 concurrent users without degradation of performance.

Adaptability: The system should allow for easy integration with third-party applications and services, enabling the addition of new features or updates without significant code refactoring.

Interoperability: The software must function seamlessly across different web browsers and devices (desktop and mobile) with no more than a 5% variance in user experience between platforms.

Security: The platform should implement industry-standard security measures, such as encryption for sensitive data and regular vulnerability assessments, achieving compliance with OWASP Top Ten security guidelines.

Testability: The software should be designed to facilitate automated testing processes. At least 80% code coverage through unit tests is the target, ensuring that critical functionalities are well-tested.

Robustness: The system should handle unexpected inputs gracefully, ensuring no system crashes or data loss occurs. It should include validation checks to prevent erroneous data submissions.

Portability: The application should be easily deployable across various cloud platforms and local servers, with a deployment time of less than 30 minutes on any supported environment.

5.5 Business Rules

User Roles and Permissions:

Admin: Can create, read, update, and delete any project submissions, manage user accounts, and view all analytics and feedback.

Faculty: Can create and manage their own projects, provide feedback on student submissions, and view student performance metrics.

Students: Can submit their projects, view feedback on their submissions, and collaborate with peers on group projects.

Guests: Can view publicly available projects but cannot submit projects or provide feedback.

Project Submission Guidelines:

All project submissions must be reviewed and approved by at least one faculty member before being publicly accessible. Projects must include specific components such as an abstract, detailed description, and relevant multimedia content (images/videos) to be considered complete.

Feedback Mechanism:

Feedback provided by faculty must be constructive and aim to enhance student learning, adhering to the platform's guidelines on tone and content. Students are required to acknowledge feedback within a specified time frame (e.g., one week) and respond with any necessary revisions.

User Authentication:

All users must register and authenticate their accounts using a valid university email address to access platform features. Passwords must meet specific security criteria (minimum length, complexity) to ensure user account security.

Data Retention Policy:

User data, including project submissions and feedback, will be retained for a period of five years after the last activity before being archived or deleted in compliance with university data protection policies.

Analytics Access:

Only admins and faculty can access project analytics to ensure that student data remains confidential and is used solely for educational purposes.

Usage Reporting:

Regular reports on user engagement, project submissions, and feedback must be generated and reviewed quarterly by the administrative team to assess the platform's effectiveness and user satisfaction.

These business rules establish the framework within which the STEM Project Showcase operates, ensuring that all users engage with the platform in a manner that promotes a positive and productive educational environment.

6. Resources

- Github
- Behance
- Google Scholar
- Google Classroom