**Software Requirements Specification**

**for**

**A Website for Final Year Students to Apply for MU Faculty Projects**

**Version 1.0**

**Prepared by**

**Group Name: <*place your group name here*>**

|  |  |  |
| --- | --- | --- |
| **Trisha Gumidelli** | **SE22UARI176** | **se22uari176@mahindrauniversity.edu.in** |
| **Swathika Reddy** | **SE22UARI172** | **se22uari172@mahindrauniversity.edu.in** |
| **Basam Nanda** | **SE22UARI028** | **se22uari028@mahindrauniversity.edu.in** |
| **Likesh Koya** | **SE22UARI210** | **se22uari210@mahindrauniversity.edu.in** |
| **Satwik Chitla** | **SE22UARI154** | **se22uari154@mahindrauniversity.edu.in** |

|  |  |
| --- | --- |
| **Instructor:** | **Avinash Arun Chauhan** |
| **Course:** | **Software Engineering** |
| **Lab Section:** | ***<place your lab section here>*** |
| **Teaching Assistant:** | **Murali Krishna Bukkasamudram** |
| **Date:** | **21.02.2025** |

**ContentsContents ii**

**Revisionsii**

**1** **Introduction 1**

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

**2** **Overall Description 2**

2.1 Product Overview 2

2.2 Product Functionality 3

2.3 Design and Implementation Constraints 3

2.4 Assumptions and Dependencies 3

**3** **Specific Requirements 4**

3.1 External Interface Requirements 4

3.2 Functional Requirements 4

3.3 Use Case Model 5

**4** **Other Non-functional Requirements 6**

4.1 Performance Requirements 6

4.2 Safety and Security Requirements 6

4.3 Software Quality Attributes 6

**5** **Other Requirements 7**

**Appendix A – Data Dictionary 8**

**Appendix B - Group Log 9**

* **Introduction**
* *This Software Requirements Specification (SRS) document outlines the objectives, functionality, and requirements of a web-based system designed to help final-year students at MU explore and apply for research or capstone projects offered by faculty members. The document provides a detailed overview of the system’s purpose, scope, and specifications to ensure successful implementation and deployment.*
* **Document Purpose**

The purpose of this document is to define the functional and non-functional requirements of the Final Year Project Application Website. It serves as a reference for developers, testers, and stakeholders involved in the project. The document describes the scope, intended audience, assumptions, constraints, and specific requirements necessary for building the system.

* **Product Scope**
* *The website is intended to provide a structured platform for students to browse, apply for, and track faculty-offered projects while allowing faculty members to manage student applications and select candidates efficiently. Key benefits include*
* *Streamlining the project application process.*
* *Improving transparency and communication between students and faculty.*
* *Reducing paperwork and administrative workload.*
* *Ensuring a fair and efficient selection process*
* Enabling the faculty to recommend external projects for students to work individually.
* Enable the faculty to group all the selected members and form a team for a smoother way of communication.
* **Intended Audience and Document Overview**

This document is intended for software developers, project managers, system testers, university faculty, and students who will use or maintain the system. It is organized into sections covering overall system description, functional and non-functional requirements, and constraints.

* **Definitions, Acronyms and Abbreviations**
* **Document Conventions**

This document follows IEEE formatting requirements, using Arial font size 11 and single spacing. Section titles are bolded, and key technical terms are italicized for emphasis.

* **References and Acknowledgments**

IEEE Software Engineering Standards

University guidelines for project applications

References to similar project management tools

* **Overall Description**
* **Product Overview**

The Final Year Project Application Website is a new, self-contained system designed to replace manual project applications with an online solution. The system will allow faculty members to create and manage project listings, while students can browse and apply for available projects. Faculty will review applications and notify selected students via the system.

* **Product Functionality**

User Authentication: Secure login for students, faculty, and admins.

Project Listings: Faculty can add, edit, and remove projects with descriptions.

Student Applications: Students can apply for projects, view status updates, and track applications.

Faculty Selection: Faculty can review applications, shortlist students, and approve final selections.

Notifications: Email or in-system alerts for application status updates.

Admin Management: User management and system maintenance.

* **Design and Implementation Constraints**

The system will follow a layered architecture to separate concerns (e.g., presentation, business logic, data access).

Frontend: Built using HTML and CSS, focusing on mobile-friendly and responsive design.

Backend: Developed using FastAPI for asynchronous API endpoints.

Data Management: Project and user data will be stored and managed using CSV files during initial development (extendable to a database in future).

Security: Secure login and token-based authentication (e.g., using OAuth2 or JWT).

Data Privacy: Sensitive data will be encrypted and transferred via HTTPS-secured endpoints.

.

* **Assumptions and Dependencies**

Faculty will actively list projects on the system.

Students will use the platform for project applications.

The university provides necessary authentication API access.

Internet access is available for users.

* **Specific Requirements**
* **External Interface Requirements**
* **User Interfaces**

Web-based, mobile-friendly dashboard for students and faculty.

* **Hardware Interfaces**

Standard web browser access (PCs, tablets, smartphones).

* **Software Interfaces**

FastAPI-powered backend for frontend interaction and CSV file manipulation.

* **Functional Requirements**

The system shall allow faculty members to create, update, and delete project listings.

The system shall allow students to browse projects and submit applications.

The system shall allow faculty members to review and approve applications.

The system shall provide automated notifications for application updates.

The system shall ensure secure login and authentication for all users.

* **Case Model**

A UML Use Case Diagram will be included, showing interactions between students, faculty, and admins with the system.

* **Other Non-functional Requirements**
* **Performance Requirements**

The system should support simultaneous access for at least 100 users.

Application submission processing time should not exceed 3 seconds.

* **Safety and Security Requirements**

Role-based access control to ensure proper user privileges.

Data encryption for user credentials and application details.

Regular security audits to prevent unauthorized data breaches.

* **Software Quality Attributes**

Reliability: The system should have 99.5% uptime for uninterrupted access.

Usability: The UI should be intuitive and easy to navigate for non-technical users.

Scalability: The platform should accommodate future feature expansions.

* **Other Requirements**

Database backups should be maintained to prevent data loss.

The platform should comply with university regulations on project selection.

**Appendix A – Data Dictionary**

A table defining key system attributes such as User ID, Project ID, Status, Role, etc. will be included.

**Appendix B - Group Log**

A log of meetings, discussions, and development updates will be maintained.

This document serves as the foundational requirement guide for the Final Year Project Application Website and will be refined as the project progresses.