# **Software Requirements Specification (SRS)**

## **ProjectHub: Collaborate, Contribute, Create**

**Version:** 1.0

**Prepared by:** Group-32

**Date:** 03-03-25

Group :

Hriday Macha-SE22UARI056, Aditya Shastry – SE22UARI008

Grisha Saini - SE22UARI218, Medha Potru – SE22UARI093

## **1. Introduction**

### **1.1 Document Purpose**

This SRS defines the functional, non-functional, and external requirements for **ProjectHub**, a web-based platform designed to help university students **discover, apply for, and collaborate** on academic and industry-level projects. It ensures structured project tracking, team formation, and mentor engagement.

### **1.2 Product Scope**

**ProjectHub** provides an **AI-driven recommendation system,** automated applications, team collaboration, and real-time tracking. The platform addresses the inefficiencies in **project discovery, team formation, and mentor connections**, offering seamless project lifecycle management.

### **1.3 Intended Audience and Document Overview**

This document is intended for **developers, project managers, university stakeholders, students, and industry partners.** It provides a structured overview of the system’s functionalities, technical constraints, and use cases.

### **1.4 Definitions, Acronyms, and Abbreviations**

* **AI:** Artificial Intelligence
* **ML:** Machine Learning
* **ProjectHub:** The proposed system for project collaboration
* **Mentor:** Industry professionals or faculty guiding students
* **SOW:** Statement of Work

### **1.5 Document Conventions**

This document follows **IEEE SRS standards**, with sections structured as per best practices.

### **1.6 References and Acknowledgments**

* **Statement of Work (SOW) - Group 32**
* **IEEE SRS Guidelines**

## **2. Overall Description**

### **2.1 Product Overview**

**ProjectHub** is a web-based platform designed to **streamline project discovery, team formation, and mentorship** for university students. It enables students to **find and apply for relevant academic and industry projects** through an AI-driven recommendation system, ensuring that opportunities align with their skills and interests. The platform also **facilitates collaboration** by connecting students with mentors, peers, and industry professionals, offering **built-in communication tools and project tracking features**. **2.2 Product Functionality**

* **Smart Project Matching:** AI-driven recommendations based on student skills, interests, and experience level.
* **One-Click Application:** Unified student profiles for seamless project applications.
* **Community Collaboration:** Peer mentorship, team formation, and Built in chat forums.
* **Direct Mentor Access:** Built-in chat and Q&A forums for guidance.
* **Productivity Tools:** Task management, milestone tracking, and collaborative document handling.

### **2.3 Design and Implementation Constraints**

* **Technology Stack:** Web-based (React, Django/MongoDB), AI for recommendations.
* **User Authentication:** Secure login via OAuth (Google, University SSO).
* **Hosting:** Cloud-based (AWS/Azure/GCP), Netlify.
* **Scalability:** Should support thousands of users simultaneously.
* **Front end:** Figma, Material Ui, JavaScript, Tailwind CSS, React JS
* **Back end:** Node JS, Express JS,MongoDB, Flask

### **2.4 Assumptions and Dependencies**

* Users should have the university mail ID to use the platform.
* AI-based recommendations require **sufficient user data**.
* **Third-party API integrations** (e.g., LinkedIn, ResearchGate) may be needed for enhanced functionality.

## **3. Specific Requirements**

### **3.1 External Interface Requirements**

* **User Interface (UI):** Intuitive dashboard for project browsing, applications, and tracking.
* **Hardware Interface:** Web-based platform, accessible via desktops/laptops.
* **Software Interface:** OAuth-based authentication, API integration for notifications and messaging.

### **3.2 Functional Requirements**

#### **F1: User Registration & Authentication**

* The system shall allow users to register using only University **Email Signup**.
* The system shall verify credentials before granting **dashboard access**.

#### **F2: Project Discovery & Matching**

* The system shall use **AI-based matching** to suggest relevant projects.
* Users shall be able to filter projects by **category, skill level, current year and mentor availability (Personal Projects)**.

#### **F3: Team Formation**

* Users can **form teams or join existing ones** based on complementary skill sets and connect on built in chat forums.

#### **F4: Application & Tracking**

* Users shall apply to projects with **one-click using pre-filled profiles**.
* The system will enable **real-time tracking** of applications and approvals from mentors.

#### **F5: Mentorship & Collaboration**

* The platform shall support **chat-based mentorship** and **discussion forums**.
* Mentors shall be able to **assign tasks and review project progress**.

#### **F6: Productivity Tools**

* Integrated **task management, deadlines, and shared documents**.
* Automated notifications for **project updates and deadlines**.

### **3.3 Use Case Model:**

Here is the **Use Case Model (3.3) for ProjectHub** based on the provided **use case diagram**.

## **3.3 Use Case Model**

### **Actors:**

* **Students**: Register, log in, apply for projects, manage profiles, and collaborate.
* **Professors/Mentors**: Guide students, participate in discussions, and review projects.
* **AI-Based System**: Provides project recommendations based on student profiles.
* **Administrator**: Manages projects, reviews applications, and ensures platform integrity.

### **3.3.1 Use Case #1: User Authentication (U1)**

#### **Purpose**

The purpose of this use case is to allow students and mentors to securely log in and access the platform.

#### **Requirements Traceability**

* F1: The system shall allow users to **register and authenticate** using Google, University Email, or Manual Signup.
* F2: The system shall verify credentials before granting access.

#### **Priority: High**

#### **Preconditions:**

* The user must be **registered** with a valid email.
* The system must have **an active authentication service**.

#### **Postconditions:**

* Upon successful authentication, the user is redirected to the **dashboard**.
* If authentication fails, an **error message** is displayed.

#### **Actors:**

* Students
* Professors/Mentors
* Administrator

#### **Flow of Events**

##### **1. Basic Flow**

1. User navigates to the **login page**.
2. User enters **email and password** or chooses **Google authentication**.
3. System **validates credentials**.
4. If successful, the user is **redirected to the dashboard**.

##### **2. Alternative Flow (Forgot Password)**

1. User clicks on **"Forgot Password"**.
2. System sends a **password reset link** to the registered email.
3. User resets the password and logs in successfully.

##### **3. Exceptions**

* **Invalid Credentials**: The system displays an error message.
* **Account Not Found**: Prompt user to **register**.

##### **Includes:**

* Profile & Settings (U3)

##### **Notes/Issues:**

* OAuth integration with Google and University SSO needs to be tested.

### **3.3.2 Use Case #2: Project Discovery & Matching (U2)**

#### **Purpose**

To allow students to search for and receive AI-driven **recommended projects**.

#### **Requirements Traceability**

* F2: AI-based **project matching**.
* F3: Search and filter functionality for **specific project criteria**.

#### **Priority: High**

#### **Preconditions:**

* User must be **logged in**.
* Projects must be **available in the database**.

#### **Postconditions:**

* The user is presented with **a list of matching projects**.
* Users can **apply or save projects for later**.

#### **Actors:**

* Students
* AI-Based System

#### **Flow of Events**

##### **1. Basic Flow**

1. Student navigates to **Projects Page**.
2. AI system **queries the database** for projects based on:
   1. Skills
   2. Academic Level
   3. Interests
3. AI displays **recommended projects**.
4. Students can **view details** or apply directly.

##### **2. Alternative Flow**

* If no projects match, the system suggests **related categories**.

##### **3. Exceptions**

* **Database not accessible**: The system displays an error message.

##### **Includes:**

* Project Management/Tracking (U4)

##### **Notes/Issues:**

* Improve recommendation accuracy using **machine learning models**.

### **3.3.3 Use Case #3: Community Collaboration (U3)**

### **Purpose**

To allow students to **collaborate, communicate, and share knowledge**.

#### **Requirements Traceability**

* F4: The system shall provide **discussion forums and chat groups**.
* F5: Peer mentorship between **juniors and seniors**.

#### **Priority: Medium**

#### **Preconditions:**

* Users must have a **valid profile**.
* Users must be **part of a project group**.

#### **Postconditions:**

* Users can **post, reply, and share files** in discussions.
* Mentors can **review progress and provide feedback**.

#### **Actors:**

* Students
* Professors/Mentors

#### **Flow of Events**

##### **1. Basic Flow**

1. Student/mentor opens **Community Collaboration** section.
2. User joins or creates a **discussion group**.
3. Users exchange **messages, files, and updates**.
4. System **sends notifications** for new posts.

##### **2. Alternative Flow**

* If no discussion exists, the user can **create a new thread**.

##### **3. Exceptions**

* **Inappropriate content detected**: Admin intervention is required.

##### **Includes:**

* Group Chat & Discussion (U5)

##### **Notes/Issues:**

* Implement **AI-based moderation** for abusive content.

### **3.3.4 Use Case #4: Mentor Access & Guidance (U4)**

#### **Purpose**

To allow mentors to guide students through **real-time discussions and feedback**.

#### **Requirements Traceability**

* F6: Mentors must be able to **interact with students via chat**.
* F7: Mentors can **track project progress**.

#### **Priority: Medium**

#### **Preconditions:**

* Mentor must be assigned to a project.
* Students must be part of an **active project**.

#### **Postconditions:**

* Mentor has access to **project reports**.
* Students can request and receive **feedback**.

#### **Actors:**

* Professors/Mentors
* Students

#### **Flow of Events**

##### **1. Basic Flow**

1. Mentor logs onto the platform.
2. Mentor accesses **assigned projects**.
3. Students submit progress reports.
4. Mentor provides **feedback** via chat or discussion boards.

##### **2. Alternative Flow**

* If a mentor is **not assigned**, students can **request mentor access**.

##### **3. Exceptions**

* **Inactive mentor account**: Admin reassigns a new mentor.

##### **Includes:**

* Project Management/Tracking (U4)

## **4. Other Non-Functional Requirements**

### **4.1 Performance Requirements**

* The system shall support **simultaneous access** by **at least 500 users**.
* Project recommendations shall be **generated within 2 seconds**.

### **4.2 Safety and Security Requirements**

* **Data encryption** (AES-256) for user information.
* **Role-based access control** (RBAC) for student, mentor, and admin functionalities.
* **Automatic logout** after **30 minutes of inactivity**.

### **4.3 Software Quality Attributes**

#### **4.3.1 Usability**

* The UI shall follow **modern design principles (Material UI/Bootstrap)** for accessibility.
* The system shall be **mobile-responsive**.

#### **4.3.2 Reliability**

* **99.9% uptime guarantee** through cloud hosting.
* **Automated backups** to prevent data loss.

#### **4.3.3 Scalability**

* Designed for **future expansion** to support more universities and industries.

## **5. Other Requirements**

* **Database Requirements:** Cloud-hosted (MongoDB).
* **Legal Compliance:** GDPR and university data policies must be followed.