Requirement Document (SRS)

## 1. Introduction

### 1.1 Purpose

The purpose of this web application is to provide a centralized platform for Computer Science (CS) students where all essential academic resources such as online lecture , books, coding tutorials, project ideas, software tools, and useful external links are available in one place. This will save time, enhance accessibility, and improve learning efficiency for students.

### 1.2 Scope

The system will be a responsive web application accessible via desktop and mobile devices. It will allow students to browse, search, and download resources based on subjects, courses, and categories. Faculty may upload materials, and students can suggest or share resources (with admin approval). The system will also provide discussion forums and announcements related to academic activities.

### 1.3 Definitions, Acronyms, and Abbreviations

* **CS**: Computer Science
* **UI**: User Interface
* **DBMS**: Database Management System
* **API**: Application Programming Interface

### 1.4 References

* IEEE SRS Standard Template
* Online academic platforms (e.g., Coursera, GitHub Education, Stack Overflow)
* University guidelines for resource sharing

### 1.5 Overview

This document describes the requirements of a CS Student Resource Hub web application, including functional and non-functional requirements, system models, and constraints.

## 2. Overall Description

### 2.1 Product Perspective

This system will act as a central hub for academic resources. It is an independent application but may integrate with third-party services like Google Drive, GitHub, and YouTube.

### 2.2 Product Functions

* Resource upload/download (PDFs, PPTs, code files, etc.)
* Categorization of resources by subject/topic
* Search and filter functionality
* User roles: Admin, Faculty, Student
* Discussion forum for Q&A
* Notifications for new resources/announcements

### 2.3 User Classes and Characteristics

* **Students**: Can browse, search, download, and suggest resources.
* **Faculty**: Can upload verified academic resources.
* **Admin**: Manages user roles, approves uploads, maintains the system.

### 2.4 Operating Environment

* **Front-end**: Web browsers (Chrome, Firefox, Edge, Safari)
* **Back-end**: Node.js / Django / Spring Boot
* **Database**: MySQL / MongoDB
* **Hosting**: Cloud-based (AWS, Azure, or university server)

### 2.5 Design and Implementation Constraints

* Must support multiple file types (PDF, DOCX, PPT, ZIP).
* Resources must be categorized properly.
* Internet connection required.

### 2.6 User Documentation

* Online user manual with screenshots.
* FAQs section.
* Tutorials for students and faculty.

## 3. Specific Requirements

### 3.1 Functional Requirements

* **Requirement 1**: The system shall allow students to search and download resources.
* **Requirement 2**: The system shall allow faculty to upload resources with metadata (title, subject, tags).
* **Requirement 3**: The system shall provide a discussion forum.
* **Requirement 4**: The system shall notify users about new uploads.

### 3.2 Non-Functional Requirements

* **Requirement 1**: The system shall be secure and require login for uploads.
* **Requirement 2**: The system shall provide 99.9% uptime.
* **Requirement 3**: The system shall be responsive and mobile-friendly.
* **Requirement 4**: The system shall handle at least 500 concurrent users.

### 3.3 Interface Requirements

* **User Interface**: Web-based, responsive, minimal design.
* **Hardware Interface**: Compatible with PCs, laptops, smartphones.
* **Software Interface**: Integration with Google Drive, GitHub, YouTube.
* **Communications Interface**: HTTP/HTTPS protocols.

## 4. System Models

* **Use Case Diagrams**: Showing interactions between Student, Faculty, and Admin.
* **Entity Relationship Diagrams (ERD)**: Resource, User, Category, Discussion tables.
* **Other Models**: Sequence diagrams for upload/download processes.

## 5. Appendices

* **Glossary**
  + **Resource**: Any academic material (PDF, code, book, etc.).
  + **Forum**: A discussion space for Q&A among students and faculty.
* **Supporting Information**
  + System development tools: VS Code, GitHub, Node.js.
  + Reference to IEEE 830 SRS standard.