**CS256: Topics in AI**

**Homework Group Project**

**AI Knowledge Hub**

**System Design Document**

Submitted By: Udayan Atreya

Mansi Patel

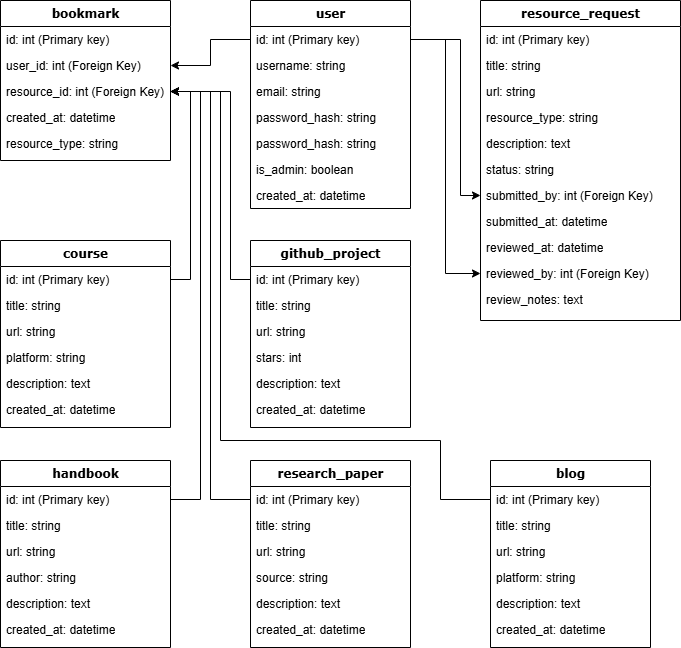
1. **Overview**

This document outlines the architecture and technology stack used in the project. The system is designed to provide a web-based application with a modern, responsive frontend, a robust backend using Python and Flask, and a persistent storage solution managed through SQLAlchemy with SQLite3. Continuous integration and deployment are facilitated through Render, while Git and GitHub manage source control and repository hosting.

1. **System Architecture**
   1. **Frontend**

* **Technologies:** HTML, CSS, JavaScript, Tailwind CSS
* **Role:**
  + Provides the user interface and interactive client-side experience.
  + Tailwind CSS is utilized to build responsive and modern designs quickly.
  + JavaScript enhances interactivity and dynamic content rendering.
  1. **Backend**
* **Technologies:** Python, Flask
* **Role:**
  + Serves as the core application server.
  + Handles HTTP requests, processes business logic, and manages API endpoints.
  + Implements server-side functionalities like authentication, routing, and data processing.
  1. **Database**
* **Technologies:** SQLAlchemy (ORM), SQLite3
* **Role:**
  + SQLAlchemy abstracts the database operations, enabling easier manipulation of data objects.
  + SQLite3 is used as the underlying relational database to persist application data.
  + The database schema includes multiple entities (e.g., User, Course, Handbook, GitHubProject, ResearchPaper, Blog, ResourceRequest, Bookmark) with defined relationships, as illustrated in the schema diagram below.

**Database Schema Diagram:**

**

* 1. **Continuous Integration and Continuous Deployment (CI/CD)**
* **Technology:** Render
* **Role:**
  + Automates the build, testing, and deployment processes.
  + Ensures that every code change is integrated, tested, and deployed seamlessly.
  1. **Source Control and Repository**
* **Technologies:** Git, GitHub
* **Role:**
  + Git manages version control to track changes in the codebase.
  + GitHub serves as the remote repository, enabling collaboration, code reviews, and issue tracking.

1. **Deployment Strategy**

* **Environment Setup:**
  + The application is deployed on Render, which manages the runtime environment, scaling, and deployment.
  + The CI/CD pipeline ensures that any new code commits trigger automated tests and deployment.
* **Maintenance:**
  + Continuous monitoring on Render helps maintain system performance and uptime.