Table of Contents

1. Introduction

The rapid expansion of software development roles demands students to continuously upgrade their practical skills. However, due to academic workload, lack of mentorship and limited peer collaboration, students often struggle to identify what skills they need to improve and how to work on real projects. Version-control platforms such as GitHub provide a rich source of real-world coding data, reflecting the student's technical involvement. Yet, most educational systems do not utilize this data for personalized learning insights.
This project aims to bridge that gap by analyzing students GitHub contributions to assess skill levels, recommend appropriate projects, and connect them with peers having complementary skills thereby fostering collaborative learning and stronger employability outcomes.

1.1 Objective of the Project

- To analyze student technical skills using GitHub contribution metrics.
- To identify and visualize individual skill gaps through ML-based profiling.
- To recommend personalized projects aligned with improvement areas.
- To match students with peers of similar or complementary skill sets for collaboration.
- To continuously track progress and update recommendations in real-time.

1.2 Description of the Project

SkillBridge is a web-based intelligent recommendation system that uses GitHub activity logs (commits, pull requests, issues resolved, languages used) to generate a dynamic Skill Vector for each student. Machine learning pipelines are utilized to rank skills, compare them with project requirements, and automatically recommend suitable project opportunities. Collaboration matching is enabled through clustering algorithms to form balanced project teams.
A continuous feedback dashboard monitors progress and updates skill gaps as students contribute to new repositories. The platform ultimately motivates students to engage in hands-on projects while reducing dependency on traditional instruction.

1.3 Scope of the Project

This project can be adopted by:

- Universities for practical skill development tracking
- Coding bootcamps for team formation and project allocation
- Student communities for finding collaborators
- Placement/support centers to evaluate readiness for real-world roles

Key Deliverables:

- Full-stack web platform
- Skill analysis model (ML-based)
- Project recommendation engine
- Peer-matching module
- Dashboard with real-time analytics

1.3.1 Use Case Model

Actors:

- Student/User
- System (Recommendation Engine)
- GitHub API

Primary Use Cases:

- Connect GitHub account
- View Skill Analysis
- Receive Project Recommendations
- Find Collaboration Partner
- Track Progress & Feedback

2. System Description

2.1 Customer/User Profiles

User Type	Description	Expected Benefits
Students	Developers wanting to improve their skills	Skill gap insights, team matching
University Faculty	Academic evaluators	Objective assessment of practical learning
Coding Groups/Clubs	Developer communities	Smart team formation, project organization
Recruiters (Future scope)	Hiring professionals	Evidence-based skill validation

2.2 Assumptions and Dependencies

- Users have an active GitHub account with publicly visible contributions.
- GitHub API must be available for authentication and data retrieval.
- Users show consistent coding behavior to ensure ML model accuracy.
- Internet connectivity required for real-time dashboard and sync.

2.3 Functional Requirements

ID	Requirement	Description
FR01	GitHub Authentication	Secure OAuth connection to user profile
FR02	Fetch Contribution Data	Extract commits, PRs, issues, languages
FR03	Generate Skill Vector	ML-based skill classification
FR04	Project Recommendation	Suggest projects based on skill gaps
FR05	Collaboration Matching	Identify peers with compatible skills
FR06	Progress Tracking	Update dashboard with development metrics

2.4 Non-Functional Requirements

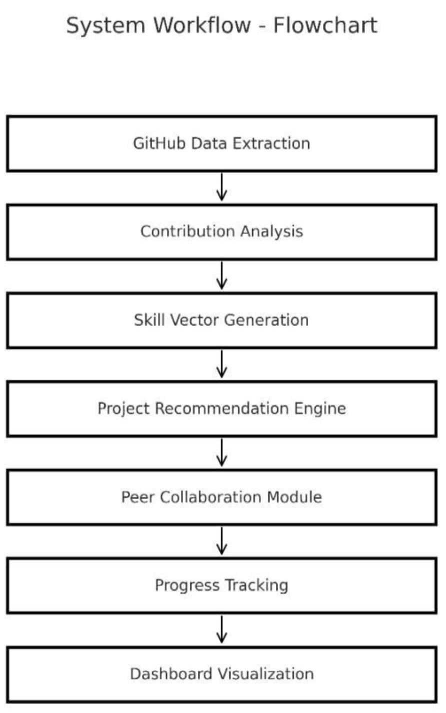
Parameter	Description
Performance	Real-time recommendation and fast analytics
Scalability	Supports growing number of users/projects
Reliability	Accurate skill analysis and system uptime
Security	OAuth security and encrypted user data
Usability	Intuitive interface for beginners
Portability	Cloud-ready architecture (AWS/others)

3. Design

3.1 System Design

The proposed SkillBridge system follows a modular and scalable architecture to ensure efficient skill analysis, personalized project recommendations, and collaboration support. The system includes a frontend for user interaction, a backend for processing GitHub data and generating recommendations, and a database for storing analyzed results. The design ensures secure API communication, real-time analytics, and continuous improvement of user skills based on updated GitHub contributions.

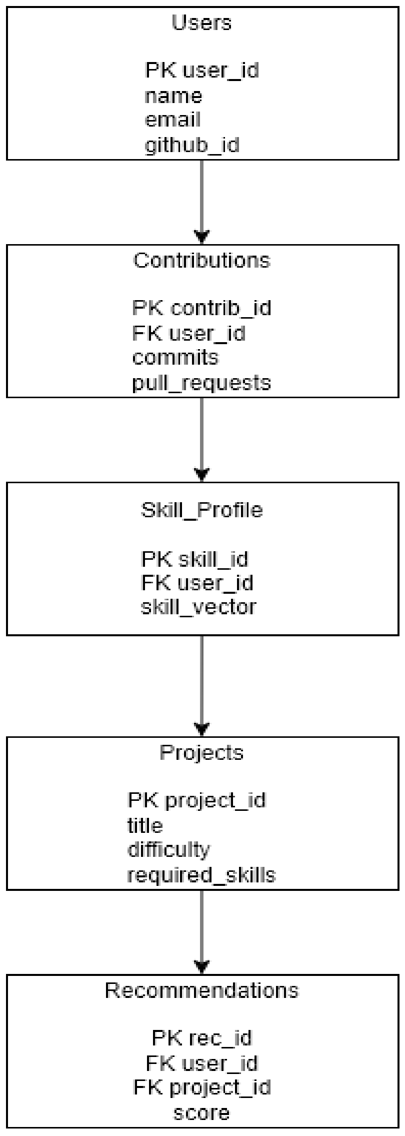
System Architecture Diagram



3.1.1 E-R Diagram

The Entity-Relationship (ER) diagram models the logical relationship between key database entities including Users, Contributions, Skill Profile, Projects, Recommendations, and Collaboration. It ensures structured data storage and efficient retrieval required for machine learning algorithms and analytics visualization.

ER Diagram - SkillBridge System

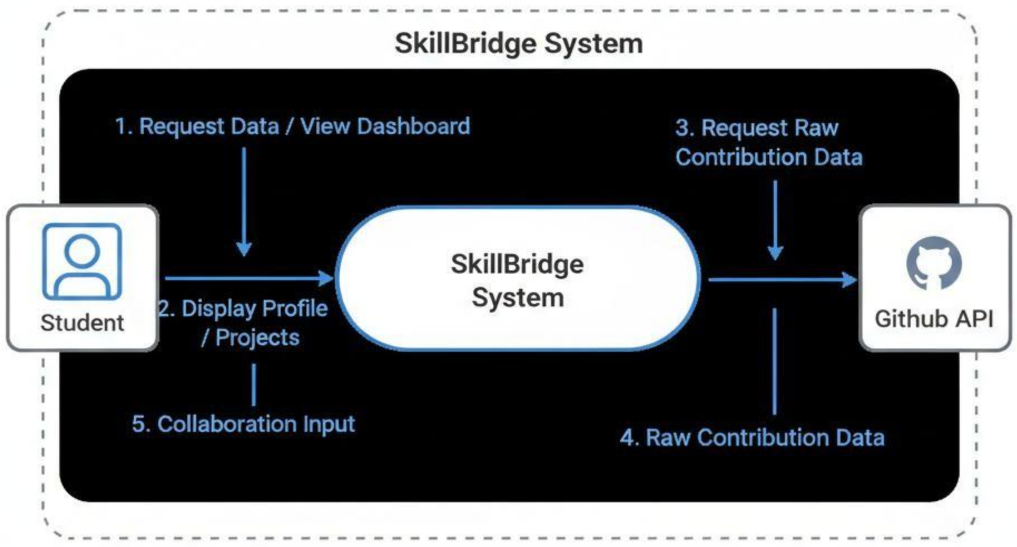


3.1.2 DFD's

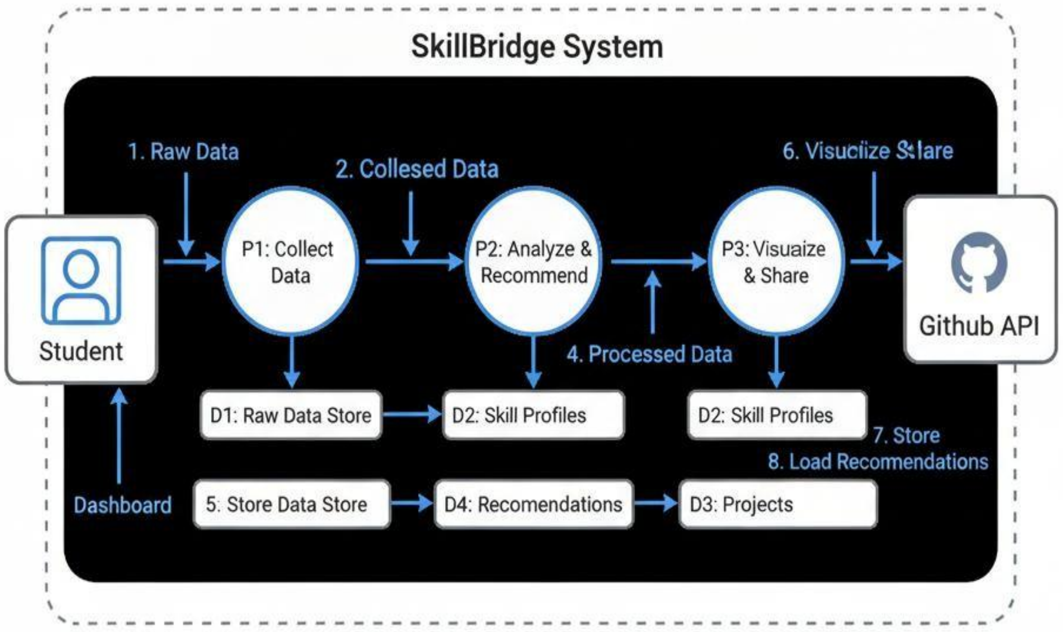
The Data Flow Diagrams represent the flow of data through the SkillBridge system.

- DFD Level-0 shows a high-level overview of interactions between the Student, GitHub API, and the SkillBridge System.
- DFD Level-1 illustrates internal processes such as data collection, skill analysis, recommendation generation, and progress visualization, including data storage interactions.

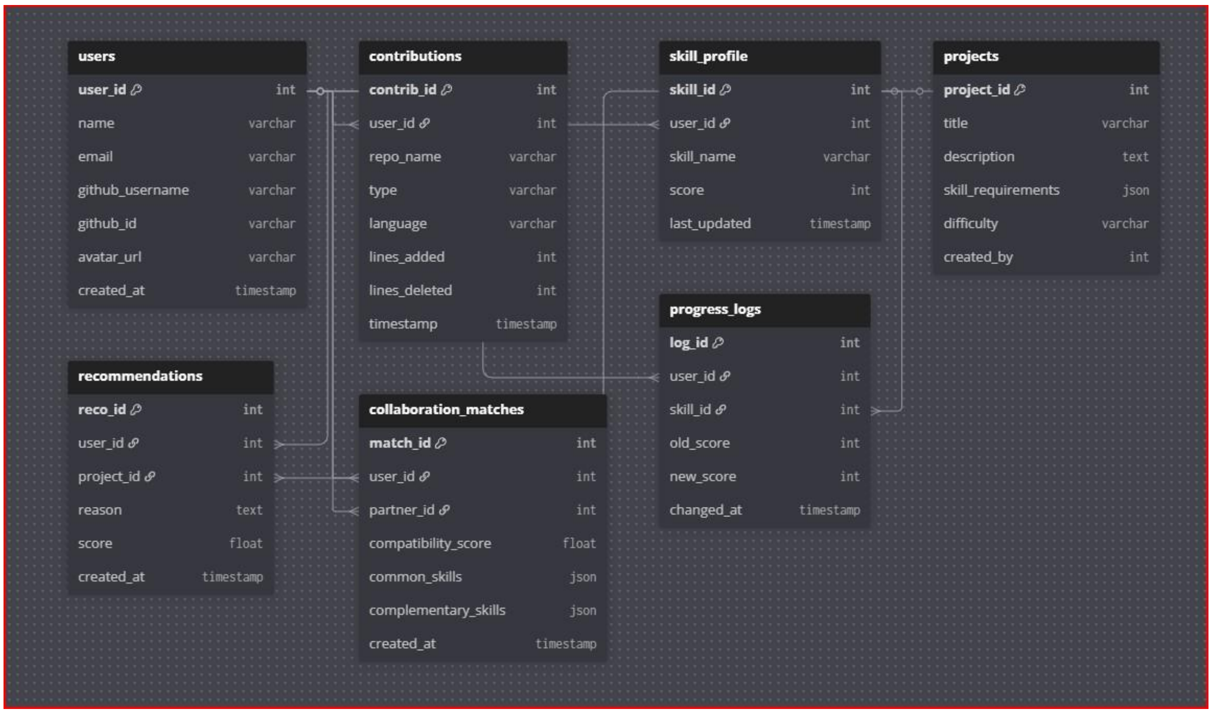
DFD LEVEL - 0



DFD LEVEL - 1



3.2 DB DESIGN



SECTION 4 – SCHEDULING AND ESTIMATES

4.1 Project Schedule

The project development is divided into structured phases ensuring systematic progress from requirement analysis to deployment.

Phase	Tasks	Duration	Status
Phase 1	Literature Review + Requirement Analysis	2 Week	Completed
Phase 2	System Design (Architecture, ERD, DFD)	3 Week	Completed
Phase 3	Backend Development + GitHub Integration	4 Weeks	In Progress
Phase 4	Machine Learning Model Development	2 Weeks	Upcoming
Phase 5	Frontend UI with Dashboard	2 Weeks	Upcoming
Phase 6	Testing, Debugging & Evaluation	1 Week	Upcoming

Phase	Tasks	Duration	Status
Phase 7	Deployment & Final Documentation	2 Week	Final Stage

4.2 Effort & Resource Estimates

The project uses free and open-source tools, minimizing cost while ensuring robust performance.

Resource	Description	Cost
Software Tools	ReactJs, PostgreSQL, Docker, BetterAuth, GitHub API	Free
Hardware	Personal laptop/Server(trying)	Existing
Cloud Hosting	Optional (AWS / Render / Netlify)	Free tier available

Section 5 – References

- Student Teamwork on Programming Projects.What can GitHub logs show us?
- A Comparative Analysis of GitHub Contributions Before and After An OSS Based Software Engineering Class
- An Investigation Into the Perceived Effectiveness of GitHub Repositories to Teach Programming Analysis of Student Pair Teamwork Using GitHub Activities
- Campus Placement Prediction and Analysis Using Machine Learning
- Campus Placement Prediction using Facebook Prophet and eXtreme Gradient Boost Algorithm
- Comprehensive Career Placement Predictor: An Analytical Tool for Optimizing Job Placement Outcomes
  Placement Prediction Using Machine Learning
- Placement Prediction & Analysis using Machine Learning
- GitHub in the Classroom: Lessons Learnt