**Software Requirement Specification for TAC Portal**

| **Name** | Mouneshwaran V |
| --- | --- |
| **Roll no** | 7376222IT202 |
| **Seat no** | 318 |
| **Project ID** | 38 |
| **Problem Statement** | Student Ranking Dashboard |

**1**. **Introduction**

**1.1. Purpose:**

The purpose of this document is to provide a comprehensive overview of the Student Ranking Dashboard, an essential tool for tracking and analyzing student performance within our college. It will give a thorough summary of the success of the students based on different coding and training activities, such as the number of LeetCodes, the placement, the rating of the training activities, the Codeforces rating, andCodeChef rating or star system. The dashboard's objectives are to monitor students' progress, promote skill development, and enable healthy competition.It includes the integration of a ranking algorithm for automated calculation of student rankings based on predefined criteria, along with features for real-time tracking of academic progress and extracurricular participation. The project scope also entails the implementation of robust security measures to safeguard student data and ensure compliance with regulatory requirements, all within the confines of specified timelines and budgetary constraints.

**2. System Overview:**

**2.1. Users:**

1. **Students**:

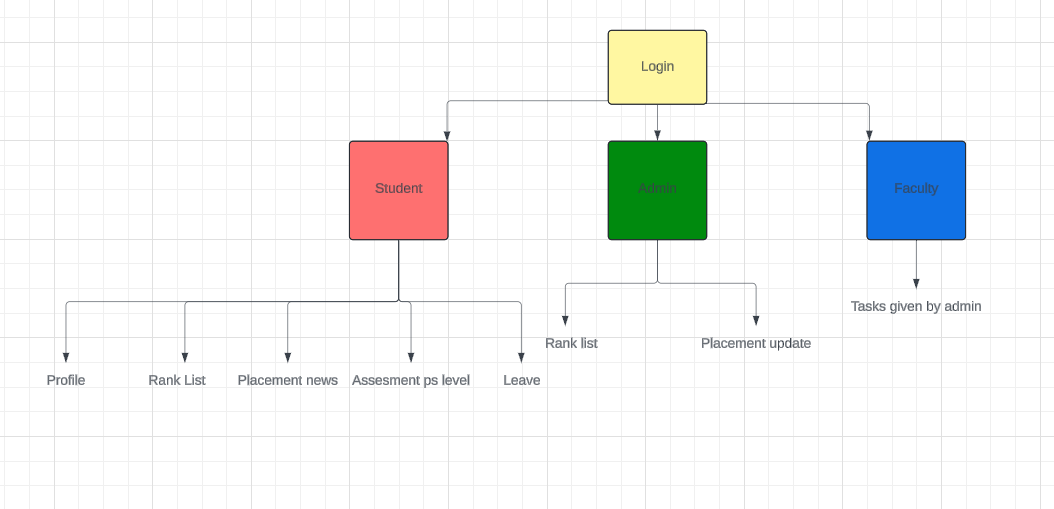
Students have access to their personal profiles.They can also access the top hundred student rank list to track their ranking relative to their peers.

**2. Faculties:**

Faculty members are empowered to assign tasks to students and provide grades or marks for completed assignments.They have access to view student profiles.Additionally, faculty members can access analytical dashboards for insights into student performance and engagement.

**3.Admins:**

Admins hold superuser privileges, allowing them to oversee and manage all aspects of the system.They have the authority to update rank of of the students or reject student marks with remarks.Admins can also manage appointments, schedule meeting, and access the system's database for administrative purposes.Furthermore, they have the capability to block student accounts, make changes to student profiles, and intervene in any system processes as necessary.



**2.2. Features:**

**User Profiles:**

Students, faculty, and administrators can register for an account and log in using their credentials.

**Task Assignment:**

Faculty members can assign tasks to individual students or groups, specifying deadlines and requirements.

**Task Submission:**

Students can submit completed tasks along with any necessary documents or materials for evaluation.

**Grading System:**

Faculty members can provide grades or marks for submitted tasks, which are then visible to students in their profiles.

**Profile Management:**

Administrators can view and edit students personal profiles, including ranking.3 information, contact details, and preferences.

**Ranking System:**

Students can view their rank relative to their peers based on the self-intensive program and their development in their fullstack project

**Task Progress Tracking:**

Students can track the progress of their assigned tasks, including deadlines, completion status, and feedback from faculty.

**Notification System:**

Users receive notifications for upcoming deadlines, task assignments, grade submissions, Placement updates and other relevant updates.

**Calendar Integration:**

Integration with a calendar system for scheduling appointments, deadlines for submission , placement activity and other important events.

**3. System Requirements Specification:**

**3.1 Functional Requirements:**

**User Profiles:**

Students, faculty, and administrators should be able to register for accounts and log in.Admins must have access control, including an analytical dashboard and dedicated features.

**Top Hundred Student Rank List:**

Students should be able to view the top hundred student rank list to track their Placement score and progress in self intensive training

**Task Assignment and Grading (Faculty):**

Faculty members should be able to assign tasks to students and provide grades or marks for completing tasks.

**Admin Privileges:**

Admins should have superuser privileges, allowing them to perform various actions:

Block a student's account.

View any student's profile.

Make changes to student profiles.

Access the database for administrative purposes.

**Database Access:**

Admins should have unrestricted access to the system's database for managing and retrieving data as necessary.

**3.2. Non-Functional Requirements:**

● **Performance**: The system must respond to user actions within 2 seconds to ensure efficient usability and must handle a concurrent user load of at least 100 users without significant performance degradation.

● **Security**: User data must be encrypted during transmission and storage, and access to sensitive functionalities should be restricted to authorized admin users through secure authentication mechanisms.

● **Usability**: The user interface should be intuitive and user-friendly, with clear and concise error messages provided to guide users in case of input errors or system failures.

● **Reliability**: The system should be available 24/7 with minimal downtime and should have a backup and recovery mechanism in place to prevent data loss in case of system failures or crashes.

● **Scalability**: The system should be designed to accommodate an increasing number of users and data volume over time, and it should be scalable to support additional features and functionalities as per future requirements.

**Backend:**

**1. Student entity**

| name | String |
| --- | --- |
| email | String |
| password | Hash code |
| Roll no | String |

**2. Faculty entity**

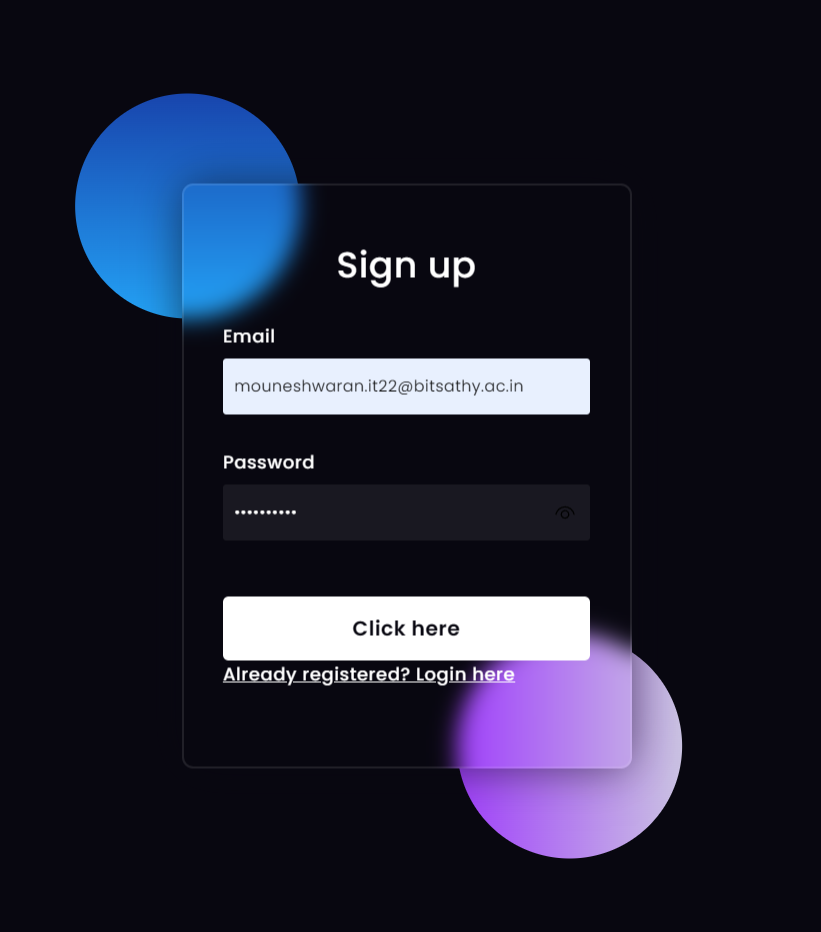
| name | String |
| --- | --- |
| email | String |
| password | Hash code |
| Faculty ID | String |

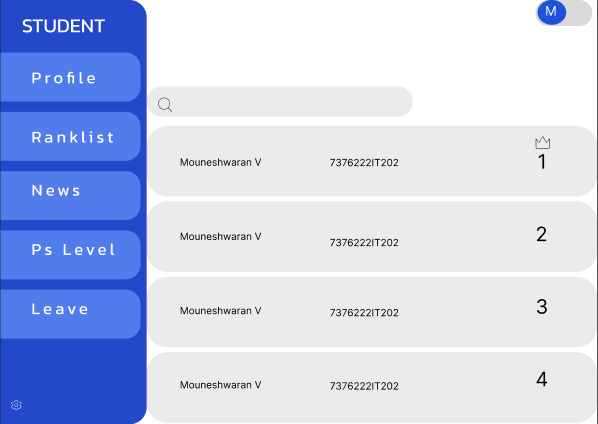
**Stack:**

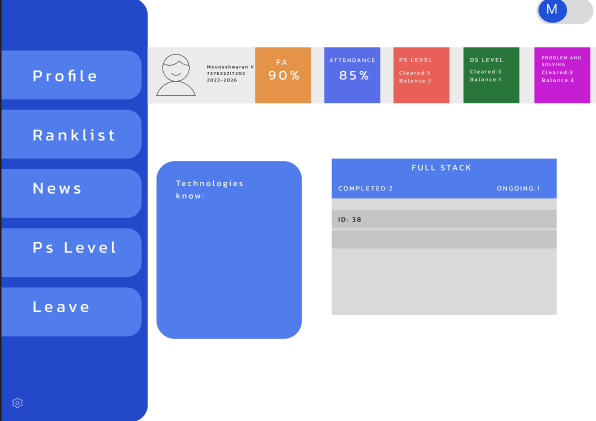
| Front End | Vue Js |
| --- | --- |
| Backend | Django |
| Data Base | Mysql |

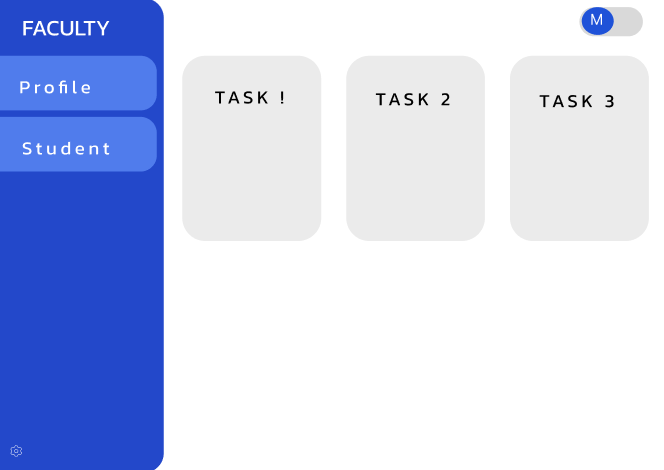
**Prototype of the Project:**

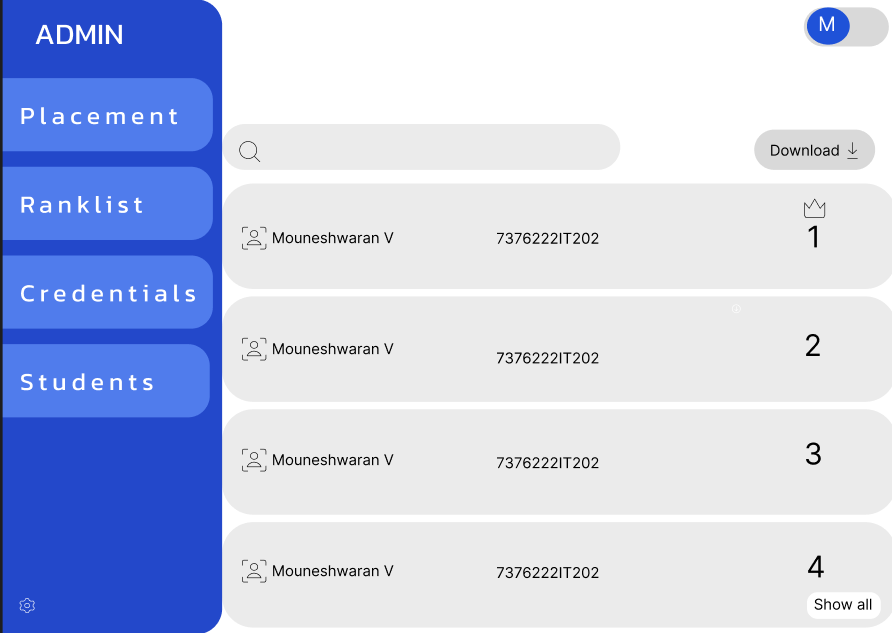
*1. Signup/Login form*

**

**

**

**

**