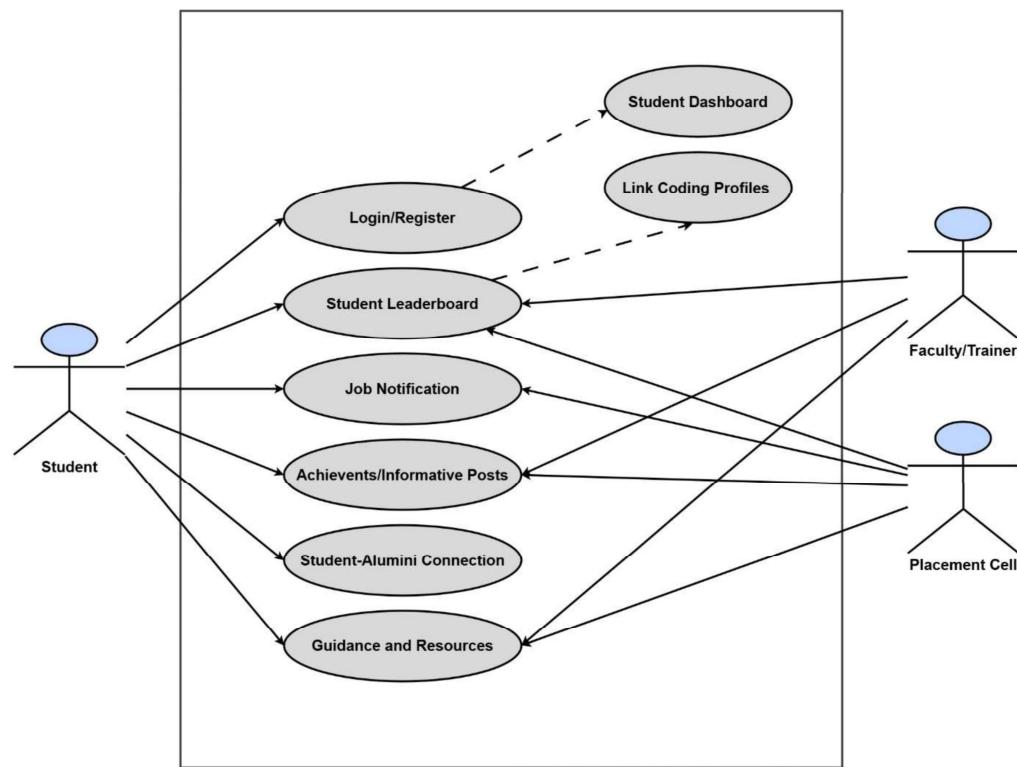


Printout

09 April 2025 13:52

- **Experiment-3**

Draw the use case diagram and specify the role of each actor.



Actor Roles in Campus Connect

1. STUDENT

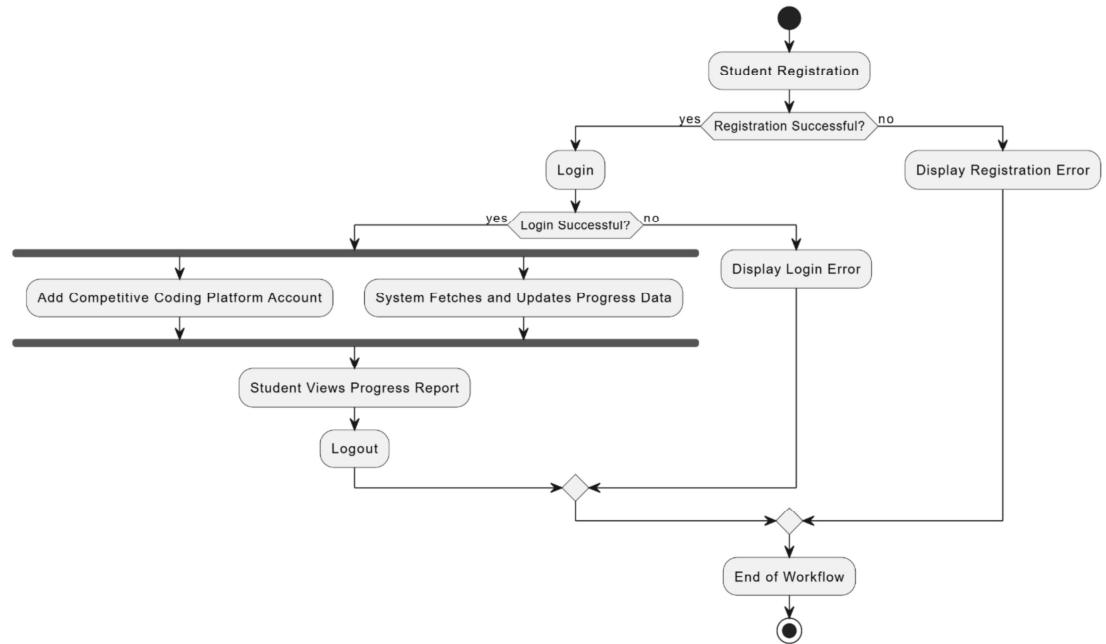
The student is an end-user who interacts with the Campus Connect to access library services. Their roles include:

- **Linking the coding platform:**
Student must link the Leetcode , gfg account with our dashboard portal
- **Job Notification :**
Student may get the notification from placement cell at job notification Page .
- **Achievements/Information Posts:**
Student may post there achievement and certification on the post portal that may visible to all student along with Faculty Member.

2. FACULTY

Experiment-5

Draw the activity diagram of Campus Connect.



Explanation of the Activity Diagram for Campus Connect

1. Start of Process

The process begins when a student logs into the Campus Connect portal (represented by the filled circle).

2. Student Accesses Profile Settings

The student navigates to their dashboard and selects the option to manage or link external coding profiles.

3. Request to Link Coding Platform

The student initiates the linking process by choosing a specific platform (e.g., LeetCode, GFG) and providing required credentials or public profile URLs.

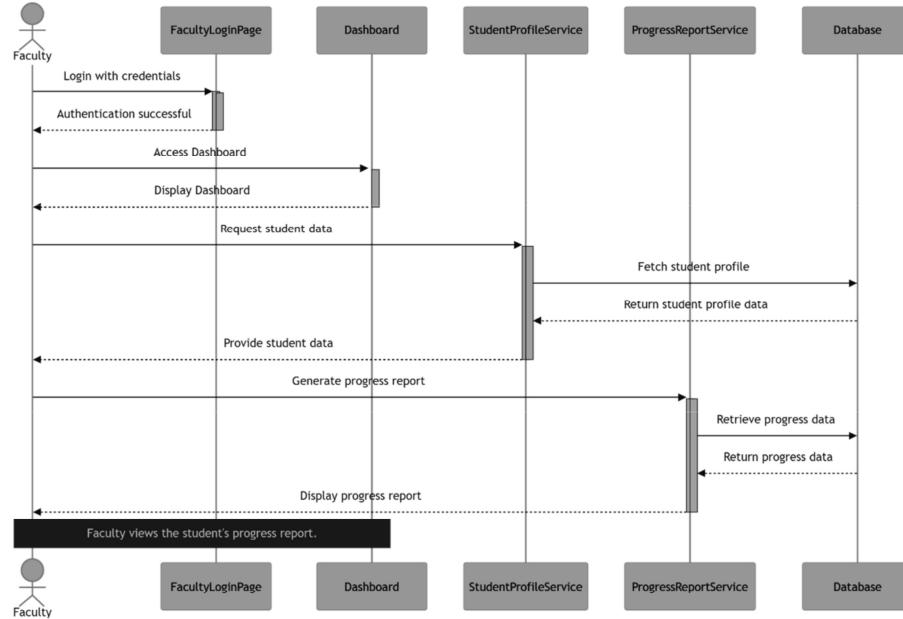
4. Is Profile Link Valid?

The system checks if the provided profile link or credentials are valid and accessible.

- Yes: If valid, the system proceeds to fetch data.

Experiment-7

Draw the sequence diagram of Campus Connect



Explanation of Collaboration Diagram

1. Student Links Coding Profile

The process starts when a student initiates a request to link their coding platform profile (e.g., LeetCode, GFG) through their dashboard.

2. Validate Platform Details

The system sends the provided profile URL or credentials to the CodingPlatform module to validate its authenticity and accessibility.

3. Platform Response

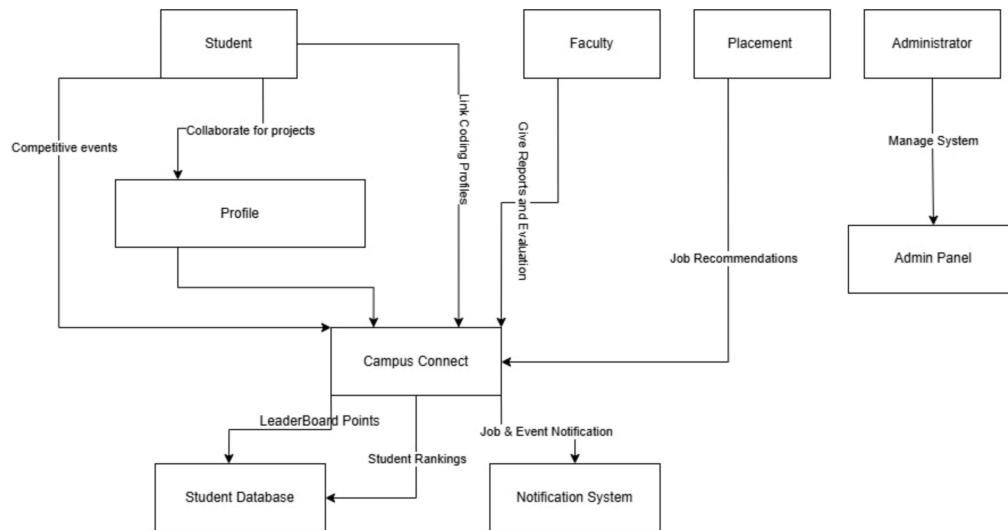
The CodingPlatform module returns a response confirming whether the provided profile is valid and publicly accessible.

4. Verify Student Eligibility

The system ensures that the student is enrolled and eligible to link external platforms (e.g., not blocked or unverified).

Experiment-8

Draw the Collaboration diagram of Campus Connect



Key Actors and Components – Campus Connect

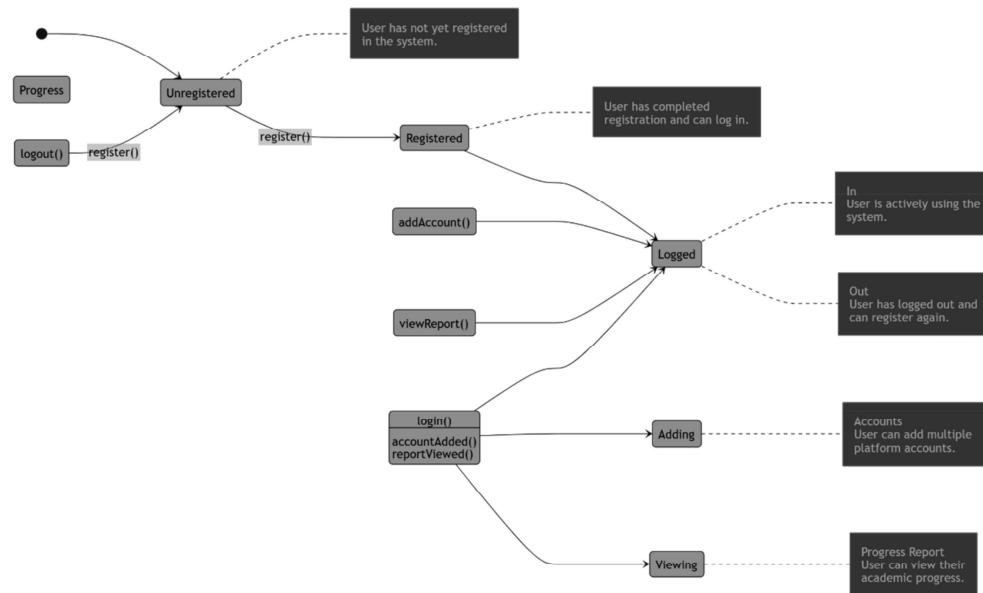
1. **Student** – The primary user who links coding profiles, applies for jobs, and shares achievements.
2. **Admin** – Faculty, HODs, or placement cell members who review student data and post job opportunities.
3. **Profile Linking Module** – Manages verification and storage of coding platform profiles.
4. **Job Application Module** – Handles job application submissions and student eligibility tracking.
5. **Achievement Module** – Records and displays student achievements across the platform.
6. **Database (DB)** – Stores student profiles, job posts, applications, and achievement data.

Process Overview:

7. The student logs in and initiates a request to link a coding platform profile (1).
8. The Profile Linking Module validates the provided profile URL and credentials (2).

Experiment-9

Draw the State Chart diagram of Campus Connect.



Explanation of State Case diagram

1. Start State:

- The system begins when a new student account is created and the student logs in for the first time.
- Action: "Student registers and accesses Campus Connect platform".

2. Profile Linking State:

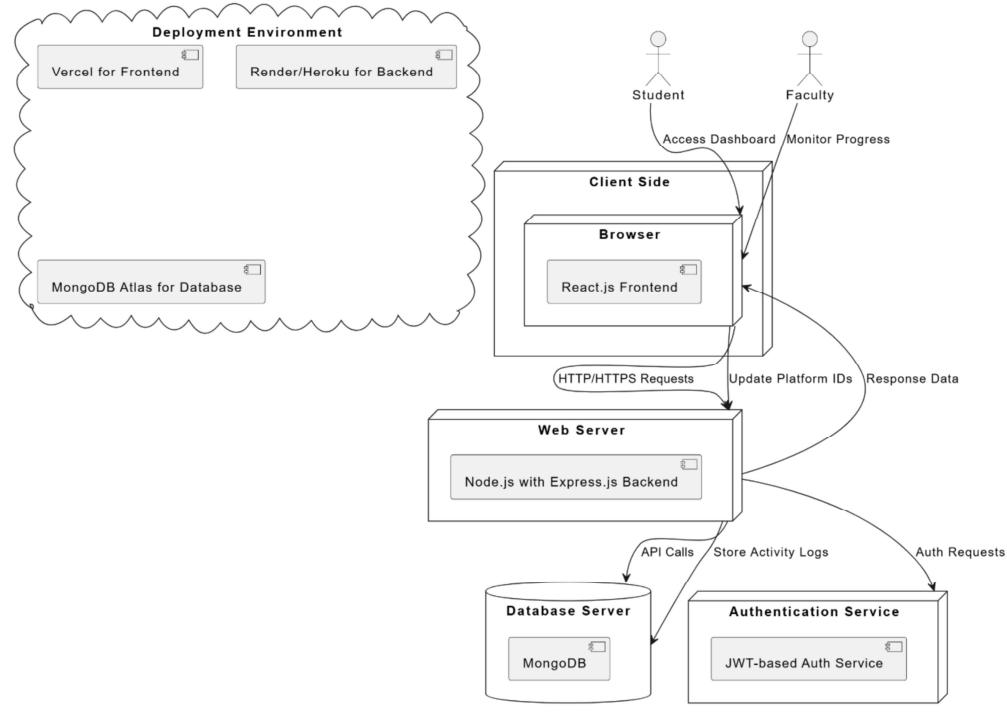
- Once logged in, the student can link their coding platform profiles (e.g., LeetCode, GFG).
- Action: "Student links coding platform profiles".
- The system transitions from the start state to this state upon successful linking and validation of the provided credentials.

3. Job Application State:

- After profile linking, the student may apply to posted job opportunities that match their skills.
- Action: "Student applies to job [eligible based on profile]".

Experiment-11

Draw the Deployment Diagram of Campus Connect



Explanation of Deployment Diagram

1. Client Node (Browser)

- Component: React.js frontend (using Next.js, Tailwind CSS, and shadcn/ui)
- Hosting Platform: Vercel or Netlify
- Users: Students and Faculty
- Functionality:
 - Provides an interactive user interface for students and faculty.
 - Students can log in, manage their profiles, and link competitive programming accounts.
 - Faculty can access dashboards and monitor student progress.
- Communication: Interacts with the backend server through HTTPS-based REST API calls.