Project Report: EduCollab - A Collaborative Learning Management System

1. Introduction

EduCollab is a web-based collaborative learning platform designed to streamline group-based projects in an academic environment. It bridges the gap between students and professors by providing a structured, transparent, and efficient system for course management, group formation, task delegation, peer review, and file sharing. The system ensures accountability by tracking individual contributions and providing professors with a clear overview of group progress and final submissions.

The core philosophy is to move beyond simple file repositories (like the initial Lab-03) into a full-fledged collaboration hub that enforces workflow, minimizes free-riding, and integrates automated group management based on student data.

2. Functional Requirements

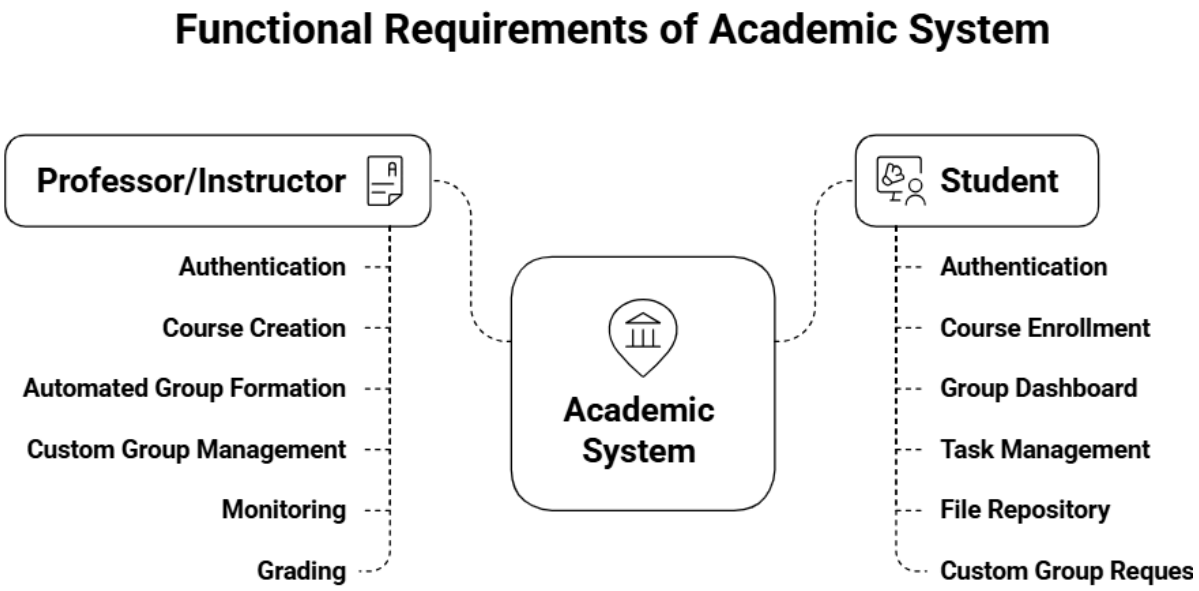
The system's functionality is divided into two primary user roles:

A. Professor/Instructor:

- Authentication: Secure login with a university email.
- Course Creation: Create new courses by providing a name and uploading a CSV file containing student IDs and CGPAs.
- Automated Group Formation: System automatically creates groups based on the uploaded CSV, assigning the student with the highest CGPA as the leader.
- Custom Group Management: Ability to view custom group requests from students and manually create groups before the automated process runs.
- Monitoring: View all groups, their members, their shared repositories, and a color-coded task board showing the status of all assigned work.
- Grading: Review and grade final group submissions.

B. Student:

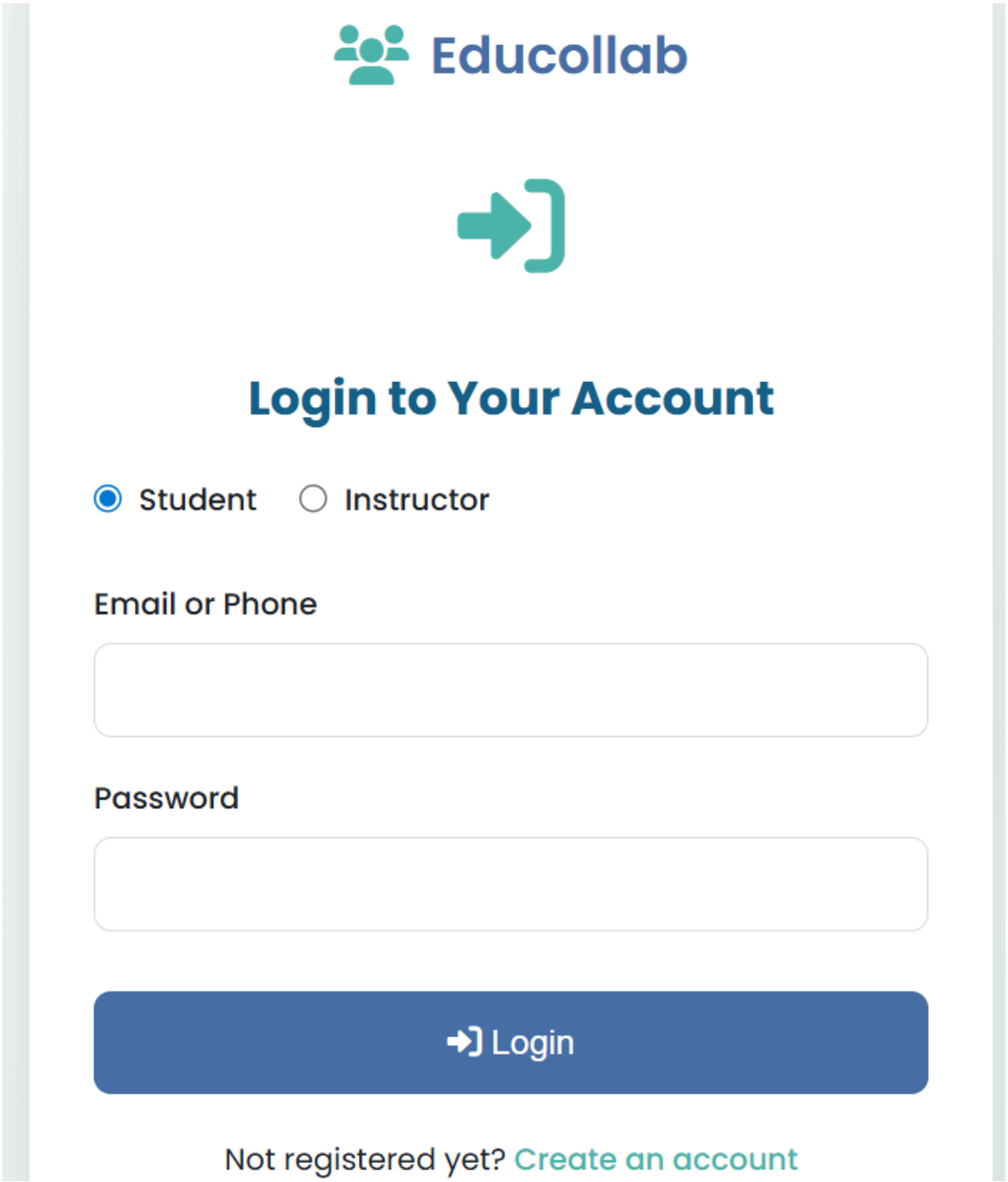
- Authentication: Secure login with university email, ID, and password.
- Course Enrollment: Join a course by entering a unique code provided by the professor. The system validates if the student's ID is in the professor's CSV.
- Group Dashboard: View their assigned group, identify the leader, and see all group members.
- Task Management: View tasks assigned by the leader, accept/reject them, upload completed work, and participate in peer review by approving teammates' submissions.
- File Repository: Access a personal repository (with public/private files) and a group-shared repository for collaborative work.
- Custom Group Request: Request to form a custom group with specific peers before the professor finalizes groups.



Login.php

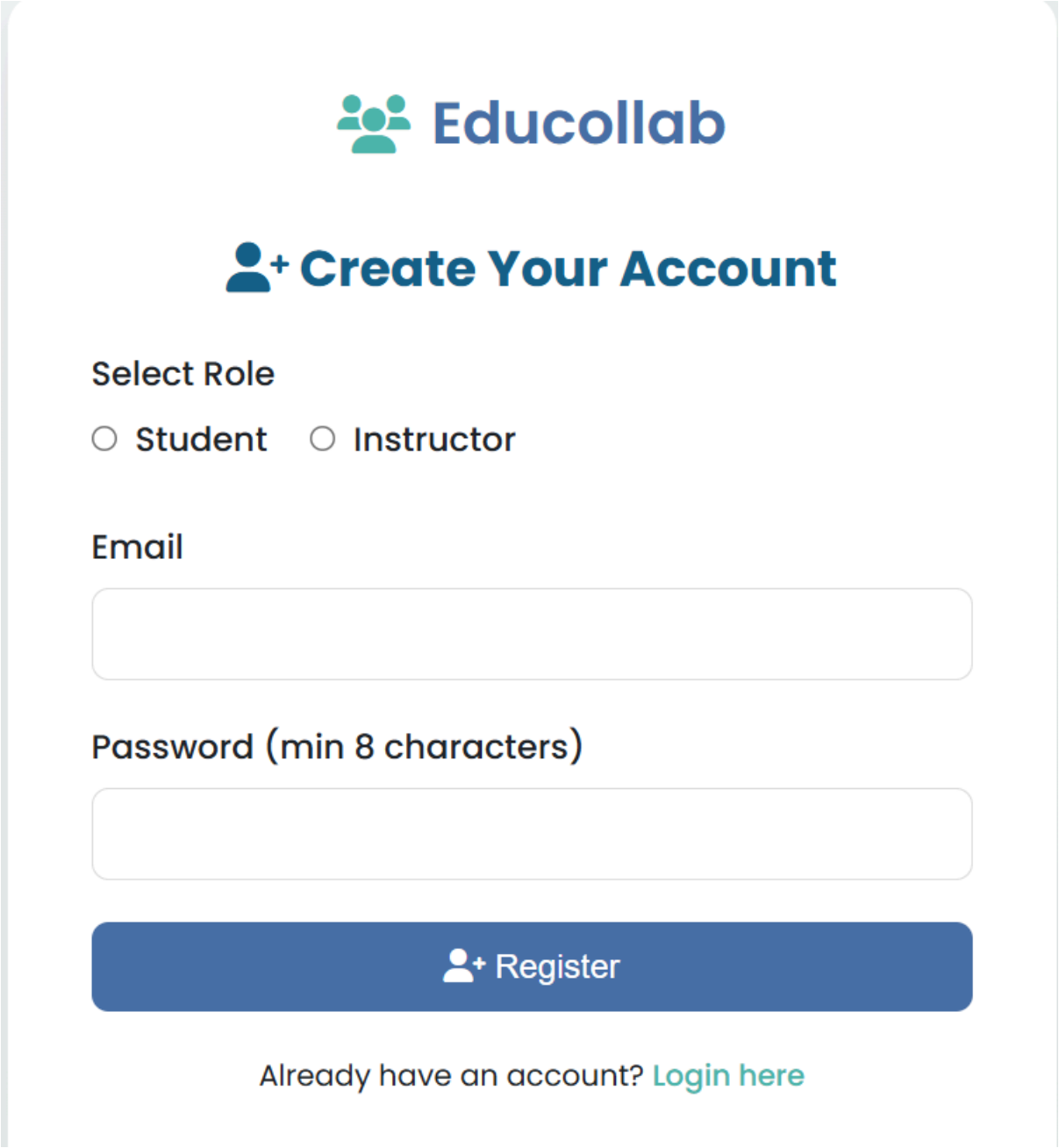
1. User selects role (student/professor)
2. User provides identifier (email or phone) and password
3. System verifies credentials against database

4. Successful login creates session variables and redirects to appropriate dashboard
5. Failed login displays descriptive error messages



Register.php

The register.php file implements a functional registration system with good security fundamentals like password hashing and SQL injection prevention. The main areas for improvement are adding CSRF protection, enhancing validation, improving error handling, and making the student email validation more flexible.



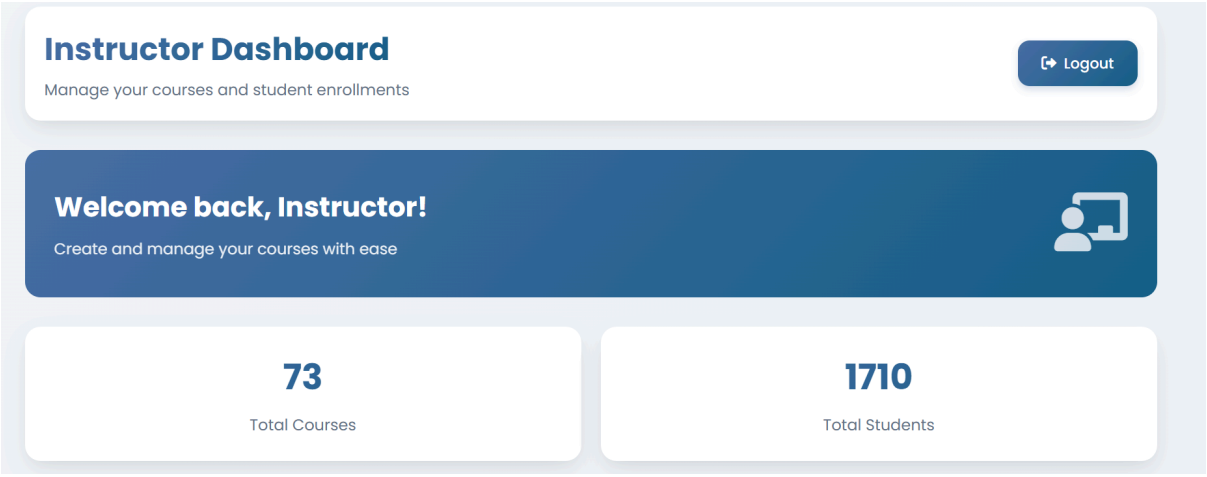
Professor_dashboard.php and course_dashboard.php

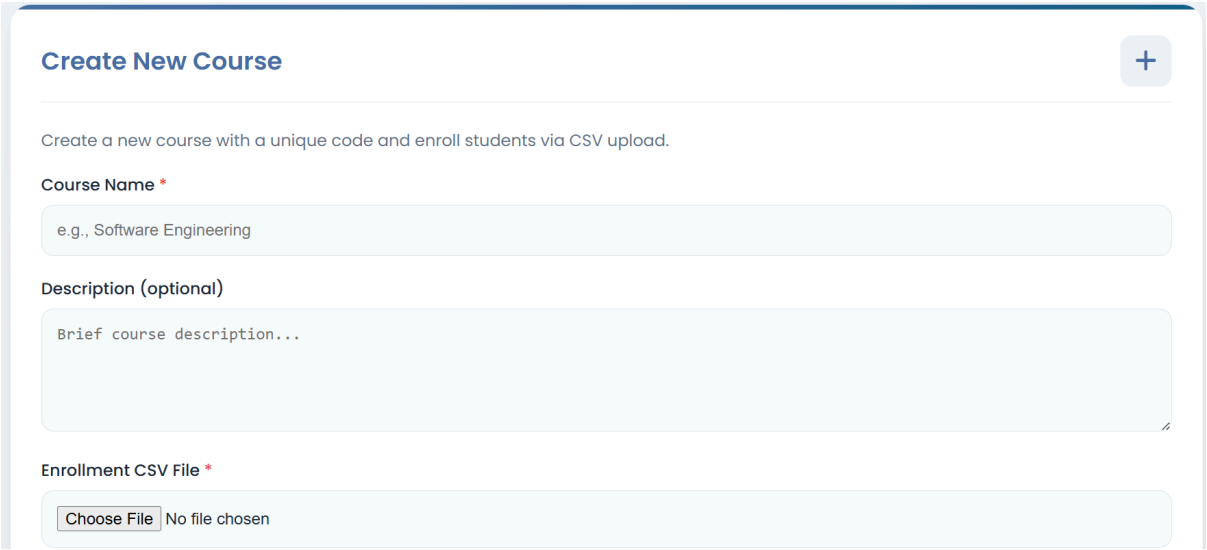
1. Course Management: Displays professor's courses with student counts

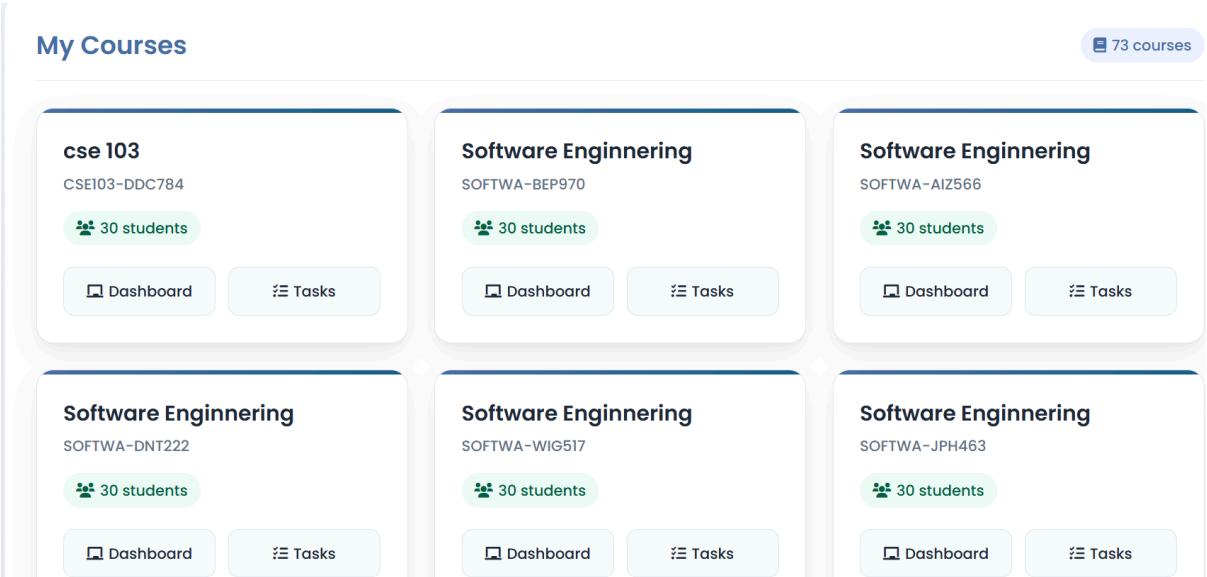
2. Statistics Overview: Shows total courses and student counts
3. Course Creation: Form to create new courses with CSV enrollment
4. Responsive Design: Works well on both desktop and mobile devices
5. User Feedback: Clear error and success messages

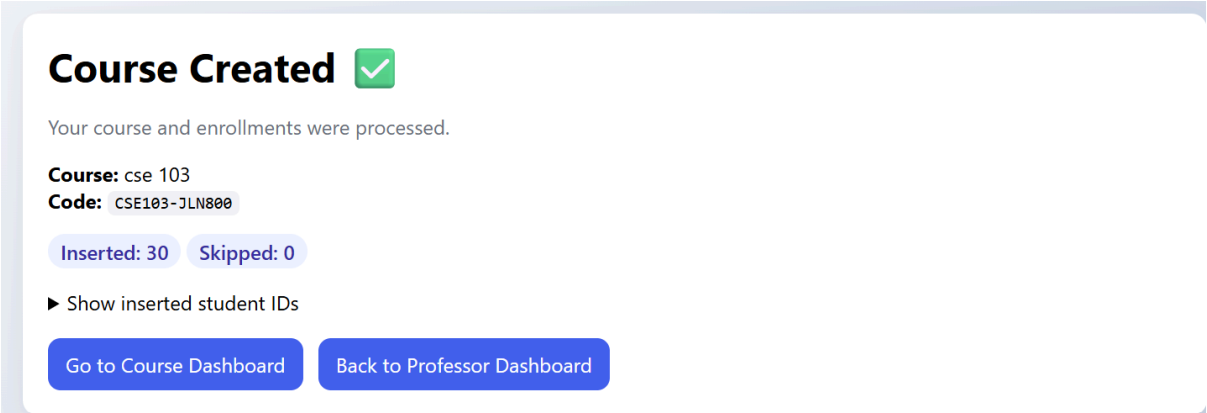
course_dashboard

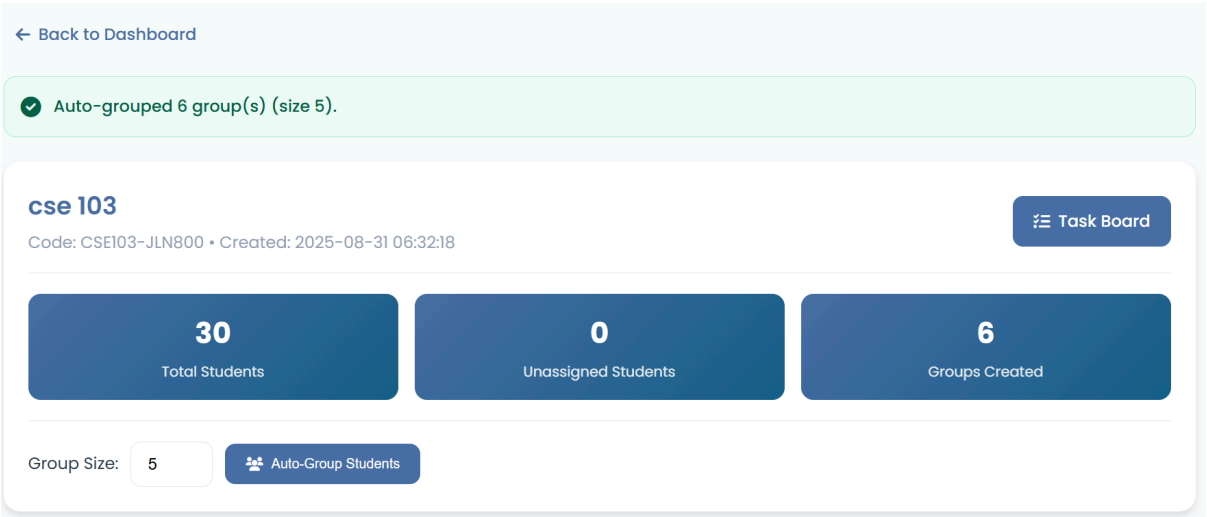
- Shows enrolled students in a table with their status (Assigned/Unassigned).
- Provides tools for auto-grouping (with configurable group size) and manual custom group creation.
- Displays all groups in a grid, allowing the professor to rename, delete, change leaders, and remove members for each group.

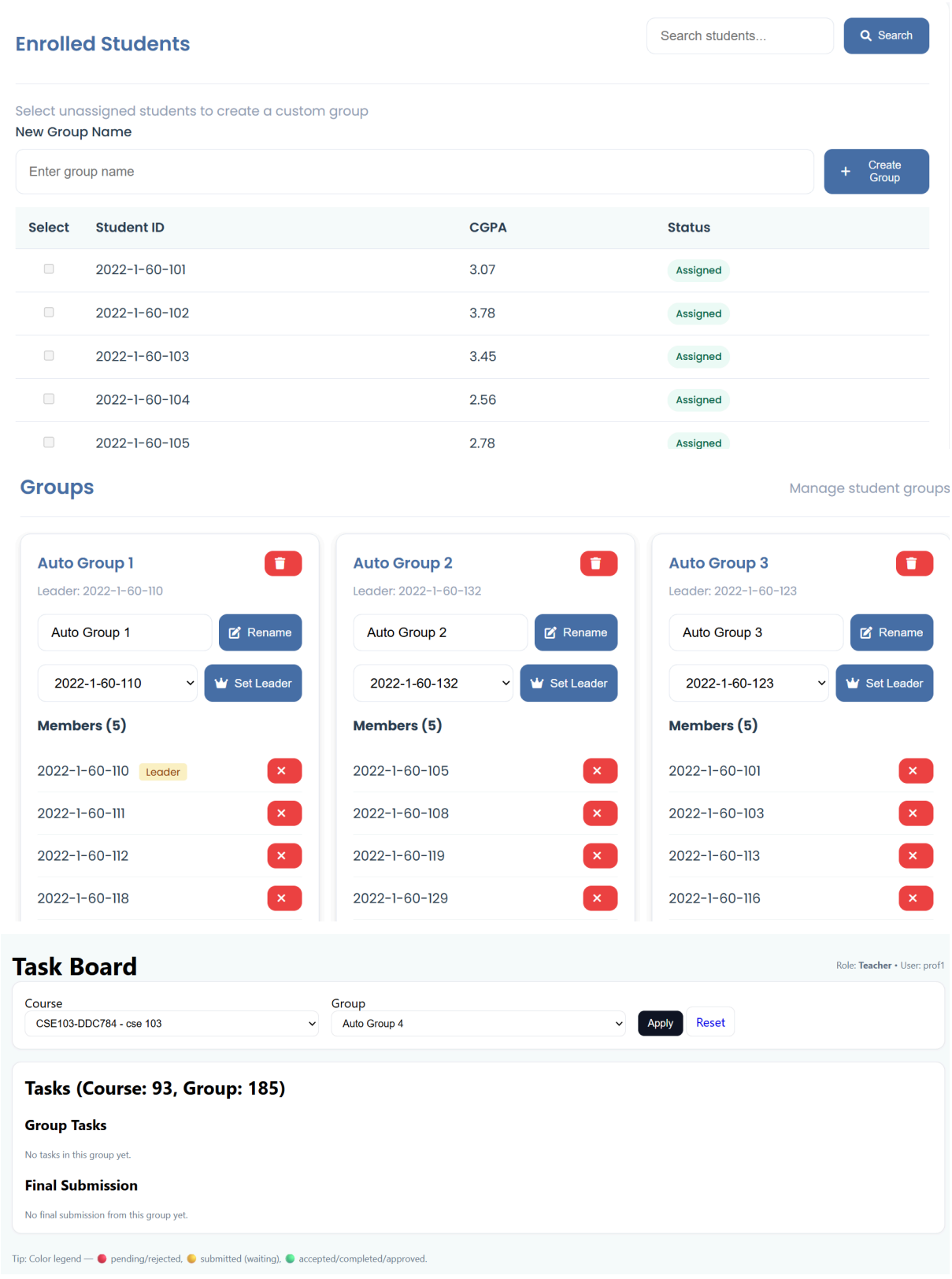





My Courses
73 courses
cse 103
CSE103-DDC784
30 students
Dashboard
Tasks
Software Enginnering
SOFTWA-BEP970
30 students
Dashboard
Tasks
Software Enginnering
SOFTWA-AIZ566
30 students
Dashboard
Tasks
Software Enginnering
SOFTWA-DNT222
30 students
Dashboard
Tasks
Software Enginnering
SOFTWA-WIG517
30 students
Dashboard
Tasks
Software Enginnering
SOFTWA-JPH463
30 students
Dashboard
Tasks


Course Created
Your course and enrollments were processed.
Course: cse 103
Code: CSE103-JLN800
Inserted: 30
Skipped: 0
Show inserted student IDs
Go to Course Dashboard
Back to Professor Dashboard

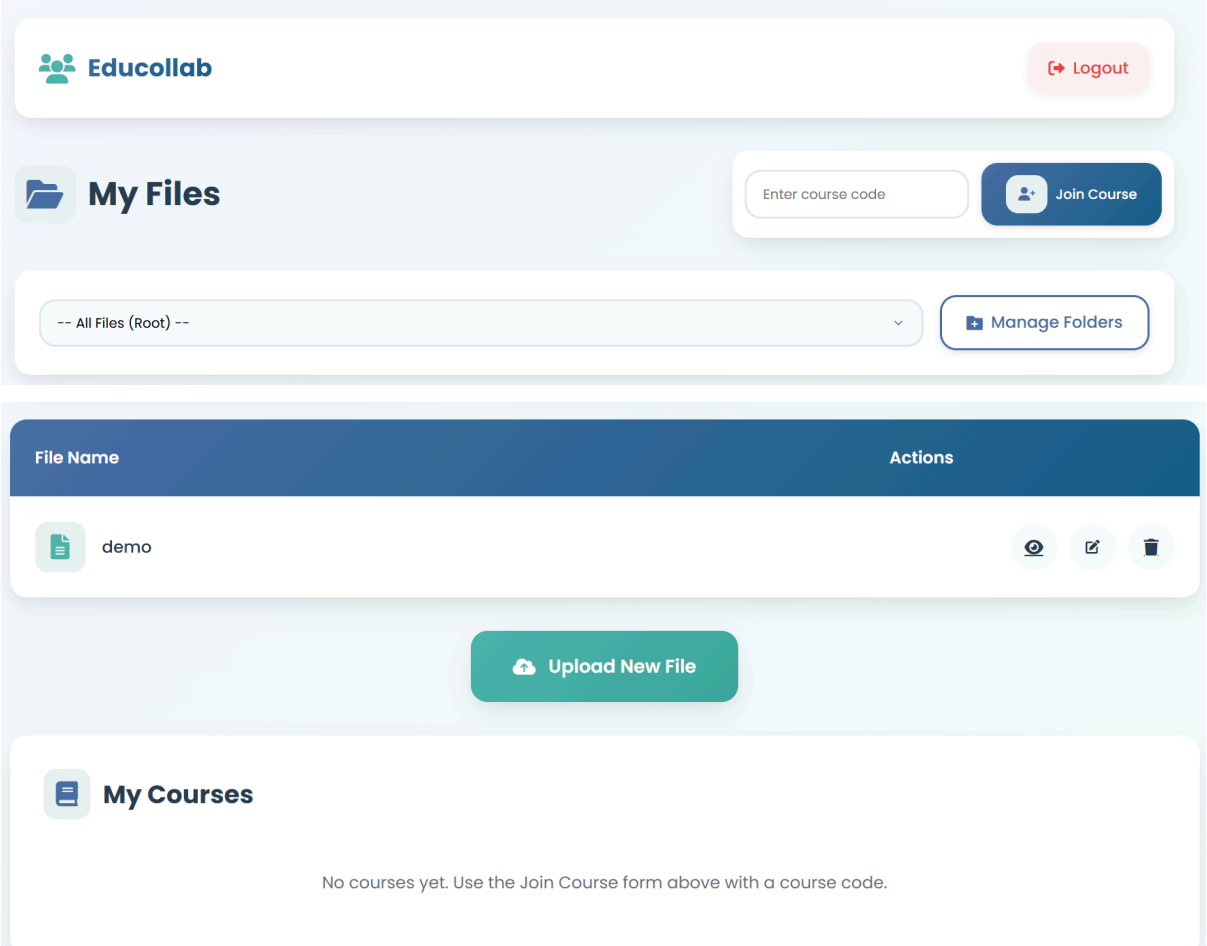

Back to Dashboard
Auto-grouped 6 group(s) (size 5).
cse 103
Code: CSE103-JLN800 • Created: 2025-08-31 06:32:18
Task Board
30
Total Students
0
Unassigned Students
6
Groups Created
Group Size:
5
Auto-Group Students



Student _ Dashboard.php

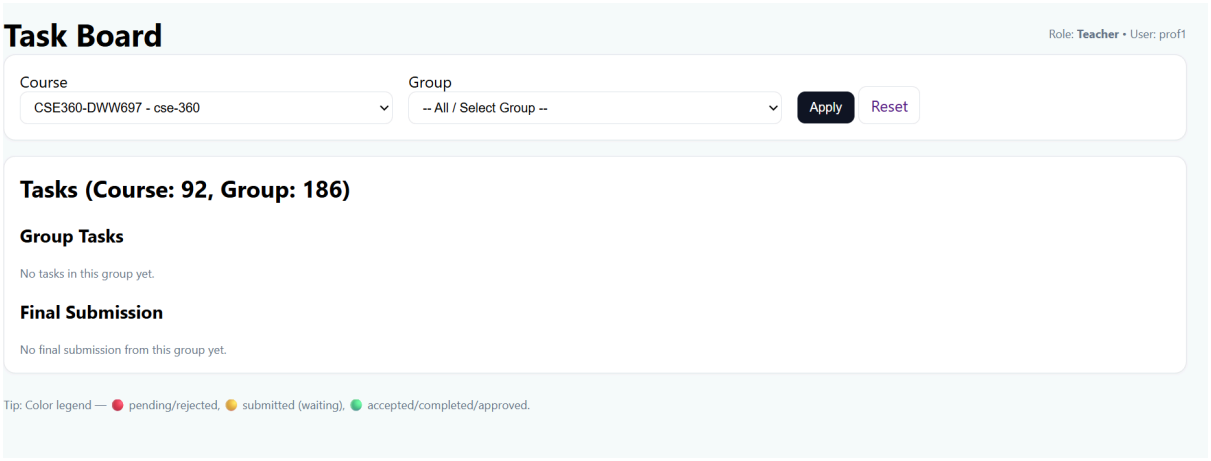
1. Course Verification: Validates course existence and student enrollment

2. Group Information: Displays group details including members and leader
3. Student Information: Shows student ID and CGPA if available
4. Navigation: Provides clear navigation back to the home page
5. Responsive Design: Works well on both desktop and mobile devices



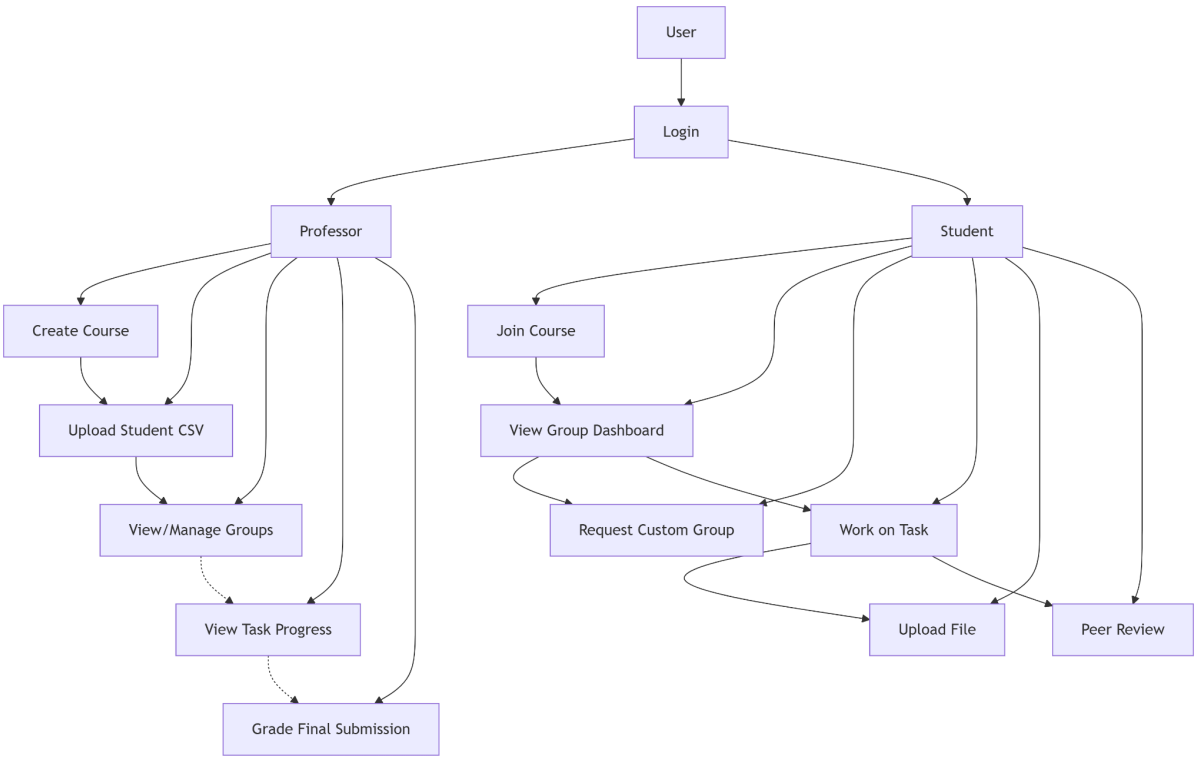
TaskBoard.php

- Student View: Lists tasks assigned to them with buttons to Accept/Reject or Upload work. Shows status badges (Red, Yellow, Green).
- Leader View: Adds an "Assign Task" form and a "Final Submission" upload area. Shows all group tasks with options to approve/reject submitted work.
- Professor View: Allows filtering by course and group. Shows the task status for all members and the group's final submission for grading.

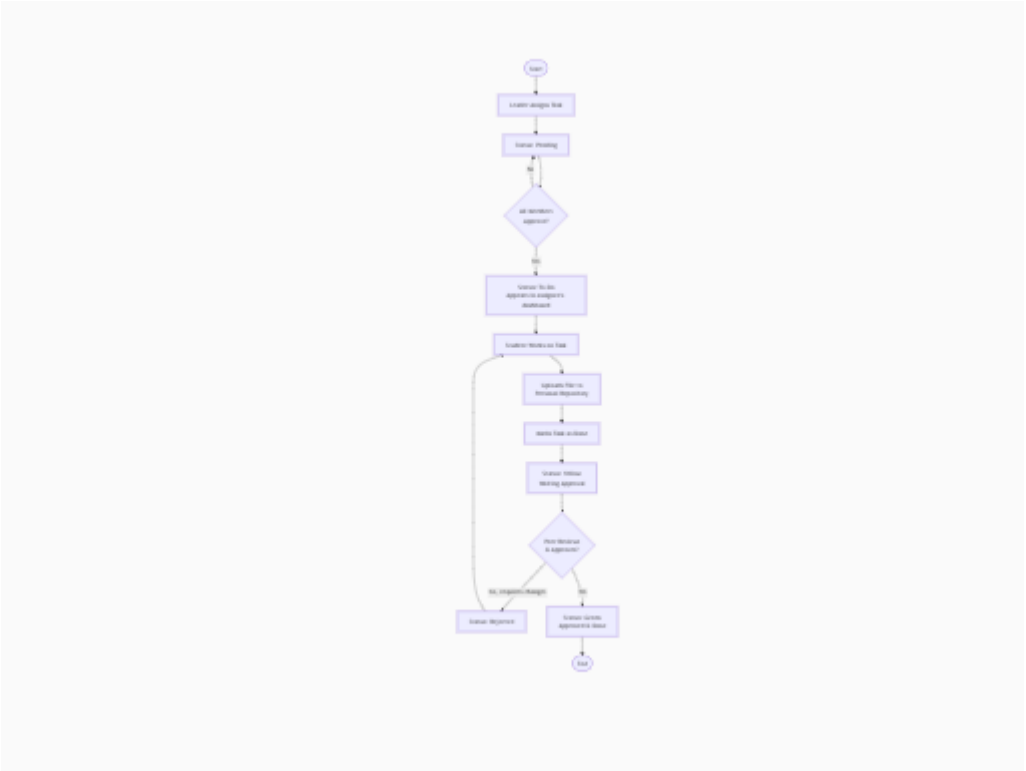


3. Basic Design Diagrams

A. Use Case Diagram



B. Activity Diagram: Student Task Workflow



4. Technology Details

- Frontend: HTML5, CSS3, JavaScript (Vanilla JS for DOM manipulation and form validation).
- Backend: PHP (Procedural with prepared statements for database interaction).
- Database: MySQL.
- Security: Password hashing using `password_hash()`, SQL injection prevention via prepared statements, session-based authentication.
- Styling: Custom CSS with a cohesive color scheme (`--educollab-primary`, `--educollab-accent`) and modern UI components (cards, grids, buttons). Font Awesome for icons and Google Fonts (Poppins) for typography.
- Key Features: File upload handling, CSV parsing, dynamic group generation algorithms, role-based access control (RBAC).

5. Team Responsibilities

The work was divided into five core modules, as per the project plan:

- Team Member 1 (Ishmat): User Verification & Authentication
    - Extended the login/register system for Student and Professor roles.
    - Implemented session management and role-based redirects (to `files.php` or `professor_dashboard.php`).
    - Ensured students could only see courses they were enrolled in via CSV validation.
- Team Member 2 (Saeikh): Course & Group Management (Professor Side)
    - Built the course creation form with CSV upload functionality.
    - Implemented the core logic for parsing CSV, auto-generating groups based on CGPA, and assigning leaders.
    - Developed the professor's course dashboard to view and manage groups.
- Team Member 3 (Ankon): Student Course Flow & Group Dashboard
    - Implemented the "Join Course" functionality using the course code.
    - Built the student's group dashboard view to see group info, members, and the leader.
    - Integrated links to the repository and task board sections built by others.
- Team Member 4 (Shanto): Task Assignment & Peer Review System
    - Designed and implemented the full task workflow (Pending → Accepted → Submitted → Completed).
    - Created the color-coded status system (Red/Yellow/Green) visible to both students and professors.
    - Built the peer review and leader approval mechanics.
- Team Member 5 (Jannat Milky): Repository Expansion & File Management
    - Expanded the basic file repository (Lab-03) into a multi-tier system (Private, Public, Final Work).
    - Implemented file upload tracking (who uploaded what and when).
    - Built the final submission system for groups and integrated it with the professor's grading view.

7. Conclusion

The EduCollab project successfully delivers a robust and functional collaborative learning platform. It meets all its core objectives: secure role-based access, efficient course and group management, a structured task and review workflow, and a transparent file repository system.

The use of fundamental web technologies (PHP, MySQL, JS, CSS) demonstrates a strong understanding of full-stack development principles. The modular division of work allowed for parallel development and integration of complex features. The system is ready for demonstration, showcasing a complete flow from professor course creation to student collaboration and final grading. Future enhancements could include real-time notifications, in-app messaging, and more advanced file preview capabilities.