CS 682 Project

Project 1 - A Portal for Managing

Students Capstone Projects

USER MANUAL

Submitted by:

Lohith Reddy Mudipalli

Sri Charan Tadiparthi

Bharath Karumanchi

Contents

I. About Capstone Hub

Capstone Hub addresses a common challenge in academic settings: coordinating capstone projects between students seeking meaningful work, organizations offering real-world problems, and instructors overseeing the entire process. Rather than relying on spreadsheets and email chains, this platform centralizes everything into role-based dashboards.

The system supports three distinct user types. Students use the platform to discover projects that match their interests and technical abilities, then rank their top choices before a deadline. Clients—whether companies, nonprofits, or research labs—pitch project ideas and later connect with the teams assigned to them. Instructors orchestrate the workflow: they review proposals, assemble teams using preference data, and coordinate presentation schedules.

II. Accessing Capstone Hub

Production URL:

https://creative-expression-production.up.railway.app/

Open this link in Chrome, Firefox, Safari, or Edge. No installation required.

For development purposes, the application can also run locally—see the next section.

III. Running Locally (Detailed Setup Guide)

This section walks you through setting up Capstone Hub on your own computer. Follow each step carefully, especially if you are new to web development.

Step 1: Install Required Software

Before starting, you need four tools installed on your machine:

1.1 Node.js (JavaScript runtime)

- Visit https://nodejs.org/
- Download the LTS (Long Term Support) version (v18 or higher)
- Run the installer and follow the prompts
- To verify installation, open Terminal (Mac) or Command Prompt (Windows) and type:

```
node --version
```

- You should see something like `v18.17.0`

1.2 MySQL (Database)

- Visit https://dev.mysql.com/downloads/mysql/
- Download MySQL Community Server for your operating system
- During installation, you will be asked to set a root password—remember this password!

- On Mac, you can also install via Homebrew:

```
brew install mysql
brew services start mysql
```

- To verify, type in terminal:

```
mysql --version
```

1.3 Git (Version control)

- Visit https://git-scm.com/downloads
- Download and install for your operating system
- On Mac, Git may already be installed. Check by typing:

```
git --version
```

1.4 Code Editor (Recommended)

- Download Visual Studio Code from https://code.visualstudio.com/
- This will help you edit configuration files easily

Step 2: Download the Project Code

Open Terminal (Mac) or Command Prompt (Windows) and navigate to where you want to store the project. For example, your Desktop:

```
cd ~/Desktop
```

Now clone (download) the project from GitHub:

```
git clone https://github.com/sricharan0912/
A-Portal-for-Managing-Students-Capstone-Projects-.git
```

This creates a folder called `A-Portal-for-Managing-Students-Capstone-Projects-`. Enter it:

```
cd A-Portal-for-Managing-Students-Capstone-Projects-
```

You should now see two main folders: `frontend` and `backend`.

Step 3: Set Up the Database

3.1 Start MySQL (if not already running)

On Mac:

```
brew services start mysql
```

On Windows: MySQL should start automatically, or use MySQL Workbench.

3.2 Log into MySQL

Open terminal and type:

```
mysql -u root -p
```

Enter your MySQL root password when prompted. You should see the MySQL prompt: `mysql>`

3.3 Create the Database

Type these commands one by one (press Enter after each):

```
CREATE DATABASE capstone_hub;
USE capstone_hub;
source backend/database/schema.sql;
```

If the source command does not work, exit MySQL by typing `exit`, then run:

```
mysql -u root -p capstone_hub < backend/database/schema.sql
```

3.4 Verify Tables Were Created

Log back into MySQL and check:

```
mysql -u root -p
USE capstone_hub;
SHOW TABLES;
```

You should see tables like `users`, `projects`, `groups`, `preferences`, etc.

Type `exit` to leave MySQL.

Step 4: Set Up Firebase Authentication

Firebase handles user login securely. You need both Admin SDK (for backend) and Client SDK (for frontend).

4.1 Create a Firebase Project

1. Go to https://console.firebase.google.com/
2. Click "Add project"
3. Enter a project name (e.g., "capstone-hub")
4. Disable Google Analytics (optional, simplifies setup)
5. Click "Create project" and wait for it to finish

4.2 Enable Email/Password Authentication

1. In your Firebase project, click "Authentication" in the left sidebar
2. Click "Get started"
3. Click "Email/Password"
4. Toggle "Enable" to ON
5. Click "Save"

4.3 Get Admin SDK Credentials (for Backend)

1. Click the gear icon (Settings) next to "Project Overview"
2. Select "Project settings"
3. Go to the "Service accounts" tab

4. Click “Generate new private key”
5. A JSON file will download—open it in a text editor
6. You will need: `project_id`, `client_email`, and `private_key`

4.4 Get Client SDK Credentials (for Frontend)

1. In Project settings, go to the “General” tab
2. Scroll down to “Your apps”
3. Click the web icon (</>) to add a web app
4. Enter a nickname (e.g., “capstone-web”)
5. Click “Register app”
6. You will see a config object with `apiKey`, `authDomain`, etc.—copy these values

Step 5: Configure Backend Environment

5.1 Open the project in VS Code:

```
code .
```

5.2 Navigate to the `backend` folder

5.3 Create a new file called `.env` (note the dot at the beginning)

5.4 Paste the following and fill in your values:

```
# Server Configuration
PORT=5050

# Database Configuration
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=your_mysql_password_here
DB_NAME=capstone_hub

# JWT Secret (make up a random string)
JWT_SECRET=any-random-secret-string-here-abc123

# Firebase Admin SDK (from the JSON file you downloaded)
FIREBASE_PROJECT_ID=your-project-id
FIREBASE_CLIENT_EMAIL=firebase-adminsdk-xxxxx@
your-project.iam.gserviceaccount.com
FIREBASE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----
\nYourKeyHere\n-----END PRIVATE KEY-----\n"
```

Important: The `FIREBASE_PRIVATE_KEY` must be wrapped in double quotes and keep the `\n` characters.

Step 6: Configure Frontend Environment

6.1 Navigate to the `frontend` folder

6.2 Create a new file called `.env`

6.3 Paste the following and fill in your Firebase web config:

```
# API URL (points to your local backend)
VITE_API_URL=http://localhost:5050

# Firebase Client SDK
# (from Firebase Console > Project Settings > Your Apps)
VITE_FIREBASE_API_KEY=AIzaSyB1234567890abcdefg
VITE_FIREBASE_AUTH_DOMAIN=your-project.firebaseapp.com
VITE_FIREBASE_PROJECT_ID=your-project-id
VITE_FIREBASE_STORAGE_BUCKET=your-project.appspot.com
VITE_FIREBASE_MESSAGING_SENDER_ID=123456789012
VITE_FIREBASE_APP_ID=1:123456789012:web:abcdef123456
```

Step 7: Install Dependencies

Open a terminal in VS Code (Terminal > New Terminal) or use your system terminal.

7.1 Install backend packages:

```
cd backend
npm install
```

Wait for installation to complete. You will see a `node_modules` folder appear.

7.2 Install frontend packages:

```
cd ../frontend
npm install
```

This may take a few minutes.

Step 8: Start the Application

You need two terminal windows running simultaneously.

Terminal 1 — Start Backend:

```
cd backend
npm run dev
```

You should see: "Server running on port 5050"

Terminal 2 — Start Frontend:

```
cd frontend
npm run dev
```

You should see a local URL like: `http://localhost:5173`

Step 9: Open the Application

Open your web browser (Chrome recommended) and go to:

http://localhost:5173

You should see the Capstone Hub landing page. You can now create an account and start using the application!

Troubleshooting Common Setup Issues

- **"command not found: node"** — Node.js is not installed or not in your PATH. Reinstall Node.js.
- **"Access denied for user root"** — Your MySQL password is incorrect. Try resetting it.
- **"ECONNREFUSED 127.0.0.1:3306"** — MySQL is not running. Start it with `brew services start mysql` (Mac) or via Services (Windows).
- **"Firebase: Error (auth/invalid-api-key)"** — Check that your Firebase API key in frontend `.env` is correct.
- **Backend crashes on startup** — Check that all `.env` values are filled in correctly, especially the Firebase private key format.
- **Port already in use** — Another application is using port 5050 or 5173. Close it or change the port in `.env`.

IV. Platform Walkthrough

Registration and Login

First-time users click “Get Started” on the landing page. The signup form collects your name, email, a password (minimum six characters), and your role. After submitting, you land directly in your dashboard.

Returning users select “Login,” enter credentials, pick their role from the dropdown, and proceed.

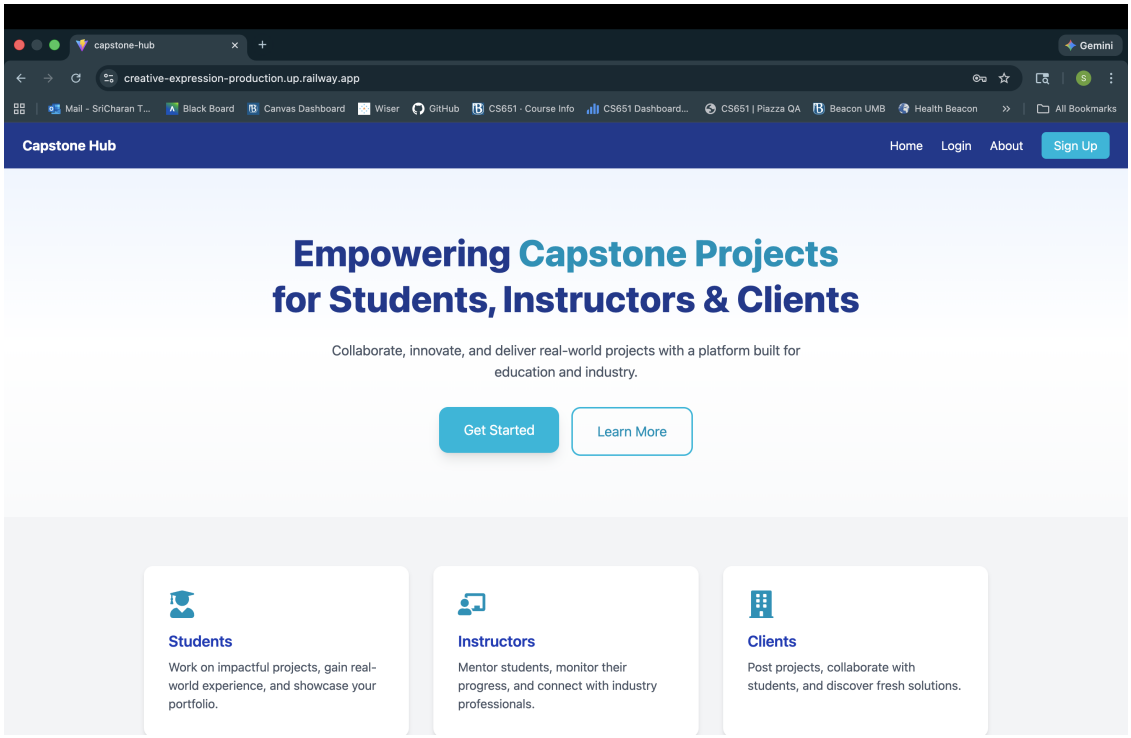

Figure 1: Landing page with Get Started button

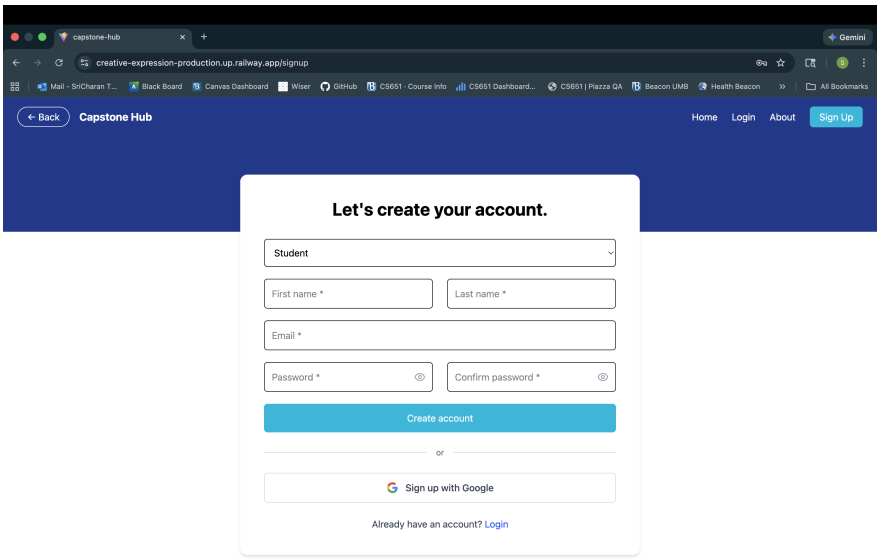

Figure 2: Signup form

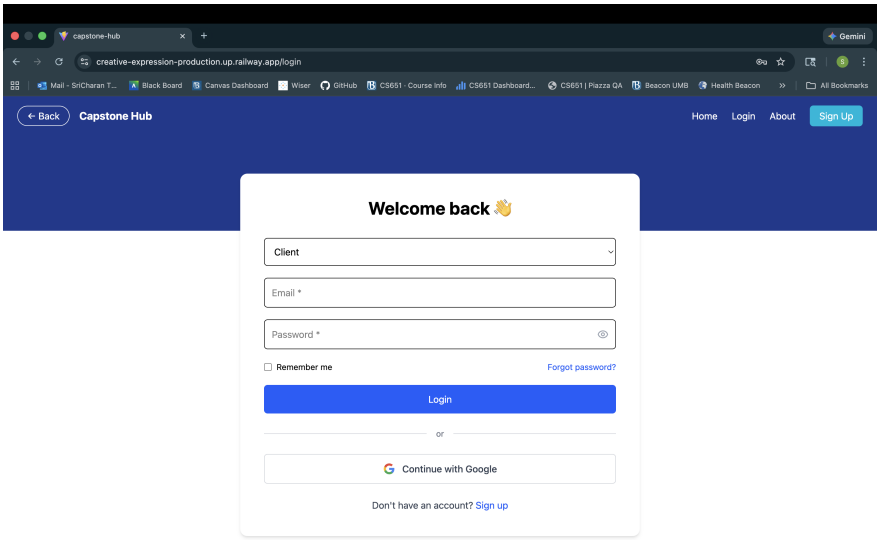

Figure 3: Login form

For Clients

Your Dashboard

Upon login, you see summary cards: how many proposals you have submitted, how many are approved, and whether any teams have been linked to your projects yet.

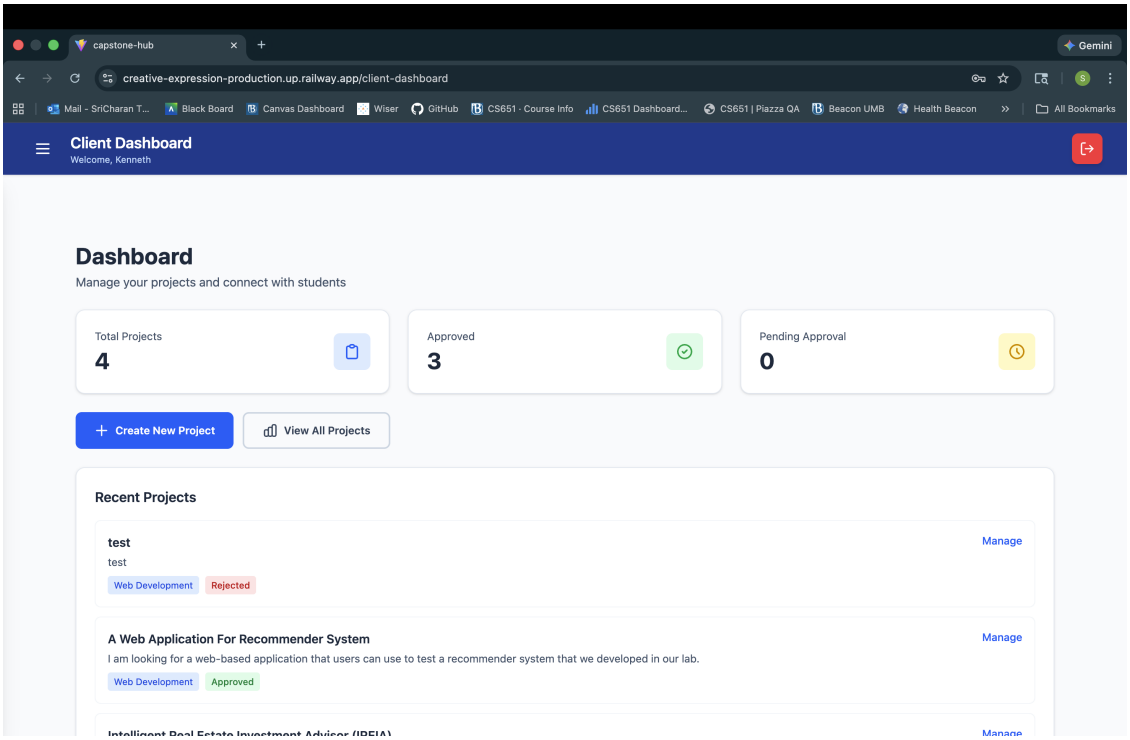

Figure 4: Client dashboard overview

Proposing a Project

Navigate to “My Projects” via the sidebar. Hit the “Create Project” button. You will fill out:

- A concise title
- A thorough description of objectives and deliverables
- Skills students should have (e.g., Python, React, SQL)
- Ideal team size
- A category label

Once submitted, the proposal enters a queue for instructor review.

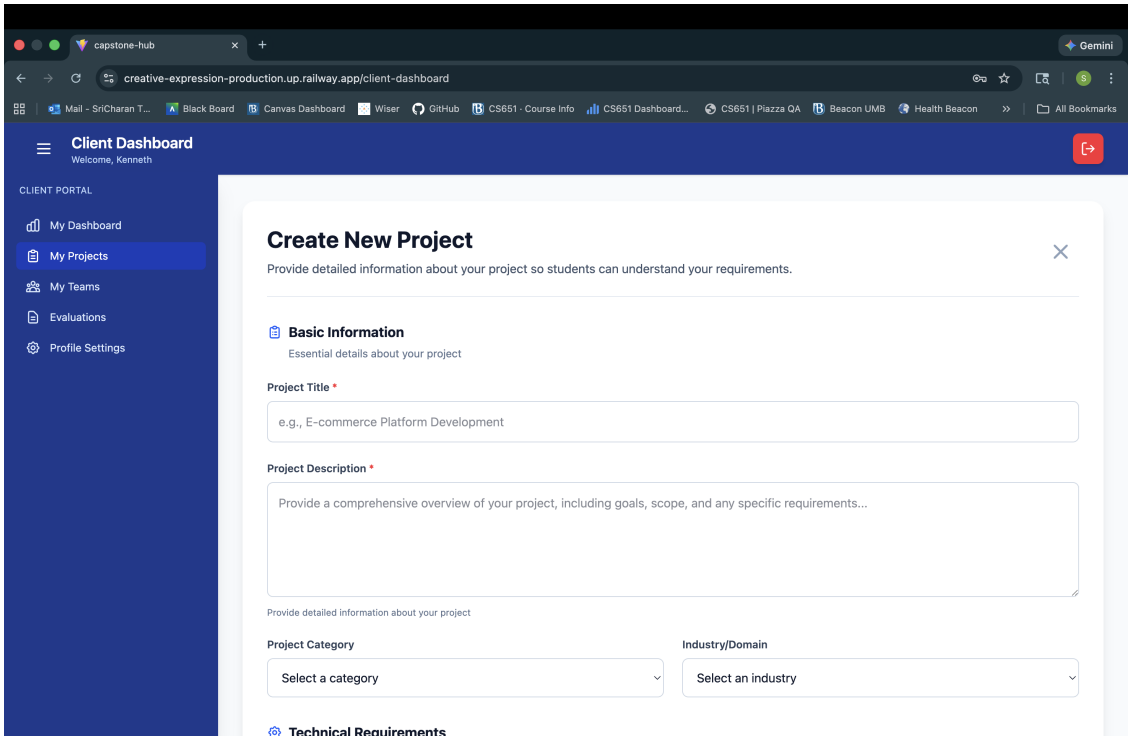

Figure 5: Project creation form - Basic Information

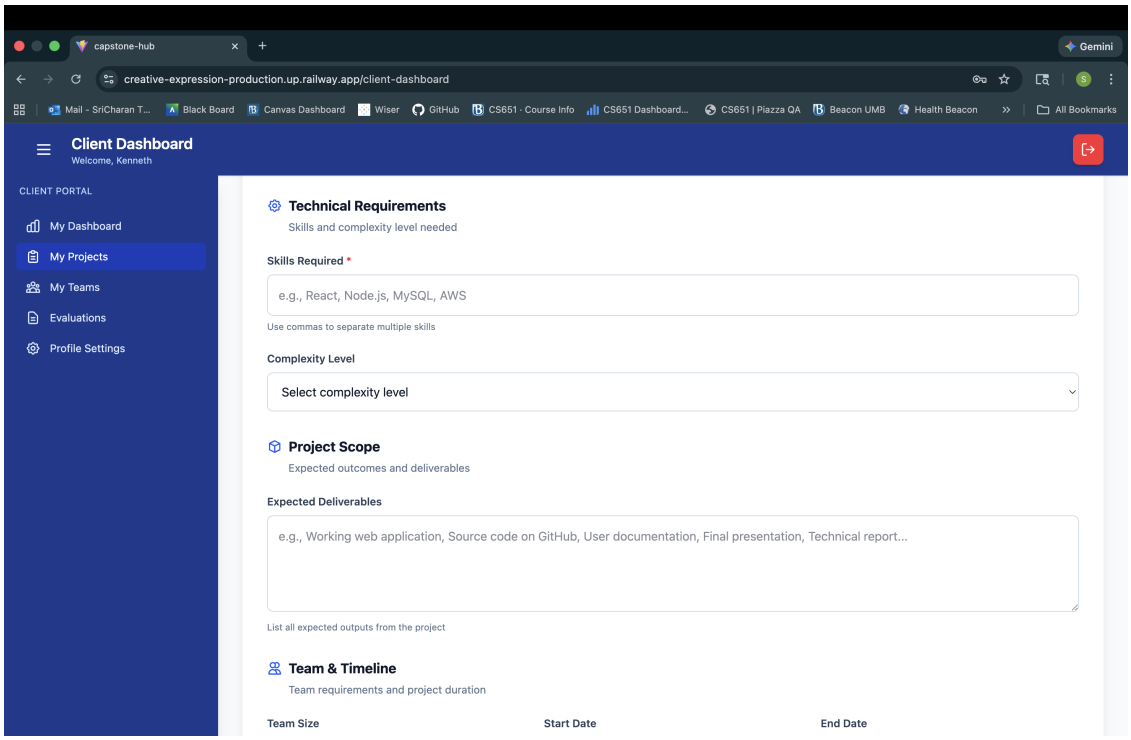

Figure 6: Project creation form - Technical Requirements

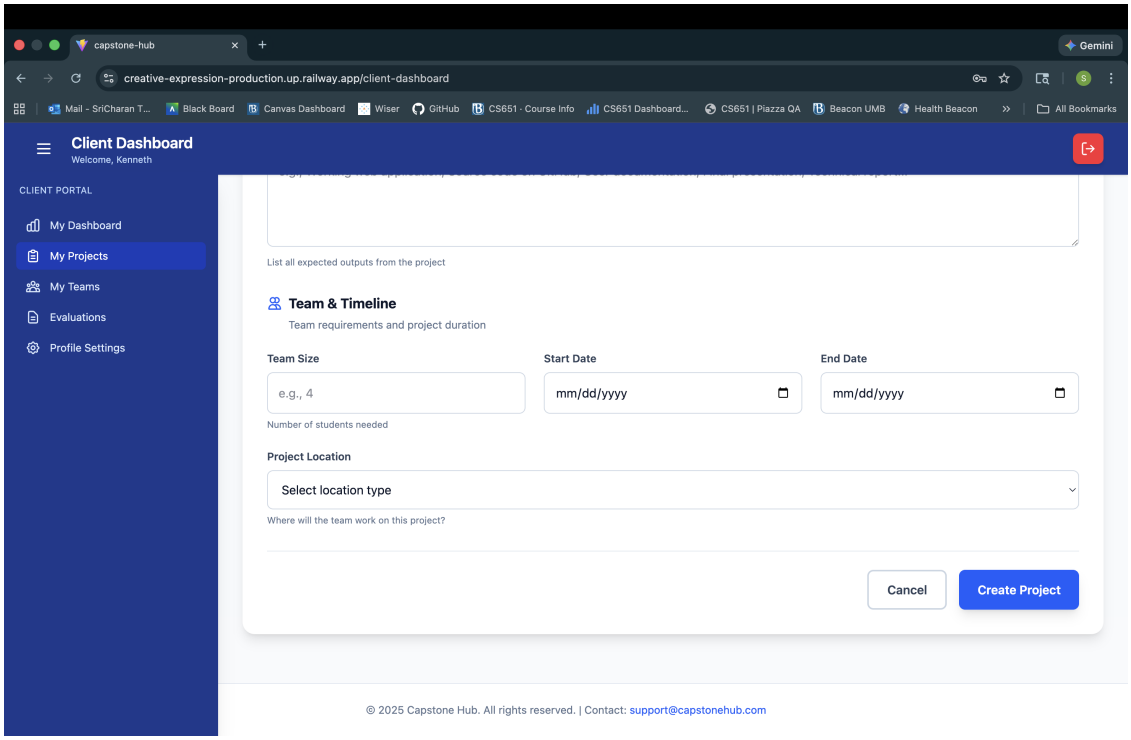

Figure 7: Project creation form - Team & Timeline

Tracking Proposal Status

Your projects appear under three tabs: Approved (green), Pending (yellow), and Rejected (red). If rejected, the instructor's feedback explains what to revise before resubmitting.

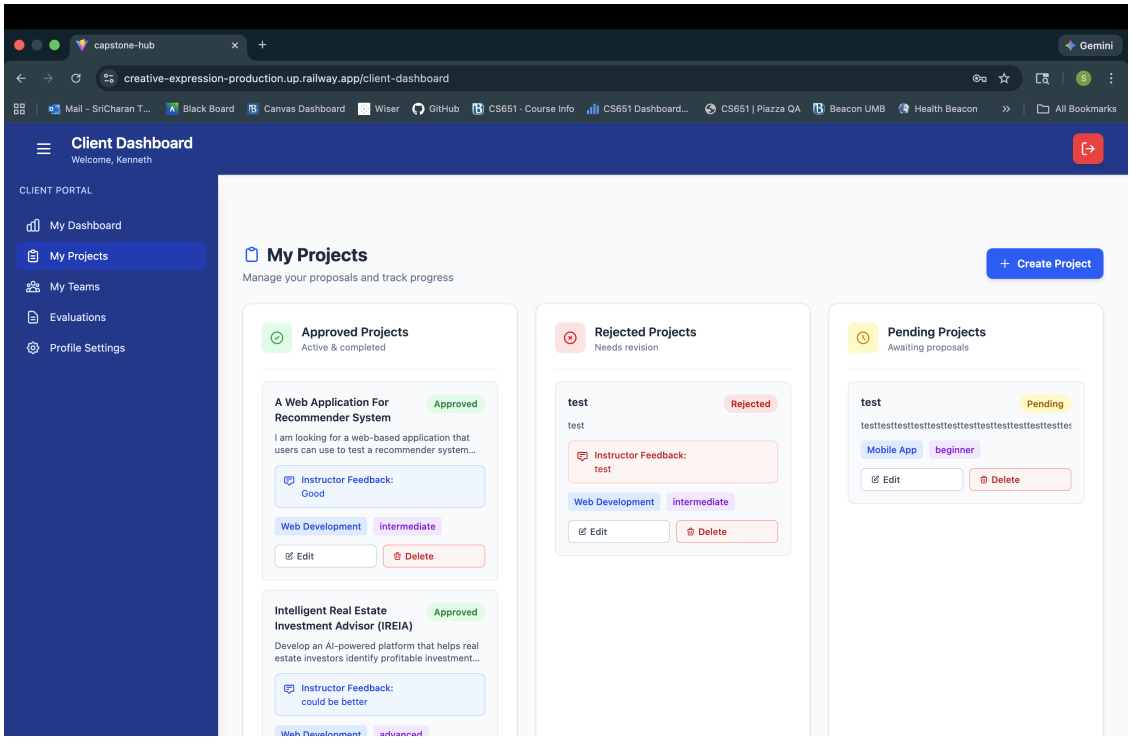

Figure 8: Project list showing status tabs

Meeting Your Team

After groups form, “My Teams” displays each assigned group: project name, member names, and email addresses so you can coordinate meetings.

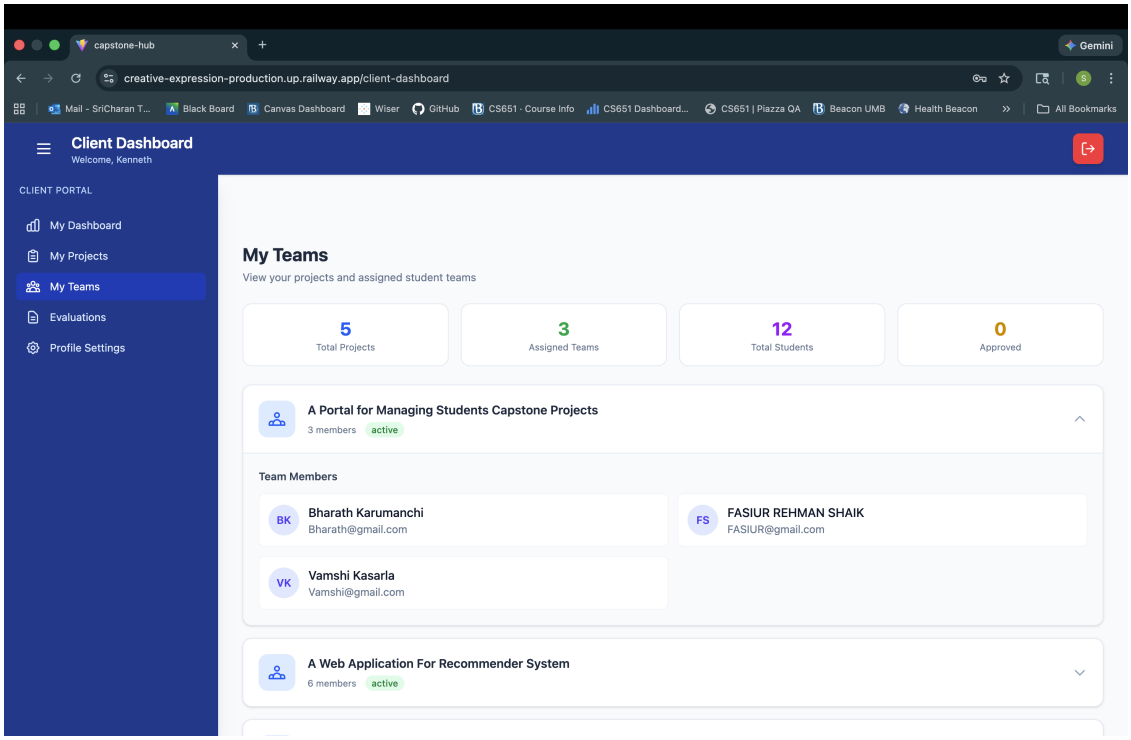

Figure 9: Teams view with member details

Evaluation Schedule

“Evaluations” lists upcoming reviews—sprints, milestones, or final presentations—along with dates, times, and locations.

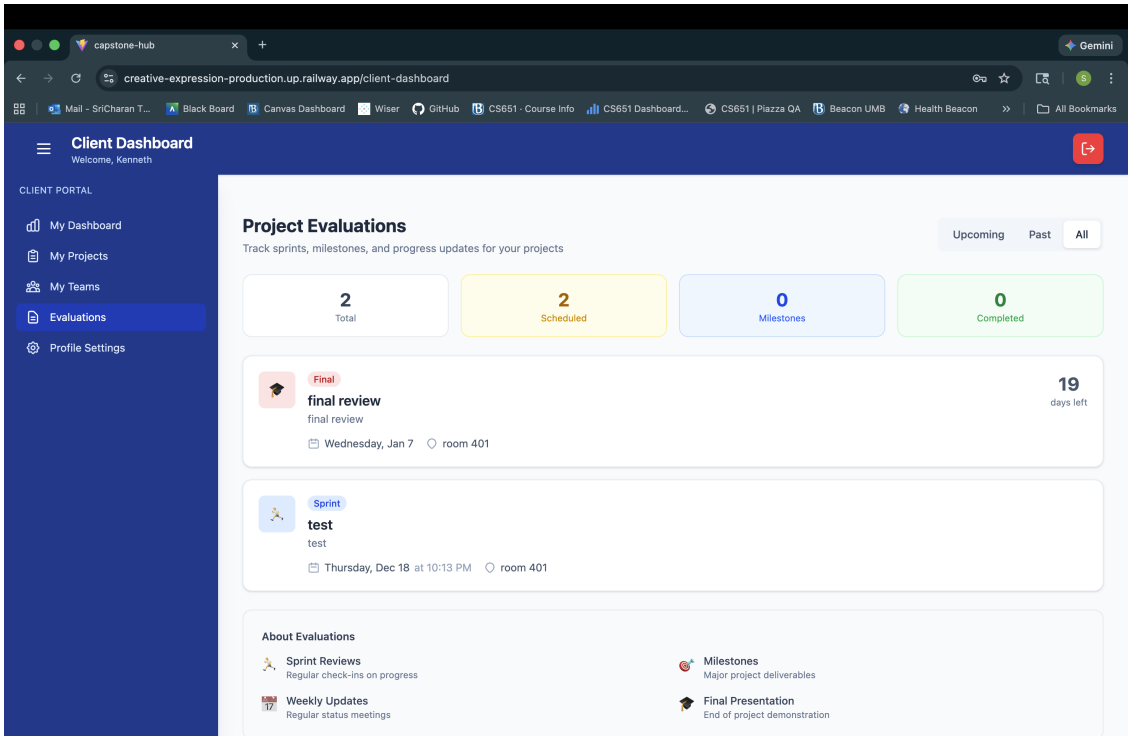

Figure 10: Client evaluation calendar

For Students

Your Dashboard

The student dashboard highlights two key items: whether you have submitted preferences yet, and how much time remains before the deadline. Quick-link cards let you jump to projects, preferences, or your group.

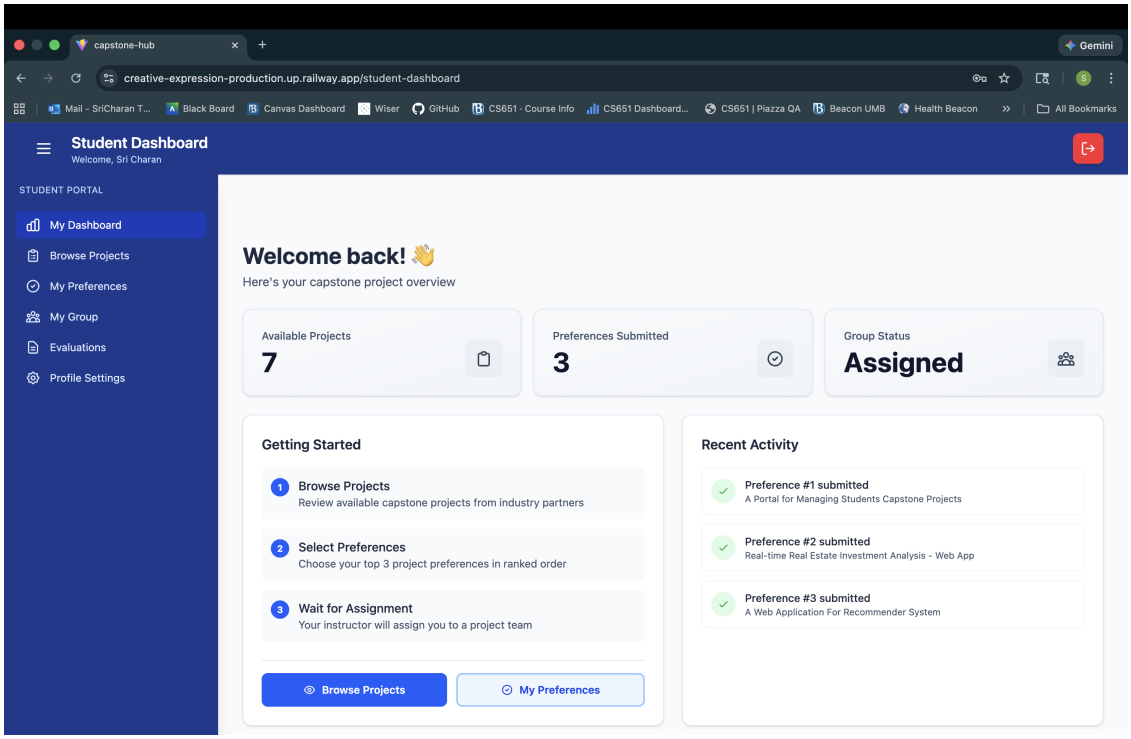

Figure 11: Student dashboard with deadline countdown

Exploring Projects

“Browse Projects” shows every approved proposal. Each card includes the title, client name, description snippet, required skills, and team size. Use this view to shortlist options.

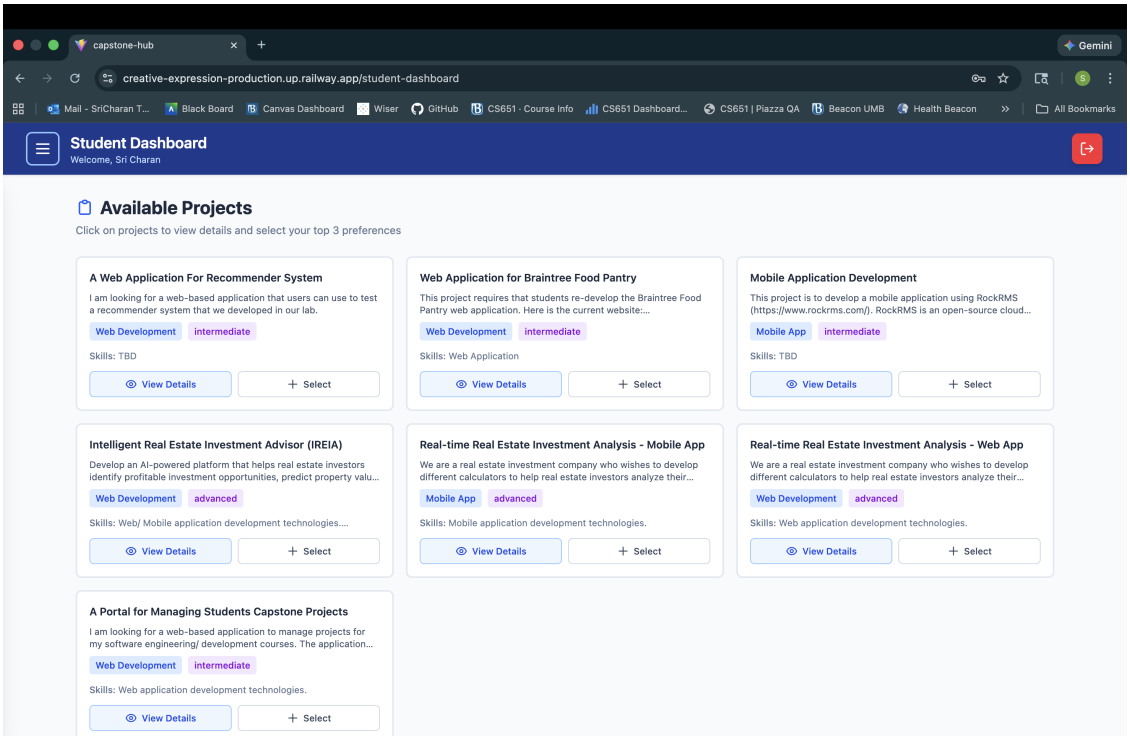

Figure 12: Project browsing grid

Ranking Your Choices

Open “My Preferences.” On the right you see available projects; click “Add” to move one into your ranked list on the left. Drag items up or down to reorder—position one is your top pick. When satisfied, click “Save Rankings.” You can revise anytime before the deadline.

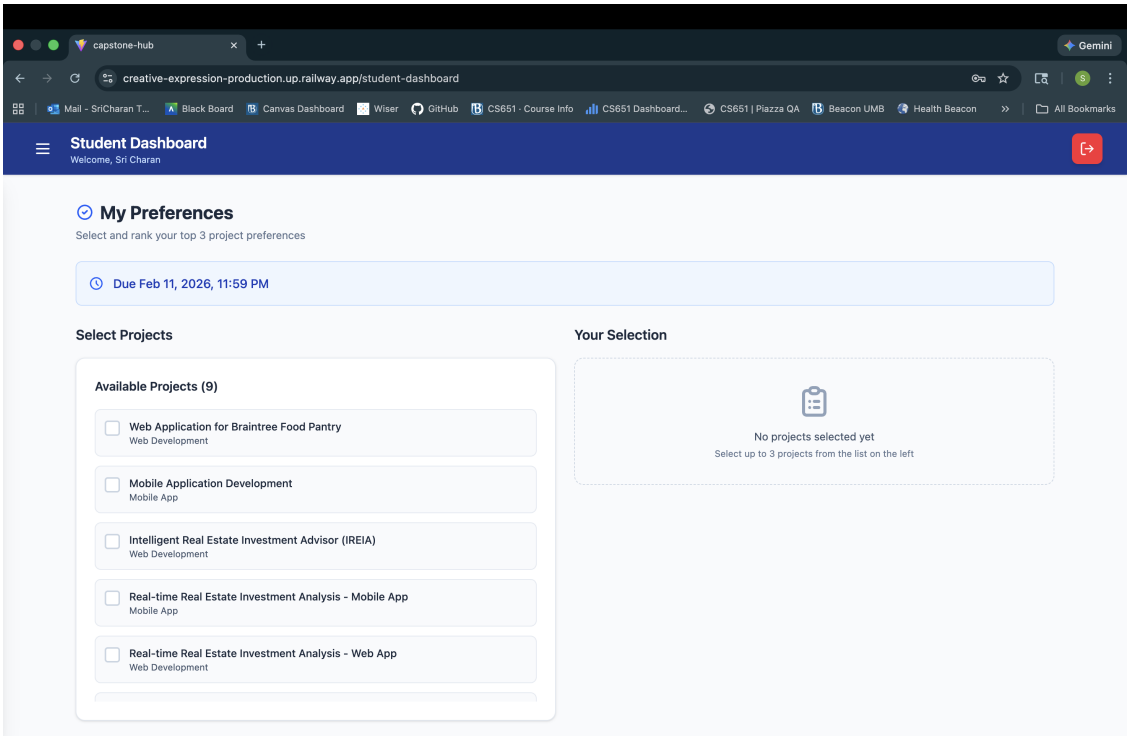

Figure 13: Preference ranking interface

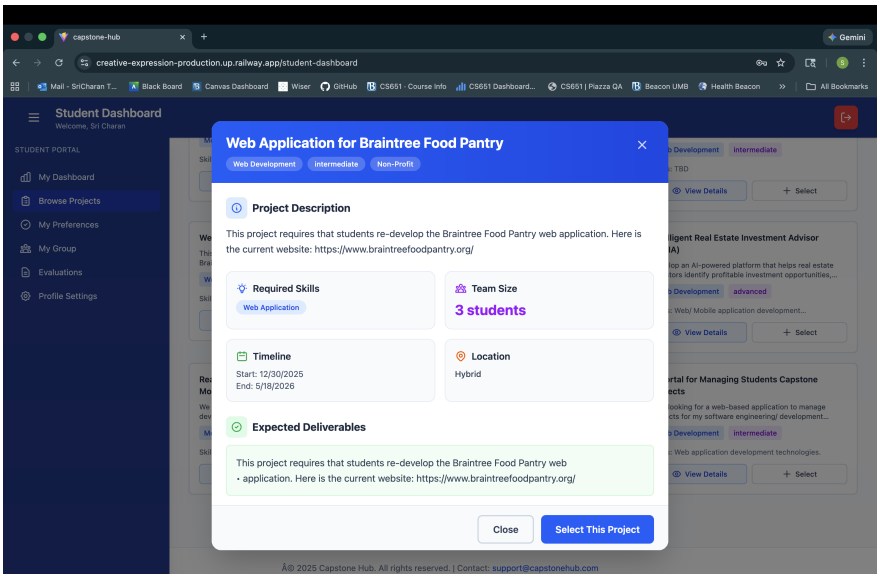

Figure 14: Project details modal

Group Assignment

Once the instructor finalizes teams, “My Group” reveals your project and teammates. Until then, this section simply notes that assignments are pending.

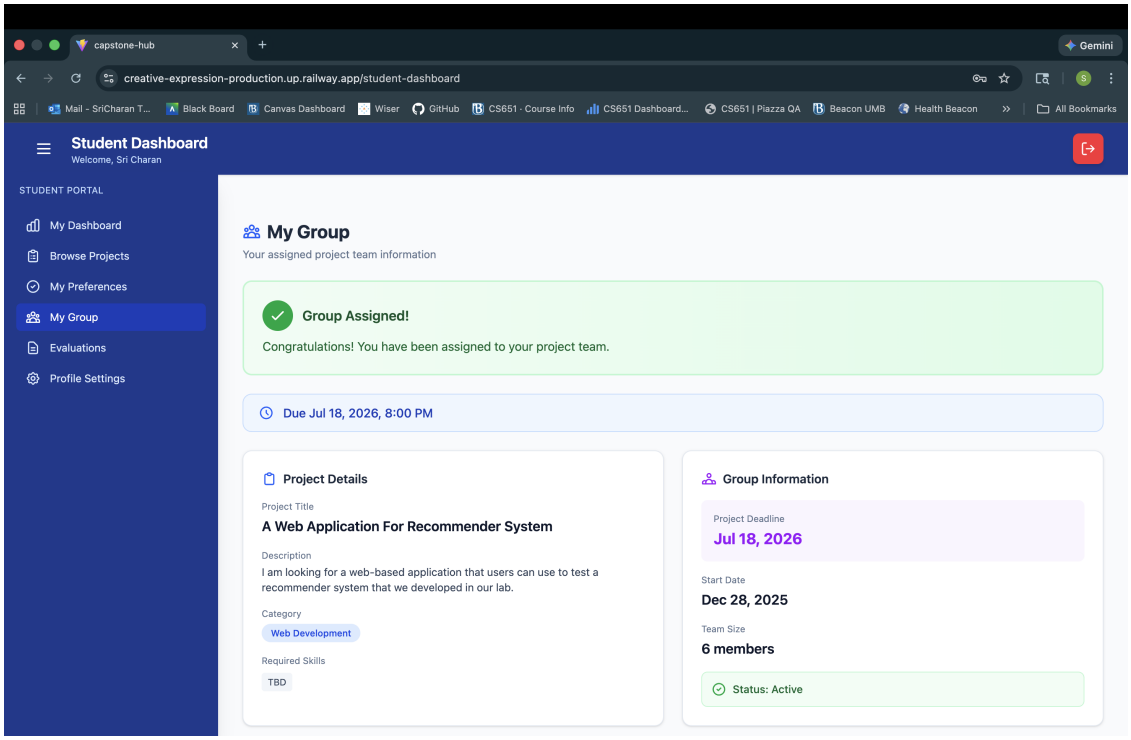

Figure 15: Assigned group details - Project Information

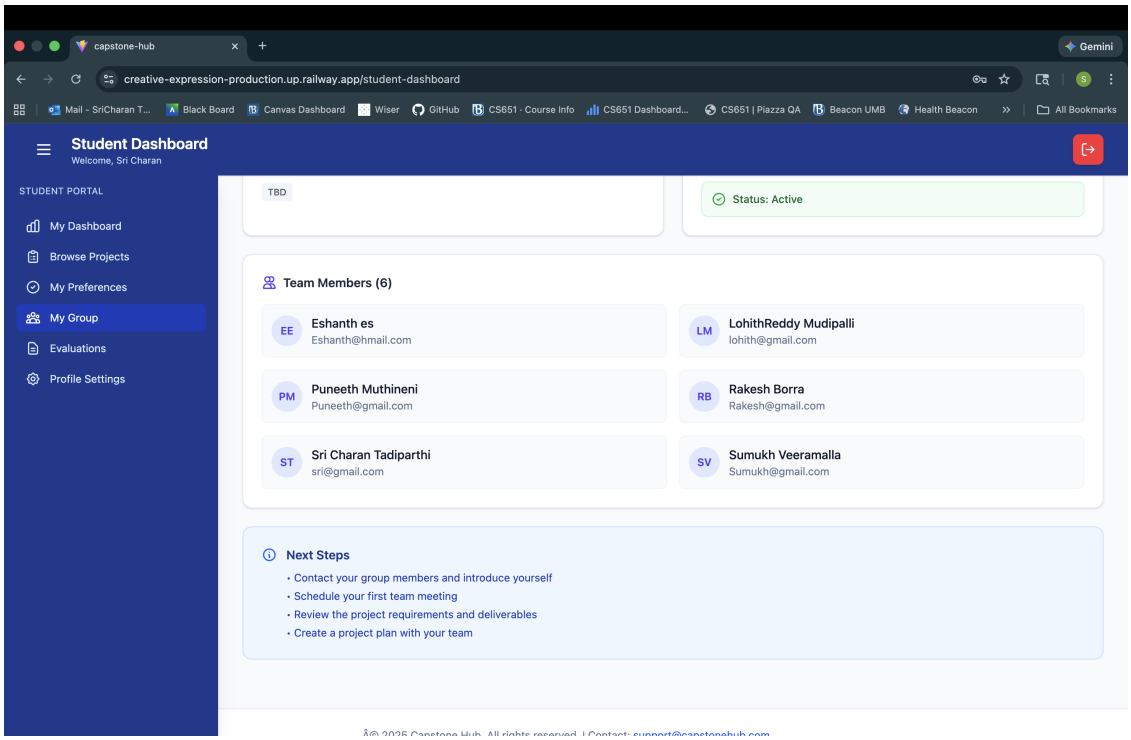

Figure 16: Assigned group details - Team Members

Evaluations

After joining a group, “Evaluations” shows your scheduled presentations. A countdown badge indicates days remaining for each event.

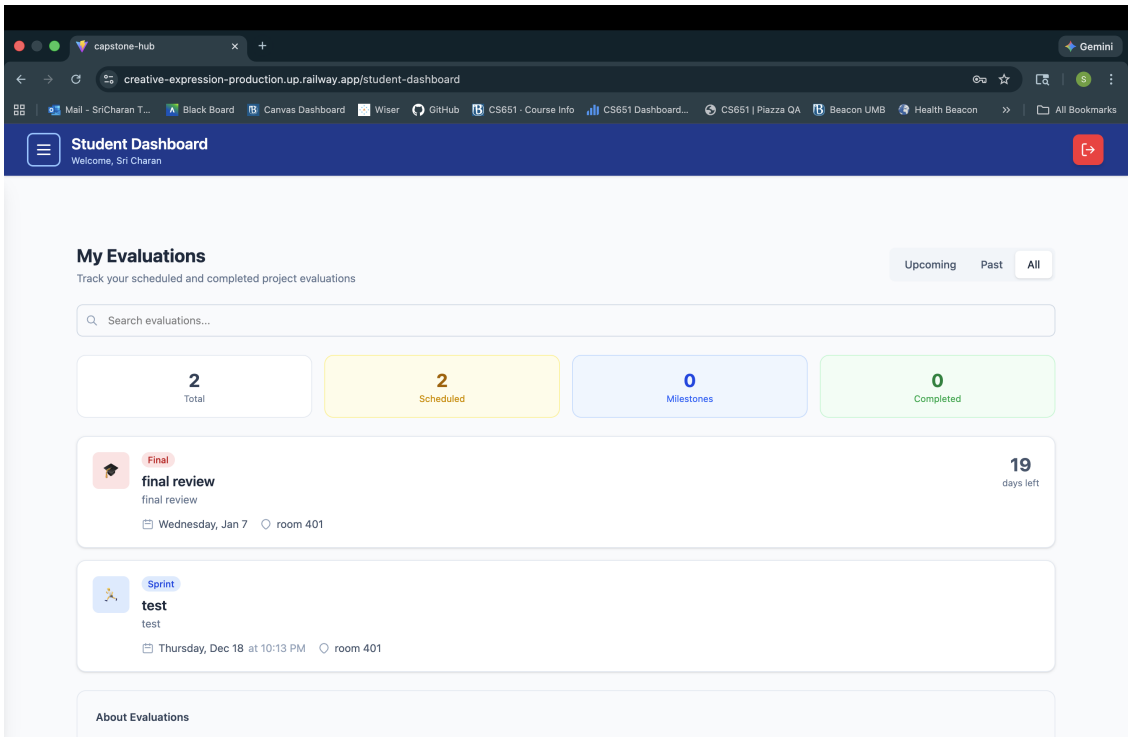

Figure 17: Upcoming evaluations with countdown

For Instructors

Your Dashboard

Summary tiles present at-a-glance metrics: total enrolled students, number of active projects, evaluations due soon, and the current preference deadline.

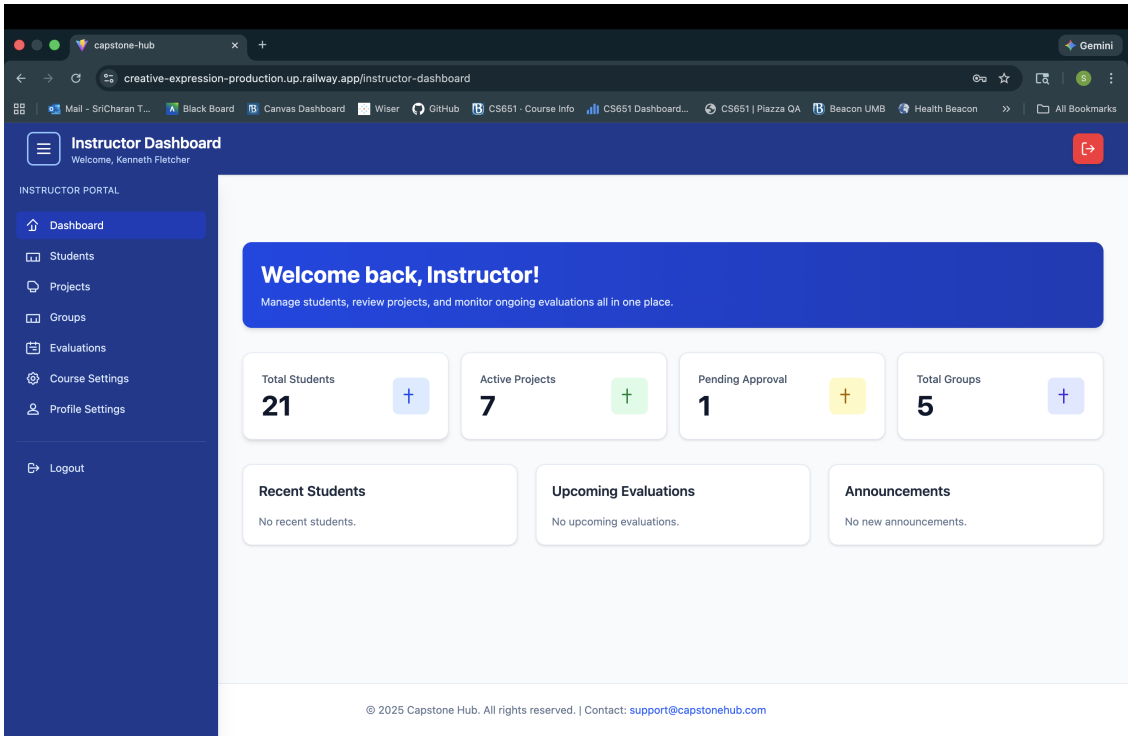

Figure 18: Instructor dashboard statistics

Student Roster

"Students" lists everyone registered with the student role. You can search by name, view assignment status, or manually add a student who missed self-registration.

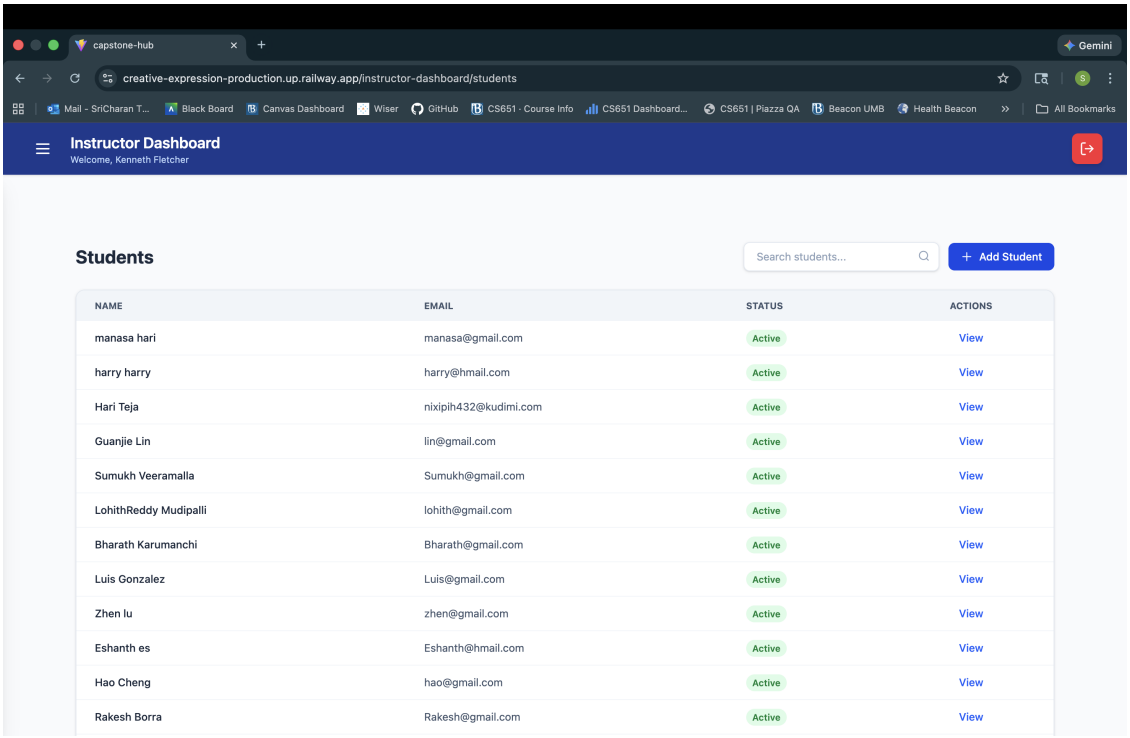

Figure 19: Student table

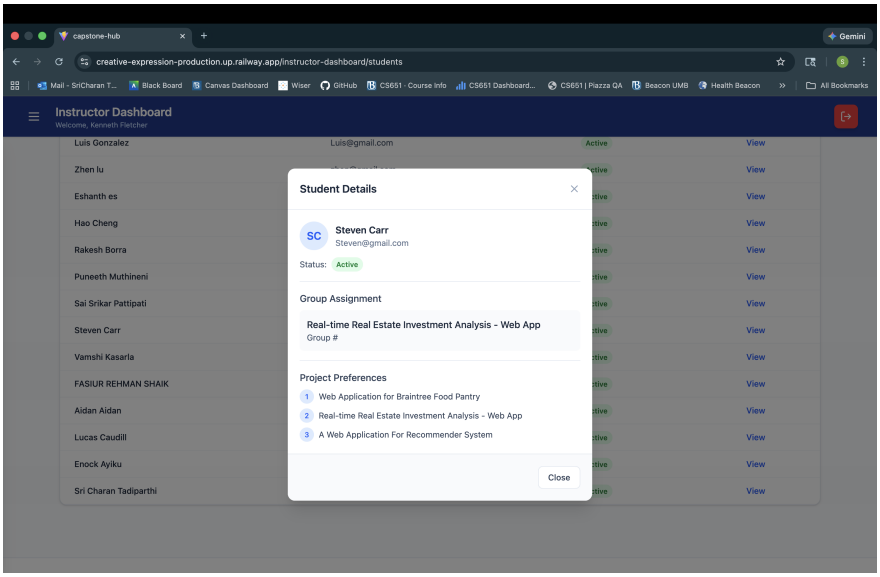

Figure 20: Student details modal

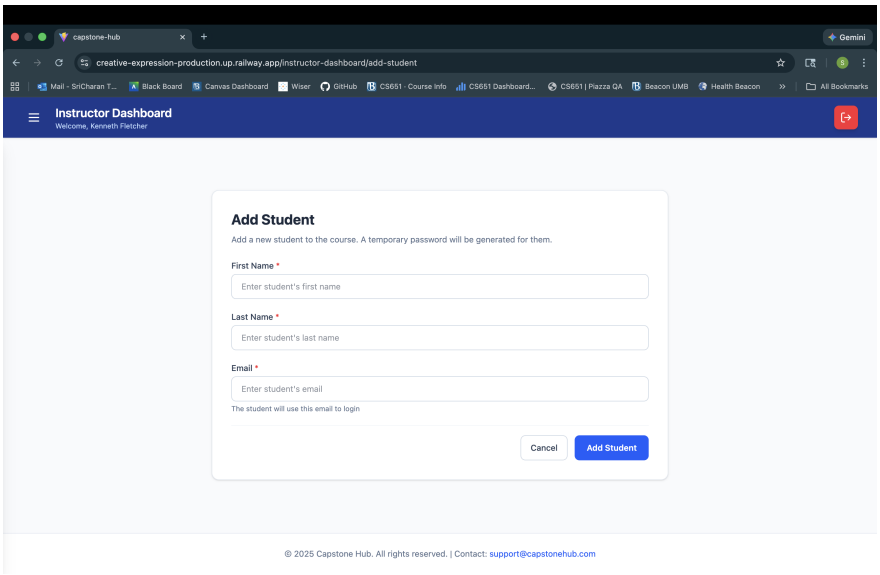

Figure 21: Add student form

Project Approval Workflow

"Projects" aggregates all client submissions. Click any card to inspect full details. From there, approve the proposal to make it visible to students, or reject it with written feedback. Rejected projects return to the client for revision.

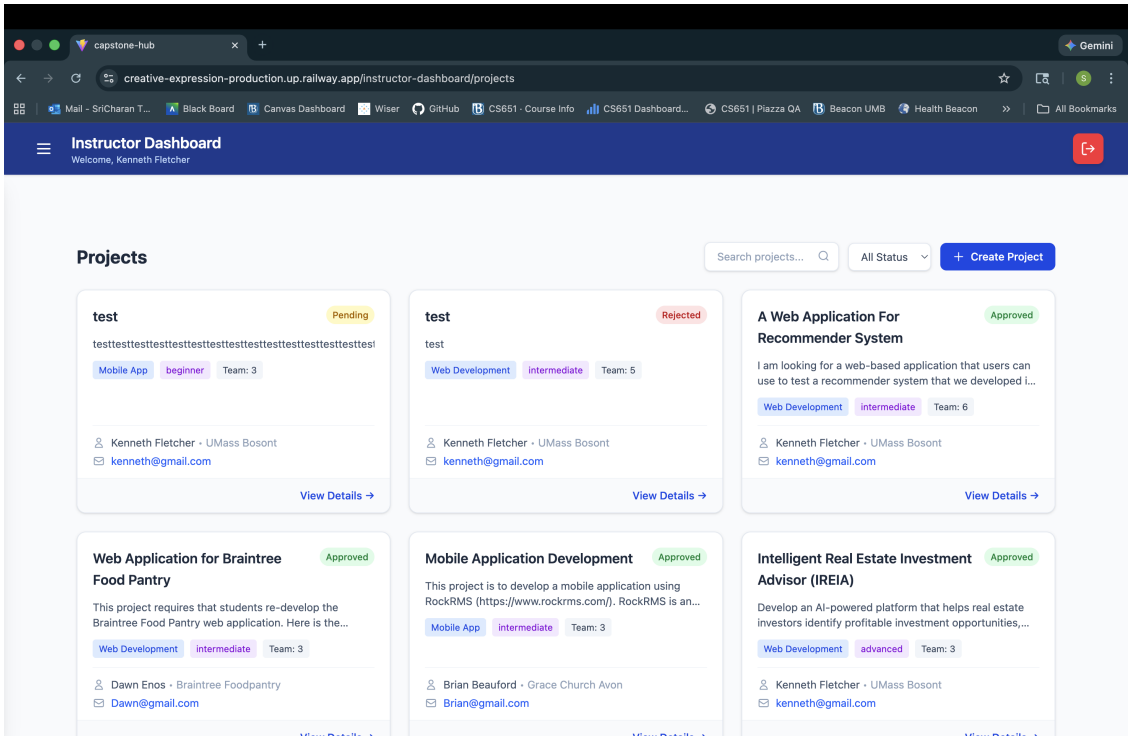

Figure 22: Projects overview

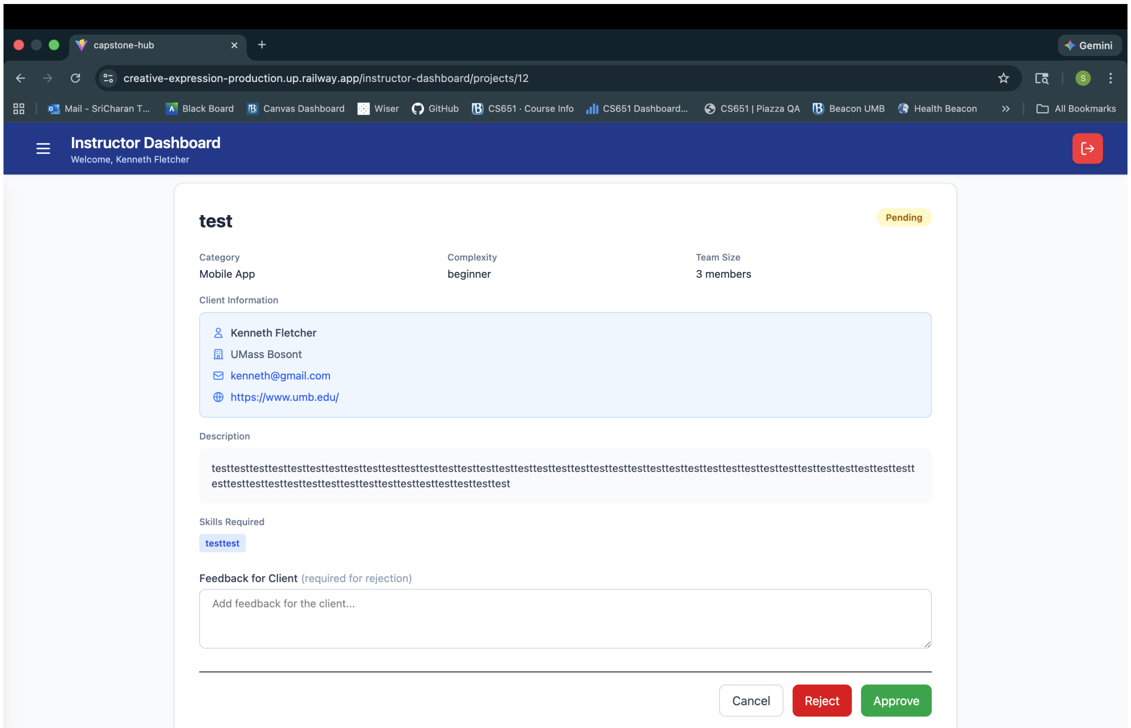

Figure 23: Project detail with approve/reject buttons

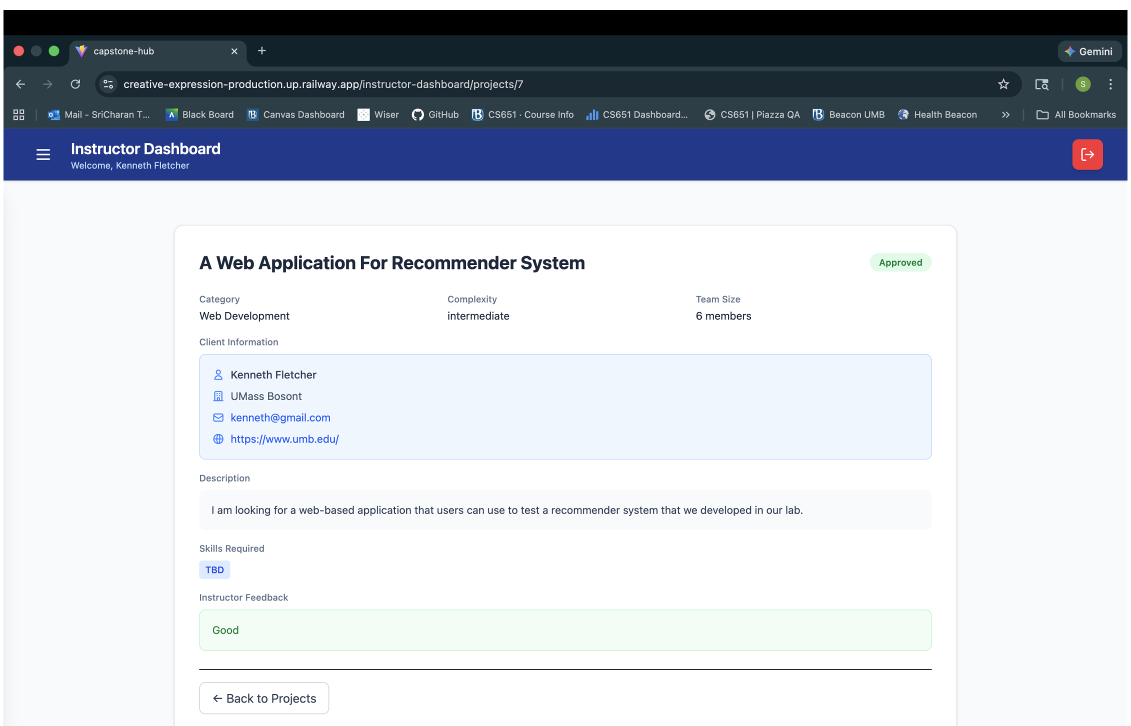

Figure 24: Approved project with instructor feedback

Building Teams Manually

Under “Groups,” click “Create Group.” Provide a name, choose a project, then multi-select students. Confirm to lock in the assignment.

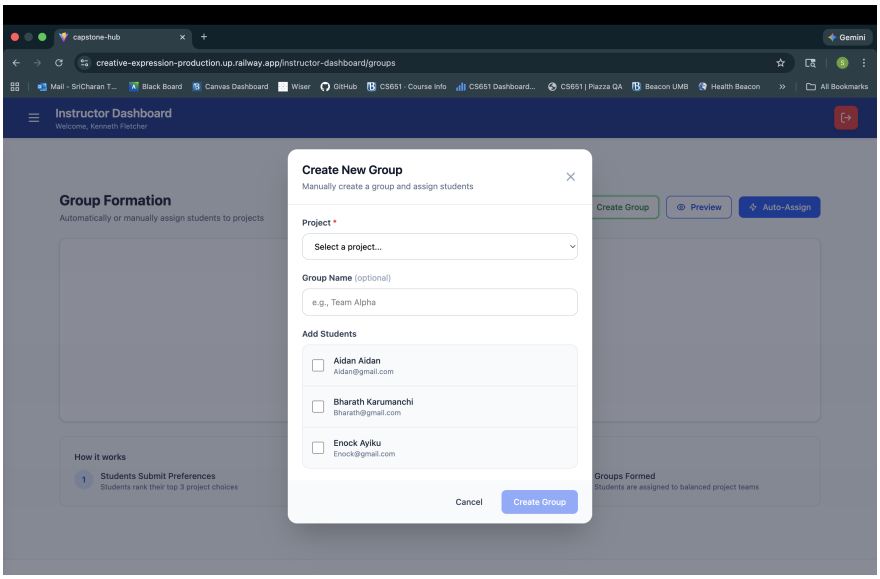

Figure 25: Manual group creation form

Automated Group Formation

Click “Automate Groups” for algorithm-driven assignments. Pick a target group size, then hit “Preview.” The system displays proposed teams alongside a satisfaction metric—the percentage of students placed in one of their preferred projects. If acceptable, finalize with one click; otherwise, adjust parameters and regenerate.

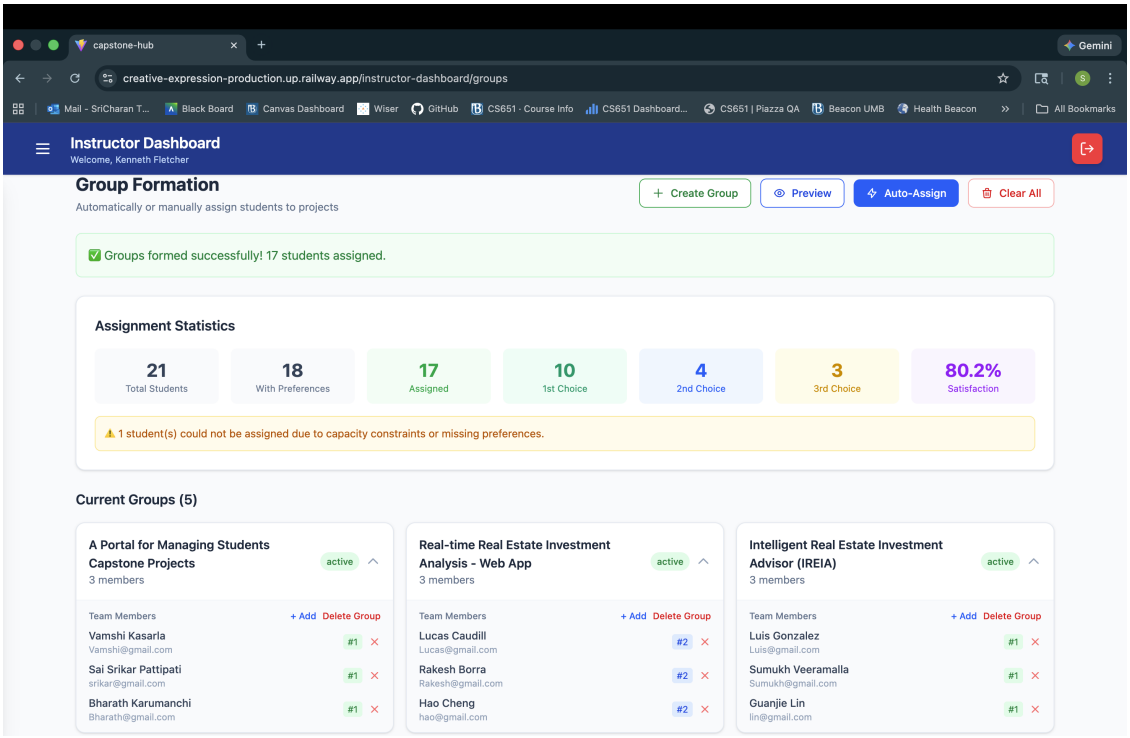

Figure 26: Groups formed after auto-assignment

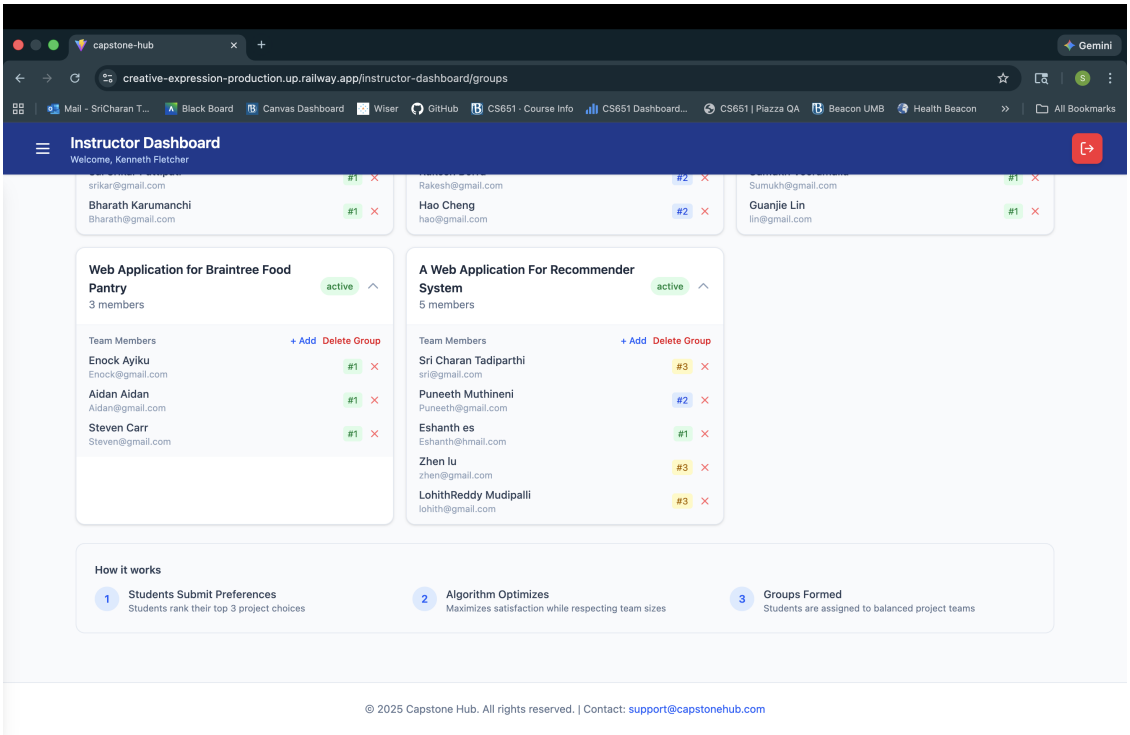

Figure 27: Auto-assignment - Groups with preference indicators

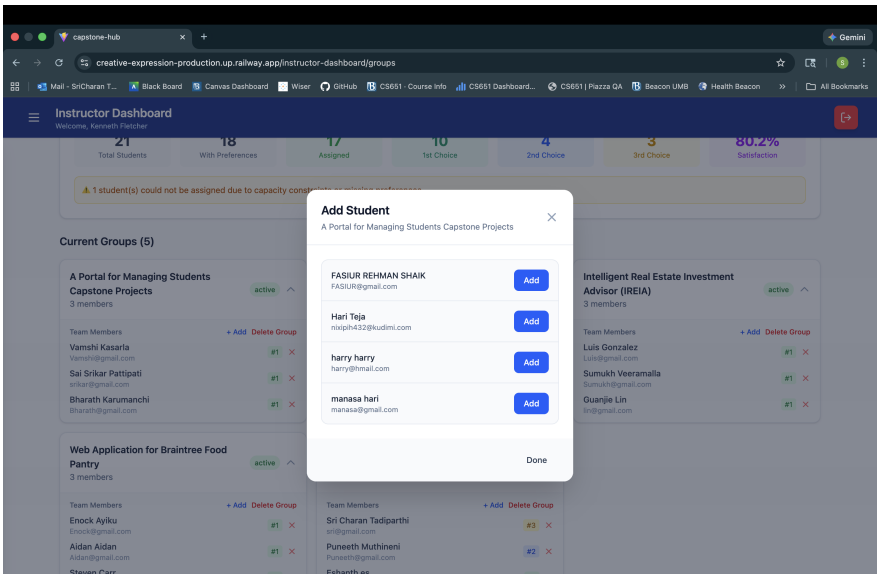

Figure 28: Auto-assignment - Add unassigned students to groups

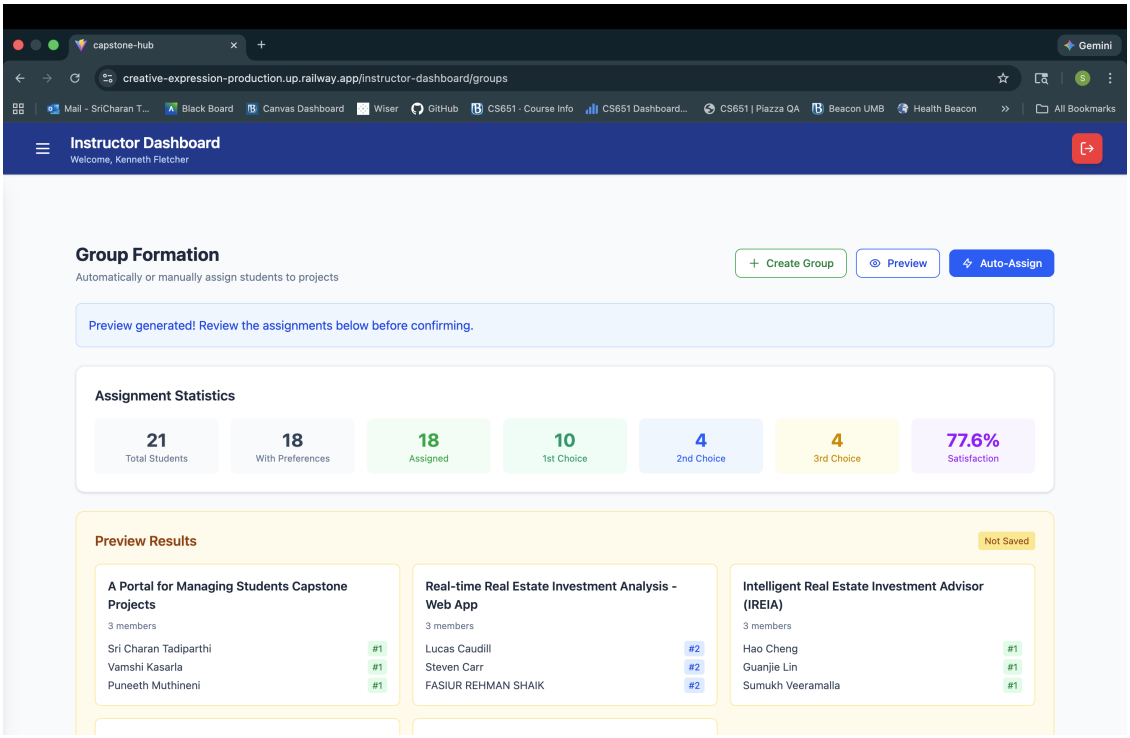

Figure 29: Preview with satisfaction score and assignment statistics

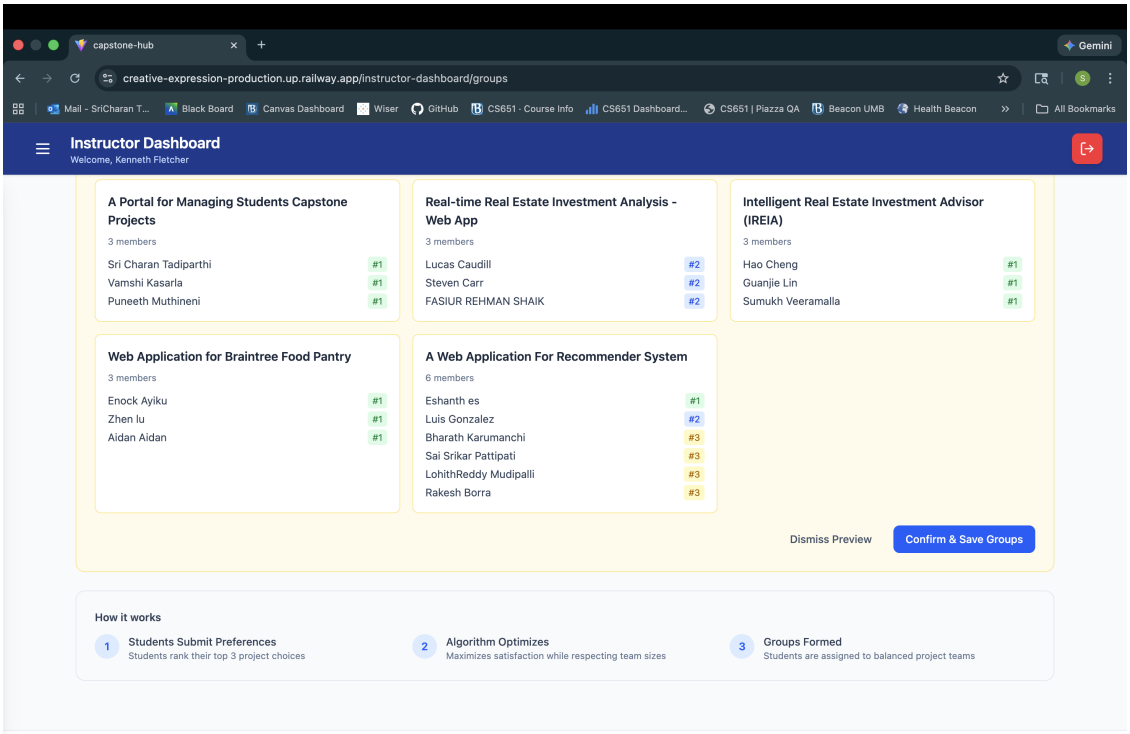

Figure 30: Preview results - Detailed group assignments

Scheduling Evaluations

“Evaluations” shows all reviews across every group. To add one, click “Schedule Evaluation,” select the team, set a date and time, choose a type (Sprint, Milestone, Weekly, Final), and specify a location or virtual meeting link.

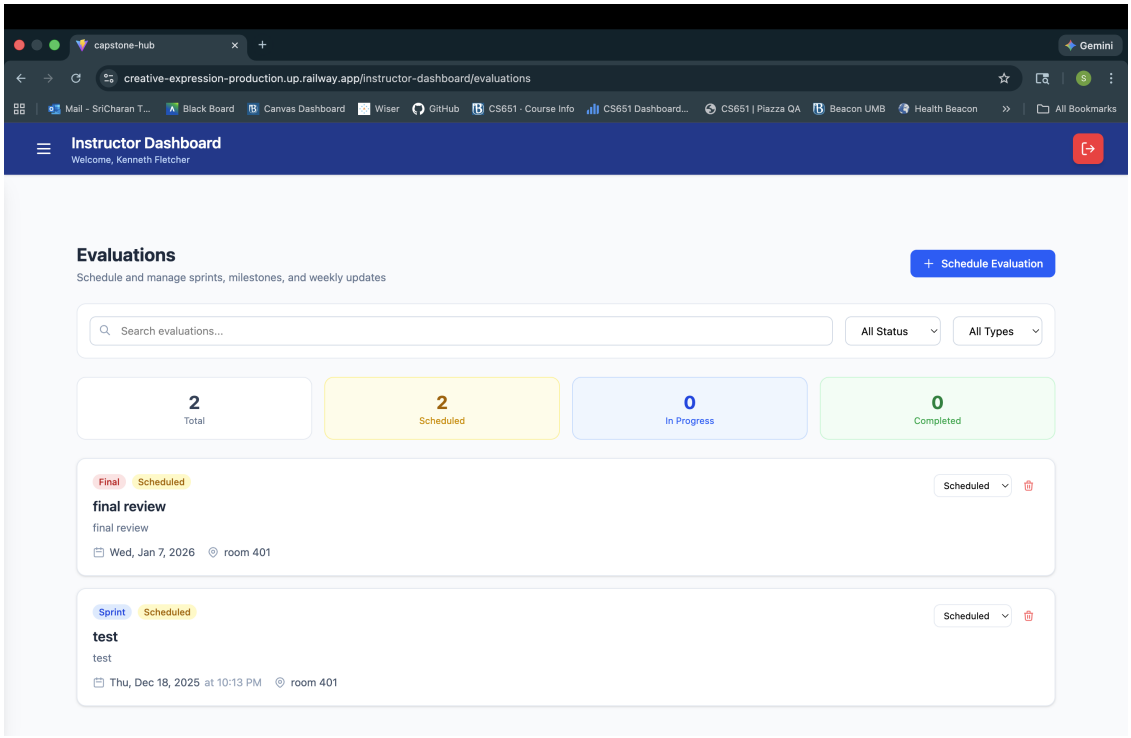

Figure 31: Evaluation list

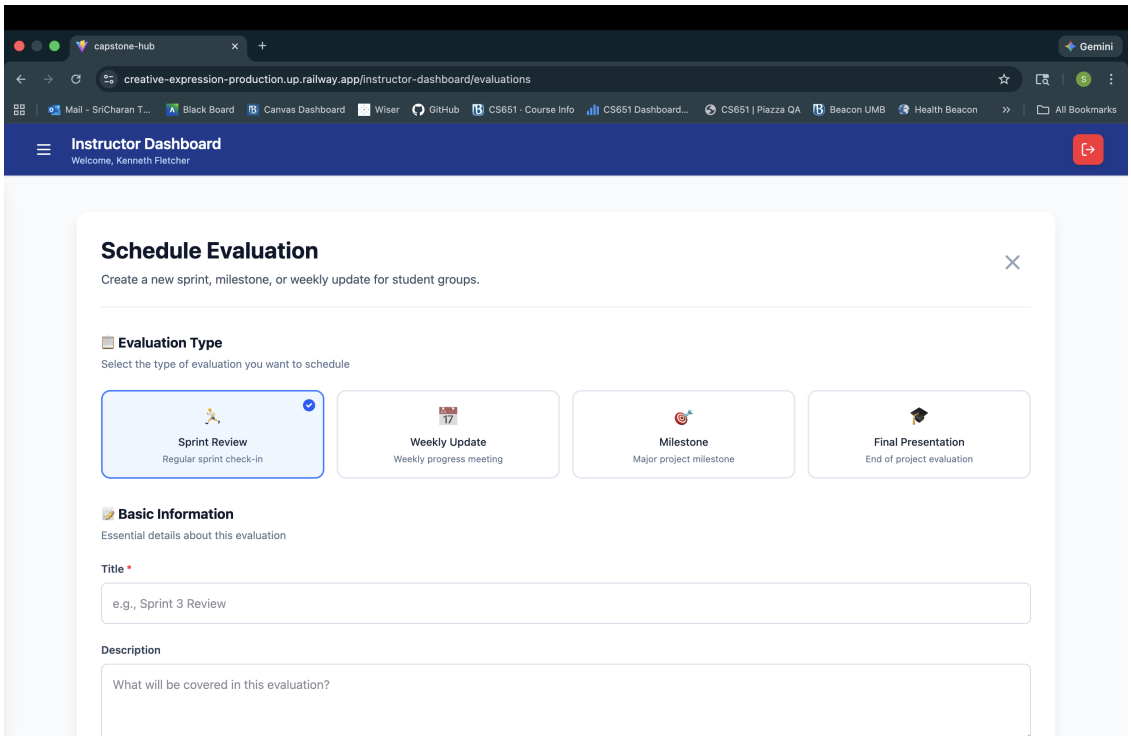

Figure 32: Schedule evaluation form - Evaluation Type

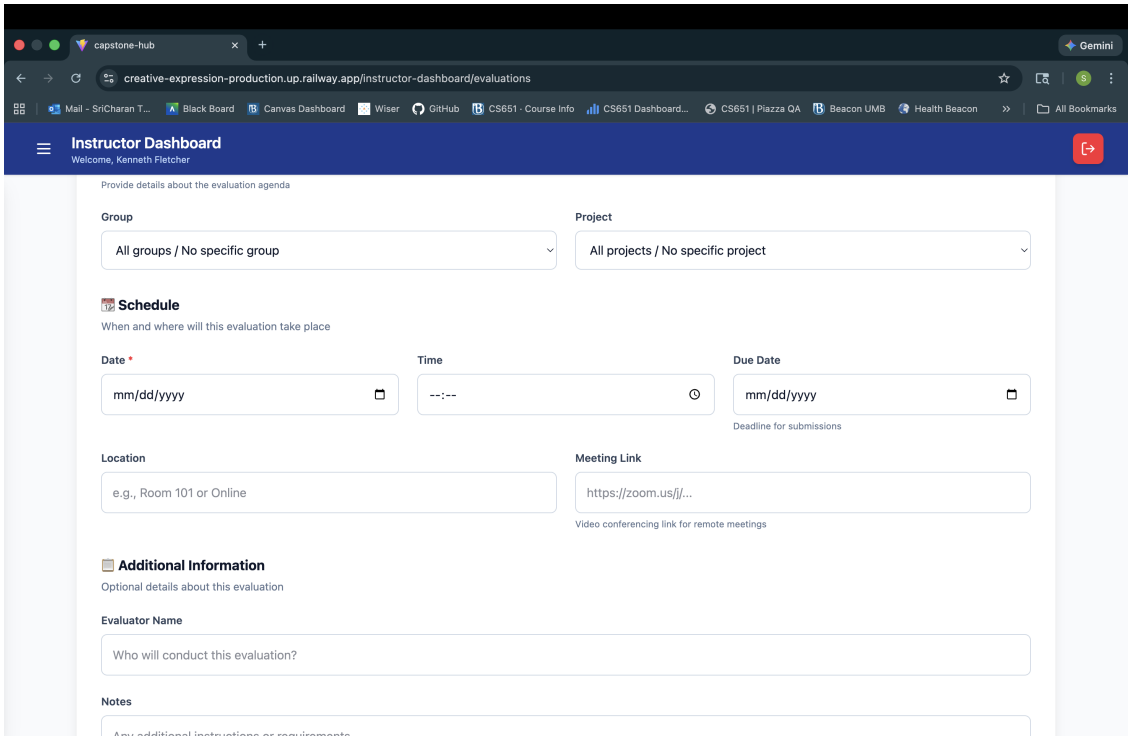

Figure 33: Schedule evaluation form - Schedule Details

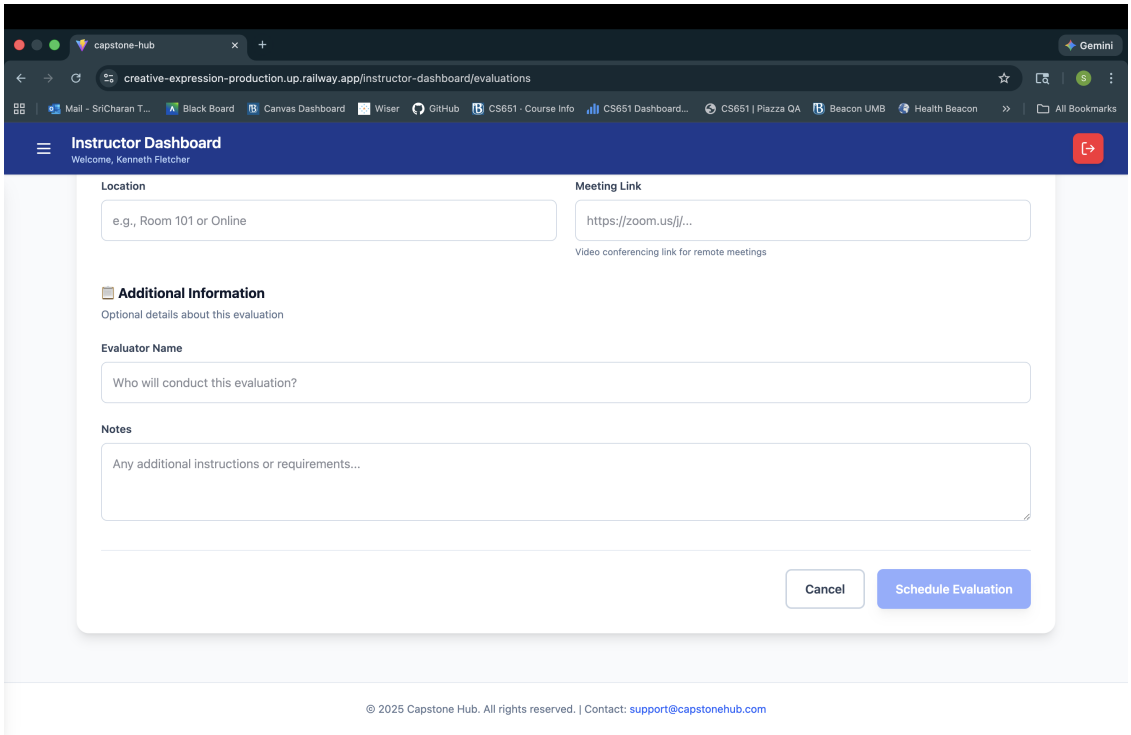

Figure 34: Schedule evaluation form - Additional Information

Setting the Preference Deadline

In “Course Settings,” pick a date and time by which students must submit rankings. Save, and the deadline propagates to every student dashboard.

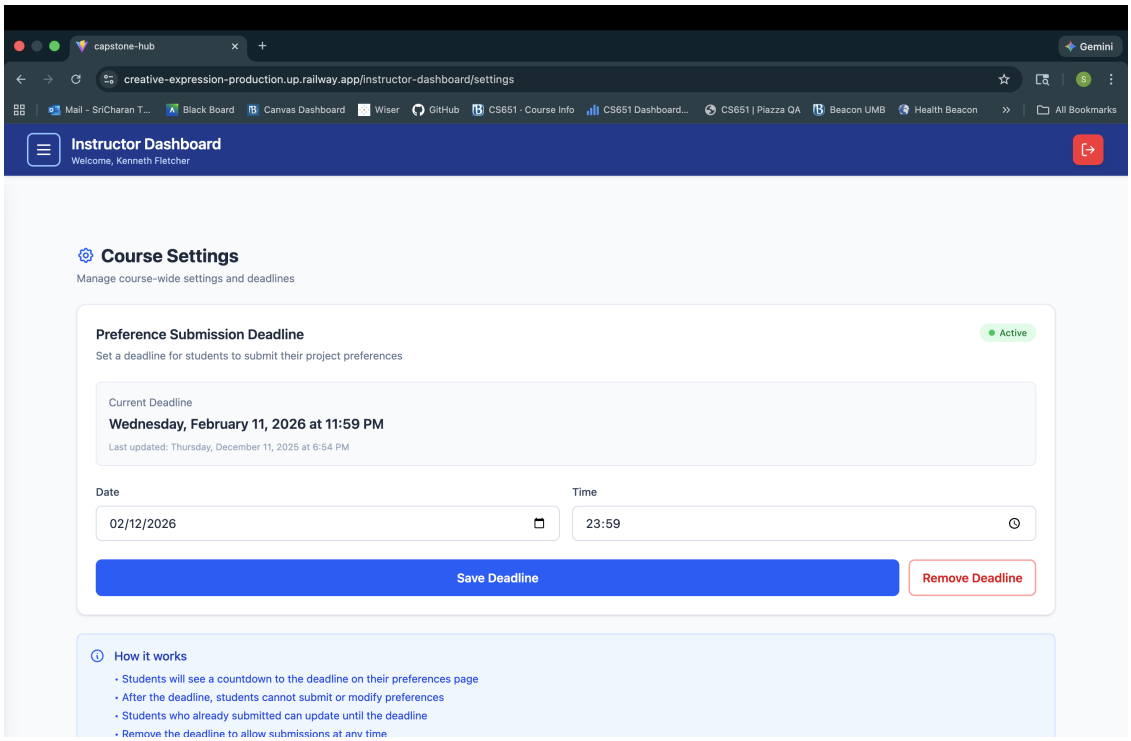

Figure 35: Deadline configuration

V. Common Issues

- **Login fails** — Double-check that your chosen role matches the one used during registration.
- **No projects visible (student)** — Only approved projects appear; pending ones remain hidden until an instructor acts.
- **Preferences won't save** — The deadline has passed. Contact your instructor.
- **Group page blank** — Teams have not been formed yet.
- **Evaluations missing** — You must be assigned to a group before evaluations appear.

VI. Technical Overview

Layer	Technology
Frontend	React 18, Vite, Tailwind CSS
Backend	Node.js, Express
Database	MySQL 8
Auth	Firebase Authentication, JWT
Hosting	Railway