Final project report (consistent with intermediate milestone documents).

Detailed report on work done from Milestone 1 through Milestone 5.
A section describing code review, issue reporting and tracking using screenshots.Components evaluated in final Project Evaluation:

Milestone-6

1. Project Scheduling Overview

Project scheduling is an important aspect of a software development project. Pivotal Tracker was used as the management tool for managing the project, assigning tasks and scheduling. A team meeting was held to plan and strategize the project roadmap and each member's responsibilities.

Agile Framework

This section explains the fundamental components of the agile framework that we have adopted for our project. We cover:

- **Sprint Length**: We use a one-week sprint length.
- **Multiple Feature Requests in a Sprint**: Our sprints can include multiple feature requests depending on the request complexity.
- **Labels**: Sprint-number, design, frontend, and backend.
- **Checklists**: Checklists are used to track and manage tasks within feature requests.

Sprint Management

This section delves into how we manage sprints within the project. Key points include:

- **Sprint Planning**: Eacn sprint kicks off with an explanation of the sprint requirements.
- **Team Involvement**: The team collaboratively selects and prioritizes feature requests.
- **Sprint Length Significance**: 1 week sprints allow us to have time bound iterations and a week-wise roadmap for the entire software development phase. It can accommodate multiple feature requests depending on their complexity.

Feature Requests

We describe how user stories and features are systematically converted into feature requests:

- **Identification**: User stories are converted to feature requests.
- **Definition**: Defining and breaking down features into actionable tasks.
- **Checklists**: To track progress within feature requests.
- **Activity:** To reflect progress, team members must share the progress in the feature request with relevant content and resources.
- **Status:** Each feature request has multiple status like unstarted, in progress, finished, accepted and rejected. It is important to keep them updated as per the current state of the progress.
- **Class Diagram:** A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Labels and Categorization

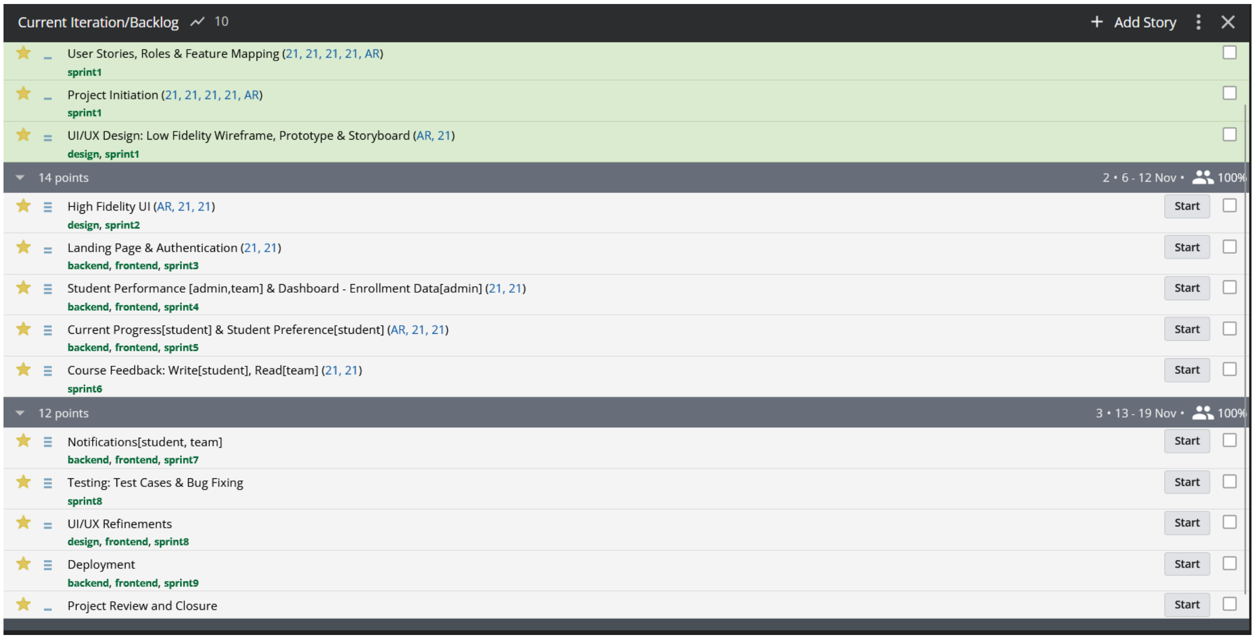
This section outlines our usage of labels in the project:

- **Sprint-number**: Helps in keeping track of the feature requests in a sprint wise manner.
- **Design, Frontend, Backend**: Three categories that define the type of feature request.

Pivotal Tracker

- The Project Manager has created feature requests from the user stories & requirements.
- Feature Requests have been defined and scheduled in a sprint wise manner.
- They are further categorized into design/frontend/backend tasks.
- Each feature request has been assigned a member by the Project Manager.
- Sprints kick off with team meetings and end with team review.
- Work progress gets mapped in the Activity section where members post their updates.
- Upon completion of sprint and team review, if all members approve the work, the feature request gets moved to accepted.

Below are some screenshots of our Project Management tool.



Current Iteration/Backlog 10 + Add Story

★ User Stories, Roles & Feature Mapping (21, 21, 21, 21, AR) □
sprint1

★ Project Initiation (21, 21, 21, 21, AR) □
sprint1

★ UI/UX Design: Low Fidelity Wireframe, Prototype & Storyboard (AR, 21) □
design, sprint1

14 points — 2 • 6 - 12 Nov • 100%

★ High Fidelity UI (AR, 21, 21) Start □
design, sprint2

★ Landing Page & Authentication (21, 21) Start □
backend, frontend, sprint3

★ Student Performance [admin,team] & Dashboard - Enrollment Data[admin] (21, 21) Start □
backend, frontend, sprint4

★ Current Progress[student] & Student Preference[student] (AR, 21, 21) Start □
backend, frontend, sprint5

★ Course Feedback: Write[student], Read[team] (21, 21) Start □
sprint6

12 points — 3 • 13 - 19 Nov • 100%

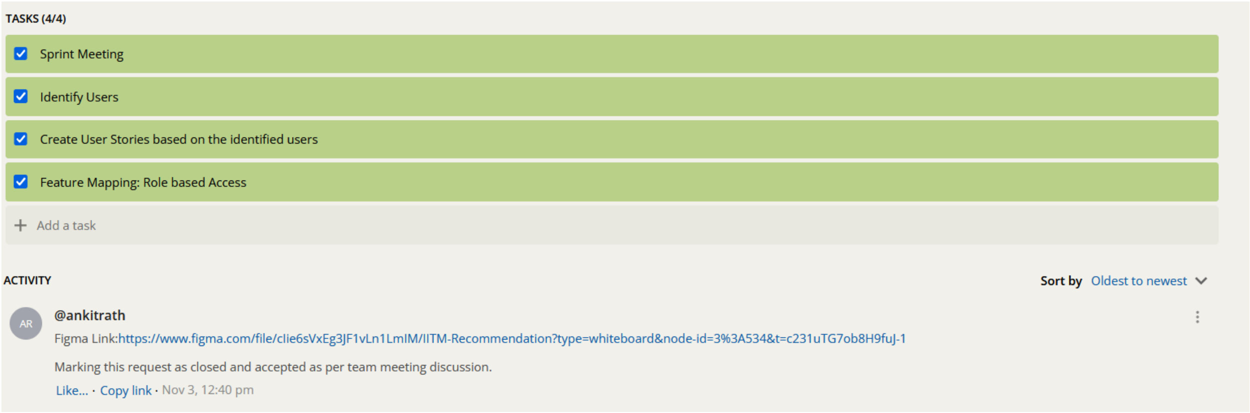
★ Notifications[student, team] Start □
backend, frontend, sprint7

★ Testing: Test Cases & Bug Fixing Start □
sprint8

★ UI/UX Refinements Start □
design, frontend, sprint8

★ Deployment Start □
backend, frontend, sprint9

★ Project Review and Closure Start □



TASKS (4/4)

☑ Sprint Meeting

☑ Identify Users

☑ Create User Stories based on the identified users

☑ Feature Mapping: Role based Access

+ Add a task

ACTIVITY Sort by Oldest to newest

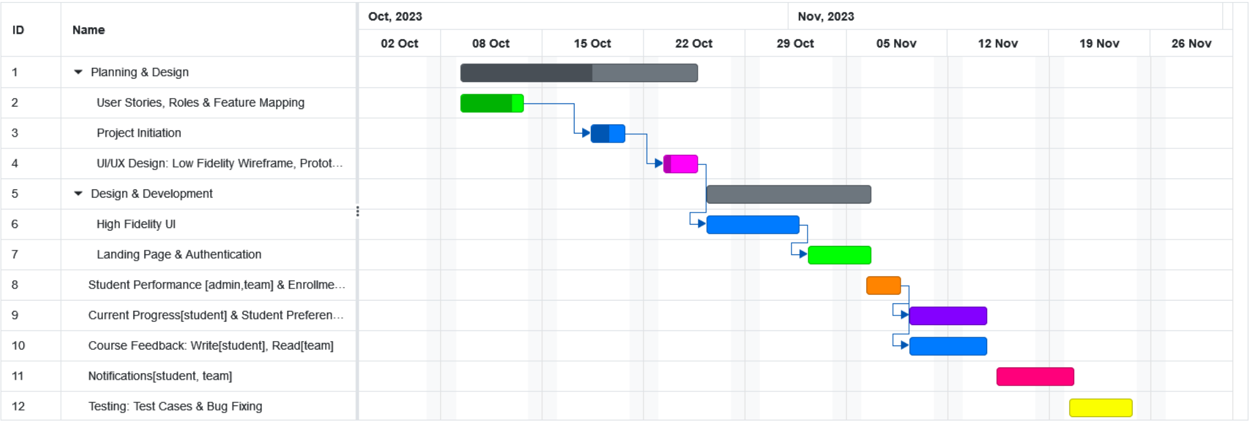
AR @ankitrath

Figma Link:https://www.figma.com/file/clie6sVxEg3JF1vLn1LmIM/IITM-Recommendation?type=whiteboard&node-id=3%3A534&t=c231uTG7ob8H9fuJ-1

Marking this request as closed and accepted as per team meeting discussion.

Like... · Copy link · Nov 3, 12:40 pm

Gantt Chart:



ID	Name
1	Planning & Design
2	User Stories, Roles & Feature Mapping
3	Project Initiation
4	UI/UX Design: Low Fidelity Wireframe, Protot...
5	Design & Development
6	High Fidelity UI
7	Landing Page & Authentication
8	Student Performance [admin,team] & Enrollme...
9	Current Progress[student] & Student Preferen...
10	Course Feedback: Write[student], Read[team]
11	Notifications[student, team]
12	Testing: Test Cases & Bug Fixing

2. Meeting Minutes

User Stories, Roles & Feature Mapping

- **Date:** 11-10-2023
- **Meeting Duration:** 30 minutes

Meeting Details/Minutes:

Agenda:

- Review the user stories provided.
- Discuss the roles and responsibilities within the team.
- Map the features to the user roles.

Meeting Summary:

1. The meeting began with a discussion of the user stories provided. Each user story was reviewed, and the team had a clear understanding of the project's primary and secondary users and their requirements.
2. Roles and responsibilities were discussed. The team identified the roles of developers, UI/UX designers and project manager. Responsibilities for each role were clarified.
    - Ankit Kumar Rath: Design, Project Manager
    - Yatin Chug: Developer, Architect
    - Chaitanya Kumaria: Quality Assurance Tester, Developer
    - Om Sharma: Developer, Technical Writer
    - Aayush Mahapatra: Developer, Architect

3. Feature mapping was done based on user roles. The following features were mapped:
    - Authentication: Required for all user roles.
    - Landing Page: Accessible by all user roles.
    - Dashboard (Current Progress): Available to all users.
    - Dashboard (Course Registration): Accessible by all users.
    - Dashboard (Course Feedback): Write access for primary and secondary users, read access for tertiary users.
    - Dashboard (Notifications): Available to all users.
    - Dashboard (Enrollment Data): Accessible by all users.
    - Dashboard (Student Performance): Read access for primary and secondary users.
    - Dashboard (Student Preference): Read access for primary and secondary users.
4. The project manager highlighted the importance of aligning the upcoming sprints with these user roles and features.

Action Items:

- Developers prepare for Sprint 2 based on feature mapping.
- UI/UX designer to start working on low-fidelity wireframes.

Project Initiation - Setup, Assigning Responsibilities

- **Date:** 18-10-2023
- **Meeting Duration:** 30 minutes

Meeting Details/Minutes:

Agenda:

- Discuss the progress of project initiation.
- Assign responsibilities for the setup phase.

Meeting Summary:

1. The meeting began with a review of the progress made during the project initiation phase. The team discussed the completion of the project charter, roles, and responsibilities matrix.
2. The project setup was discussed, including the establishment of project management tools, such as Trello, and the setup of the development environment.
3. Roles and responsibilities were reconfirmed, ensuring that all team members understood their roles and were ready to take on their assigned tasks.

Action Items:

- Developers to complete the setup phase and be ready for Sprint 4.

Low Fidelity Wireframe, Storyboard, Wireframe Prototype

- **Date:** 23-10-2023
- **Meeting Duration:** 30 minutes

Meeting Details/Minutes:

Agenda:

- Review the progress of low-fidelity wireframes.
- Discuss the initial storyboard and wireframe prototype.

Meeting Summary:

1. The meeting began with a review of the progress made on the low-fidelity wireframes. The UI/UX designer presented their initial designs, and the team provided feedback.
2. The team discussed the initial storyboard, ensuring that it accurately represented the user flow and interaction.
3. The wireframe prototype was demonstrated to the team, allowing everyone to interact with the design and provide feedback.
4. The project manager emphasized the importance of user feedback and iterative design in the upcoming sprints.

Action Items:

- UI/UX designer to refine wireframes based on feedback.
- Developers prepare for Sprint 3, focusing on project initiation and setup.

3. Design of Components

The system has been divided into multiple components. Each component has been defined below:

Landing Page: Users can view the details of the website using the landing page, this is the first page that will be shown to the user.

Authentication: Users can login into the system using the authentication component. Based on the user role, we can decide which components can be visible to the user and what cannot be accessed by the user.

Current Progress: This component allows the user, specifically the student to view the current progress they have made. This includes showing completed courses, grades obtained and due courses.

Course Recommendation: The course recommendation component allows the user to specify their preference based on their commitment and gives them an ideal suggestion for the course they should select.

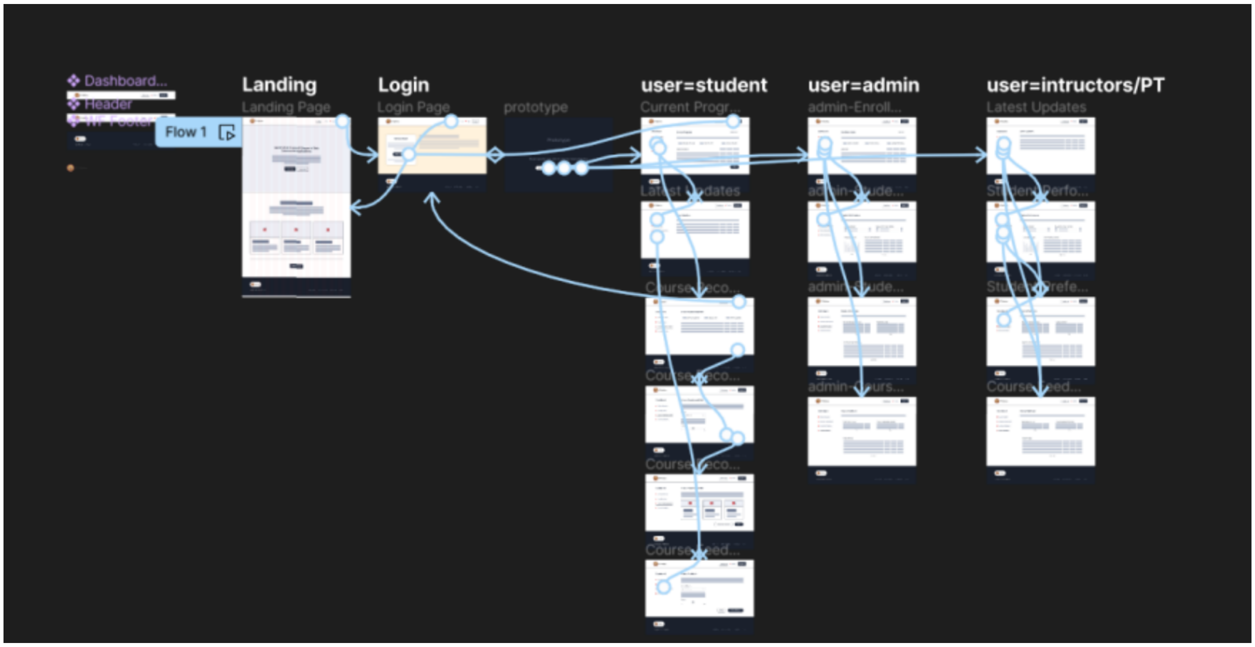
Course Feedback: The component allows students to give feedback for the courses they have completed. The planning team can view this feedback.

Notifications: Allows users to view the latest updates. Team can view alerts based on student performance.

Enrollment Data: Admin can upload, delete and view the enrollment data. This is a key requirement as multiple components depend on this component.

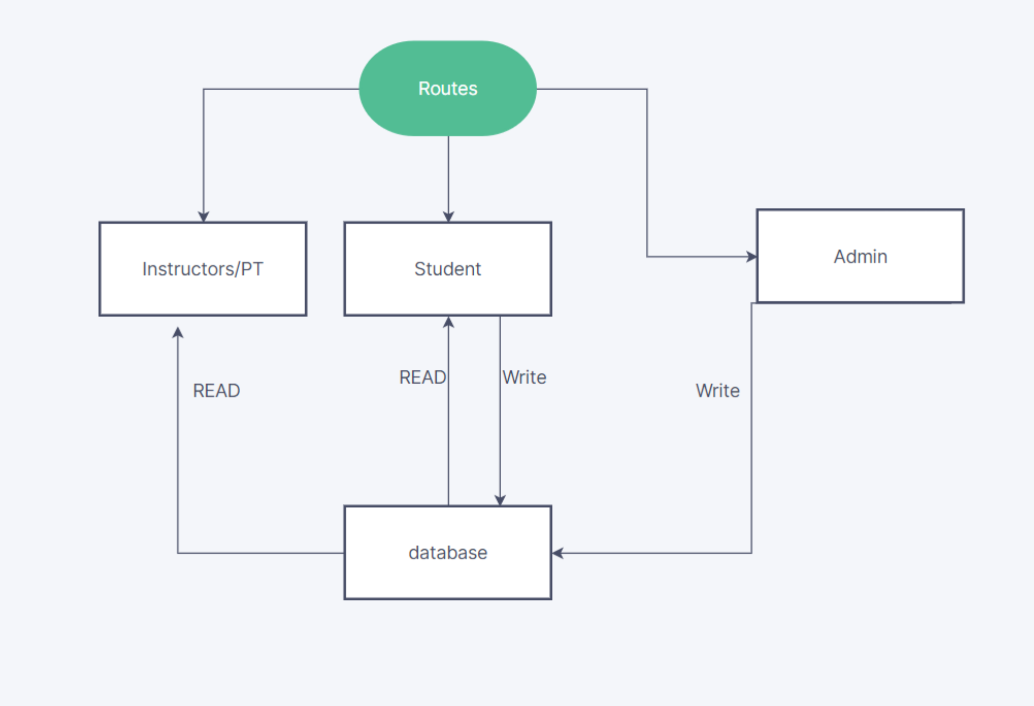
Student Performance: The team can view student performance. It includes, course wise performance, best performing students and pass fail ratio.

4. Software Design



Class Diagram:

- There will be 4 classes
    - **Routes**: Which consists of all the routes of the single file application
    - **Instructors**: The instructor will have read access to the database
    - **Admin:** admin will update the student database so, they will have write access to the database. They will be pushing the data
    - **student :** Both read and write access to the database. They will enter there info, and see there progress and recommendations.



The overall design of the project.
Teamwork that includes code review, issue reporting and tracking etc.
Efficient coding practices.
Consistent design through different milestones.

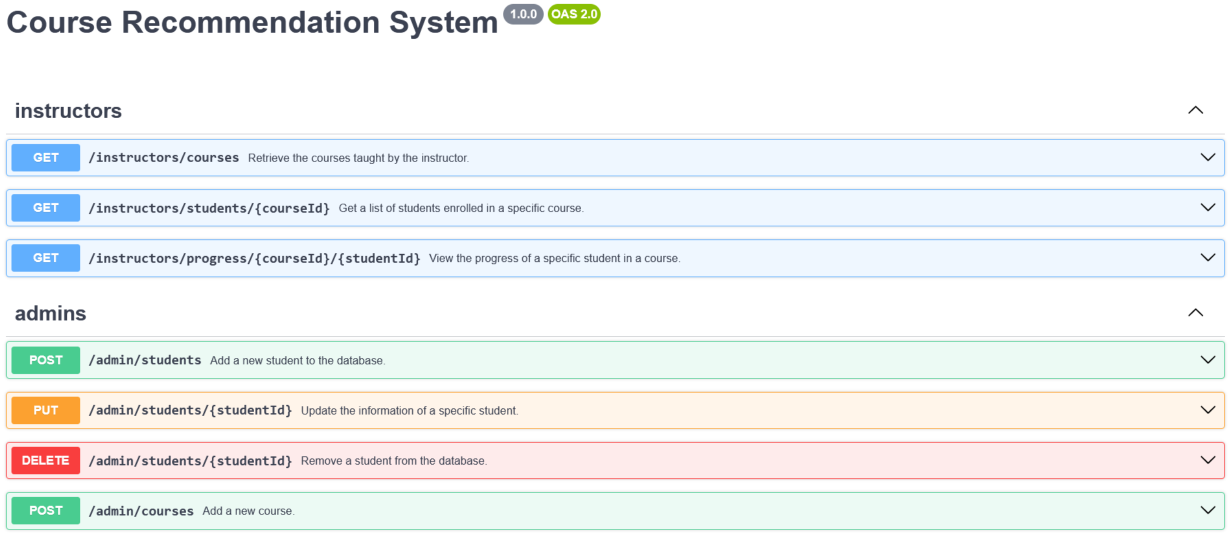
Milestone 4: API Endpoints

Deadline: End of Week 8 (November 17th)

Description: This milestone involved the creation of API endpoints for instructor, admin and students.

Code Review, Issue Reporting, and Tracking:

A screenshot of the Swagger documentation for the API can be found in the attached image.



Course Recommendation System 1.0.0 OAS 2.0

instructors

GET `/instructors/courses` Retrieve the courses taught by the instructor.

GET `/instructors/students/{courseId}` Get a list of students enrolled in a specific course.

GET `/instructors/progress/{courseId}/{studentId}` View the progress of a specific student in a course.

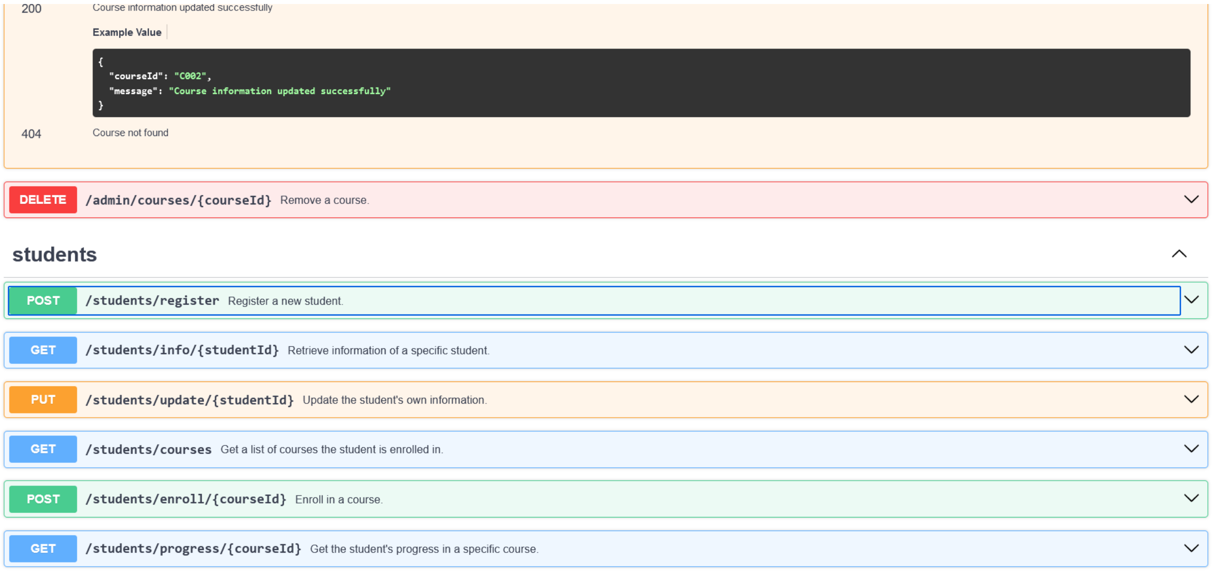
admins

POST `/admin/students` Add a new student to the database.

PUT `/admin/students/{studentId}` Update the information of a specific student.

DELETE `/admin/students/{studentId}` Remove a student from the database.

POST `/admin/courses` Add a new course.



200 Course information updated successfully

Example Value

```
{
  "courseId": "C002",
  "message": "Course information updated successfully"
}
```

404 Course not found

DELETE /admin/courses/{courseId} Remove a course.

students

POST /students/register Register a new student.

GET /students/info/{studentId} Retrieve information of a specific student.

PUT /students/update/{studentId} Update the student's own information.

GET /students/courses Get a list of courses the student is enrolled in.

POST /students/enroll/{courseId} Enroll in a course.

GET /students/progress/{courseId} Get the student's progress in a specific course.

All code was reviewed using the following process:
Developers submitted their code for review using pull requests.
Other developers reviewed the code and provided feedback.
The original developer made changes based on the feedback.
The code was then merged into the master branch.
Issues were tracked using a project management tool, i.e Jira.

Description of API Endpoints:

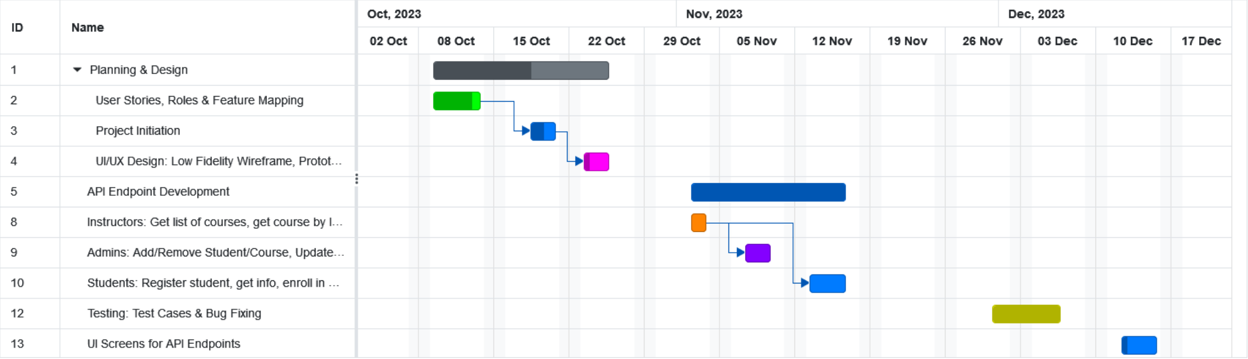
The API endpoints are designed to be RESTful and easy to use.
They are documented using Swagger, which makes it easy for developers to understand how to use them.
The API endpoints are secure and can only be accessed by authorized users.

Milestone 5: Test cases were defined. (check code and given summary):

Summary: This milestone focused on defining and documenting comprehensive test cases and building a test suite for the project's API endpoints. Each endpoint was covered with exhaustive scenarios, ensuring thorough functionality and error handling.

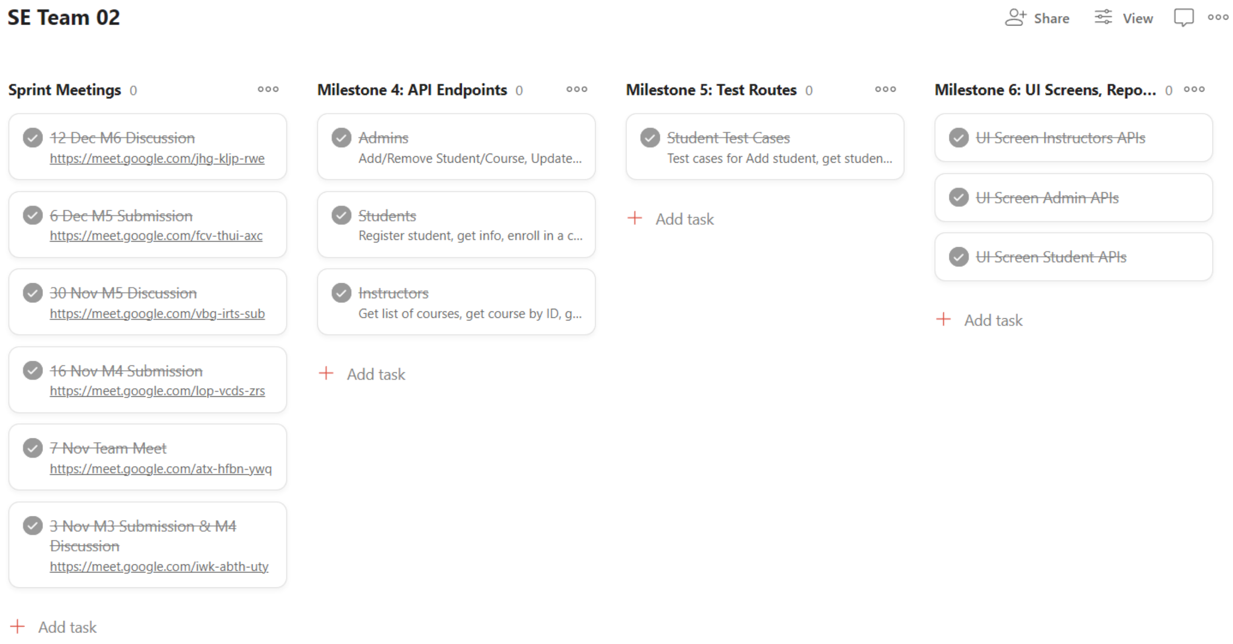
Milestone 6: UI screens for api endpoints were created.

Screenshots/Images:



ID	Name
1	Planning & Design
2	User Stories, Roles & Feature Mapping
3	Project Initiation
4	UI/UX Design: Low Fidelity Wireframe, Protot…
5	API Endpoint Development
8	Instructors: Get list of courses, get course by I…
9	Admins: Add/Remove Student/Course, Update…
10	Students: Register student, get info, enroll in …
12	Testing: Test Cases & Bug Fixing
13	UI Screens for API Endpoints

Issue Tracking: Kanban Board



SE Team 02

Sprint Meetings 0

- ~~12 Dec M6 Discussion~~
  https://meet.google.com/jhg-kljp-rwe
- ~~6 Dec M5 Submission~~
  https://meet.google.com/fcv-thui-axc
- ~~30 Nov M5 Discussion~~
  https://meet.google.com/vbg-irts-sub
- ~~16 Nov M4 Submission~~
  https://meet.google.com/lop-vcds-zrs
- ~~7 Nov Team Meet~~
  https://meet.google.com/atx-hfbn-ywq
- ~~3 Nov M3 Submission & M4 Discussion~~
  https://meet.google.com/iwk-abth-uty

+ Add task

Milestone 4: API Endpoints 0

- ~~Admins~~
  Add/Remove Student/Course, Update...
- ~~Students~~
  Register student, get info, enroll in a c...
- ~~Instructors~~
  Get list of courses, get course by ID, g...

+ Add task

Milestone 5: Test Routes 0

- ~~Student Test Cases~~
  Test cases for Add student, get studen...

+ Add task

Milestone 6: UI Screens, Repo... 0

- ~~UI Screen Instructors APIs~~
- ~~UI Screen Admin APIs~~
- ~~UI Screen Student APIs~~

+ Add task