SOFTWARE ENGINEERING

Project

MAY 2024 - TEAM 27

CONTENTS

1 Meet Our Team

2 Problem Statement

3 Type of Users

4 User Stories

5 Wireframes

6 Schedules & Designs

7 API Routes

8 Implementation Details

9 Technology Used

10 How to Run?

MEET OUR TEAM

DEV KHATRI

SATISH JAISWAL

PREM KUMAR

SHRIRANG SAPATE

SRISHTI SINGH

SHRIKANTH V

PRANAV R

SHRISTI TIWARI

PROBLEM STATEMENT

Problem Statement

With the rapid advancement of generative AI (GenAI) technologies, there are promising possibilities of effective integration of GenAI into these learning environments. The goal is to examine ways in which GenAI can be effectively integrated into learning environments such as in the SEEK portal.

1. Learning Environment

- Learners can watch video lectures.
- They can get announcements regarding the course via the portal.

Problem Statement

2. Coding Environment

- They can code in the given code editor and can get instant output on the code written by them.
- They can submit practice and graded programming assignments and can verify how accurate their code is.

3. GenAI Support for Learners

- Learners can ask the chatbot about the queries they are having in the course.
- Learners can ask the chatbot about the doubts they are having in the video lectures in the course.
- Chatbot can give the corrected version of code to the learners in the practice assignments section for learning purposes.

Problem Statement

4. Admin Role

- Admin can do CRUD operations for video lectures
- Admin can post announcements which will be visible to learners.
- Admin can manage learners(all the crud operations for learners)

TYPE OF USERS

Users	Remark
Learners/Students	Learners can watch video lectures, can get course support, can code in the given code editor, can submit practice and graded programming assignments, ask the chatbot about the submitted code, get help from chatbot in practice assignments, can get corrected version of code from chatbot for learning purposes, can ask the doubts they are having in the video lectures, the chatbot will answer the queries they are having in the lecture video.
Admin	Admin can do CRUD operations for video lectures, post announcements which will be visible to learners, can manage learners(all the crud operations for learners).

USER STORIES

USER STORIES WITHOUT GEN AI

Scenario 1: Asking video lecture doubts

A A learner watches a video lecture for the course and post a doubt in forum.

B Instructor/TA check the doubts and answers the query.

Scenario 2: Accessing Course Support

A A learner needs help with understanding the course rules and regulations. Learners mail the support team.

B Support team navigate to the mail and answers the query.

Scenario 3: Solving a Programming Assignment Problem

A A learner logs into the SEEK portal.

B Navigates to the course page and selects the programming question.

C Reads the problem statement and writes code in the provided editor.

D Submits the code and receives immediate feedback on test cases passed/failed.

E Learner tries to understand why certain test cases failed and attempts to debug and resubmit.

USER STORIES WITH GEN AI

Scenario 1: Asking video lecture doubts

A A learner watches a video lecture for the course and ask the doubt to the chatbot.

B The GenAI assistant immediately answers the query of the learner in detail.

Scenario 2: Accessing Course Support

A A learner needs help with understanding the course rules and regulations. Learners ask the chatbot.

B The GenAI assistant immediately answers the query of the learner in detail.

Scenario 3: Solving a Programming Assignment Problem

A A learner logs into the SEEK portal.

B Navigates to the course page and selects the programming question.

C Reads the problem statement and writes code in the provided editor.

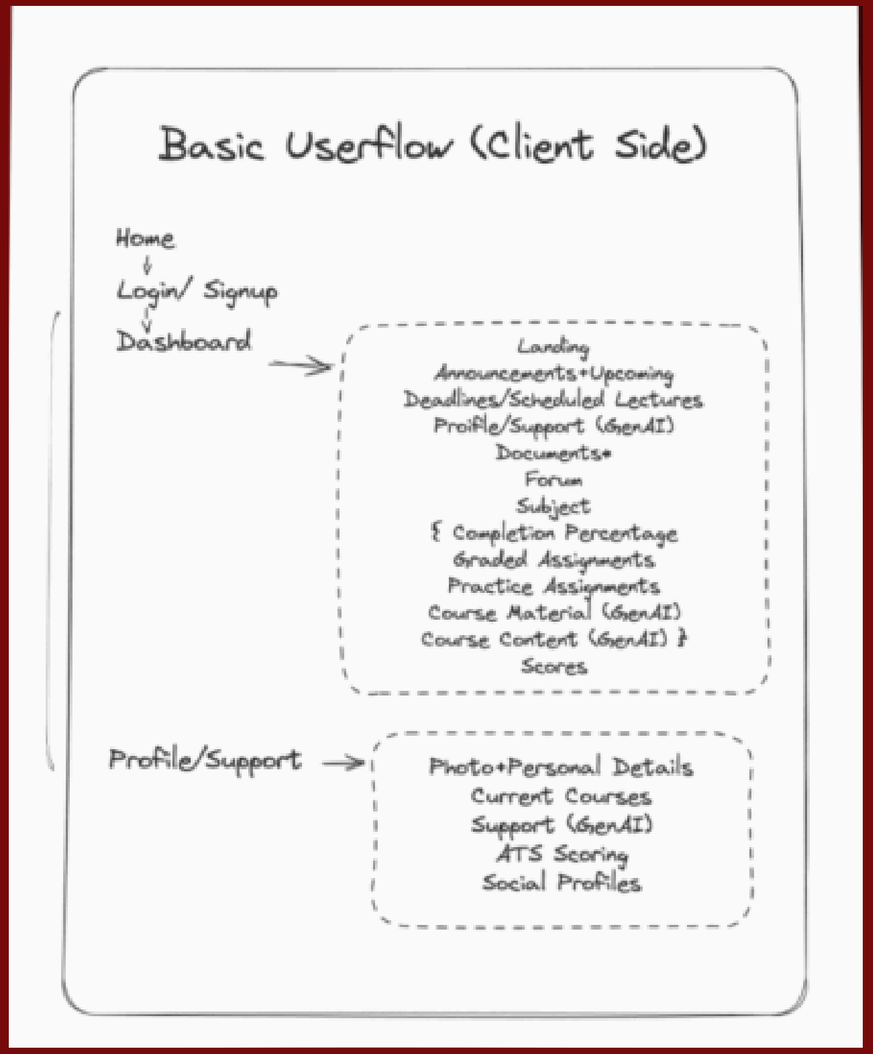
D Submits the code and receives immediate feedback on test cases passed/failed.

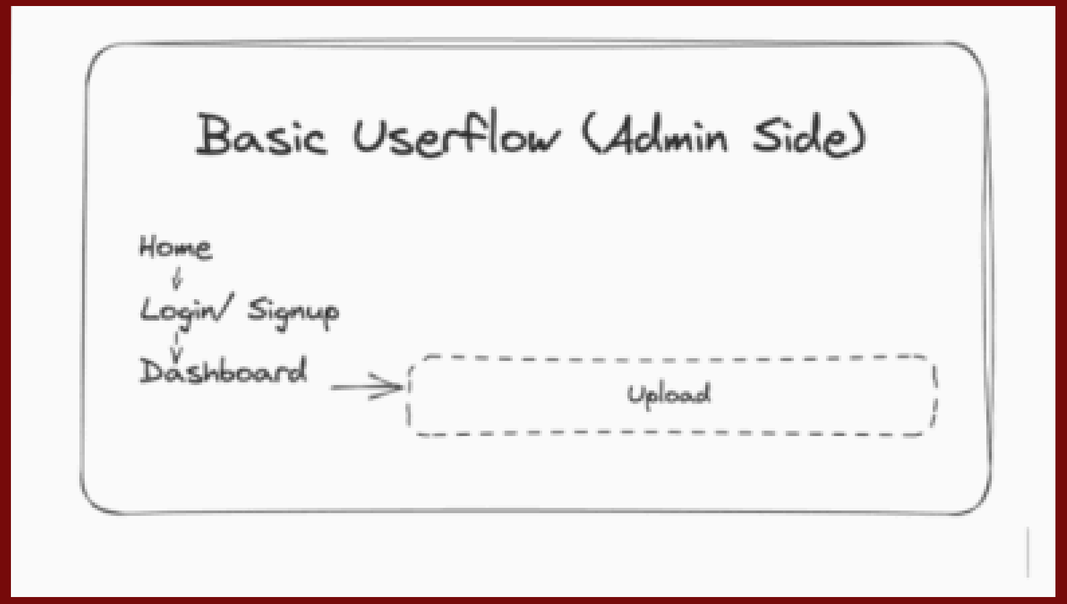
E GenAI suggests possible reasons for failed test cases and offers debugging tips.

F Learner attempts to debug and resubmit with the help of GenAI's suggestions.

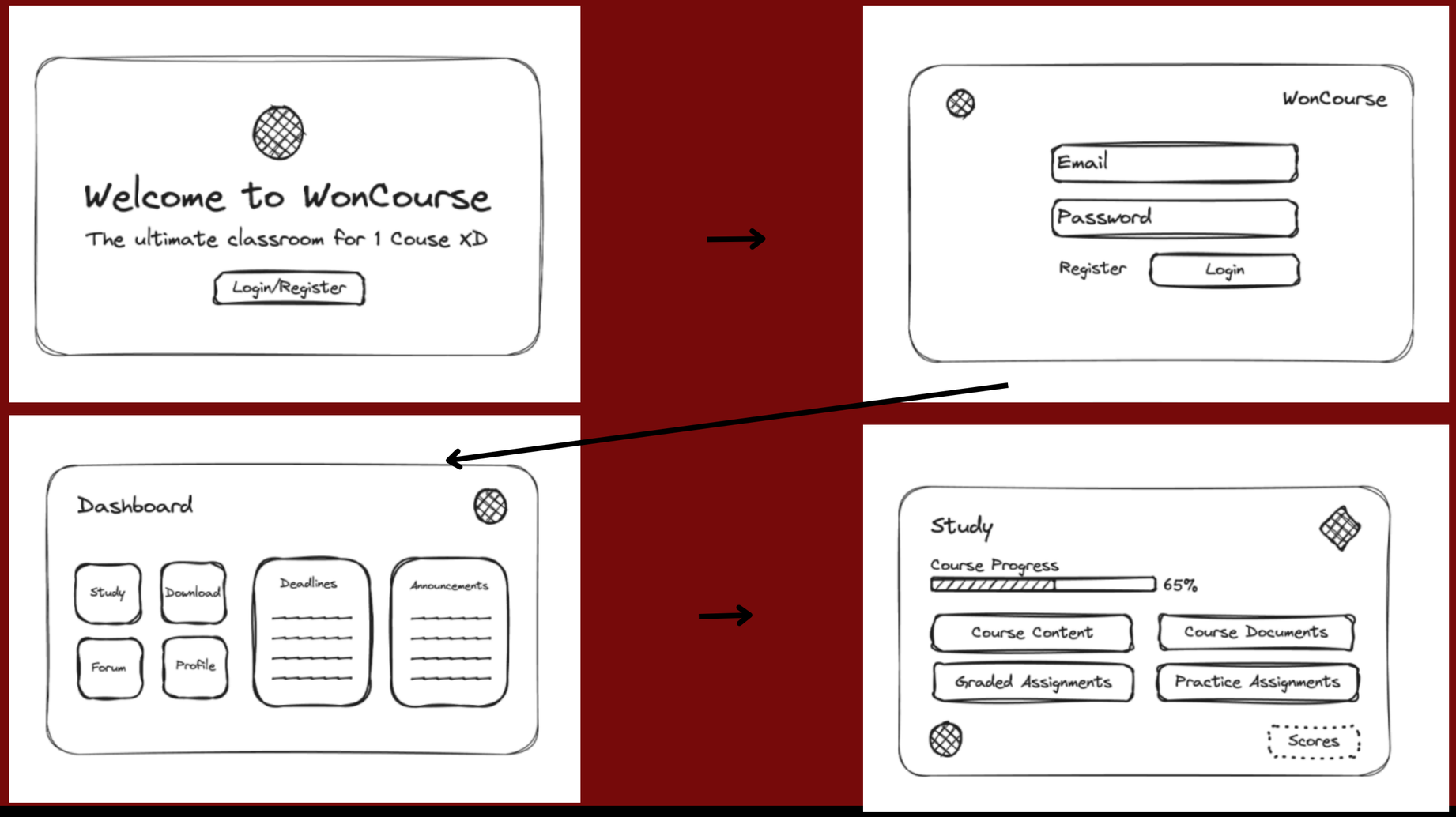
WIREFRAMES

USERFLOW FOR THE APP





USERFLOW FOR THE APP



PROJECT SCHEDULES AND DESIGN

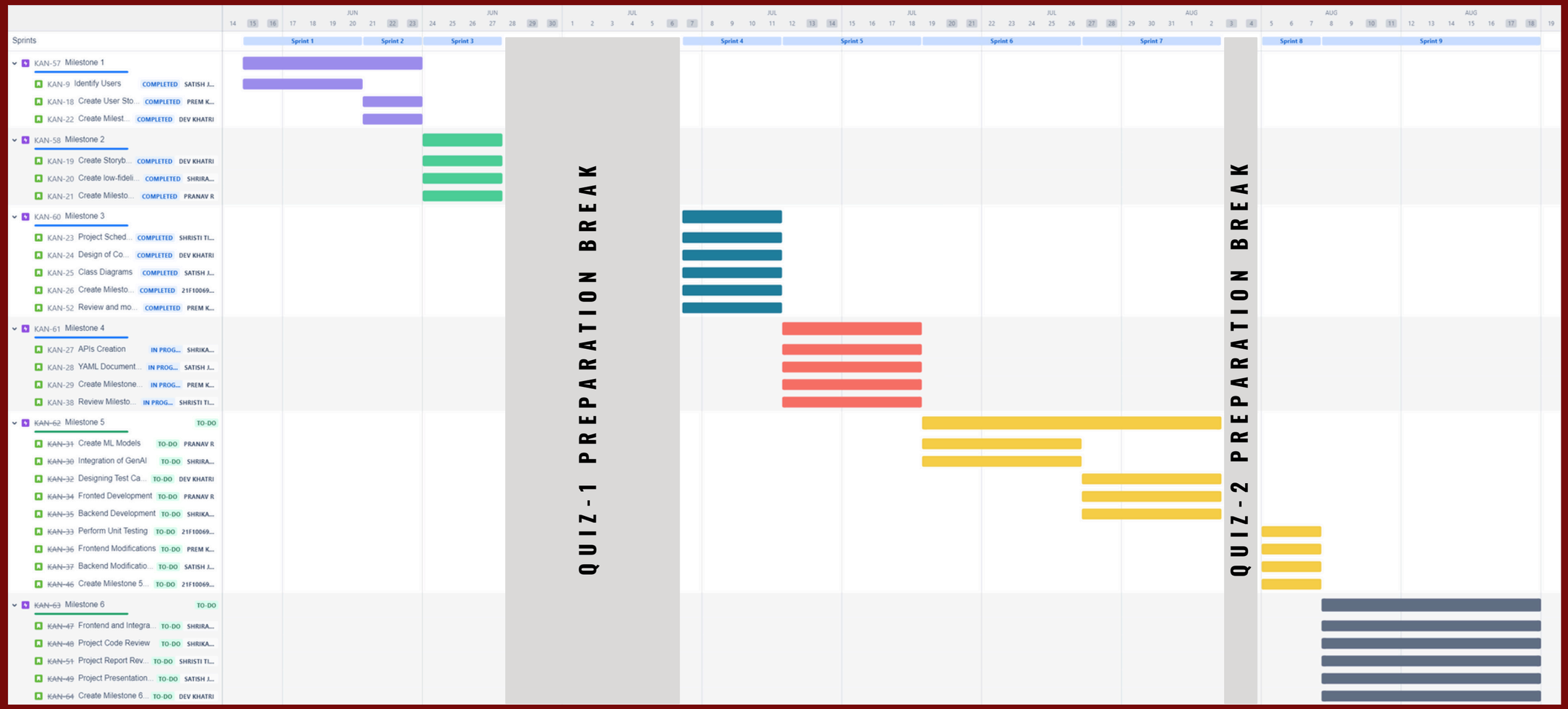
Sprint Schedule

Sprint No.	Topic/Task	Timeline	Assigned To
Sprint 1	Identify Users & Requirements	15 June, 2024 - 20 June, 2024	Satish
Sprint 2	Create User Stories	21 June, 2024 - 23 June, 2024	Dev and Prem
Sprint 3	Develop Storyboards & Wireframes	24 June, 2024 - 27 June, 2024	Shriraang, Dev and Pranav
Sprint 4	Schedule & Design Components	07 July, 2024 - 11 July, 2024	Srishti S, Satish, Dev, Prem and Shristi T
Sprint 5	Design API & Integrate GenAI	12 July, 2024 - 18 July, 2024	Shrikanth, Satish, Prem and Shristi T

Sprint Schedule

Sprint No.	Topic/Task	Timeline	Assigned To
Sprint 6	Implement API	19 July, 2024 - 26 July, 2024	Pranav and Shrikanth
Sprint 7	Design Test Cases	27 July, 2024 - 02 August, 2024	Dev, Pranav and Shrikanth
Sprint 8	Conduct Unit Testing	05 August, 2024 - 07 August, 2024	Prem, Srishti S and Satish
Sprint 9	Integrate Frontend & Backend	08 August, 2024 - 18 August, 2024	All
Sprint 10	Integration of GenAi	08 August, 2024 - 18 August, 2024	Shriraang and Dev

GANTT CHART


Sprints
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6
Sprint 7
Sprint 8
Sprint 9
KAN-57 Milestone 1
KAN-9 Identify Users COMPLETED SATISH J...
KAN-18 Create User Sto... COMPLETED PREM K...
KAN-22 Create Milest... COMPLETED DEV KHATRI
KAN-58 Milestone 2
KAN-19 Create Storyb... COMPLETED DEV KHATRI
KAN-20 Create low-fideli... COMPLETED SHRIRA...
KAN-21 Create Milesto... COMPLETED PRANAV R
KAN-60 Milestone 3
KAN-23 Project Sched... COMPLETED SHRISTI TI...
KAN-24 Design of Co... COMPLETED DEV KHATRI
KAN-25 Class Diagrams COMPLETED SATISH J...
KAN-26 Create Milesto... COMPLETED 21F10069...
KAN-52 Review and mo... COMPLETED PREM K...
KAN-61 Milestone 4
KAN-27 APIs Creation IN PROG... SHRIKA...
KAN-28 YAML Document... IN PROG... SATISH J...
KAN-29 Create Milestone... IN PROG... PREM K...
KAN-38 Review Milesto... IN PROG... SHRISTI TI...
KAN-62 Milestone 5 TO-DO
KAN-31 Create ML Models TO-DO PRANAV R
KAN-30 Integration of GenAI TO-DO SHRIRA...
KAN-32 Designing Test Ca... TO-DO DEV KHATRI
KAN-34 Fronted Development TO-DO PRANAV R
KAN-35 Backend Development TO-DO SHRIKA...
KAN-33 Perform Unit Testing TO-DO 21F10069...
KAN-36 Frontend Modifications TO-DO PREM K...
KAN-37 Backend Modificatio... TO-DO SATISH J...
KAN-46 Create Milestone 5... TO-DO 21F10069...
KAN-63 Milestone 6 TO-DO
KAN-47 Frontend and Integra... TO-DO SHRIRA...
KAN-48 Project Code Review TO-DO SHRIKA...
KAN-51 Project Report Rev... TO-DO SHRISTI TI...
KAN-49 Project Presentation... TO-DO SATISH J...
KAN-64 Create Milestone 6... TO-DO DEV KHATRI
QUIZ-1 PREPARATION BREAK
QUIZ-2 PREPARATION BREAK

API ROUTES

API ROUTES

POST	/	Home API
POST	/register	Register a new user
POST	/login	Login user
GET	/user	Get users
DELETE	/user/{user_id}	Delete user
GET	/study	Get study content
PUT	/study	Edit course

API ROUTES

GET	/study/lectures	Get lectures
POST	/study/lectures	Add lecture
DELETE	/study/lectures/{lecture_id}	Delete lecture
GET	/dashboard/admin	Admin dashboard
POST	/execute	Execute code
POST	/logout	Logout
POST	/videochat	Video Chat
POST	/codechat	Code Chat
POST	/supportchat	Support Chat

IMPLEMENTATION DETAILS

Implementation Details of the Project

Backend Implementation:

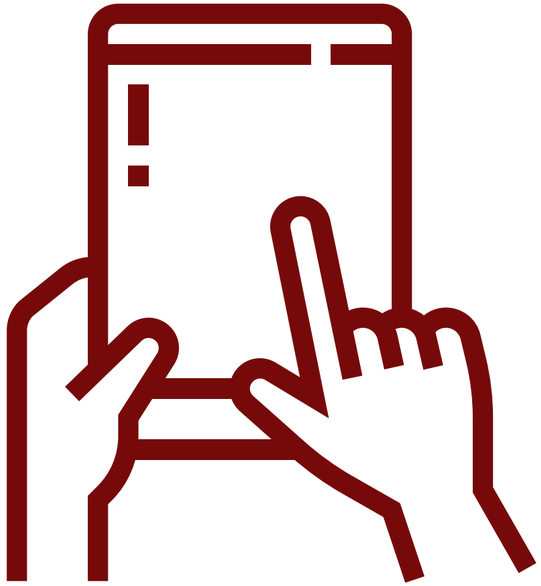
- Developed a RESTful API to manage user authentication, course content, and GenAI interactions.
- Implemented GenAI features such as feedback on programming assignments, study plan generation, and personalized support.

Frontend Implementation:

- Built a user-friendly dashboard that integrates GenAI functionalities for enhanced user interaction.
- Ensured smooth navigation between various components like course content, assignments, and user profiles.

Database Schema:

- Designed a comprehensive database schema to manage users, courses, assignments, and support requests.
- Incorporated relationships between tables to support the GenAI-driven features.



TECHNOLOGY USED

Tools and Technologies Used

1. Technologies for the backend:

- Flask, Flask Restful (For creating API endpoints), Flask SQLAlchemy, Pytest (Only for testing), Swagger Editor (Only for API documentation), Thunderclient (For checking API endpoints manually)

2. Technologies for the frontend:

- Vue 3 CLI, Vue Router, Vuex Store, JavaScript, Bootstrap (for aesthetics and styling), HTML and CSS, ESlint to help with debugging the frontend, Axios and Fetch API for requests.

3. General technologies used:

- GitHub (for versioning, code management, tracking, reviewing, issues, etc.), Canva (for making reports), Jira (for project management), draw.io (app.diagrams.net/) (for creating UML diagrams)

SOFTWARE ENGINEERING

Project

Project By :

Dev Khatri (21f3001150)
Satish Jaiswal (21f2000142)
Prem Kumar (21f1000531)
Srishti Singh (21f1006972)
Shrikanth V (21f1002328)
Shrirang Sapate (21f1002870)
Pranav R (21f1004199)
Shristi Tiwari (21f2000589)

TEAM 27

Thank You!