



SOFTWARE ENGINEERING PROJECT

Presented by : Team 10

Indian Institute of Technology, Madras | 2024

TABLE OF CONTENT

- Team Members
- Problem Statement
- Type of Users
- User Stories
- Wireframes
- Project Schedule and Design
- API Routes
- Test Codes
- Technology Used



TEAM MEMBERS

- Hari Prapan
 - Pragya Singh
 - Ojasv Singhal
 - Anujj Prem
 - Mahak Thakre
 - Kavya Dwivedi
 - Santosh Kumar Verma
 - Bindu Yadav
- 

PROBLEM STATEMENT

Effective Integration of Generative AI Into Programming learning Environments

- Design and Implement a learning portal with Gen-AI Integrated capabilities for learners that empowers them to take control of their education and achieve their full potential.
- The portal contains learning videos, assignments, resources, practice and graded assignments, and programming quizzes.
- The learning environment provides opportunities for self-learning and enabling learners to learn at their own pace and schedule.

TYPES OF USERS

- 1** **Primary Users - Students/Learner** who can who enroll in courses and utilize the learning portal to access learning materials, complete assignments, take quizzes, and participate in discussions.
- 2** **Secondary Users - Course Instructor** manages learning experience. Create and manage courses, upload content like lectures and assignments. Respond to student's queries and communicate with students for general announcements and resolving other queries.
- 3** **Tertiary Users - Subject Matter Experts, EdTech Companies** and other higher authorities for exploration, feedback, and other support etc.

USER STORIES – STUDENT

- **I want : The portal to provide detailed feedback on my programming assignments, highlighting errors and suggesting improvements within 5 minutes of the deadline.**
- **I want : The LMS to generate concise summaries of lecture videos, summarizing key points and concepts in under 200 words.**
- **I want : The LMS to provide a detailed comparison of my coding assignment against the optimal solution, including a similarity score and areas for improvement within 5 minutes of the deadline.**
- **I want : Access to a course-specific chatbot within the portal that can provide real-time assistance with a response time of under 30 seconds.**
- **I want : Gen-AI to create a dynamic study plan based on my course load and deadlines.**

USER STORIES – STUDENT

- **I want : Gen-AI to provide instant feedback on my practice assignments.**
- **I want : The IDE in the SEEK portal to suggest code completions, improvements and syntax corrections.**
- **I want : Gen-AI to provide hints for fixing issues related to failed hidden test cases in my programming assignments.**
- **I want : Gen-AI to analyze my previous wrong answers and suggest my weak topics.**
- **I want : Gen-AI to enhance lecture transcripts in my native language.**

USER STORIES – INSTRUCTOR

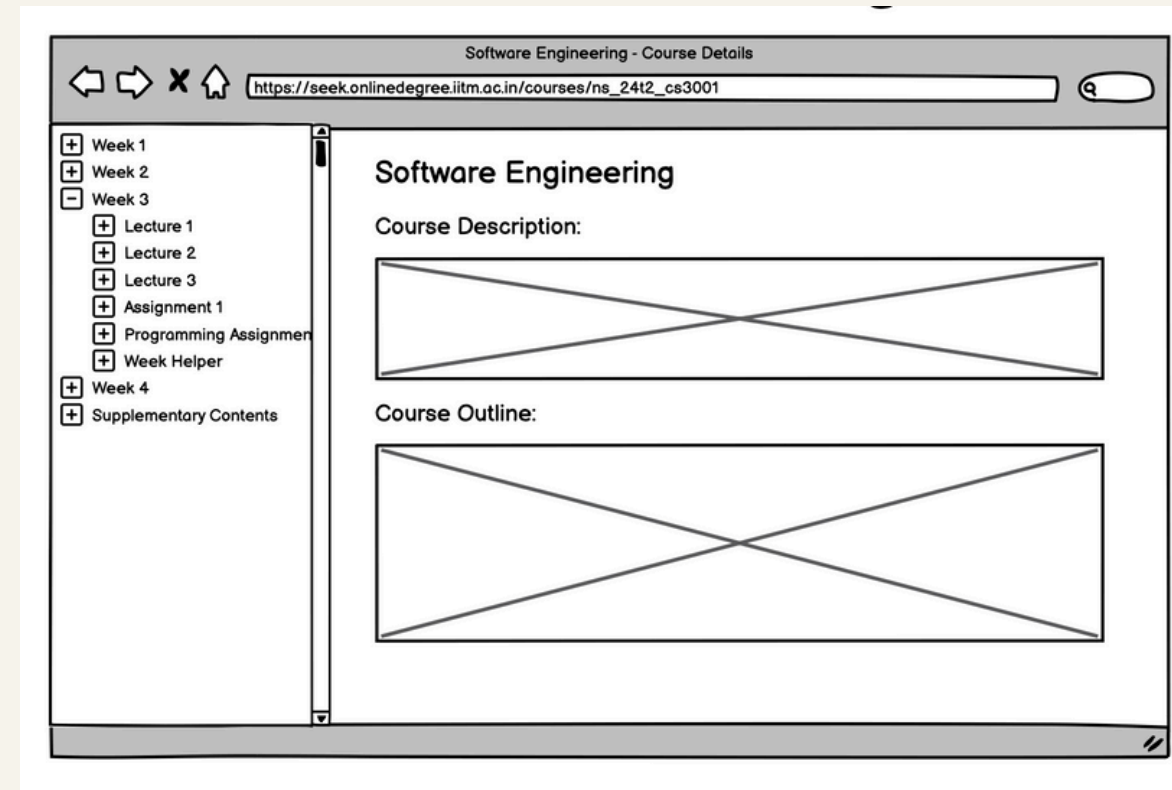
- **I want : Gen-AI to only be a tool for small queries and not spoon-feed the answers to students.**
- **I want : The portal to auto-grade programming assignments with 90% accuracy and provide comprehensive reports to students within 10 minutes of the deadline.**
- **I want : Integrating a chatbot that uses NLP to answer students' questions in real-time.**
- **I want : The portal to include a course-specific chatbot that can accurately answer at least 80% of student queries within 30 seconds.**

USER STORIES – LEARNER

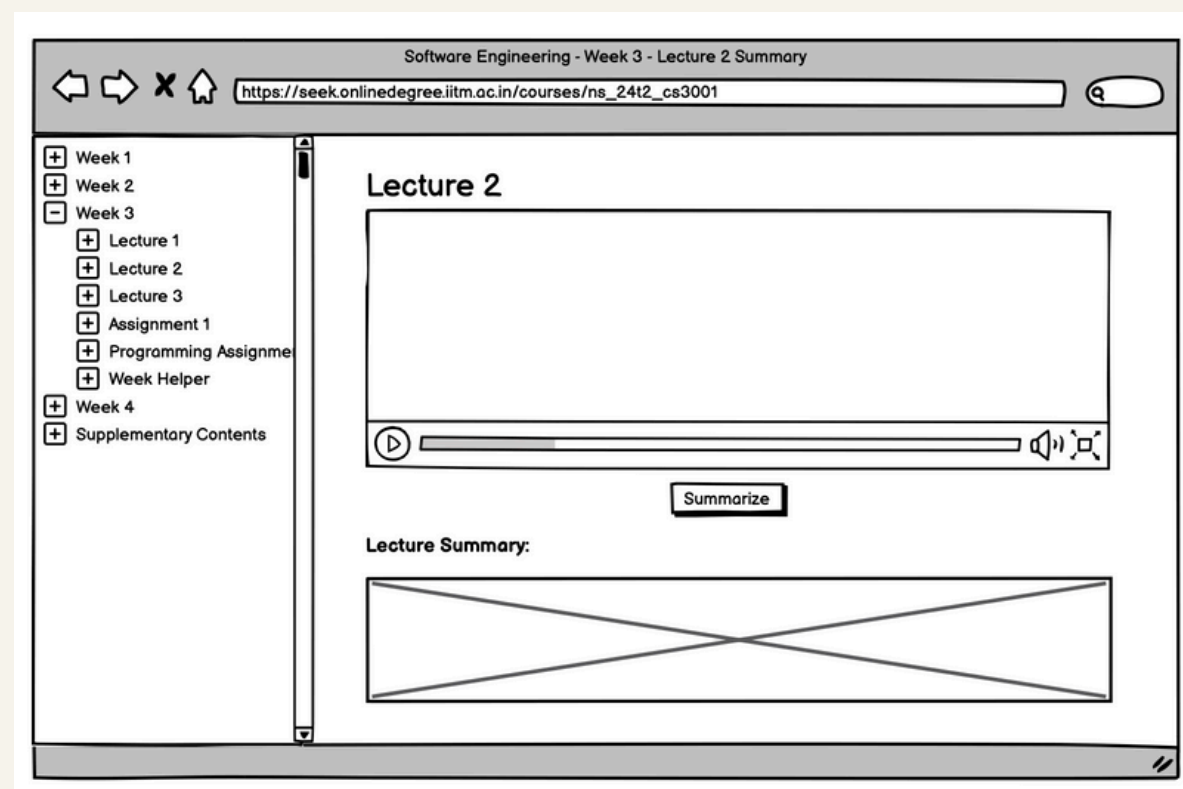
- **I want : To use a Gen-AI tool for note-taking and highlighting key points.**
- **I want : AI-moderated study groups where students can collaborate on assignments and quiz preparations.**
- **I want : Using AI to create gamified elements such as badges, leaderboards, and personalized challenges.**
- **I want: Using AI to automate the grading of assignments and quizzes, providing instant feedback.**

WIREFRAMES

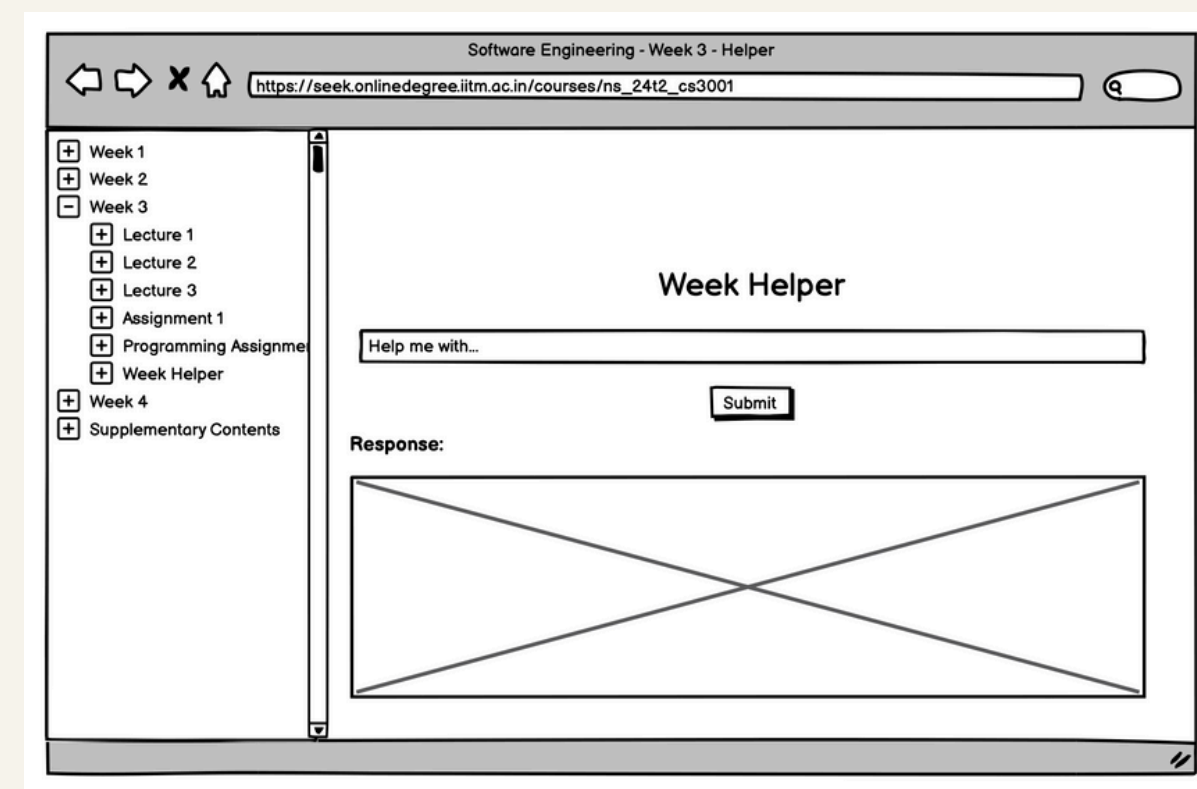
COURSE HOME PAGE



LECTURE SUMMARY



WEEK HELPER



ASSIGNMENT

Software Engineering - Week 3 - Assignment 1

https://seek.onlinedegree.iitm.ac.in/courses/ns_24t2_cs3001

- + Week 1
- + Week 2
- Week 3
 - + Lecture 1
 - + Lecture 2
 - + Lecture 3
 - + Assignment 1
 - + Programming Assignment
 - + Week Helper
- + Week 4
- + Supplementary Contents

Assignment 1 (Graded)

Question 1

Question 2

Question 3

Submit

PROGRAMMING ASSIGNMENT

Software Engineering - Week 3 - Programming Assignment 1

https://seek.onlinedegree.iitm.ac.in/courses/ns_24t2_cs3001

- + Week 1
- + Week 2
- Week 3
 - + Lecture 1
 - + Lecture 2
 - + Lecture 3
 - + Assignment 1
 - + Programming Assignment
 - + Week Helper
- + Week 4
- + Supplementary Contents

Programming Assignment

Write a program to...

Print ("Hello")

Submit

Code Report:

Improvements:

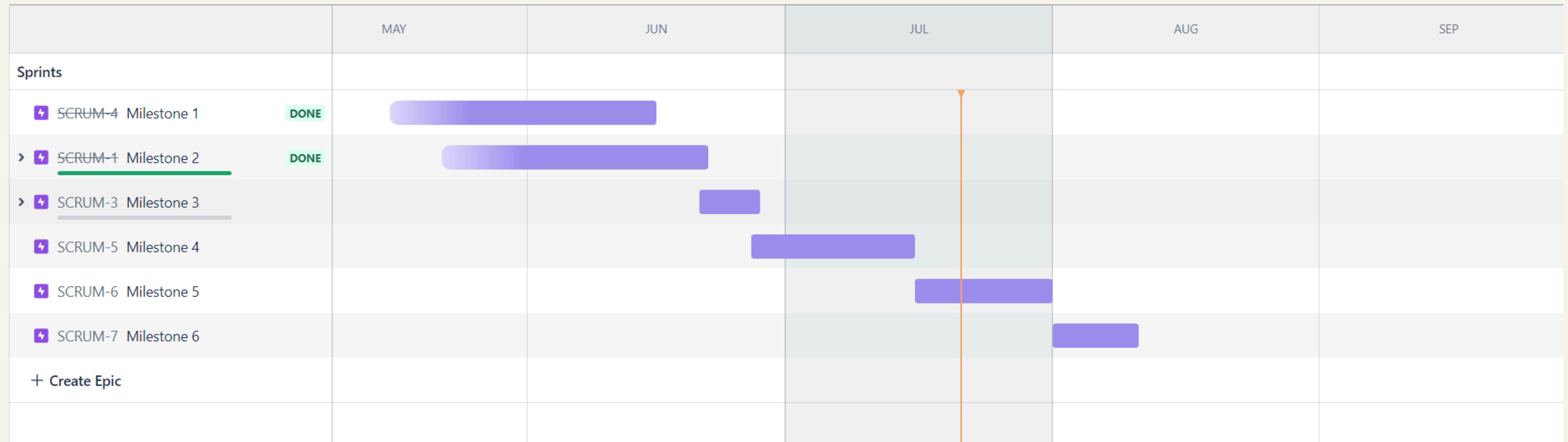
Improved Code:

PROJECT SCHEDULES AND DESIGN

S. No.	Milestone	Sub-Tasks	Sprint	Assigned To
1	Identify User Requirements	Identify users	1	Hari, Ojasv
		User stories	1	Kavya, Santosh
		Report	1	Mahak, Bindu
2	User Interfaces	Storyboard	2	Pragya, Kavya
		Wireframes	2	Anujj
		Report	2	Mahak, Bindu
3	Scheduling and Design	Project Scheduling	3	Hari
		Components design	3	Santosh
		Software Design	3	Ojasv
		Minutes of meetings	3	Kavya
		Report	3	All

S. No.	Milestone	Sub-Tasks	Sprint	Assigned To
4	API Integrations	API Design	4	Ojasv
		API Description	4	Anujj
		YAML File	4	All
5	Testing	Test Case Design	5	Pragya, Bindu, Hari, Anujj
		Unit Testing	5	Ojasv, Kavya, Santosh, Bindu
		Report	5	All
6	Final Submission	Frontend design	6	Pragya, Kavya, Mahak
		Backend Implementation	6	Hari, Ojasv
		Report	7	All
		Presentation	7	Kavya, Anujj, Santosh

GANTT CHART OF SCHEDULE



API ROUTES

POST**/lecture** Create a new lecture**PUT****/lecture/{id}** Update a lecture**DELETE****/lecture/{id}** Delete a lecture**GET****/summary/{lectureId}** Get a summarized transcript of a YouTube lecture**GET****/info** Get all weeks and their associated data**POST****/helper** Get AI assistance for a query within a specific week's context**POST****/week** Create a new week

PUT `/week/{id}` Update a week



DELETE `/week/{id}` Delete a week



POST `/assignment` Create a new assignment



PUT `/assignment/{id}` Update an assignment



DELETE `/assignment/{id}` Delete an assignment



POST `/question` Create a new question



PUT `/question/{id}` Update a question



DELETE `/question/{id}` Delete a question



GET

/progassg Get all programming assignments

POST

/progassg Create a new programming assignment

PUT

/progassg/{id} Update a programming assignment

DELETE

/progassg/{id} Delete a programming assignment

POST

/progassg_suggestions Get code improvement suggestions and explanations for a programming assignment

POST

/submit_assg Submit answers to an assignment and receive AI-powered feedback and grading

TESTING

1. `/info` Endpoints

Test	Description	Example
1.1 test_get_info_successful	Ensures that the <code>/info</code> endpoint returns a 200 status code and contains the expected "weeks" key in the response	If the request is made to <code>/info</code> , the API should return a 200 status code and include a "weeks" key in the response data.
1.2 test_get_info_invalid_endpoint	Verifies that requesting an invalid endpoint returns a non-200 status code.	If the request is made to <code>/info_invalid</code> , the API should return a status code indicating failure (e.g., 404).

2. `/helper` Endpoints

Test	Description	Example
2.1 test_helper_successful	Checks if the <code>/helper</code> endpoint correctly processes a valid query and weekId, returning a 200 status code and a markdown response.	If the request includes a valid query and weekId, the API should return a 200 status code and markdown content.
2.2 test_helper_invalid_week	Ensures that the <code>/helper</code> endpoint returns a 404 status code when an invalid weekId is provided.	If the request includes an invalid weekId, the API should return a 404 status code indicating week not found.
2.3 test_helper_invalid_request	Verifies that the <code>/helper</code> endpoint returns a 400 status code when required parameters are missing.	If the request is missing required parameters, the API should return a 400 status code indicating a bad request.

3. `/summary/{lectureId}` Endpoints

Test	Description	Example
3.1 test_get_summary_successful	Checks if the <code>/summary</code> endpoint returns a 200 status code and a markdown summary for a valid lecture ID.	If the request includes a valid <code>lectureId</code> , the API should return a 200 status code and markdown summary content.
3.2 test_get_summary_invalid_lecture	Ensures that the <code>/summary</code> endpoint returns a 404 status code for a non-existent lecture ID.	If the request includes a non-existent <code>lectureId</code> , the API should return a 404 status code indicating lecture not found.

4. `/week` Endpoints

Test	Description	Example
4.1 test_create_week_successful	Verifies that the POST <code>/week</code> endpoint correctly creates a new week and returns a 201 status code with the expected data.	If a valid week creation request is made, the API should return a 201 status code and the created week data.
4.2 test_create_week_invalid_request	Ensures that the POST <code>/week</code> endpoint returns a 400 status code when required parameters are missing.	If the week creation request is missing required parameters, the API should return a 400 status code indicating a bad request.
4.3 test_update_week_successful	Checks if the PUT <code>/week/{id}</code> endpoint correctly updates an existing week and returns a 200 status code.	If a valid week update request is made for an existing week, the API should return a 200 status code.
4.4 test_update_week_invalid_id	Verifies that the PUT <code>/week/{id}</code> endpoint returns a 404 status code when trying to update a non-existent week.	If the week update request is made for a non-existent week, the API should return a 404 status code indicating week not found.
4.5 test_delete_week_successful	Ensures that the DELETE <code>/week/{id}</code> endpoint correctly deletes an existing week and returns a 200 status code.	If a valid delete request is made for an existing week, the API should return a 200 status code.
4.6 test_delete_week_invalid_id	Checks if the DELETE <code>/week/{id}</code> endpoint returns a 404 status code when trying to delete a non-existent week.	If the delete request is made for a non-existent week, the API should return a 404 status code indicating week not found.

5. /lecture Endpoints

Test	Description	Example
5.1 test_create_lecture_successful	Verifies that the POST /lecture endpoint correctly creates a new lecture and returns a 201 status code with the expected data.	If a valid lecture creation request is made, the API should return a 201 status code and the created lecture data.
5.2 test_create_lecture_invalid_request	Ensures that the POST /lecture endpoint returns a 400 status code when required parameters are missing.	If the lecture creation request is missing required parameters, the API should return a 400 status code indicating a bad request.
5.3 test_update_lecture_successful	Checks if the PUT /lecture/{id} endpoint correctly updates an existing lecture and returns a 200 status code.	If a valid lecture update request is made for an existing lecture, the API should return a 200 status code.
5.4 test_update_lecture_invalid_id	Verifies that the PUT /lecture/{id} endpoint returns a 404 status code when trying to update a non-existent lecture.	If the lecture update request is made for a non-existent lecture, the API should return a 404 status code indicating lecture not found.
5.5 test_delete_lecture_successful	Ensures that the DELETE /lecture/{id} endpoint correctly deletes an existing lecture and returns a 200 status code.	If a valid delete request is made for an existing lecture, the API should return a 200 status code.
5.6 test_delete_lecture_invalid_id	Checks if the DELETE /lecture/{id} endpoint returns a 404 status code when trying to delete a non-existent lecture.	If the delete request is made for a non-existent lecture, the API should return a 404 status code indicating lecture not found.

6. /assignment Endpoints

Test	Description	Example
6.1 test_create_assignment_successful	Verifies that the POST /assignment endpoint correctly creates a new assignment and returns a 201 status code with the expected data.	If a valid assignment creation request is made, the API should return a 201 status code and the created assignment data.
6.2 test_create_assignment_invalid_request	Ensures that the POST /assignment endpoint returns a 400 status code when required parameters are missing.	If the assignment creation request is missing required parameters, the API should return a 400 status code indicating a bad request.
6.3 test_update_assignment_successful	Checks if the PUT /assignment/{id} endpoint correctly updates an existing assignment and returns a 200 status code.	If a valid assignment update request is made for an existing assignment, the API should return a 200 status code.
6.4 test_update_assignment_invalid_id	Verifies that the PUT /assignment/{id} endpoint returns a 404 status code when trying to update a non-existent assignment.	If the assignment update request is made for a non-existent assignment, the API should return a 404 status code indicating assignment not found.
6.5 test_delete_assignment_successful	Ensures that the DELETE /assignment/{id} endpoint correctly deletes an existing assignment and returns a 200 status code.	If a valid delete request is made for an existing assignment, the API should return a 200 status code.
6.6 test_delete_assignment_invalid_id	Checks if the DELETE /assignment/{id} endpoint returns a 404 status code when trying to delete a non-existent assignment.	If the delete request is made for a non-existent assignment, the API should return a 404 status code indicating assignment not found.

TECHNOLOGY STACK

TECHNOLOGY STACK

List of Technologies used:

- **Vue.js**
- **Flask**
- **SQLite**
- **npm**
- **Gemini-1.5-Flash**

Indian Institute of Technology, Madras | 2024

THANK YOU

Presented By : Team 10