SE - SEP 2023 Team - 3

A Presentation by

Prashant Kumar Mishra
Saurabh Patil
Sandip Biswas
Aditi Krishana
Anushka Krishna
Mukesh Kumar Singh

Table Of Content

User Story

Storyboard

Wireframe

Tracking & Communication

Code Structure

Application Demo

Project timeline :
Sep-2023 - Dec-2023

Milestone 1: User Story: Primary User

As a student, **I want** to input my academic interests, learning goals, schedules, and commitments, **so that** the system considers my current preferences and constraints to recommend a personalized learning path	**As a** student, **I want** to provide feedback on the recommended learning paths, **so that** the system can utilize my sentiments and ratings in the refinement of the recommendation engine.	**As a** student, **I want** to explore different popular learning paths taken by my peers who are similar to me, **so that** I can gain insights, broaden my academic horizons, and make informed decisions about my own learning journey
As a student, **I want** the system to consider my past performance in previous courses, so that the recommendations align with my academic strengths and weaknesses.	**As a** student, **I want** to prevent other students from accessing my personal data, **so that** I have confidence in the confidentiality and integrity of my data.	**As a** student, **I want** to prevent other students from accessing my personal data, **so that** I have confidence in the confidentiality and integrity of my data.ta.
As a student, **I want** to share feedback on each course I have taken or finished, **so that** the system can consider my sentiments and ratings to enhance the recommendation engine.	**As a** student, **I want** to prevent other students from accessing my personal data, **so that** I have confidence in the confidentiality and integrity of my data.	

Milestone 1: User Story: Secondary Users

As an admin,
I want to upload enrollment data, such as student's learning profile, including their past academic performance, interests, and goals from previous terms,
so that the system can use it to generate learning path recommendations.

As an admin,
I want to efficiently manage courses, including their prerequisites, corequisites, and required minimum hours of study, by performing actions such as inserting, updating, and deleting,
so that learning path recommendations align with academic requirements and regulations.

As an admin,
I want to provide feedback or ratings on recommended learning paths,
so that the recommendation engine can be reinforced and fine-tuned for improved suggestions

As an admin,
I want to provide feedback or ratings on recommended learning paths,
so that the recommendation engine can be reinforced and fine-tuned for improved suggestions

As an admin,
I want to provide feedback or ratings on recommended learning paths,
so that the recommendation engine can be reinforced and fine-tuned for improved suggestions

As an admin,
I want to provide feedback or ratings on recommended learning paths,
so that the recommendation engine can be reinforced and fine-tuned for improved suggestions

As an admin,
I want to view or download a summary report of the uploaded enrollment data,
so that I can verify its accuracy.

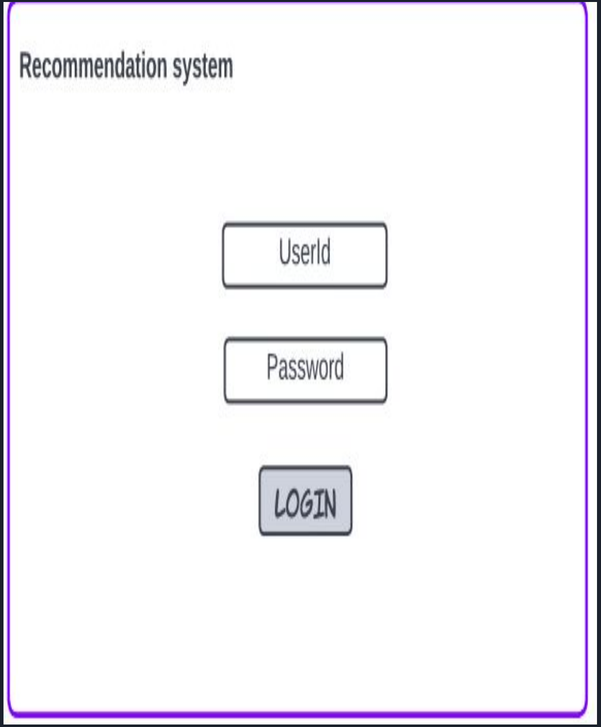
Milestone 1: User Story: Tertiary Users

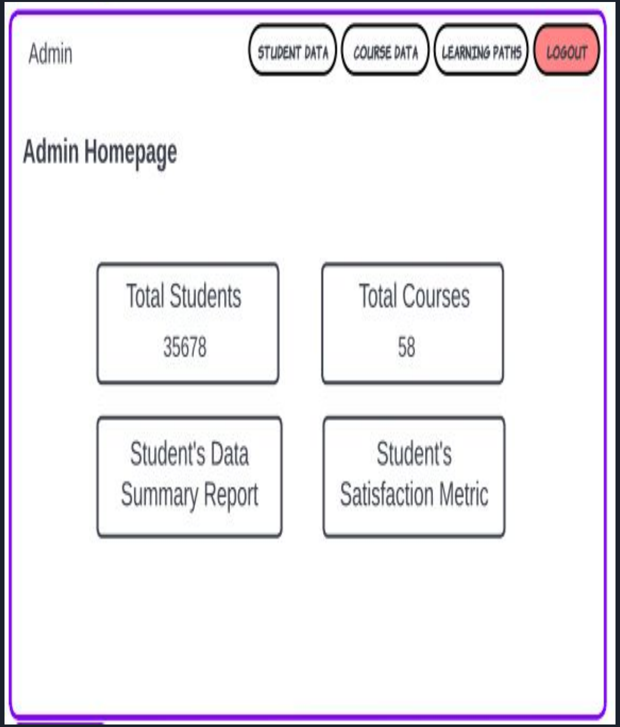
As a course professor or instructor,
I want to get alerted if the rating of my course is less than 3,
so that I can evaluate the effectiveness of my teaching methods and make improvement

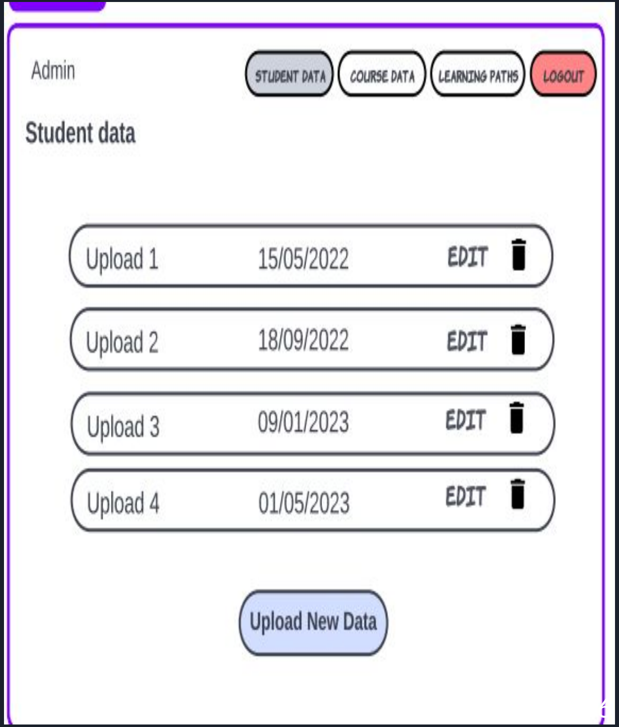
As a member of the student affairs or academic team,
I want to access data on student engagement with recommended learning paths,
so that I can verify the efficiency and effectiveness of the recommendation engine

Milestone 2: Wireframe

Wireframe Link



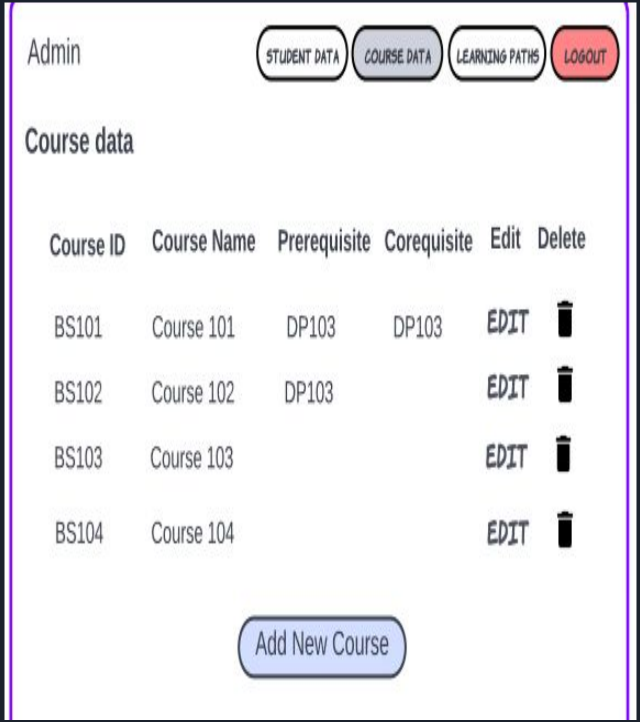




Upload 1	15/05/2022	EDIT
Upload 2	18/09/2022	EDIT
Upload 3	09/01/2023	EDIT
Upload 4	01/05/2023	EDIT

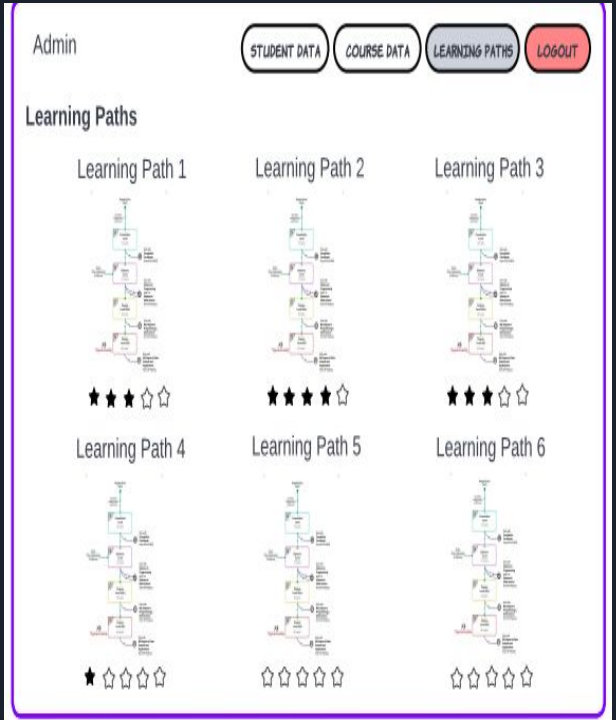
Milestone 2: Wireframe

Wireframe Link



Course data

Course ID	Course Name	Prerequisite	Corequisite	Edit	Delete
BS101	Course 101	DP103	DP103	EDIT	🗑
BS102	Course 102	DP103		EDIT	🗑
BS103	Course 103			EDIT	🗑
BS104	Course 104			EDIT	🗑



Learning Paths

Learning Path 1 ★★★☆☆

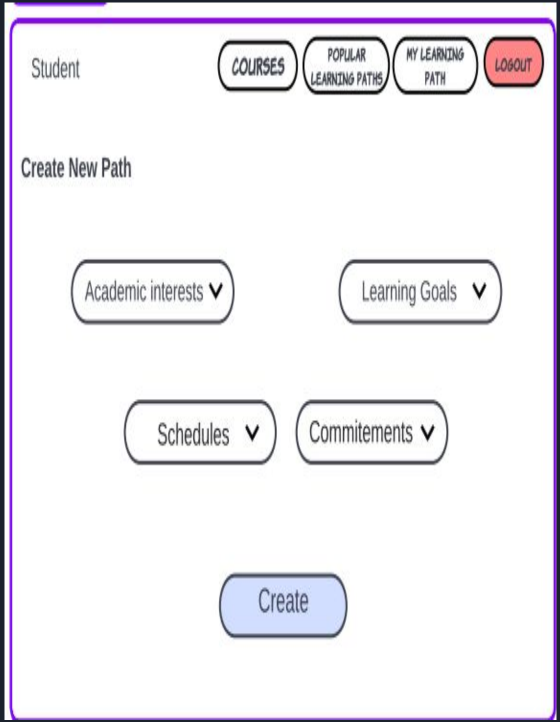
Learning Path 2 ★★★★☆

Learning Path 3 ★★★☆☆

Learning Path 4 ★☆☆☆☆

Learning Path 5 ☆☆☆☆☆

Learning Path 6 ☆☆☆☆☆



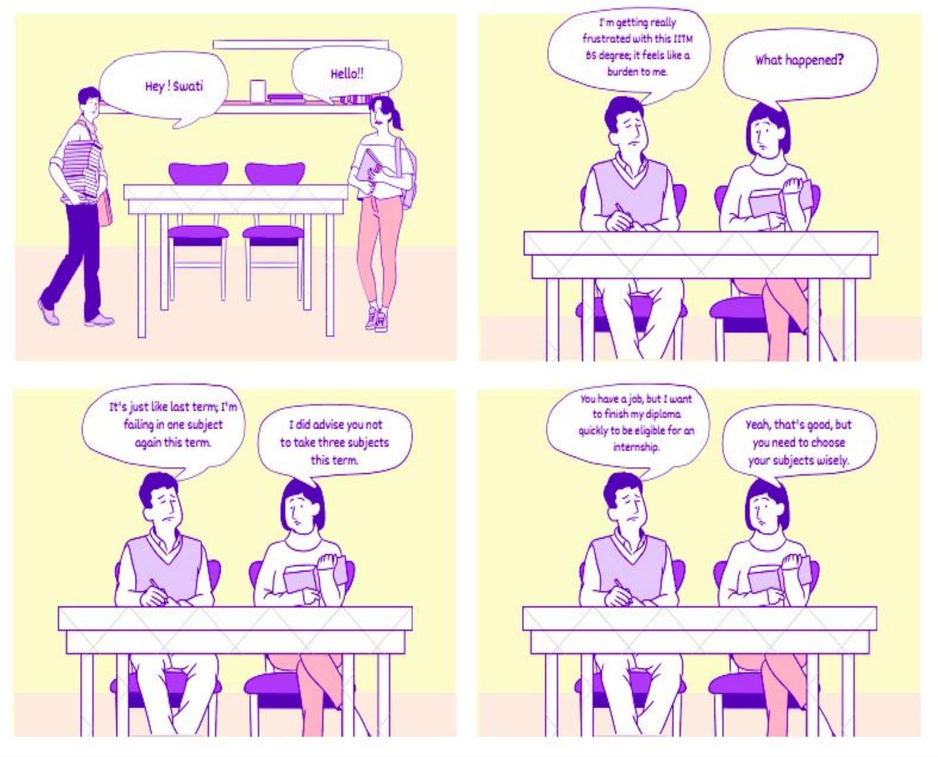
Create New Path

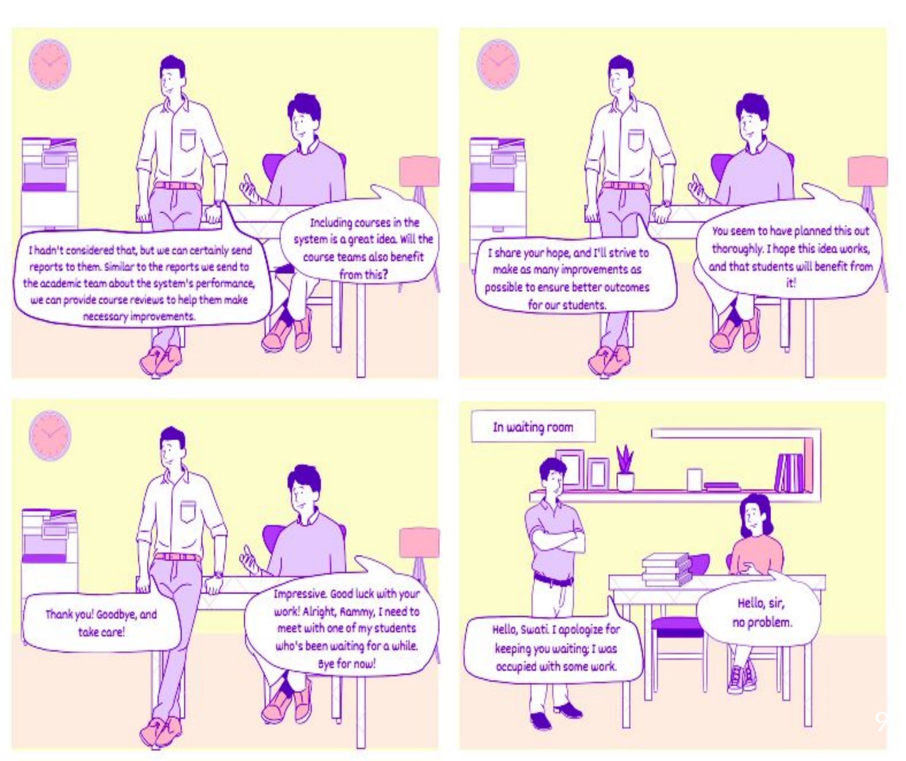
Milestone 2: StoryBoard Animation

Animation Link

Milestone 2: StoryBoard Video & PPT

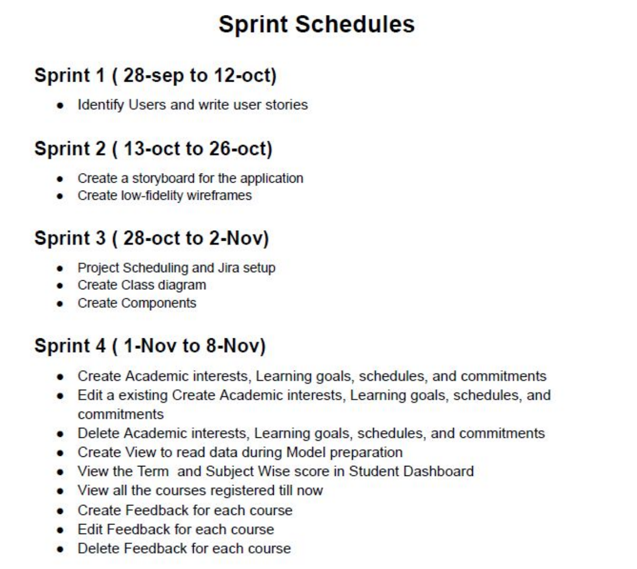
Video **PPT**


Hey ! Swati
Hello!!
I'm getting really frustrated with this IITM BS degree; it feels like a burden to me.
What happened?
It's just like last term; I'm failing in one subject again this term.
I did advise you not to take three subjects this term.
You have a job, but I want to finish my diploma quickly to be eligible for an internship.
Yeah, that's good, but you need to choose your subjects wisely.


I hadn't considered that, but we can certainly send reports to them. Similar to the reports we send to the academic team about the system's performance, we can provide course reviews to help them make necessary improvements.
Including courses in the system is a great idea. Will the course teams also benefit from this?
I share your hope, and I'll strive to make as many improvements as possible to ensure better outcomes for our students.
You seem to have planned this out thoroughly. I hope this idea works, and that students will benefit from it!
Thank you! Goodbye, and take care!
Impressive. Good luck with your work! Alright, Rammy, I need to meet with one of my students who's been waiting for a while. Bye for now!
In waiting room
Hello, Swati. I apologize for keeping you waiting; I was occupied with some work.
Hello, sir, no problem.

Milestone 3: Project Schedule

We planned to complete the entire project in 8 Sprints, spanning across the Months of September, October, November and December 2023, based on the user stories created in the previous milestones



Sprint Schedules

Sprint 1 (28-sep to 12-oct)

- Identify Users and write user stories

Sprint 2 (13-oct to 26-oct)

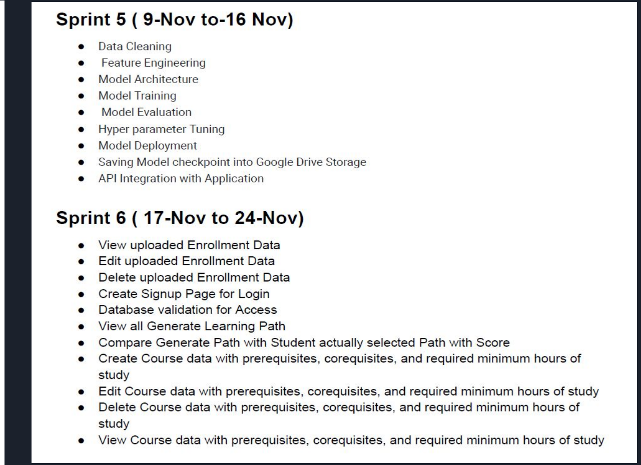
- Create a storyboard for the application
- Create low-fidelity wireframes

Sprint 3 (28-oct to 2-Nov)

- Project Scheduling and Jira setup
- Create Class diagram
- Create Components

Sprint 4 (1-Nov to 8-Nov)

- Create Academic interests, Learning goals, schedules, and commitments
- Edit a existing Create Academic interests, Learning goals, schedules, and commitments
- Delete Academic interests, Learning goals, schedules, and commitments
- Create View to read data during Model preparation
- View the Term and Subject Wise score in Student Dashboard
- View all the courses registered till now
- Create Feedback for each course
- Edit Feedback for each course
- Delete Feedback for each course

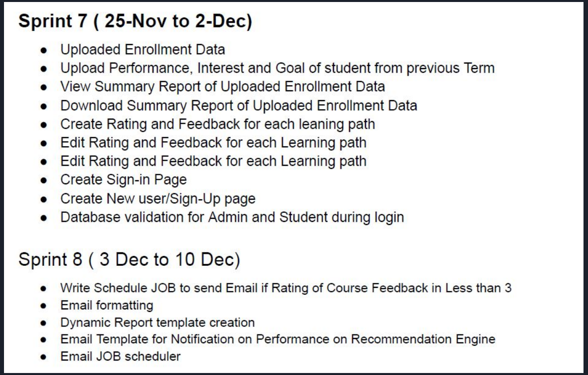


Sprint 5 (9-Nov to-16 Nov)

- Data Cleaning
- Feature Engineering
- Model Architecture
- Model Training
- Model Evaluation
- Hyper parameter Tuning
- Model Deployment
- Saving Model checkpoint into Google Drive Storage
- API Integration with Application

Sprint 6 (17-Nov to 24-Nov)

- View uploaded Enrollment Data
- Edit uploaded Enrollment Data
- Delete uploaded Enrollment Data
- Create Signup Page for Login
- Database validation for Access
- View all Generate Learning Path
- Compare Generate Path with Student actually selected Path with Score
- Create Course data with prerequisites, corequisites, and required minimum hours of study
- Edit Course data with prerequisites, corequisites, and required minimum hours of study
- Delete Course data with prerequisites, corequisites, and required minimum hours of study
- View Course data with prerequisites, corequisites, and required minimum hours of study



Sprint 7 (25-Nov to 2-Dec)

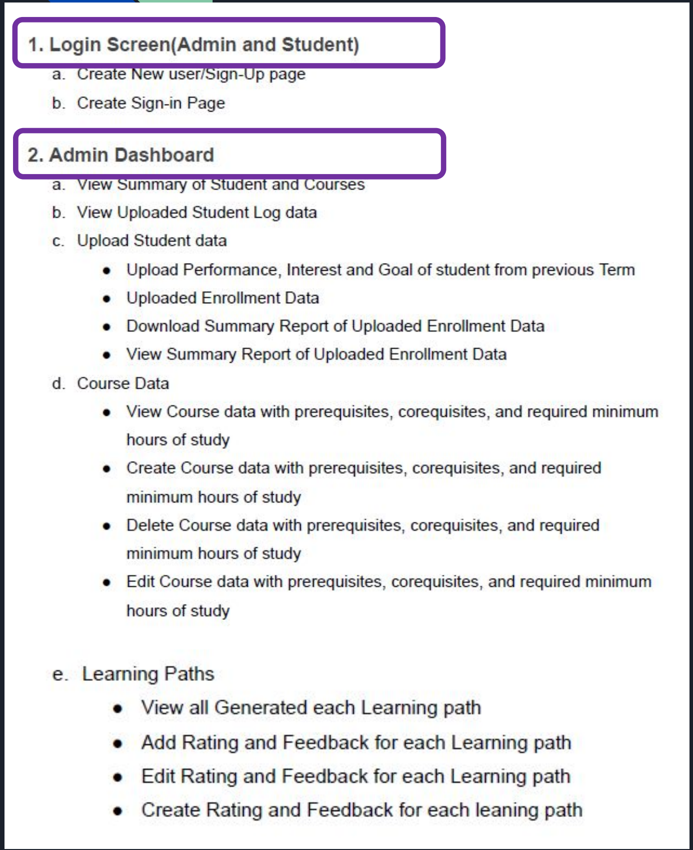
- Uploaded Enrollment Data
- Upload Performance, Interest and Goal of student from previous Term
- View Summary Report of Uploaded Enrollment Data
- Download Summary Report of Uploaded Enrollment Data
- Create Rating and Feedback for each leaning path
- Edit Rating and Feedback for each Learning path
- Edit Rating and Feedback for each Learning path
- Create Sign-in Page
- Create New user/Sign-Up page
- Database validation for Admin and Student during login

Sprint 8 (3 Dec to 10 Dec)

- Write Schedule JOB to send Email if Rating of Course Feedback in Less than 3
- Email formatting
- Dynamic Report template creation
- Email Template for Notification on Performance on Recommendation Engine
- Email JOB scheduler

Milestone 3: Design of Components

Description of components of our system based on the user stories created in the previous milestones



1. Login Screen(Admin and Student)

a. Create New user/Sign-Up page

b. Create Sign-in Page

2. Admin Dashboard

a. View Summary of Student and Courses

b. View Uploaded Student Log data

c. Upload Student data

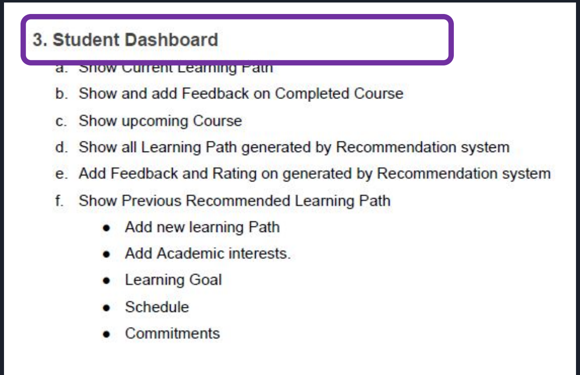
- Upload Performance, Interest and Goal of student from previous Term
- Uploaded Enrollment Data
- Download Summary Report of Uploaded Enrollment Data
- View Summary Report of Uploaded Enrollment Data

d. Course Data

- View Course data with prerequisites, corequisites, and required minimum hours of study
- Create Course data with prerequisites, corequisites, and required minimum hours of study
- Delete Course data with prerequisites, corequisites, and required minimum hours of study
- Edit Course data with prerequisites, corequisites, and required minimum hours of study

e. Learning Paths

- View all Generated each Learning path
- Add Rating and Feedback for each Learning path
- Edit Rating and Feedback for each Learning path
- Create Rating and Feedback for each leaning path



3. Student Dashboard

a. Show Current Learning Path

b. Show and add Feedback on Completed Course

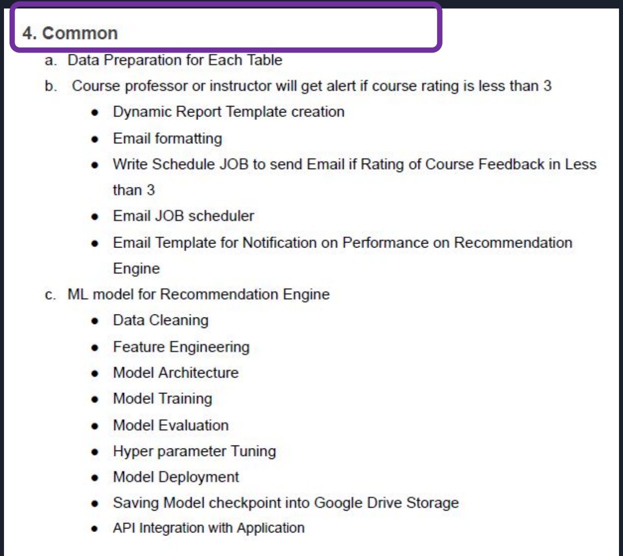
c. Show upcoming Course

d. Show all Learning Path generated by Recommendation system

e. Add Feedback and Rating on generated by Recommendation system

f. Show Previous Recommended Learning Path

- Add new learning Path
- Add Academic interests.
- Learning Goal
- Schedule
- Commitments



4. Common

a. Data Preparation for Each Table

b. Course professor or instructor will get alert if course rating is less than 3

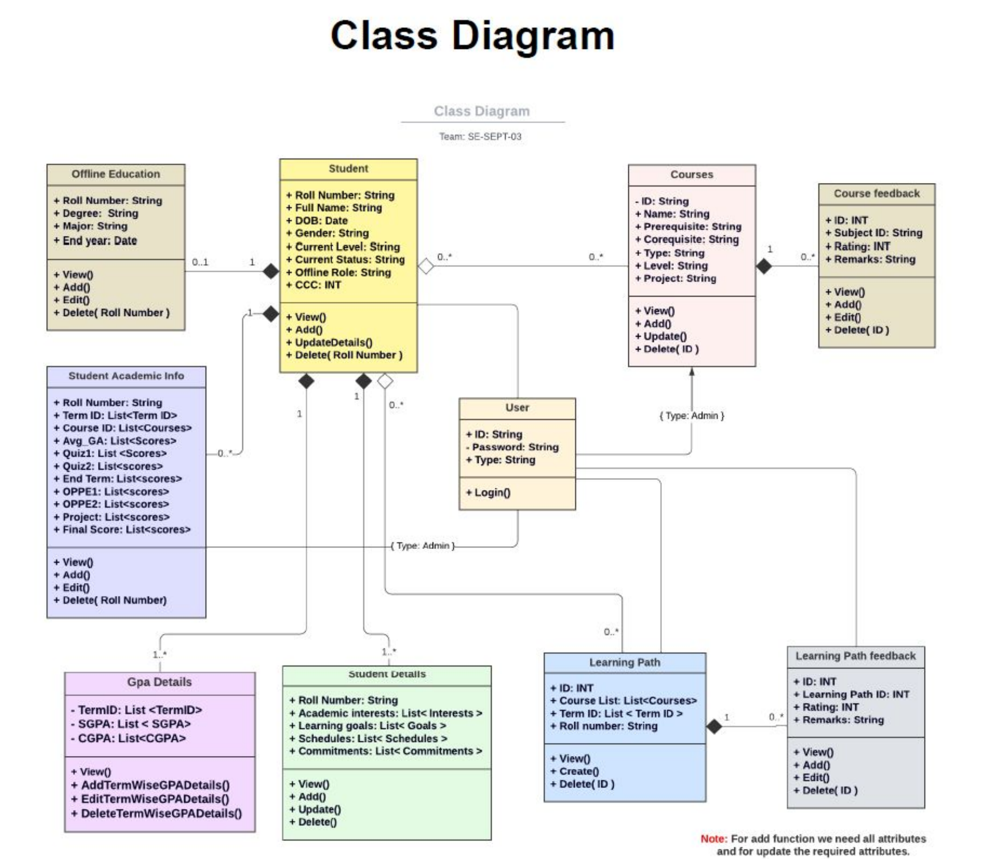
- Dynamic Report Template creation
- Email formatting
- Write Schedule JOB to send Email if Rating of Course Feedback in Less than 3
- Email JOB scheduler
- Email Template for Notification on Performance on Recommendation Engine

c. ML model for Recommendation Engine

- Data Cleaning
- Feature Engineering
- Model Architecture
- Model Training
- Model Evaluation
- Hyper parameter Tuning
- Model Deployment
- Saving Model checkpoint into Google Drive Storage
- API Integration with Application

Milestone 3: Software Design

Basic class diagrams of the proposed system



Class Diagram

Class Diagram

Team: SE-SEPT-03

Offline Education
+ Roll Number: String
+ Degree: String
+ Major: String
+ End year: Date

+ View()
+ Add()
+ Edit()
+ Delete(Roll Number)

Student
+ Roll Number: String
+ Full Name: String
+ DOB: Date
+ Gender: String
+ Current Level: String
+ Current Status: String
+ Offline Role: String
+ CCC: INT

+ View()
+ Add()
+ UpdateDetails()
+ Delete(Roll Number)

Courses
- ID: String
+ Name: String
+ Prerequisite: String
+ Corequisite: String
+ Type: String
+ Level: String
+ Project: String

+ View()
+ Add()
+ Update()
+ Delete(ID)

Course feedback
+ ID: INT
+ Subject ID: String
+ Rating: INT
+ Remarks: String

+ View()
+ Add()
+ Edit()
+ Delete(ID)

Student Academic Info
+ Roll Number: String
+ Term ID: List<Term ID>
+ Course ID: List<Courses>
+ Avg_GA: List<Scores>
+ Quiz1: List <Scores>
+ Quiz2: List<scores>
+ End Term: List<scores>
+ OPPE1: List<scores>
+ OPPE2: List<scores>
+ Project: List<scores>
+ Final Score: List<scores>

+ View()
+ Add()
+ Edit()
+ Delete(Roll Number)

User
+ ID: String
- Password: String
+ Type: String

+ Login()

Gpa Details
- TermID: List <TermID>
- SGPA: List < SGPA>
- CGPA: List<CGPA>

+ View()
+ AddTermWiseGPADetails()
+ EditTermWiseGPADetails()
+ DeleteTermWiseGPADetails()

Student Details
+ Roll Number: String
+ Academic interests: List< Interests >
+ Learning goals: List< Goals >
+ Schedules: List< Schedules >
+ Commitments: List< Commitments >

+ View()
+ Add()
+ Update()
+ Delete()

Learning Path
+ ID: INT
+ Course List: List<Courses>
+ Term ID: List < Term ID >
+ Roll number: String

+ View()
+ Create()
+ Delete(ID)

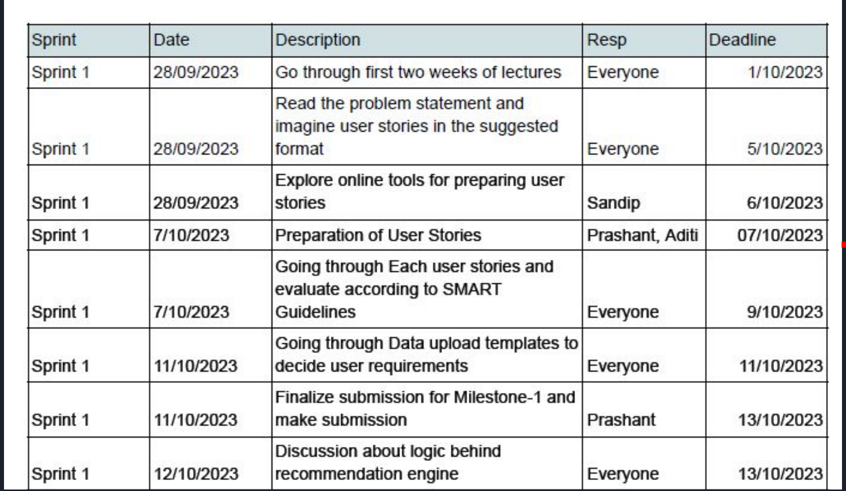
Learning Path feedback
+ ID: INT
+ Learning Path ID: INT
+ Rating: INT
+ Remarks: String

+ View()
+ Add()
+ Edit()
+ Delete(ID)

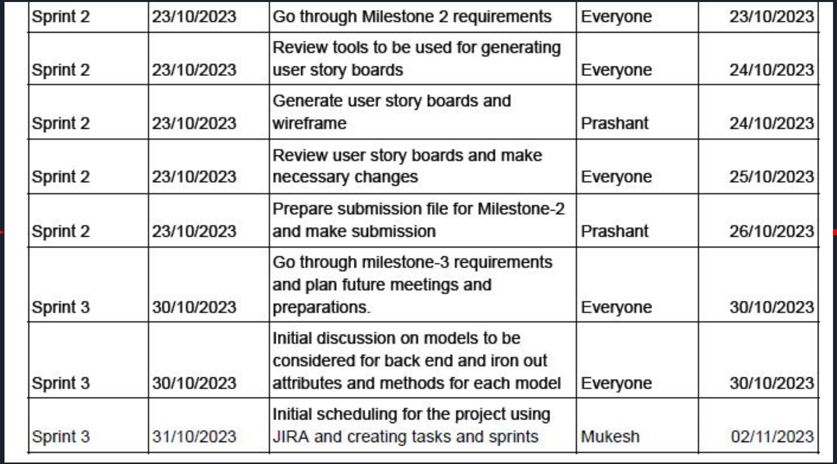
Note: For add function we need all attributes and for update the required attributes.

Milestone 3: Scrum Meetings

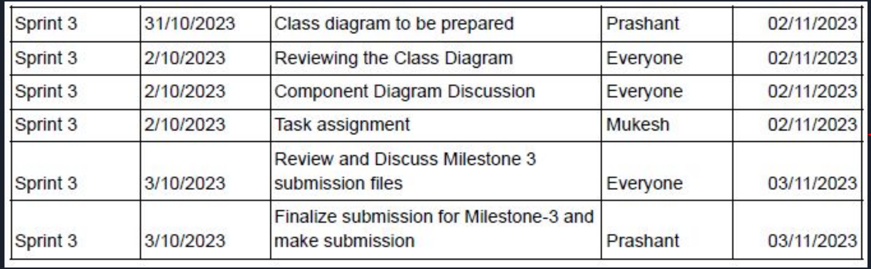
Details/Minutes of a few scrum meetings



Sprint	Date	Description	Resp	Deadline
Sprint 1	28/09/2023	Go through first two weeks of lectures	Everyone	1/10/2023
Sprint 1	28/09/2023	Read the problem statement and imagine user stories in the suggested format	Everyone	5/10/2023
Sprint 1	28/09/2023	Explore online tools for preparing user stories	Sandip	6/10/2023
Sprint 1	7/10/2023	Preparation of User Stories	Prashant, Aditi	07/10/2023
Sprint 1	7/10/2023	Going through Each user stories and evaluate according to SMART Guidelines	Everyone	9/10/2023
Sprint 1	11/10/2023	Going through Data upload templates to decide user requirements	Everyone	11/10/2023
Sprint 1	11/10/2023	Finalize submission for Milestone-1 and make submission	Prashant	13/10/2023
Sprint 1	12/10/2023	Discussion about logic behind recommendation engine	Everyone	13/10/2023



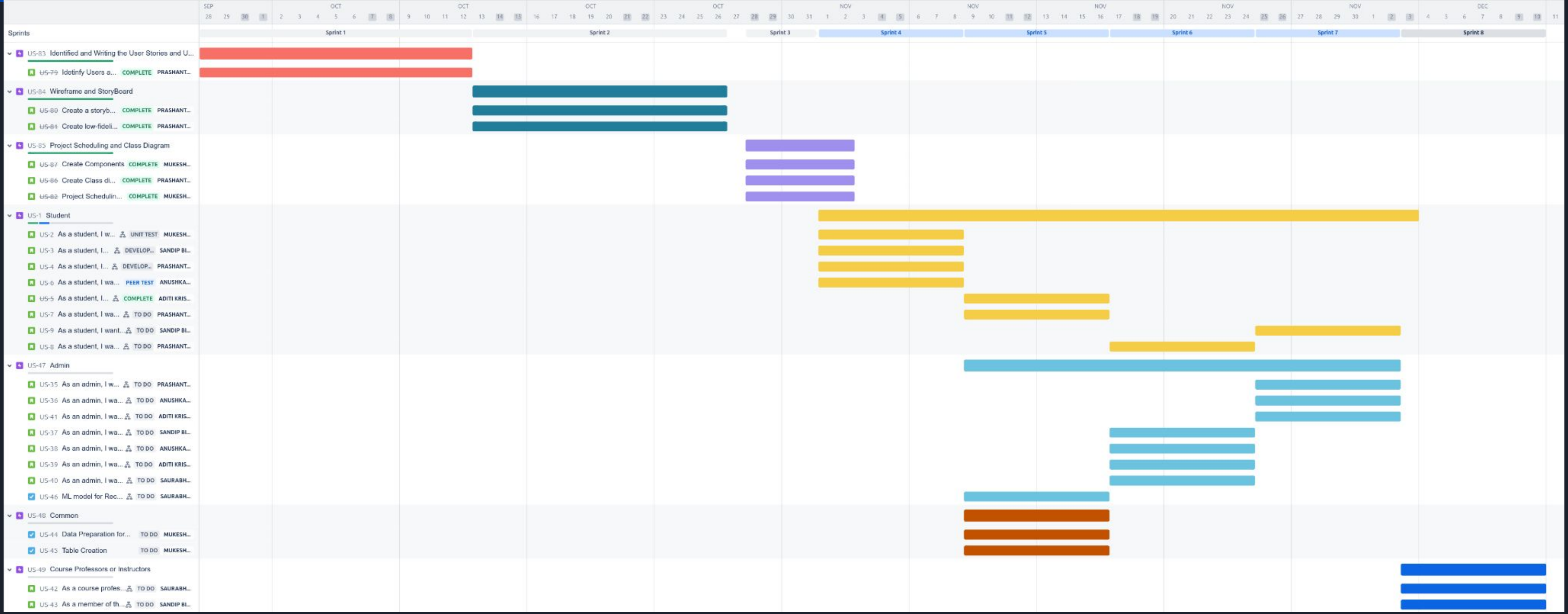
Sprint 2	23/10/2023	Go through Milestone 2 requirements	Everyone	23/10/2023
Sprint 2	23/10/2023	Review tools to be used for generating user story boards	Everyone	24/10/2023
Sprint 2	23/10/2023	Generate user story boards and wireframe	Prashant	24/10/2023
Sprint 2	23/10/2023	Review user story boards and make necessary changes	Everyone	25/10/2023
Sprint 2	23/10/2023	Prepare submission file for Milestone-2 and make submission	Prashant	26/10/2023
Sprint 3	30/10/2023	Go through milestone-3 requirements and plan future meetings and preparations.	Everyone	30/10/2023
Sprint 3	30/10/2023	Initial discussion on models to be considered for back end and iron out attributes and methods for each model	Everyone	30/10/2023
Sprint 3	31/10/2023	Initial scheduling for the project using JIRA and creating tasks and sprints	Mukesh	02/11/2023



Sprint 3	31/10/2023	Class diagram to be prepared	Prashant	02/11/2023
Sprint 3	2/10/2023	Reviewing the Class Diagram	Everyone	02/11/2023
Sprint 3	2/10/2023	Component Diagram Discussion	Everyone	02/11/2023
Sprint 3	2/10/2023	Task assignment	Mukesh	02/11/2023
Sprint 3	3/10/2023	Review and Discuss Milestone 3 submission files	Everyone	03/11/2023
Sprint 3	3/10/2023	Finalize submission for Milestone-3 and make submission	Prashant	03/11/2023

Milestone 3: Gantt Chart

Project Planning

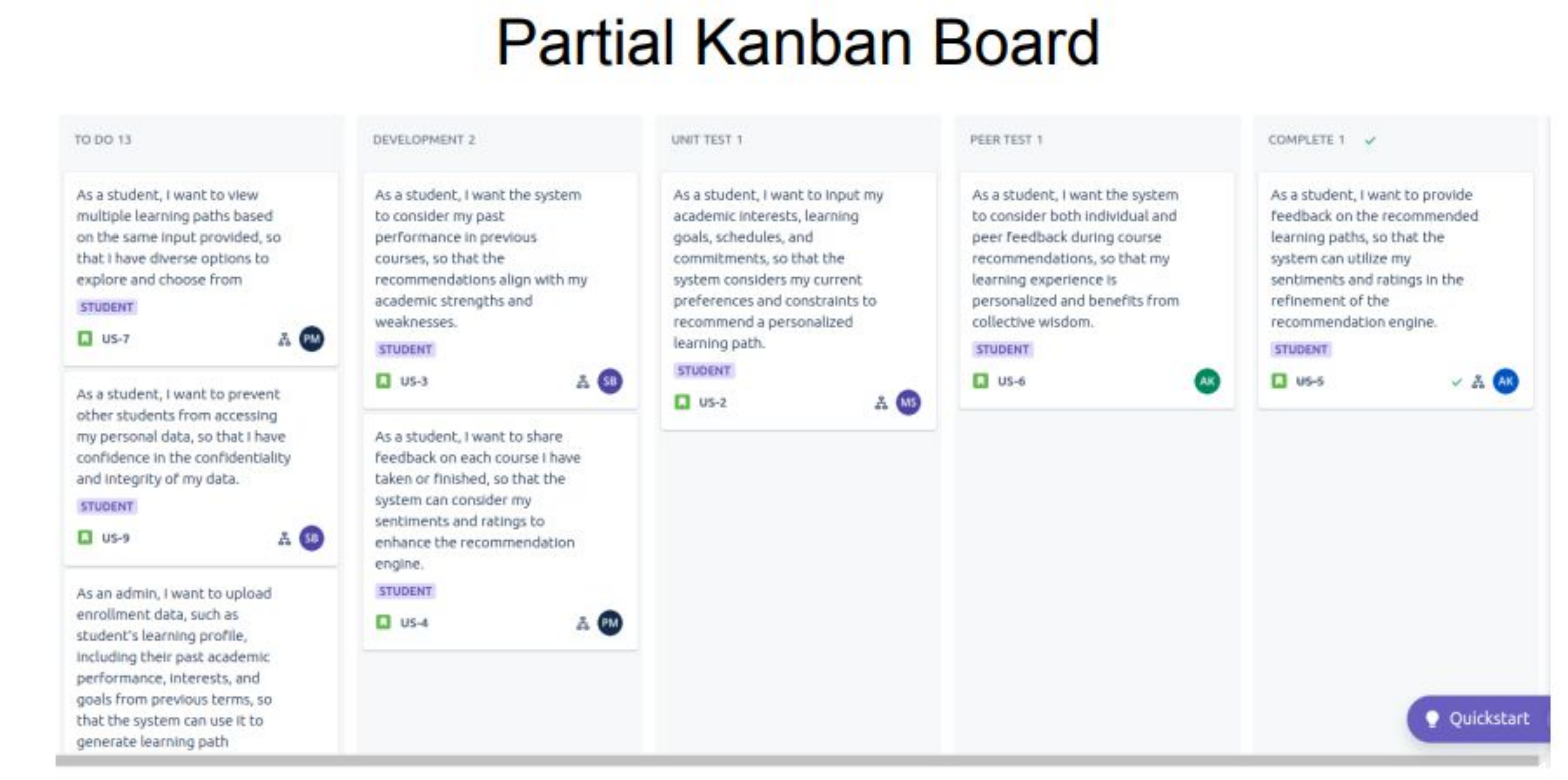


Sprints

US-83	Identified and Writing the User Stories and U...		
~~US-79~~	Idetinfy Users a...	COMPLETE	PRASHANT...
US-84	Wireframe and StoryBoard		
~~US-80~~	Create a storyb...	COMPLETE	PRASHANT...
~~US-81~~	Create low-fideli...	COMPLETE	PRASHANT...
US-85	Project Scheduling and Class Diagram		
~~US-87~~	Create Components	COMPLETE	MUKESH...
~~US-86~~	Create Class di...	COMPLETE	PRASHANT...
~~US-82~~	Project Schedulin...	COMPLETE	MUKESH...
US-1	Student		
US-2	As a student, I w...	UNIT TEST	MUKESH...
US-3	As a student, I...	DEVELOP...	SANDIP BI...
US-4	As a student, I...	DEVELOP...	PRASHANT...
US-6	As a student, I wa...	PEER TEST	ANUSHKA...
~~US-5~~	As a student, I...	COMPLETE	ADITI KRIS...
US-7	As a student, I wa...	TO DO	PRASHANT...
US-9	As a student, I want...	TO DO	SANDIP BI...
US-8	As a student, I wa...	TO DO	PRASHANT...
US-47	Admin		
US-35	As an admin, I w...	TO DO	PRASHANT...
US-36	As an admin, I wa...	TO DO	ANUSHKA...
US-41	As an admin, I wa...	TO DO	ADITI KRIS...
US-37	As an admin, I wa...	TO DO	SANDIP BI...
US-38	As an admin, I wa...	TO DO	ANUSHKA...
US-39	As an admin, I wa...	TO DO	ADITI KRIS...
US-40	As an admin, I wa...	TO DO	SAURABH...
US-46	ML model for Rec...	TO DO	SAURABH...
US-48	Common		
US-44	Data Preparation for...	TO DO	MUKESH...
US-45	Table Creation	TO DO	MUKESH...
US-49	Course Professors or Instructors		
US-42	As a course profes...	TO DO	SAURABH...
US-43	As a member of th...	TO DO	SANDIP BI...

Milestone 3: Kanban Board

Project Planning



Partial Kanban Board

TO DO 13

As a student, I want to view multiple learning paths based on the same input provided, so that I have diverse options to explore and choose from

STUDENT

US-7 PM

As a student, I want to prevent other students from accessing my personal data, so that I have confidence in the confidentiality and integrity of my data.

STUDENT

US-9 SB

As an admin, I want to upload enrollment data, such as student's learning profile, including their past academic performance, interests, and goals from previous terms, so that the system can use it to generate learning path

DEVELOPMENT 2

As a student, I want the system to consider my past performance in previous courses, so that the recommendations align with my academic strengths and weaknesses.

STUDENT

US-3 SB

As a student, I want to share feedback on each course I have taken or finished, so that the system can consider my sentiments and ratings to enhance the recommendation engine.

STUDENT

US-4 PM

UNIT TEST 1

As a student, I want to input my academic interests, learning goals, schedules, and commitments, so that the system considers my current preferences and constraints to recommend a personalized learning path.

STUDENT

US-2 MS

PEER TEST 1

As a student, I want the system to consider both individual and peer feedback during course recommendations, so that my learning experience is personalized and benefits from collective wisdom.

STUDENT

US-6 AK

COMPLETE 1 ✓

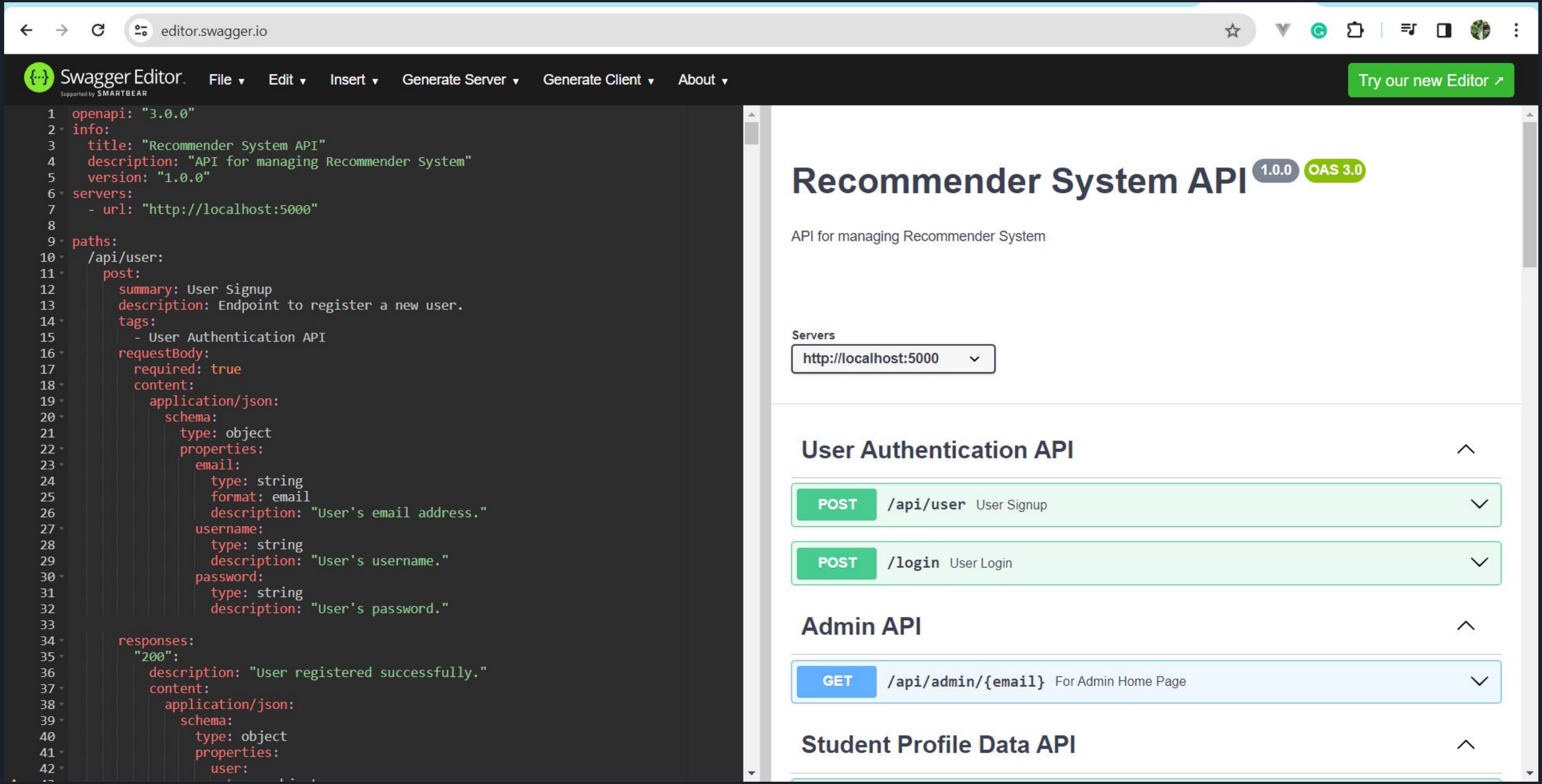
As a student, I want to provide feedback on the recommended learning paths, so that the system can utilize my sentiments and ratings in the refinement of the recommendation engine.

STUDENT

US-5 ✓ AK

Milestone 4: **YAML Documentation**

Link



```yaml
openapi: "3.0.0"
info:
  title: "Recommender System API"
  description: "API for managing Recommender System"
  version: "1.0.0"
servers:
  - url: "http://localhost:5000"

paths:
  /api/user:
    post:
      summary: User Signup
      description: Endpoint to register a new user.
      tags:
        - User Authentication API
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                email:
                  type: string
                  format: email
                  description: "User's email address."
                username:
                  type: string
                  description: "User's username."
                password:
                  type: string
                  description: "User's password."

      responses:
        "200":
          description: "User registered successfully."
          content:
            application/json:
              schema:
                type: object
                properties:
                  user:
```

Recommender System API 1.0.0 OAS 3.0

API for managing Recommender System

Servers
http://localhost:5000

User Authentication API

POST /api/user User Signup

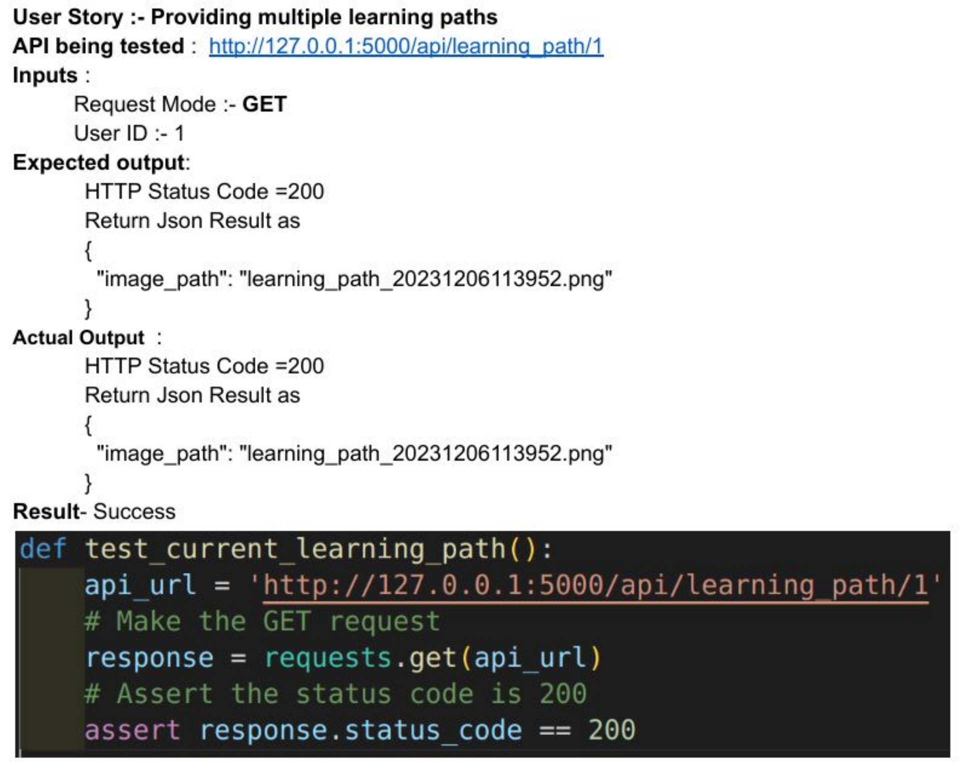
POST /login User Login

Admin API

GET /api/admin/{email} For Admin Home Page

Student Profile Data API

Milestone 5: **API Testing**



User Story :- Providing multiple learning paths

API being tested : http://127.0.0.1:5000/api/learning_path/1

Inputs :

Request Mode :- **GET**

User ID :- 1

Expected output:

HTTP Status Code =200

Return Json Result as

```
{
  "image_path": "learning_path_20231206113952.png"
}
```

Actual Output :

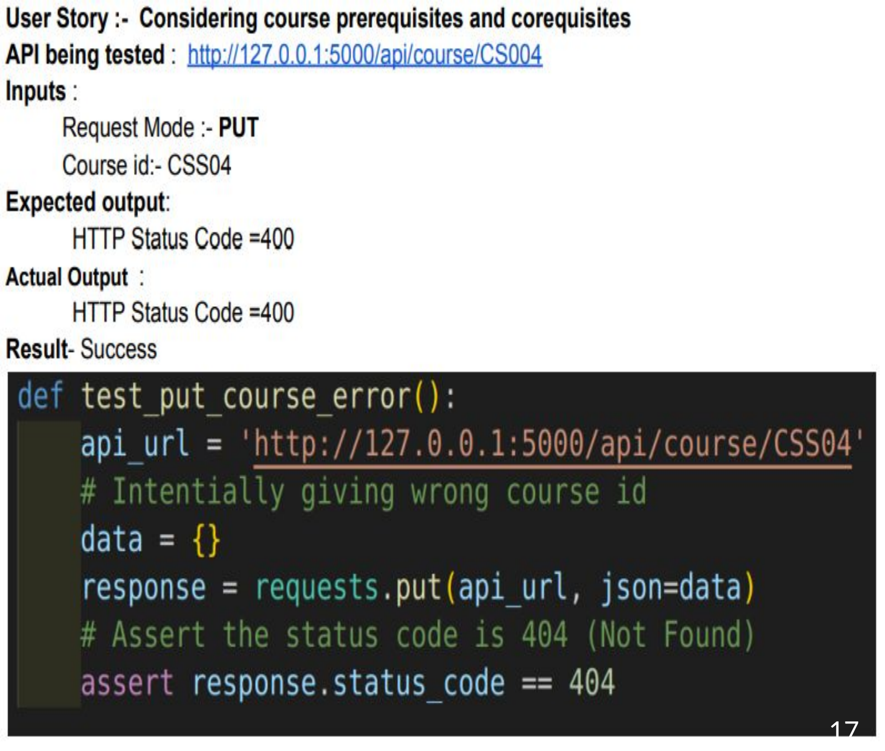
HTTP Status Code =200

Return Json Result as

```
{
  "image_path": "learning_path_20231206113952.png"
}
```

Result- Success

```python
def test_current_learning_path():
    api_url = 'http://127.0.0.1:5000/api/learning_path/1'
    # Make the GET request
    response = requests.get(api_url)
    # Assert the status code is 200
    assert response.status_code == 200
```



User Story :- Considering course prerequisites and corequisites

API being tested : http://127.0.0.1:5000/api/course/CS004

Inputs :

Request Mode :- **PUT**

Course id:- CSS04

Expected output:

HTTP Status Code =400

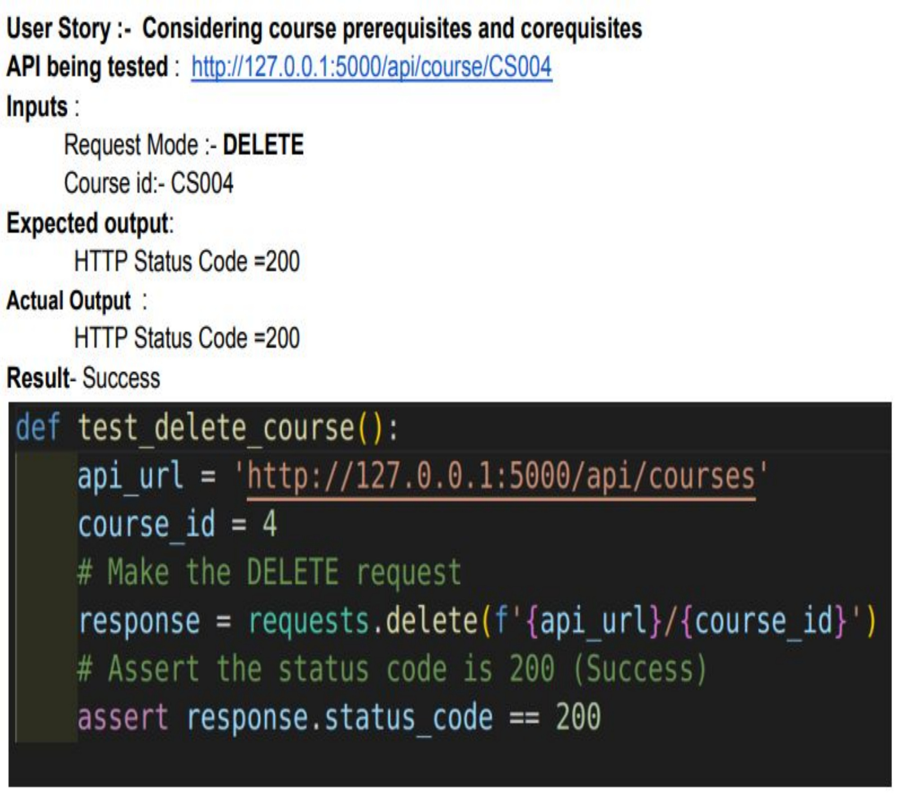
Actual Output :

HTTP Status Code =400

Result- Success

```python
def test_put_course_error():
    api_url = 'http://127.0.0.1:5000/api/course/CSS04'
    # Intentially giving wrong course id
    data = {}
    response = requests.put(api_url, json=data)
    # Assert the status code is 404 (Not Found)
    assert response.status_code == 404
```

Milestone 5: **API Testing**



User Story :- Considering course prerequisites and corequisites

API being tested : http://127.0.0.1:5000/api/course/CS004

Inputs :

Request Mode :- **DELETE**

Course id:- CS004

Expected output:

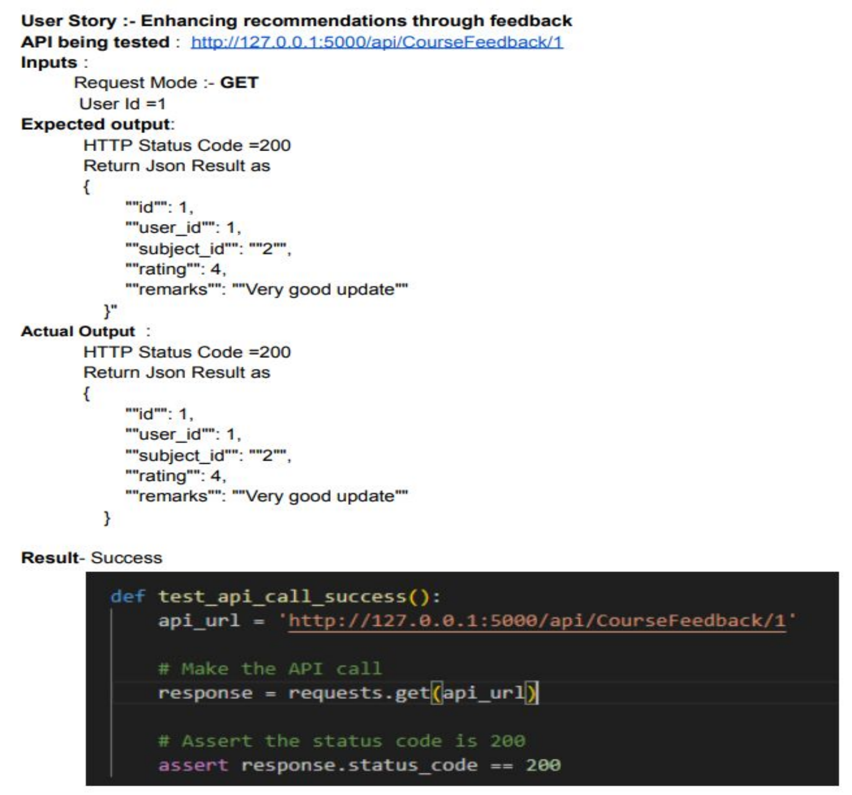
HTTP Status Code =200

Actual Output :

HTTP Status Code =200

Result- Success

```
def test_delete_course():
    api_url = 'http://127.0.0.1:5000/api/courses'
    course_id = 4
    # Make the DELETE request
    response = requests.delete(f'{api_url}/{course_id}')
    # Assert the status code is 200 (Success)
    assert response.status_code == 200
```



User Story :- Enhancing recommendations through feedback

API being tested : http://127.0.0.1:5000/api/CourseFeedback/1

Inputs :

Request Mode :- **GET**

User Id =1

Expected output:

HTTP Status Code =200

Return Json Result as

```
{
    ""id"": 1,
    ""user_id"": 1,
    ""subject_id"": ""2"",
    ""rating"": 4,
    ""remarks"": ""Very good update""
  }"
```

Actual Output :

HTTP Status Code =200

Return Json Result as

```
{
    ""id"": 1,
    ""user_id"": 1,
    ""subject_id"": ""2"",
    ""rating"": 4,
    ""remarks"": ""Very good update""
  }
```

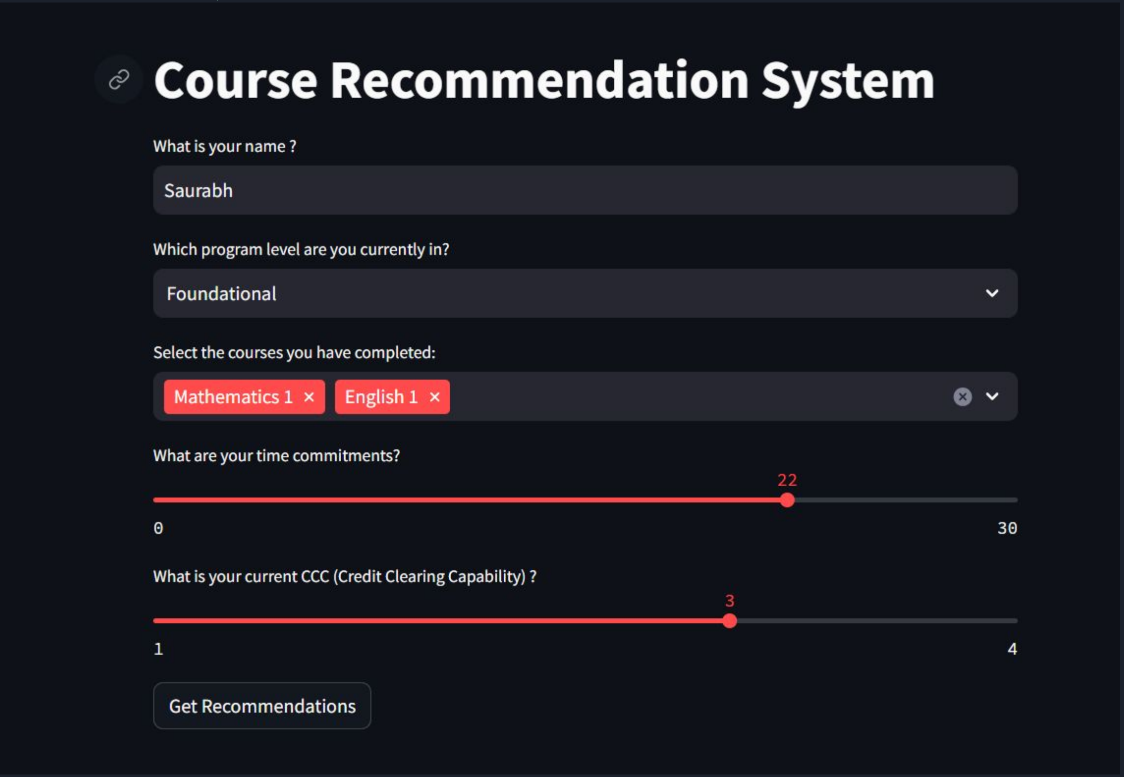
Result- Success

```
def test_api_call_success():
    api_url = 'http://127.0.0.1:5000/api/CourseFeedback/1'

    # Make the API call
    response = requests.get(api_url)

    # Assert the status code is 200
    assert response.status_code == 200
```

Course Recommendation System



Course Recommendation System

What is your name ?

Saurabh

Which program level are you currently in?

Foundational

Select the courses you have completed:

Mathematics 1 × English 1 ×

What are your time commitments?

22

0 30

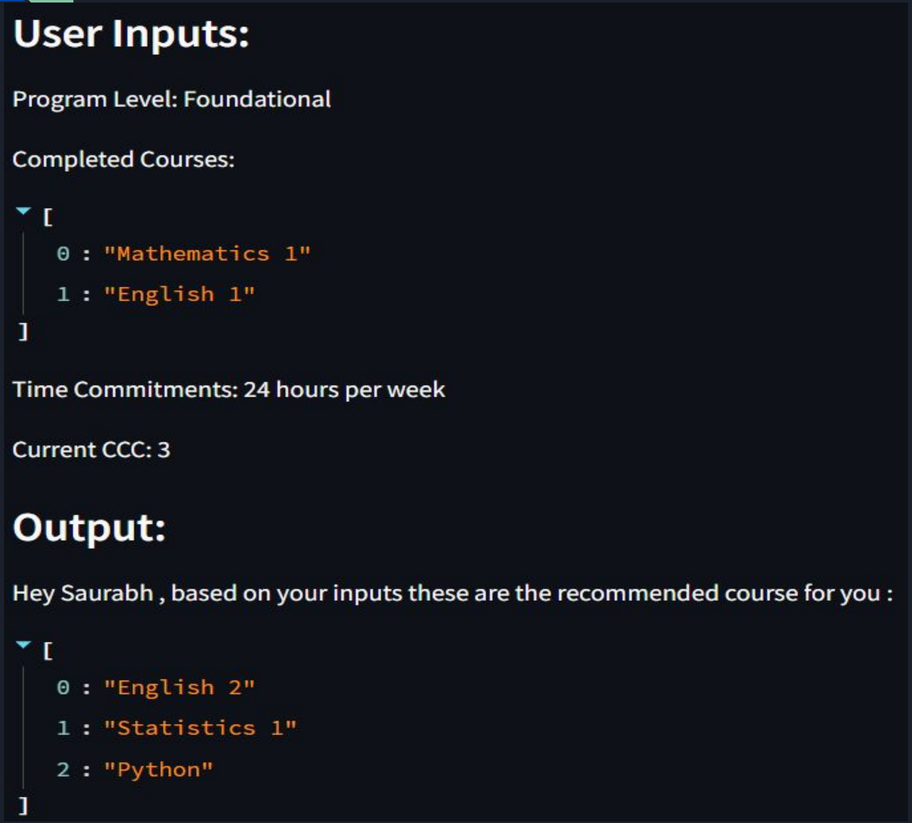
What is your current CCC (Credit Clearing Capability) ?

3

1 4

- This Course Recommendation system is created using rule based logics
- Used Streamlit framework for frontend and fast prototyping.
- Recommendation engine takes these parameters :
    - Previously takes courses
    - Time commitments
    - CCC (credit clearing capability)
- Used Difficulty level of courses as metric for recommendation.
- Used Sentiments of Peer Reviews to decided to difficulty level of courses.
- Based on the CCC values user variable no of course recommendations.
- Hosted on Huggingface spaces for flexibility.

Course Recommendation System-Output



User Inputs:

Program Level: Foundational

Completed Courses:

```
▾ [
    0 : "Mathematics 1"
    1 : "English 1"
]
```

Time Commitments: 24 hours per week

Current CCC: 3

Output:

Hey Saurabh , based on your inputs these are the recommended course for you :

```
▾ [
    0 : "English 2"
    1 : "Statistics 1"
    2 : "Python"
]
```

- Based on the user input the system displays the input parameters and the list of recommended courses.

- Recommendation engine also take into account the co-requisites and prerequisites of particular courses

- **Working Mechanism:** Based on the inputs , the system first filter out the remaining courses and classify them into different difficulty level (Easy , Medium , Hard) based on the sentiment analysis done on course feedback data and generated the recommendation based on time commitment and CCC in that term.

Application Demo

Thank you !!