SoftwareEngineering
Group 19

Final Project Report

This document is the final project report for our IITM BS Course recommendation system, developed for our Software Engineering project. It includes all our previous milestone submissions and all other relevant details as required by the problem statement. This report is consistent with the intermediate milestone documents

Chandana Nisankara 21f1005727 **Dhanya K** 21f1001504 **Patil Dhairyasheel** 21f1006987 **Saquib Hashmi** 21f1004600 **Varun Sunderarajan** 21f2001516

Detailed Report On Work Done From Milestone 1 To Milestone 5:

The following pages include all the milestone submissions made as part of our previous milestones. The milestones have been separated by pages, which demarcate the different sections and indicate the milestone that follows.

Video presentation links:

- **Backend:** https://github.com/bsc-iitm/soft-engg-project-sept-2023-se-sept-19-1/blob/main/Milestone-6-Final-Submission/backend_demo.mp4
- **Frontend:** https://github.com/bsc-iitm/soft-engg-project-sept-2023-se-sept-19-1/blob/main/Milestone-6-Final-Submission/frontend_demo-720.mp4

MILESTONE 1 SUBMISSIONS FOLLOW

Honour Code

I **Chandana Nisankara** with roll no. **21f1005727** declare that I will not use any ideas, writings, code or work that is not my own or my group's with the intention of claiming it my or my group's work. For all the work that I will submit as part of this project, I will not share it outside my group with anybody directly or indirectly or upload it to any of the public forums on the internet.
I acknowledge that failing in any of the above constitutes plagiarism and in that case, the institute will take appropriate disciplinary action.
Sign: Chandana Nisankara
Date:12-10-2023

I **Dhanya K** with roll no. **21f1001504** declare that I will not use any ideas, writings, code or work that is not my own or my group's with the intention of claiming it my or my group's work. For all the work that I will submit as part of this project, I will not share it outside my group with anybody directly or indirectly or upload it to any of the public forums on the internet.
I acknowledge that failing in any of the above constitutes plagiarism and in that case, the institute will take appropriate disciplinary action.
Sign: Dhanya K
Date: 12-10-2023

I **Patil Dhairyasheel** with roll no. **21f1006987** declare that I will not use any ideas, writings, code or work that is not my own or my group's with the intention of claiming it my or my group's work. For all the work that I will submit as part of this project, I will not share it outside my group with anybody directly or indirectly or upload it to any of the public forums on the internet.
I acknowledge that failing in any of the above constitutes plagiarism and in that case, the institute will take appropriate disciplinary action.
Sign:Patil Dhairyasheel
Date:12-10-2023

I **Saquib Hashmi** with roll no.**21f1004600** declare that I will not use any ideas, writings, code or work that is not my own or my group's with the intention of claiming it my or my group's work. For all the work that I will submit as part of this project, I will not share it outside my group with anybody directly or indirectly or upload it to any of the public forums on the internet.
I acknowledge that failing in any of the above constitutes plagiarism and in that case, the institute will take appropriate disciplinary action.
Sign: Saquib Hashmi
Date:12-10-2023

I **Varun Sunderarajan** with roll no. **21f2001516** declare that I will not use any ideas, writings, code or work that is not my own or my group's with the intention of claiming it my or my group's work. For all the work that I will submit as part of this project, I will not share it outside my group with anybody directly or indirectly or upload it to any of the public forums on the internet.
I acknowledge that failing in any of the above constitutes plagiarism and in that case, the institute will take appropriate disciplinary action.
Sign: Varun Sunderarajan
Date: 12-10-2023

Identifying the various types of Users

The following are the various types of users of the Learning Path Recommendation system:

Primary Users:
- Students
- Administrator

Secondary Users:
- Course Instructor
- Data Analyst
- Academic advisor
- IT support
- Institutional Researchers

Tertiary Users:
- Developers

User stories

Here are the user stories with proper sequence numbers:

1. As a student,
   I want to be able to log into my student account securely,
   So that I can access and utilize the features and resources available within the system.

2. As a student,
   I want to access my learning profile, where I can access past scores and details of quizzes
   So that I can have details of past completed courses.

3. As a student,
   I want to receive notifications and reminders for important academic dates, such as registration deadlines,
   So that I can stay organized and avoid missing critical events.

4. As a student,
   I want to receive personalized course recommendations based on my academic goals, interests,
   So that I can make informed decisions about my course selections.

5. As a student,
   I want to input my domain preferences (programming/mathematics/business analytics), achievements and/or professional details,
   So that the system can generate better personalized course recommendations for me.

6. As a student,
   I want to input my details like weekly available time , preferences ,
   So that the system can generate personalized combinations according to data for me.

7. As a student,
   I want to be able to see course wise feedback given by others,
   So that it can help me to choose my preferences.

8. As a student,
   I want to give feedback in the form of text as well as rate the courses based on their difficulty level,
   So that I can help the program fine-tune its algorithm and also for the students to see my feedback and for me to reflect on my own feedback in future.

9. As a student,
   I want all my personal data which are collected by the software to be anonymous,
   So that my privacy is protected.

10. As an admin,
    I want to be able to log into the admin panel securely,
    So that I can access and manage the system's administrative functions.

11. As an admin,
    I want to upload enrollment data from previous terms in a bulk format,
    So that the system can analyze patterns and improve its recommendation algorithm.

12. As an admin,
    I want to monitor student feedback,
    So that the code and conduct is maintained.

13. As an admin,
    I want to assist students with account-related issues,
    So that I can provide support and maintain system security.

14. As a data analyst,
    I want to access all the necessary data from the database
    So that I can check the correctness of the algorithm and make changes accordingly.

15. As an IT support staff member,
    I want to receive alerts on system errors or issues,
    So that I can promptly resolve them to ensure the system runs smoothly.

16. As an Academic advisor ,
    I want to receive analytics and feedback
    So that I can make data-driven improvements to the course content and structure.

MILESTONE 2 SUBMISSIONS FOLLOW

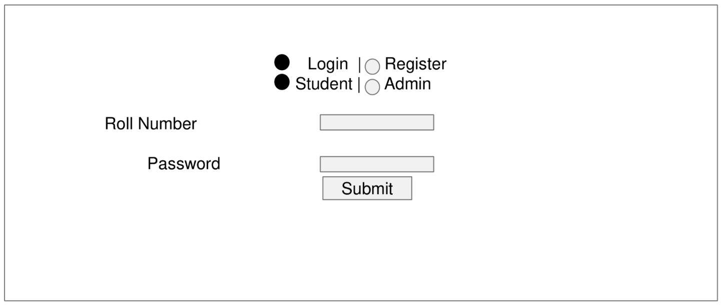
Wireframes - Common Views

In this document, we described the wireframes for the common views that are intended for all users of the application.

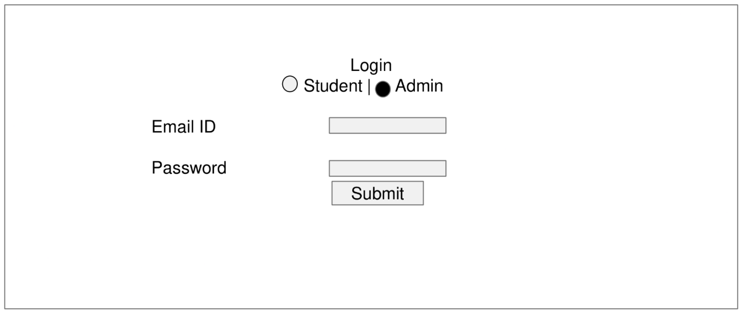
Login Page:

This is the login view that appears when a user first visits the course recommendation system website. It allows students, admins, and other users to login by entering their credentials. Students login with their roll numbers and passwords. Admins login with their email IDs and passwords. There is also a registration link that allows new users to create accounts.

User Login (Student)



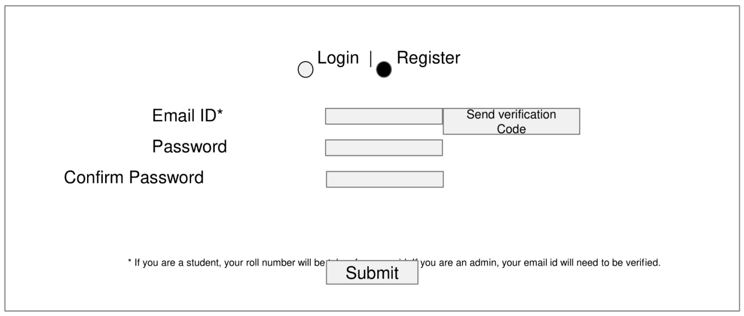
User Login (Admin)



Registration Page:

This page allows new users to register for the course recommendation system. Users enter details like email ID, password, and confirm password to create an account. For students, their roll numbers are automatically captured during registration. Admin users need to validate their email IDs after registration.

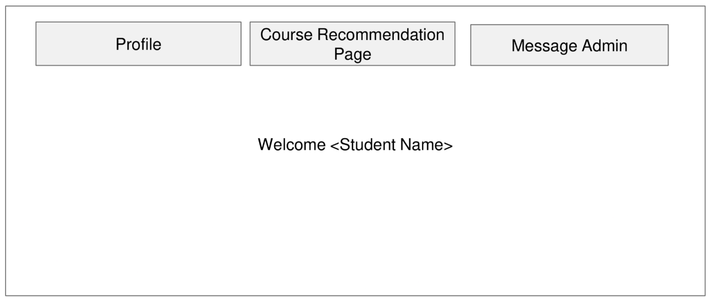
Register


Login | ● Register
Email ID*
Send verification Code
Password
Confirm Password
* If you are a student, your roll number will be [illegible] you are an admin, your email id will need to be verified.
Submit

Student Dashboard:

This is the landing page for students after they successfully login. It greets them by name and provides profile, course recommendations, messaging, and other options. From here, students can access their profile details, view suggested courses, give feedback, and contact admins.

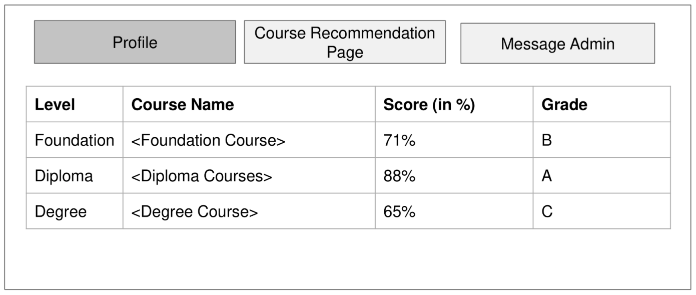
Student Dashboard



Student Profile Page:

This page shows students their profile information. It displays their current course details like foundation, diploma and degree courses enrolled, scores obtained, and grades. It serves as the student's overview within the system.

Student Profile

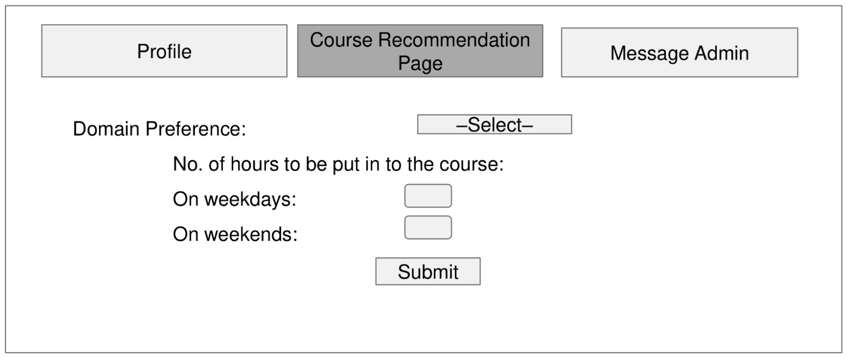


Level	Course Name	Score (in %)	Grade
Foundation	<Foundation Course>	71%	B
Diploma	<Diploma Courses>	88%	A
Degree	<Degree Course>	65%	C

Course Recommendation Page:

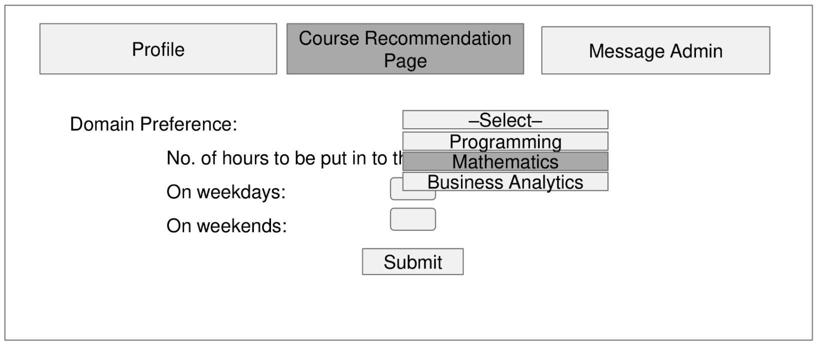
This is where students select parameters like domain preference, study hours per week, etc. to receive course recommendations suited to their needs. The system analyzes these inputs and suggests the top courses along with estimated weekly hours and difficulty level. Students can view more course details or give feedback.

Course Recommendation Page

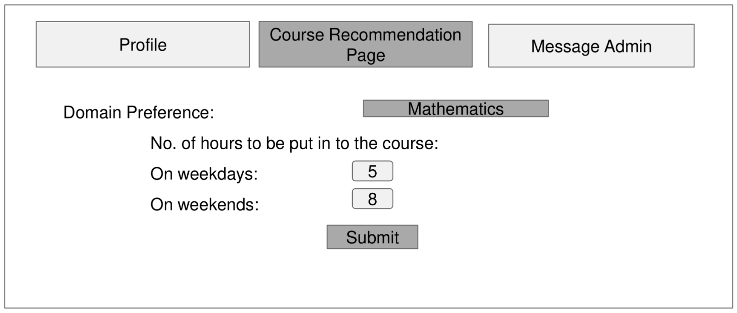


Domain Preference:	–Select–
No. of hours to be put in to the course:	
On weekdays:	
On weekends:	

Course Recommendation Page



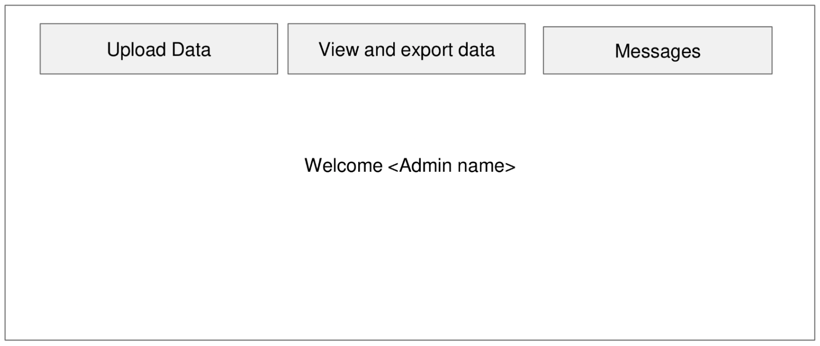
Course Recommendation Page



Admin Dashboard Page:

This is the admin landing page for managing the system. Key features enable data uploads, exports, viewing analytical reports, managing student registrations & profiles, addressing queries through the messaging module. Overall, it equips admins with controls to handle core functions.

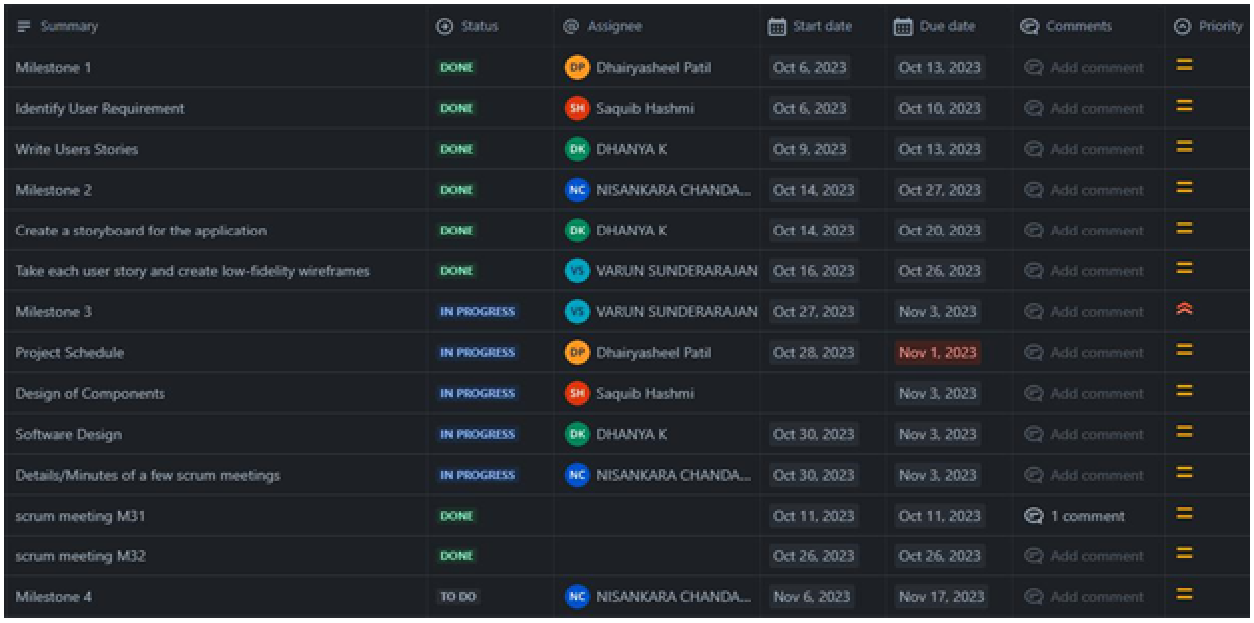
Admin Dashboard



MILESTONE 3 SUBMISSIONS FOLLOW

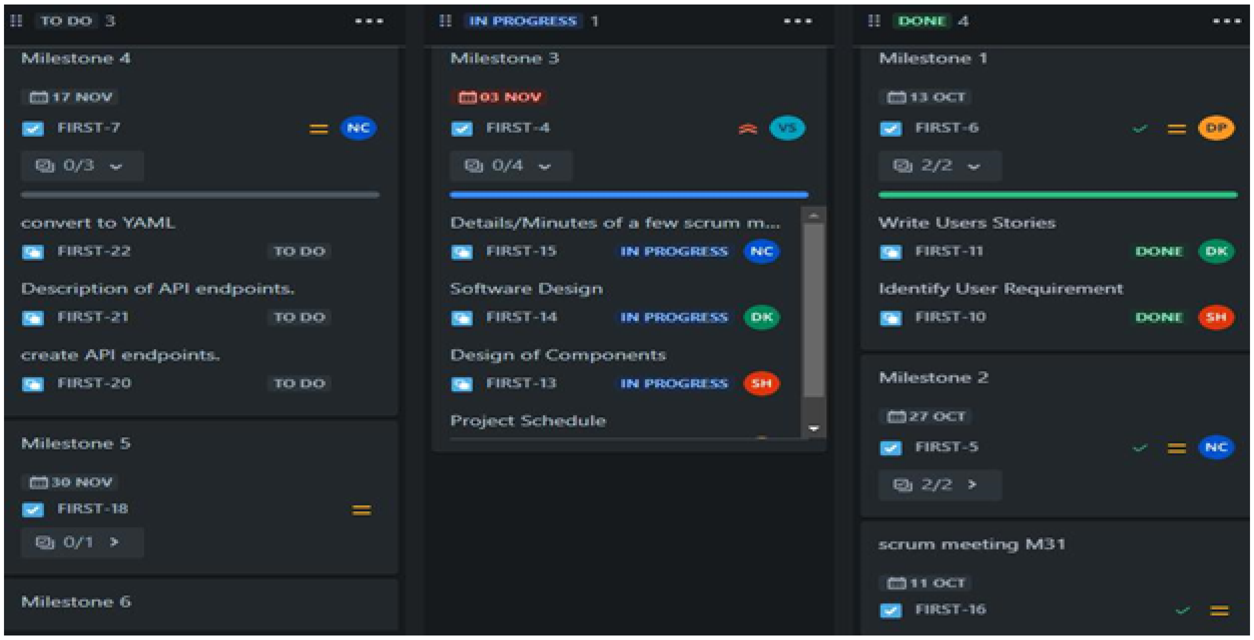
Scheduling and Design

Project Schedule



Summary	Status	Assignee	Start date	Due date	Comments	Priority
Milestone 1	DONE	DP Dhairyasheel Patil	Oct 6, 2023	Oct 13, 2023	Add comment	=
Identify User Requirement	DONE	SH Saquib Hashmi	Oct 6, 2023	Oct 10, 2023	Add comment	=
Write Users Stories	DONE	DK DHANYA K	Oct 9, 2023	Oct 13, 2023	Add comment	=
Milestone 2	DONE	NC NISANKARA CHANDA...	Oct 14, 2023	Oct 27, 2023	Add comment	=
Create a storyboard for the application	DONE	DK DHANYA K	Oct 14, 2023	Oct 20, 2023	Add comment	=
Take each user story and create low-fidelity wireframes	DONE	VS VARUN SUNDERARAJAN	Oct 16, 2023	Oct 26, 2023	Add comment	=
Milestone 3	IN PROGRESS	VS VARUN SUNDERARAJAN	Oct 27, 2023	Nov 3, 2023	Add comment	⌃
Project Schedule	IN PROGRESS	DP Dhairyasheel Patil	Oct 28, 2023	Nov 1, 2023	Add comment	=
Design of Components	IN PROGRESS	SH Saquib Hashmi		Nov 3, 2023	Add comment	=
Software Design	IN PROGRESS	DK DHANYA K	Oct 30, 2023	Nov 3, 2023	Add comment	=
Details/Minutes of a few scrum meetings	IN PROGRESS	NC NISANKARA CHANDA...	Oct 30, 2023	Nov 3, 2023	Add comment	=
scrum meeting M31	DONE		Oct 11, 2023	Oct 11, 2023	1 comment	=
scrum meeting M32	DONE		Oct 26, 2023	Oct 26, 2023	Add comment	=
Milestone 4	TO DO	NC NISANKARA CHANDA...	Nov 6, 2023	Nov 17, 2023	Add comment	=

We used the Jira Work Management app to collaborate and schedule our tasks and then assign different tasks to each member. We added scrum meetings as well in the schedule where we scheduled meetings and then updated it with the minutes later for reference.



TO DO 3

Milestone 4
17 NOV
FIRST-7 = NC
0/3

convert to YAML
FIRST-22 TO DO

Description of API endpoints.
FIRST-21 TO DO

create API endpoints.
FIRST-20 TO DO

Milestone 5
30 NOV
FIRST-18 =
0/1

Milestone 6

IN PROGRESS 1

Milestone 3
03 NOV
FIRST-4 VS
0/4

Details/Minutes of a few scrum m...
FIRST-15 IN PROGRESS NC

Software Design
FIRST-14 IN PROGRESS DK

Design of Components
FIRST-13 IN PROGRESS SH

Project Schedule

DONE 4

Milestone 1
13 OCT
FIRST-6 ✓ = DP
2/2

Write Users Stories
FIRST-11 DONE DK

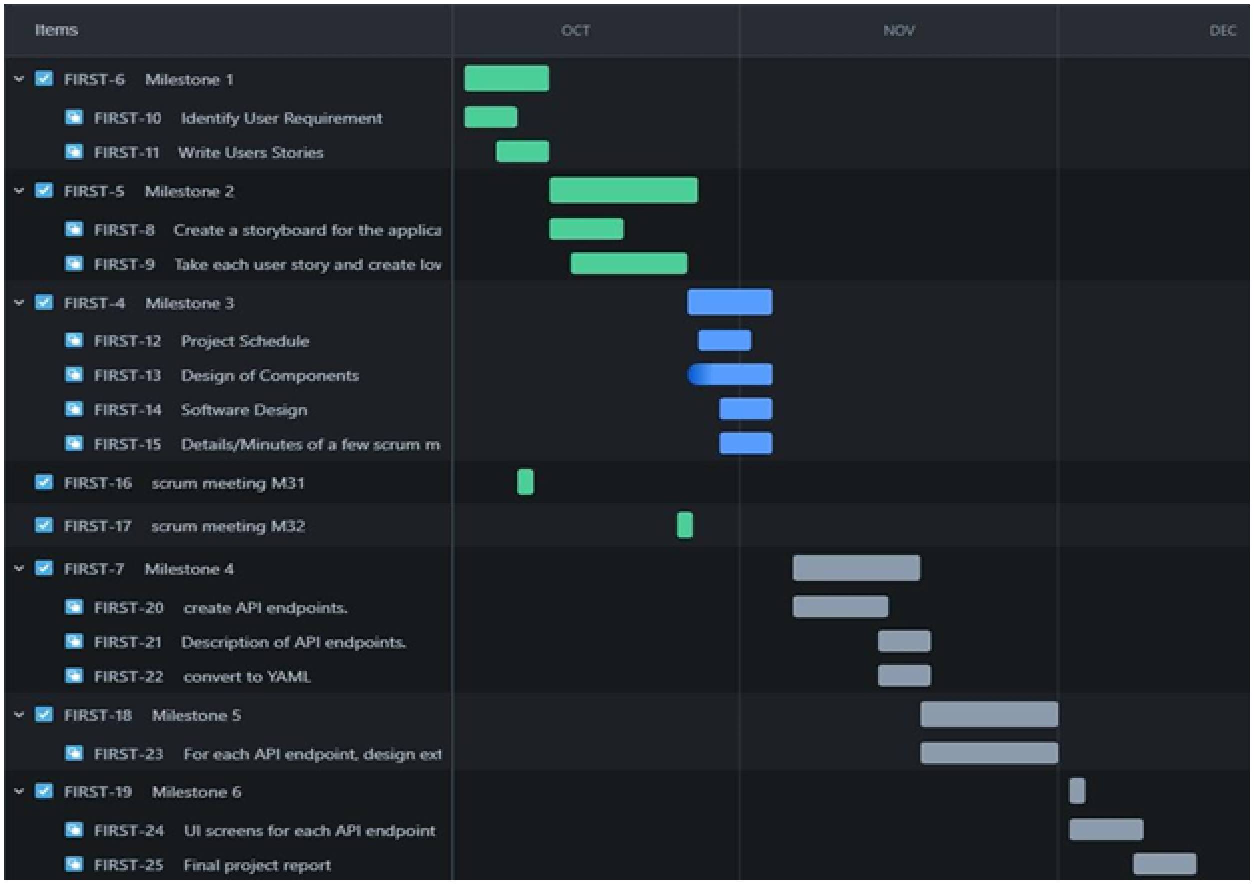
Identify User Requirement
FIRST-10 DONE SH

Milestone 2
27 OCT
FIRST-5 ✓ = NC
2/2

scrum meeting M31
11 OCT
FIRST-16 ✓ =

The app also gives us a task board where we can check all the tasks that are completed or are pending and their respective details.

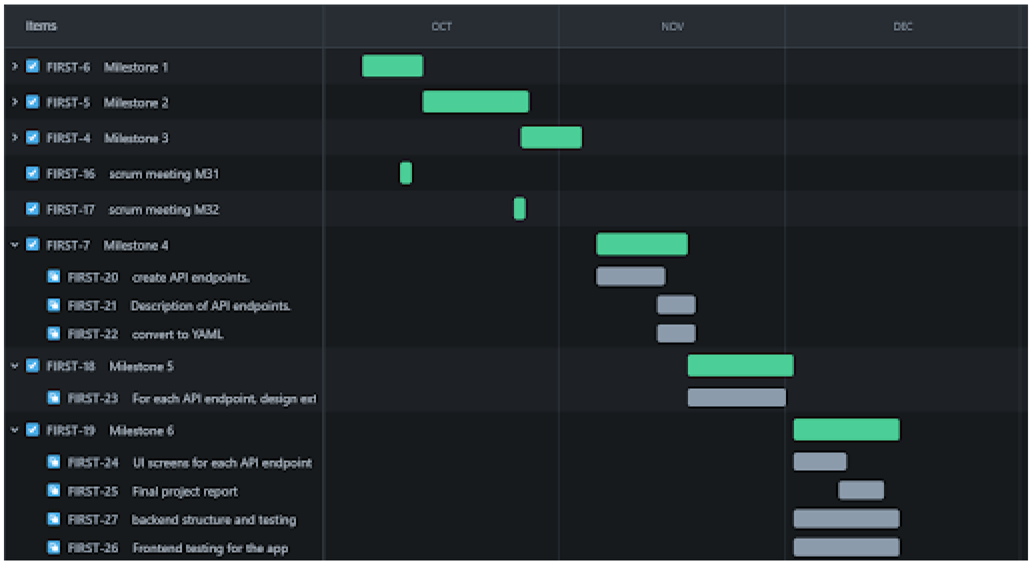
GANTT CHART



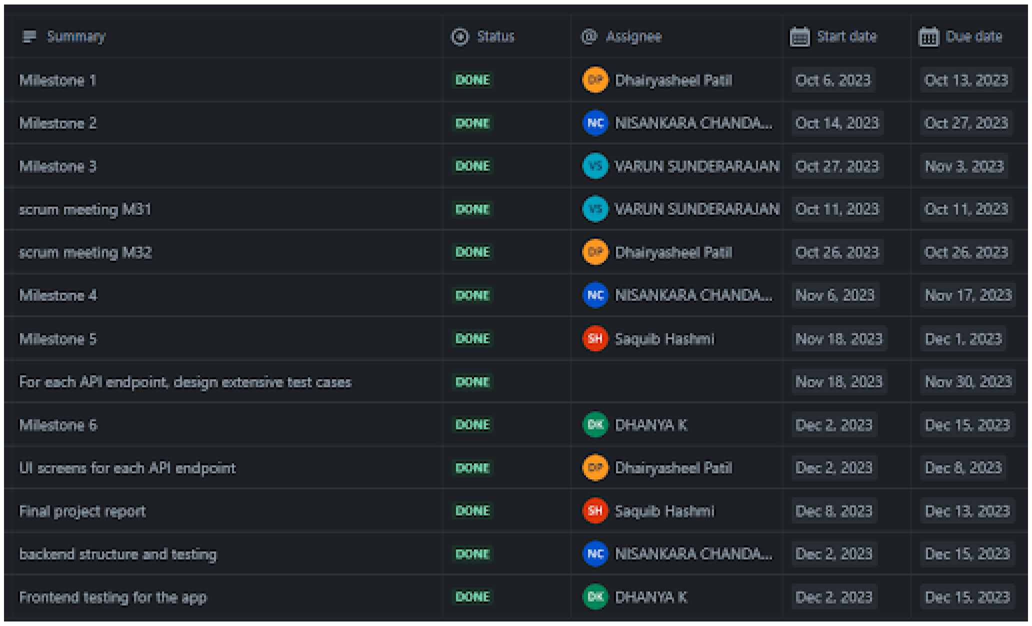
FIRST-6 Milestone 1
FIRST-10 Identify User Requirement
FIRST-11 Write Users Stories
FIRST-5 Milestone 2
FIRST-8 Create a storyboard for the applica
FIRST-9 Take each user story and create low
FIRST-4 Milestone 3
FIRST-12 Project Schedule
FIRST-13 Design of Components
FIRST-14 Software Design
FIRST-15 Details/Minutes of a few scrum m
FIRST-16 scrum meeting M31
FIRST-17 scrum meeting M32
FIRST-7 Milestone 4
FIRST-20 create API endpoints.
FIRST-21 Description of API endpoints.
FIRST-22 convert to YAML
FIRST-18 Milestone 5
FIRST-23 For each API endpoint, design ext
FIRST-19 Milestone 6
FIRST-24 UI screens for each API endpoint
FIRST-25 Final project report

We took advantage of the timeline feature in the JIRA software and generated a GANTT Chart. It gave us visual representation of tasks against time which helped us in keeping track of our timeline. Thanks to the software we could keep an easy track on start and end date of a certain task and keep the due dates in check as well as keep a track on the completed and pending tasks and the priority it must be taken care of. There is a comment section in every task as well where the members can comment and share their feedback.

Additional Screenshots from JIRA:



FIRST-6 Milestone 1
FIRST-5 Milestone 2
FIRST-4 Milestone 3
FIRST-16 scrum meeting M31
FIRST-17 scrum meeting M32
FIRST-7 Milestone 4
FIRST-20 create API endpoints.
FIRST-21 Description of API endpoints.
FIRST-22 convert to YAML
FIRST-18 Milestone 5
FIRST-23 For each API endpoint, design ext
FIRST-19 Milestone 6
FIRST-24 UI screens for each API endpoint
FIRST-25 Final project report
FIRST-27 backend structure and testing
FIRST-26 Frontend testing for the app



Summary	Status	Assignee	Start date	Due date
Milestone 1	DONE	DP Dhairyasheel Patil	Oct 6, 2023	Oct 13, 2023
Milestone 2	DONE	NC NISANKARA CHANDA...	Oct 14, 2023	Oct 27, 2023
Milestone 3	DONE	VS VARUN SUNDERARAJAN	Oct 27, 2023	Nov 3, 2023
scrum meeting M31	DONE	VS VARUN SUNDERARAJAN	Oct 11, 2023	Oct 11, 2023
scrum meeting M32	DONE	DP Dhairyasheel Patil	Oct 26, 2023	Oct 26, 2023
Milestone 4	DONE	NC NISANKARA CHANDA...	Nov 6, 2023	Nov 17, 2023
Milestone 5	DONE	SH Saquib Hashmi	Nov 18, 2023	Dec 1, 2023
For each API endpoint, design extensive test cases	DONE		Nov 18, 2023	Nov 30, 2023
Milestone 6	DONE	DK DHANYA K	Dec 2, 2023	Dec 15, 2023
UI screens for each API endpoint	DONE	DP Dhairyasheel Patil	Dec 2, 2023	Dec 8, 2023
Final project report	DONE	SH Saquib Hashmi	Dec 8, 2023	Dec 13, 2023
backend structure and testing	DONE	NC NISANKARA CHANDA...	Dec 2, 2023	Dec 15, 2023
Frontend testing for the app	DONE	DK DHANYA K	Dec 2, 2023	Dec 15, 2023

Design of Components

User Authentication and Authorization:

- This component allows both students and administrators to securely log in to their respective accounts.

- It manages user access permissions and roles, ensuring that each user sees only the appropriate information and features.

Student Profile Management:

Provides students with the ability to access and update their profiles, including personal details, academic goals, and domain preferences.

Learning Profile and Progress:

Allows students to access their learning profiles, view past scores and details of quizzes, and track their progress in completed courses.

Notifications and Reminders:

Sends notifications and reminders to students for important academic dates, such as registration deadlines, ensuring they stay organized and do not miss critical events.

Course Recommendation Engine:

Utilizes machine learning algorithms to generate personalized course recommendations for students based on their academic goals, interests, and input preferences.

Feedback and Rating System:

- Allows students to provide feedback on courses, rate them based on difficulty, and leave text-based reviews.
- Helps in fine-tuning the recommendation algorithm and provides valuable information to other students.

Data Anonymization:

- Ensures that all personal data collected by the system is anonymized to protect the privacy of students, as per their request

Admin Panel:

- ○ Provides administrators with secure access to the admin panel for managing the system's administrative functions.
- ○ Provides admin with access to analytics, including negative feedback, which aids in data-driven improvements to course content and structure.

Data Import and Analysis:

- ○ Allows administrators to upload enrolment data from previous terms in bulk format, which is then used for pattern analysis and to improve the recommendation algorithm.

Analytics and Reporting:

- Offers administrators the ability to view and export analytics on course enrolments, course popularity, and student feedback.
- Provides insights for curriculum improvement and reporting to academic advisors.

User Support and Monitoring:

- Supports administrators in monitoring student feedback and assisting students with account-related issues.
- Ensures code of conduct is maintained within the system.

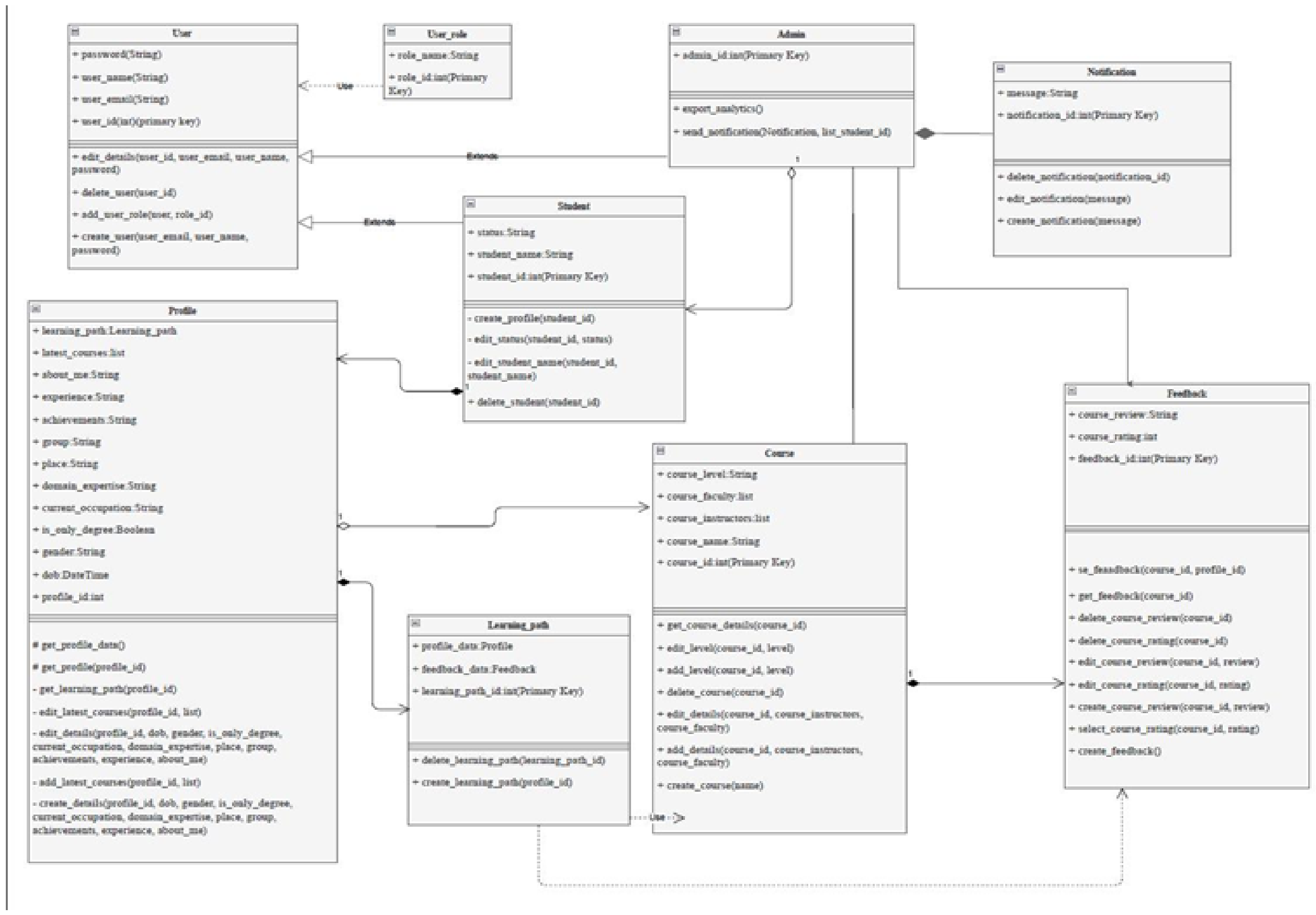
Data Access for Data Analysts:

Provides data analysts with access to the necessary data from the database to evaluate the correctness of the recommendation algorithm and make necessary changes.

IT Support Alerts:

Alerts IT support staff members about system errors or issues, enabling them to promptly resolve issues to ensure the system runs smoothly.

Class Diagram


User
+ password(String)
+ user_name(String)
+ user_email(String)
+ user_id(int)(primary key)
+ edit_details(user_id, user_email, user_name, password)
+ delete_user(user_id)
+ add_user_role(user, role_id)
+ create_user(user_email, user_name, password)
User_role
+ role_name:String
+ role_id:int(Primary Key)
Use
Admin
+ admin_id:int(Primary Key)
+ export_analytics()
+ send_notification(Notification, list_student_id)
Extends
Notification
+ message:String
+ notification_id:int(Primary Key)
+ delete_notification(notification_id)
+ edit_notification(message)
+ create_notification(message)
Student
+ status:String
+ student_name:String
+ student_id:int(Primary Key)
- create_profile(student_id)
- edit_status(student_id, status)
- edit_student_name(student_id, student_name)
+ delete_student(student_id)
Extends
Profile
+ learning_path:Learning_path
+ latest_courses:list
+ about_me:String
+ experience:String
+ achievements:String
+ group:String
+ place:String
+ domain_expertise:String
+ current_occupation:String
+ is_only_degree:Boolean
+ gender:String
+ dob:DateTime
+ profile_id:int
# get_profile_data()
# get_profile(profile_id)
- get_learning_path(profile_id)
- edit_latest_courses(profile_id, list)
- edit_details(profile_id, dob, gender, is_only_degree, current_occupation, domain_expertise, place, group, achievements, experience, about_me)
- add_latest_courses(profile_id, list)
- create_details(profile_id, dob, gender, is_only_degree, current_occupation, domain_expertise, place, group, achievements, experience, about_me)
Learning_path
+ profile_data:Profile
+ feedback_data:Feedback
+ learning_path_id:int(Primary Key)
+ delete_learning_path(learning_path_id)
+ create_learning_path(profile_id)
Use
Course
+ course_level:String
+ course_faculty:list
+ course_instructors:list
+ course_name:String
+ course_id:int(Primary Key)
+ get_course_details(course_id)
+ edit_level(course_id, level)
+ add_level(course_id, level)
+ delete_course(course_id)
+ edit_details(course_id, course_instructors, course_faculty)
+ add_details(course_id, course_instructors, course_faculty)
+ create_course(name)
Feedback
+ course_review:String
+ course_rating:int
+ feedback_id:int(Primary Key)
+ se_feedback(course_id, profile_id)
+ get_feedback(course_id)
+ delete_course_review(course_id)
+ delete_course_rating(course_id)
+ edit_course_review(course_id, review)
+ edit_course_rating(course_id, rating)
+ create_course_review(course_id, review)
+ select_course_rating(course_id, rating)
+ create_feedback()
1
1
1
1
1

<u>SCRUM Meetings Schedule and Minutes</u>

<u>Meeting 1</u> - 29th September - 19:00

Introduction of the group members, We went through the project statement and had a discussion on the overall idea of the project. We discussed the first live session and what we understood from the problem statement. Discussed our strengths as to which tasks should be assigned to whom.

<u>Meeting 2</u> - 3rd October - 16:30

Collaborated on github, discussed the live session 2 and shared our ideas as to what we individually understood about the project and decided to meet after watching all the lectures till week 2. Decided to come up with our own versions of users for milestone 1 before meeting next.

Meeting 3 - 9th October - 20:00

Discussed our individual versions Users, collectively compared and finalized our Users. Decided to individually work on user stories and then compare it in the next meeting.

Meeting 4 - 12th October - 08:00

Compared and finalized all the user stories. Made the final pdf for submission of milestone-1. Rechecked the pdf and successfully submitted milestone1 along with the honor code pdf.

Meeting 5 - 16th October - 21:00

Short meeting on discussion of milestone 2 requirements. Went through the statement and tried to understand what exactly needs to be done. Decided to meet next after completing the required theory study for the same.

Meeting 6 - 21st October - 16:00

Discussed on milestone 2, divided wireframe and storyboard work in 2 groups. Brainstormed on both and came up with ideas for same. Went through ideas and softwares where we can make both of them.

Meeting 7 - 26th October - 19:00

Our basic Wireframe and storyboard was ready, and made small changes which were required. Finalized everything. And then made the submission after going through everything.

Meeting 8 - 30th October - 20:00

Went through Milestone-3 requirements. Discussed what needs to be done. Brainstormed on Component list and class diagram. Divided the work as per expertise and decided to meet after completion of individual work.

Meeting 9 - 2nd November - 20:30

Discussed on the progress of the milestone 3 requirements. Finalized all the requirements, took screenshots from jira, collaborated all the individual tasks in a single file. Finalized the final submission pdf with all the required images and details.

MILESTONE 4 SUBMISSION: YAML file only, no PDF available!

Kindly check the Github repository for the submission.

MILESTONE 5 SUBMISSIONS FOLLOW

Descriptions of relevant API tests are given below.

Page being tested: http://127.0.0.1:8080/login/
Inputs:
- ➢ **Request Method:** POST
- ➢ **JSON:** {"username":"john123","password":"pass123"}

Expected Output:
- ➢ **HTTP Status Code:** 200

Actual Output:
- ➢ **HTTP Status Code:** 200

Result: Success

```python
def test_login_success():
  url = base_url + "login"
  payload = {"username":"john123","password":"pass123"}
  response = requests.post(url, json=payload)
  assert response.status_code == 200
```

Page being tested: http://127.0.0.1:8080/login
Inputs:
- ➢ **Request Method:** POST
- ➢ **JSON:** {"username":"john123","password":"wrongpass")

Expected Output:
- ➢ **HTTP Status Code:** 401
- ➢ **JSON:** "Invalid credentials"

Actual Output:
- ➢ **HTTP Status Code:** 401
- ➢ **JSON:** "Invalid credentials"

Result: Success

```python
def test_login_failure():
  url = base_url + "login"
  payload = {"username":"john123","password":"wrongpass"}
  response = requests.post(url, json=payload)
  assert response.status_code == 401
  assert response.json() == "Incorrect login"
```

Page being tested: http://127.0.0.1:8080/profile
Inputs:

- **Request Method:** GET
- **Header:** {"authToken":"123abc"}

Expected Output:

- **HTTP Status Code:** 200

Actual Output:

- **HTTP Status Code:** 200

Result: Success

```python
def test_get_student_profile():
  url = base_url + "profile"
  headers = {"authToken":"123abc"}
  response = requests.get(url, headers=headers)
  assert response.status_code == 200
  assert "profile_id" in response.json()
```

Page being tested: http://127.0.0.1:8080/profile
Inputs:

- **Request Method:** GET
- **Header:** {"authToken":"123abc"}

Expected Output:

- **HTTP Status Code:** 401
- **JSON:** "Invalid credentials"

Actual Output:

- **HTTP Status Code:** 401
- **JSON:** "Invalid credentials"

Result: Success

```python
def test_get_profile_unauthorized():
  url = base_url + "profile"
  headers = {"authToken":"invalidtoken"}
  response = requests.get(url, headers=headers)
  assert response.status_code == 401
  assert response.json() == "Invalid credentials"
```

Page being tested: http://127.0.0.1:8080/recommendations
Inputs:
- ➢ **Request Method:** GET
- ➢ **Header:** `{"authToken":"456def"}`

Expected Output:
- ➢ **HTTP Status Code:** 200

Actual Output:
- ➢ **HTTP Status Code:** 200

Result: Success

```python
def test_get_recommendations_success():
  url = base_url + "recommendations"
  headers = {"authToken":"456def"}
  response = requests.get(url, headers=headers)
  assert response.status_code == 200
  assert type(response.json()) is list
```

Page being tested: http://127.0.0.1:8080/recommendations
Inputs:
- ➢ **Request Method:** GET
- ➢ **Header:** `{"authToken":"invalid"}`

Expected Output:
- ➢ **HTTP Status Code:** 401
- ➢ **JSON:** `"Invalid credentials"`

Actual Output:
- ➢ **HTTP Status Code:** 401
- ➢ **JSON:** `"Invalid credentials"`

Result: Success

```python
def test_get_recommendations_unauthorized():
  url = base_url + "recommendations"
  headers = {"authToken":"invalid"}
  response = requests.get(url, headers=headers)
  assert response.status_code == 401
  assert response.json() == "Invalid credentials"
```

Page being tested: http://127.0.0.1:8080/feedback/{courseId)}
Inputs:

- **Request Method:** POST
- **Header:** {"authToken":"123abc"}
- **JSON:** {"rating":4,"comment":"Good course"}

Expected Output:

- **HTTP Status Code:** 200

Actual Output:

- **HTTP Status Code:** 200

Result: Success

```python
def test_submit_feedback():
  url = base_url + "feedback/5252"
  headers = {"authToken":"123abc"}
  payload = {"rating":4,"comment":"Good course"}
  response = requests.post(url, json=payload, headers=headers)
  assert response.status_code == 200
```

Page being tested: http://127.0.0.1:8080/feedback/{courseId)}
Inputs:

- **Request Method:** POST
- **Header:** {"authToken":"123abc"}
- **JSON:** {"rating":4,"comment":"Good course"}

Expected Output:

- **HTTP Status Code:** 404
- **JSON:** "Course not found!"

Actual Output:

- **HTTP Status Code:** 404
- **JSON:** "Course not found!"

Result: Success

```python
def test_submit_feedback_not_found():
  url = base_url + "feedback/999"
  headers = {"authToken":"123abc"}
  payload = {"rating":4, "comment":"Good course"}
  response = requests.post(url, json=payload, headers=headers)
  assert response.status_code == 404
  assert response.json() == "Course not found!"
```

Page being tested: http://127.0.0.1:8080/data/anonymize
Inputs:

- ➢ **Request Method:** POST
- ➢ **Header:** `{"authToken":"123abc"}`

Expected Output:

- ➢ **HTTP Status Code:** 200

Actual Output:

- ➢ **HTTP Status Code:** 200

Result: Success

```
Python
def test_anonymize_data_success():
  url = base_url + "data/anonymize"
  headers = {"authToken":"123abc"}
  response = requests.post(url, headers=headers)
  assert response.status_code == 200
```

Page being tested: http://127.0.0.1:8080/data/anonymize
Inputs:

- ➢ **Request Method:** POST
- ➢ **Header:** `{"authToken":"123abc"}`

Expected Output:

- ➢ **HTTP Status Code:** 401
- ➢ **JSON:** `"Invalid credentials"`

Actual Output:

- ➢ **HTTP Status Code:** 401
- ➢ **JSON:** `"Invalid credentials"`

Result: Success

```
Python
def test_anonymize_data_unauthorized():
  url = base_url + "data/anonymize"
  headers = {"authToken":"invalid"}
  response = requests.post(url, headers=headers)
  assert response.status_code == 401
  assert response.json() == "Invalid credentials"
```

Page being tested: http://127.0.0.1:8080/analytics

Inputs:

- **Request Method:** GET
- **Header:** `{"authToken":"admin123"}`

Expected Output:

- **HTTP Status Code:** 200

Actual Output:

- **HTTP Status Code:** 200

Result: Success

```python
def test_get_analytics_success():
  url = base_url + "analytics"
  headers = {"authToken":"admin123"}
  response = requests.get(url, headers=headers)
  assert response.status_code == 200
  assert type(response.json()) is dict
```

Page being tested: http://127.0.0.1:8080/analytics

Inputs:

- **Request Method:** GET
- **Header:** `{"authToken":"invalid"}`

Expected Output:

- **HTTP Status Code:** 401
- **JSON:** "Invalid credentials"

Actual Output:

- **HTTP Status Code:** 401
- **JSON:** "Invalid credentials"

Result: Success

```python
def test_get_analytics_unauthorized():
  url = base_url + "analytics"
  headers = {"authToken":"invalid"}
  response = requests.get(url, headers=headers)
  assert response.status_code == 401
  assert response.json() == "Invalid credentials"
```

Technologies and Tools Used: Overall Development Process

● **Git and Github:** As asked in the problem statement, git was used as the very basis for the entire project. Github repositories were used right from the very beginning, and milestone submissions were made therein. The steps involved in git usage became second-nature to us by the end of the project.

● **Postman:** Postman was the go-to tool that we used for testing out API endpoints while building them. The easy-to-use and feature-rich nature of Postman meant that we didn't have to face much trouble in testing and debugging our APIs.

● **Visual Studio Code:** It was the IDE of choice for all of us. It is quite powerful and its extensions were very useful.

● **Swagger OAS Editor:** We've used Swagger to create and edit our API specification YAML document.

● **Firefox, Safari, and Chrome**:We used these browsers and their inbuilt developer consoles and extensions while coding

● **DB Browser for SQLite:** It was used to manage the backend SQLite database during the development and testing process.

Technologies and Tools Used: Frontend

Vue.js 3: Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications. It was used to create a functional application around our wireframes.

Bootstrap: Bootstrap was used throughout the entire app, styling purposes. Various Bootstrap elements, such as buttons, tables, cards, etc. have been used.
HTML/CSS/JS: The basic tech stack for web pages

Technologies and Tools Used: Backend

● **Python:** The programming language that brought it all together. Almost 100% of the submissions as part of our repository is Python code. Python-based frameworks like Flask were the basis for the application. Various Python libraries, such as uuid, datetime, humanize, etc. were used.

● **Flask:** As mentioned in the problem statement, the Flask web framework was used for the backend server. Various Flask sub-components, such as Flask-Security-Too (for the authentication mechanisms), Flask-RESTful (for the API), Flask-CORS (for enabling cross-origin requests), etc. have been used. ● SQLite: It was used as the database for storing the app data at the backend. Various tables corresponding to the users, tickets, articles (FAQs), notifications, etc. were created

● **SQLAlchemy:** Flask-SQLAlchemy was used as the ORM to enable the interactions between the app at the backend Flask API and the SQLite database. The various data models were made using the appropriate classes in SQLAlchemy.

Where is our application hosted?

Our application (i.e the backend server itself) isn't hosted on the cloud. It runs on localhost, on port 8000. The codebase itself is hosted on Github, though.

Instructions to run the application .

Backend:-

````
# Local Setup
- Clone the project
- Run `setup.sh`

# Local Development Run
- `local_run.sh` It will start the flask app in `development`. Suited for
local development

# Replit run
- Go to shell and run
    `pip install --upgrade poetry`
- Click on `main.py` and click button run
- Sample project is at https://replit.com/@thejeshgn/flask-template-app
- The web app will be availabe at
https://flask-template-app.thejeshgn.repl.co
- Format https://<replname>.<username>.repl.co
````

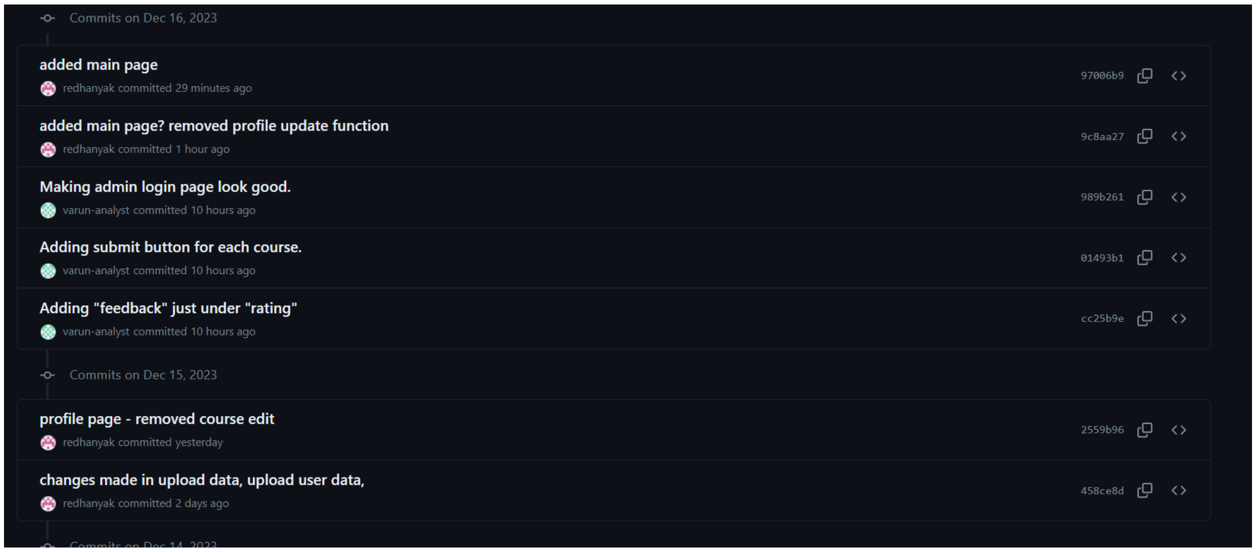
Frontend:-

~~~
# frontend

## Project setup
```
npm install
```

### Compiles and hot-reloads for development
```
npm run serve
```

### Compiles and minifies for production
```
npm run build
```

### Lints and fixes files
~~~

```
```
npm run lint
```

### Customize configuration
See [Configuration Reference](https://cli.vuejs.org/config/).
```

Code Review & Issue Tracking Screenshots

The following screenshots demonstrate the various code review, issue reporting, and issue tracking actions that we've undertaken. Specifically, we have used tools such as Trello, Google Spaces, Github Code Review, and Google Meet for our needs in this regard.



Commits on Dec 16, 2023

added main page
redhanyak committed 29 minutes ago 97006b9

added main page? removed profile update function
redhanyak committed 1 hour ago 9c8aa27

Making admin login page look good.
varun-analyst committed 10 hours ago 989b261

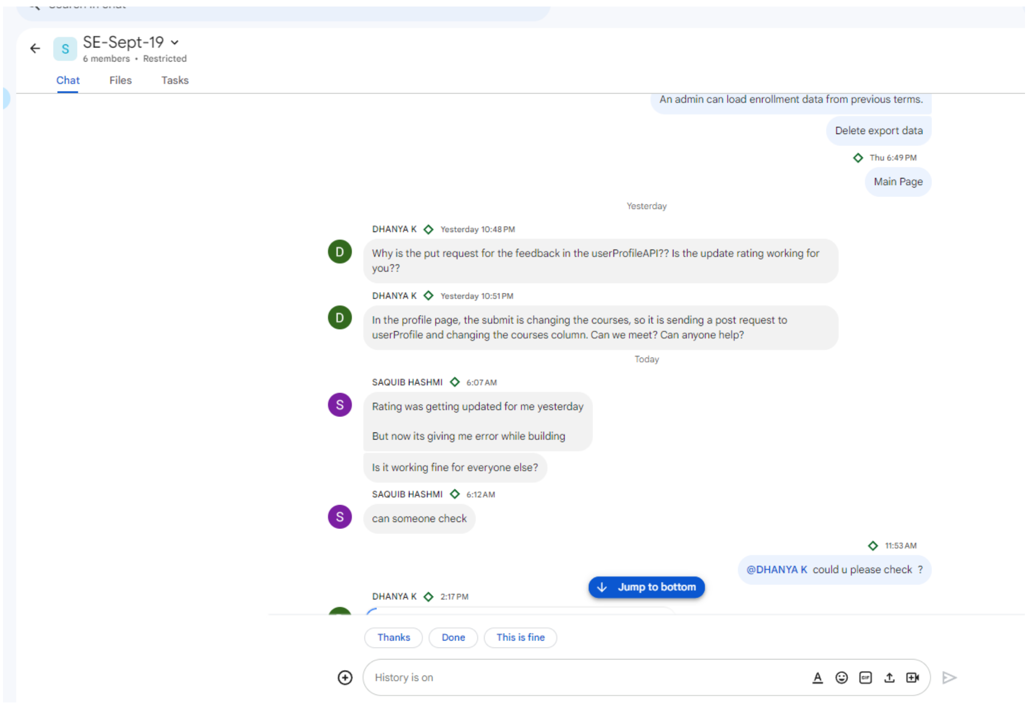
Adding submit button for each course.
varun-analyst committed 10 hours ago 01493b1

Adding "feedback" just under "rating"
varun-analyst committed 10 hours ago cc25b9e

Commits on Dec 15, 2023

profile page - removed course edit
redhanyak committed yesterday 2559b96

changes made in upload data, upload user data,
redhanyak committed 2 days ago 458ce8d



SE-Sept-19
6 members • Restricted

An admin can load enrollment data from previous terms.

Delete export data

Thu 6:49 PM

Main Page

Yesterday

DHANYA K Yesterday 10:48 PM
Why is the put request for the feedback in the userProfileAPI?? Is the update rating working for you??

DHANYA K Yesterday 10:51 PM
In the profile page, the submit is changing the courses, so it is sending a post request to userProfile and changing the courses column. Can we meet? Can anyone help?

Today

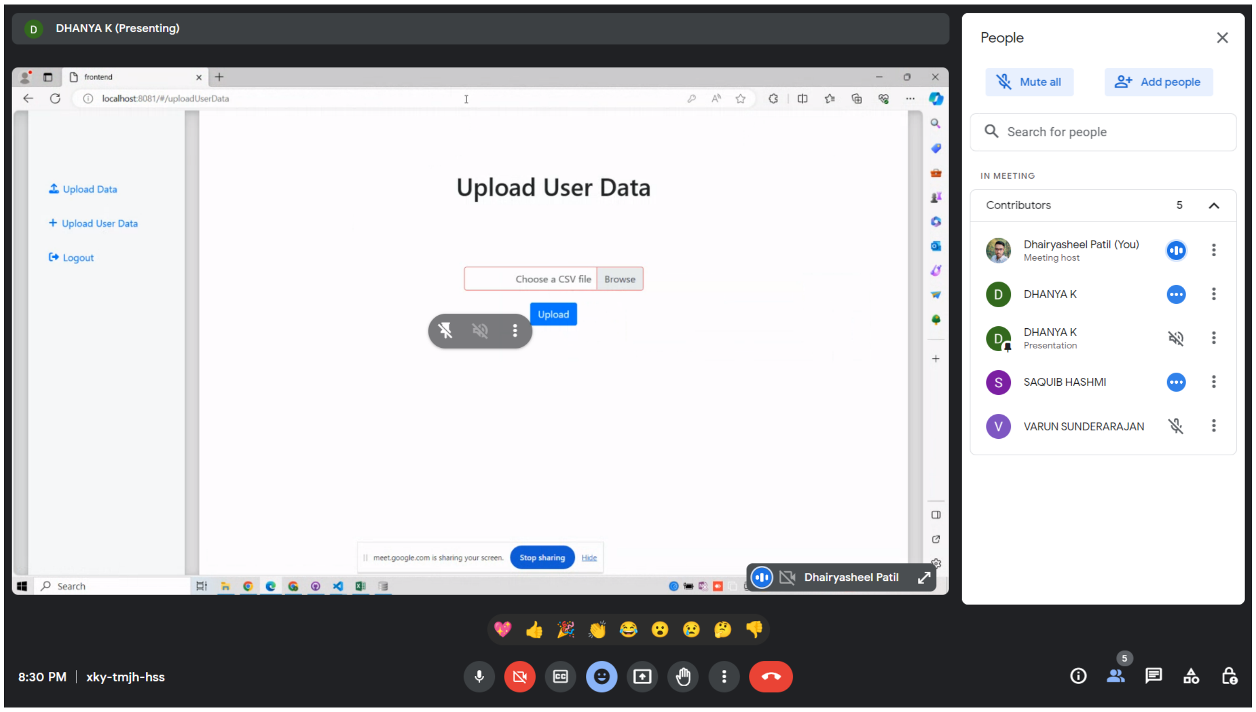
SAQUIB HASHMI 6:07 AM
Rating was getting updated for me yesterday
But now its giving me error while building
Is it working fine for everyone else?

SAQUIB HASHMI 6:12 AM
can someone check

11:53 AM
@DHANYA K could u please check ?

DHANYA K 2:17 PM

An issue being discussed on Google Space



One of the Scrum meetings being held on Google Meet.