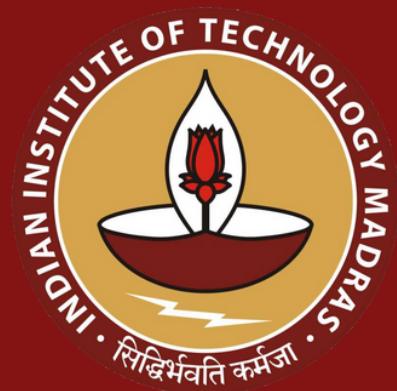


Software Engineering

Project



TEAM 16

Prepared By:

Anish Maity (21f3000417)
Abhishek Darji (22f1000678)
Achin Aggarwal (22f1001390)
Sarath Sasidharan Pillai (22f1000152)
Atharva Sarbhukan (22f1000533)
Rituparno Sen (21f1004275)
Dr. Ambrish Naik (22f2000424)



SCHEDULING AND DESIGN

TABLE OF CONTENTS

- 01 PROJECT SCHEDULE**
- 02 PROJECT SCHEDULING TOOL**
- 03 SCRUM MEETINGS - MINUTES/DETAILS**
- 04 COMPONENT DESIGN - STUDENT & ADMIN**
- 05 SOFTWARE DESIGN - DB SCHEMA & CLASS DIAGRAM**
- 06 UI PAGES**

PROJECT SCHEDULE

1. Sprint Schedule

Sprint No.	Topic/Task	Timeline
Sprint 1	Identifying User & Requirements	14 Jan, 2024 - 28 Jan, 2024
Sprint 2	User Interfaces	29 Jan, 2024 - 9 Feb, 2024
Sprint 3	Scheduling and Design	10 Feb, 2024 - 20 Feb, 2024
Sprint 4	API Endpoints	24 Feb, 2024 - 2 Mar, 2024
Sprint 5	Testing All the Test Cases	3 Mar, 2024 - 12 Mar, 2024
Sprint 6	Integrate Frontend & Backend	19 Mar, 2024 - 30 Mar, 2024

2. About the Sprints

- **Sprint 1 :- Identifying User & Requirements**

Identify and categorize users (primary, secondary, and tertiary) by analyzing their roles and interactions within the system. Gather inputs from key stakeholders to ensure comprehensive requirement collection.

Develop detailed user stories for new features and integrations, following SMART guidelines to ensure clarity, feasibility, and alignment with project goals. Complete Milestone 1 Report Vetting and Submission.

Dates: 14 Jan, 2024 - 28 Jan, 2024

- **Sprint 2 :- User Interfaces**

Develop a storyboard to illustrate the user journey and create low-fidelity wireframes for user stories, applying usability guidelines and heuristics, and complete Milestone 2 Report Vetting and Submission.

Dates: 29 Jan, 2024 - 9 Feb, 2024

- **Sprint 3 :- Scheduling and Design**

Develop a project schedule, including sprint timelines, scrum meeting plans, task distribution, and Trello Board, or Gantt charts using Jira tool.

Design system components, create basic class diagrams, develop most UI pages, and document scrum meeting details. Complete Milestone 3 Report Vetting and Submission.

Dates: 10 Feb, 2024 - 20 Feb, 2024

- **Sprint 4 :- API Endpoints**

Design and implement API architecture by creating new endpoints and integrating relevant APIs based on user stories.

Document API details, including integrated and custom-built endpoints, and submit YAML specifications along with implementation code. Complete Milestone 4 Report Vetting and Submission.

Dates: 24 Feb, 2024 - 2 Mar, 2024

- **Sprint 5 :- Testing All the Test Cases**

Design comprehensive test cases for each API endpoint, covering inputs, expected outputs, actual results, and success/failure validation.

Perform unit testing to ensure functionality, performance, and reliability of the system. Complete Milestone 5 Report Vetting and Submission.

Dates: 3 Mar, 2024 - 12 Mar, 2024

- **Sprint 6 :- Integrate Frontend & Backend**

Combine frontend and backend components into a cohesive system. Complete Milestone 6 Report Vetting and Submission.

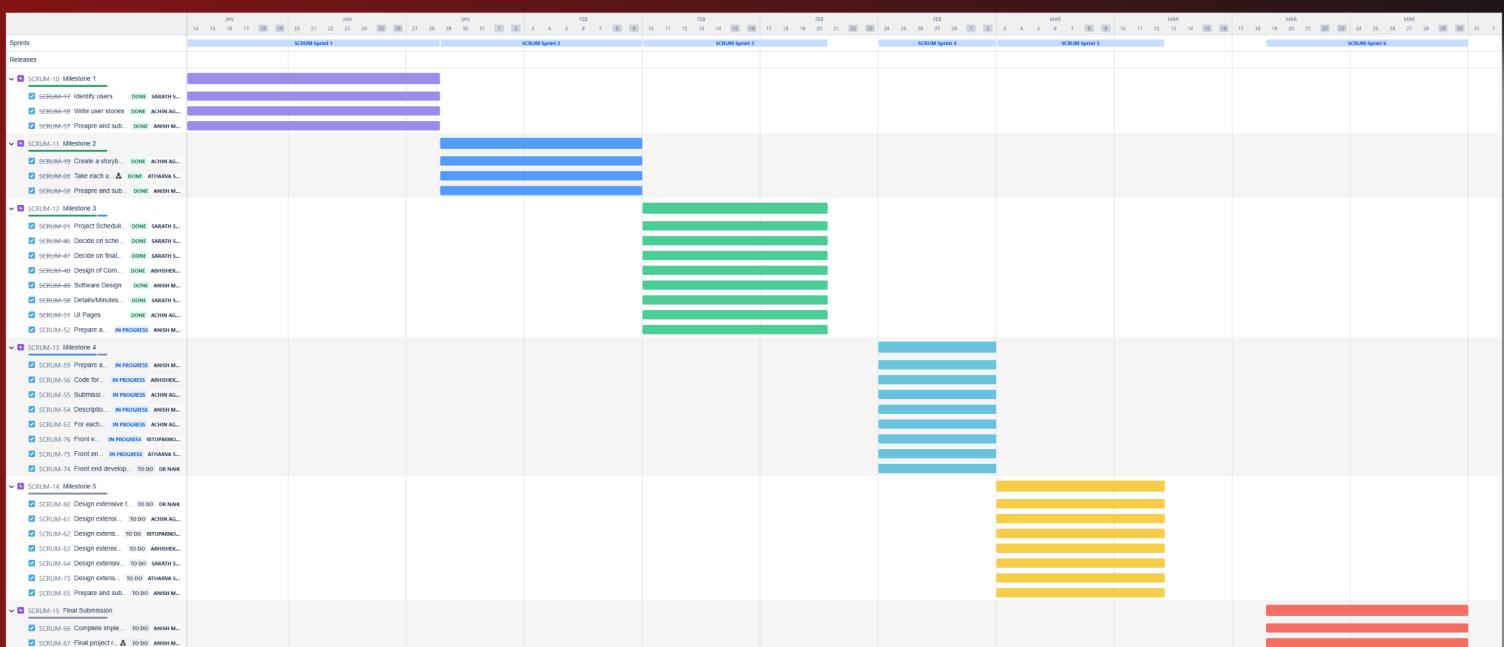
Dates: 19 Mar, 2024 - 30 Mar, 2024

SCRUM BOARD

All sprints

Search AA AM SP AS AD RS +2 Epic Sprint

TO DO 10	IN PROGRESS 8	DONE 13
Front end development- Dashboard MILESTONE 4 SCRUM-74	Prepare and submit Milestone 4 report MILESTONE 4 SCRUM-59	Identify users MILESTONE 1 SCRUM-17
Design extensive test cases for API 1-5 MILESTONE 5 SCRUM-60	Code for the chat bot APIs (implementation) MILESTONE 4 SCRUM-56	Write user stories MILESTONE 1 SCRUM-18
Design extensive test cases for API 6-10 MILESTONE 5 SCRUM-61	Submission of YAML (for the APIs created by dev-team) MILESTONE 4 SCRUM-55	Preapre and submit Milestone 1 report MILESTONE 1 SCRUM-57
Design extensive test cases for API 11-15 MILESTONE 5 SCRUM-62	Description of API endpoints. (As per the problem statement) MILESTONE 4 SCRUM-54	Create a storyboard MILESTONE 2 SCRUM-19
Design extensive test cases for API 16-20 MILESTONE 5 SCRUM-63	For each user story, create new API endpoints or use appropriate API endpoints from libraries MILESTONE 4 SCRUM-53	Take each user story and create low-fidelity wireframes MILESTONE 2 SCRUM-20
Design extensive test cases for API 21-25 MILESTONE 5 SCRUM-64	Prepare and submit milestone 3 PDF Report MILESTONE 3 SCRUM-52	Preapre and submit Milestone 2 report MILESTONE 2 SCRUM-58
Design extensive test cases for API 26 onwards MILESTONE 5 SCRUM-73	Front end development- Connection MILESTONE 4 SCRUM-76	Project Scheduling MILESTONE 3 SCRUM-21
Prepare and submit Milestone 5 report MILESTONE 5 SCRUM-65	Front end development- Chat bot MILESTONE 4 SCRUM-75	Decide on schedules MILESTONE 3 SCRUM-46
Complete implementation along with a working prototype. FINAL SUBMISSION SCRUM-66		Decide on final sceduling tool MILESTONE 3 SCRUM-47
Final project report (consistent with intermediate milestone documents). FINAL SUBMISSION SCRUM-67		Design of Components MILESTONE 3 SCRUM-48
+ Create issue		Software Design MILESTONE 3 SCRUM-49
		Details/Minutes of a few scrum meetings MILESTONE 3 SCRUM-50
		UI Pages MILESTONE 3 SCRUM-51



Project Scheduling

Tool - Jira

The screenshot displays a Jira project management interface with six sprints:

- SCRUM Sprint 1 (14 Jan – 28 Jan):** Contains 3 issues. All tasks are marked as "DONE".
 - SCRUM-17: Identify users (Milestone 1)
 - SCRUM-18: Write user stories (Milestone 1)
 - SCRUM-19: Preapre and submit Milestone 1 report (Milestone 1)
- SCRUM Sprint 2 (29 Jan – 9 Feb):** Contains 3 issues. All tasks are marked as "DONE".
 - SCRUM-19: Create a storyboard (Milestone 2)
 - SCRUM-20: Take each user story and create low-fidelity wireframes (Milestone 2)
 - SCRUM-50: Preapre and submit Milestone 2 report (Milestone 2)
- SCRUM Sprint 3 (10 Feb – 20 Feb):** Contains 8 issues. Most tasks are marked as "DONE", except for one which is "IN PROGRESS".
 - SCRUM-21: Project Scheduling (Milestone 3)
 - SCRUM-46: Decide on schedules (Milestone 3)
 - SCRUM-47: Decide on final sceduling tool (Milestone 3)
 - SCRUM-48: Design of Components (Milestone 3)
 - SCRUM-49: Software Design (Milestone 3)
 - SCRUM-50: Details/Minutes of a few scrum meetings (Milestone 3)
 - SCRUM-51: UI Pages (Milestone 3)
 - SCRUM-52: Prepare and submit milestone 3 PDF Report (Milestone 3)
- SCRUM Sprint 4 (24 Feb – 2 Mar):** Contains 8 issues. Most tasks are marked as "IN PROGRESS", except for one which is "TO DO".
 - SCRUM-59: Prepare and submit Milestone 4 report (Milestone 4)
 - SCRUM-56: Code for the chat bot APIs (implementation) (Milestone 4)
 - SCRUM-55: Submission of YAML (for the APIs created by dev-team) (Milestone 4)
 - SCRUM-54: Description of API endpoints. (As per the problem statement) (Milestone 4)
 - SCRUM-53: For each user story, create new API endpoints or use appropriate API endpoints from libraries (Milestone 4)
 - SCRUM-76: Front end development- Connection (Milestone 4)
 - SCRUM-75: Front end development- Chat bot (Milestone 4)
 - SCRUM-74: Front end development- Dashboard (Milestone 4)
- SCRUM Sprint 5 (3 Mar – 12 Mar):** Contains 7 issues. Most tasks are marked as "TO DO", except for one which is "IN PROGRESS".
 - SCRUM-60: Design extensive test cases for API 1-5 (Milestone 5)
 - SCRUM-61: Design extensive test cases for API 6-10 (Milestone 5)
 - SCRUM-62: Design extensive test cases for API 11-15 (Milestone 5)
 - SCRUM-63: Design extensive test cases for API 16-20 (Milestone 5)
 - SCRUM-64: Design extensive test cases for API 21-25 (Milestone 5)
 - SCRUM-73: Design extensive test cases for API 26 onwards (Milestone 5)
 - SCRUM-65: Prepare and submit Milestone 5 report (Milestone 5)
- SCRUM Sprint 6 (19 Mar – 30 Mar):** Contains 2 issues. Both tasks are marked as "TO DO".
 - SCRUM-66: Complete implementation along with a working prototype. (Final Submission)
 - SCRUM-67: Final project report (consistent with intermediate milestone documents). (Final Submission)

Scrum Meeting

Minutes/Details

Scrum Meetings Schedule: Every Monday, Saturday (21:00 - 23:30 PM)

Mode: Google Meet

Attendees:

- Achin Aggarwal
- Anish Maity
- Atharva Sarbhukan
- Abhishek Darji
- Sarath Sasidharan Pillai
- Rituparno Sen
- Dr. Ambrish Naik

• Sprint 1 Scrum meetings minutes/details:

The team will focus on identifying and defining different types of users (primary, secondary, and tertiary) based on their roles and interactions within the system. Each team member will contribute by gathering insights and inputs from key stakeholders to ensure a thorough requirement collection.

Once the users are identified, the team will divide tasks to create detailed user stories for the new features and integrations, following SMART guidelines for clarity and feasibility.

In the meeting, we will collaboratively review and finalize the user stories, ensuring alignment with the project goals. Additionally, all members will participate in reviewing the Milestone 1 Report. Once the report is finalized, the team will complete the vetting process and proceed with the final submission.

Dates: 14 Jan, 2024 - 28 Jan, 2024

• Sprint 2 Scrum meetings minutes/details:

The team will collaborate to develop a storyboard illustrating the user journey and create low-fidelity wireframes for each user story. Each member will contribute ideas and suggestions to ensure the wireframes align with usability guidelines and heuristics.

A discussion will take place to review the storyboard and wireframes, where feedback will be provided for improvements. The team will finalize the designs based on the discussion and ensure they meet project requirements.

After the final review, the Milestone 2 Report will be vetted and submitted.

Dates: 29 Jan, 2024 - 9 Feb, 2024

- **Sprint 3 Scrum meetings minutes/details:**

The team will collaborate to finalize the project schedule, including sprint timelines, scrum meeting plans, and task distribution. Project management tools like Jira, Trello, or Gantt charts will be used to track progress effectively.

A discussion will be held to design key system components and create basic class diagrams to define system structure. Additionally, most UI pages will be developed, ensuring alignment with project requirements.

After the final review, the Milestone 3 Report will be vetted and submitted.

Dates: 10 Feb, 2024 - 20 Feb, 2024

- **Sprint 4 Scrum meetings minutes/details:**

The team will collaborate to design and implement the API architecture, ensuring that new endpoints are created and relevant APIs are integrated based on user stories. Each team member will contribute to defining the API structure and functionality.

A discussion will take place to document API details, including integrated and custom-built endpoints, along with YAML specifications. The team will then proceed with the development and deployment of the designed APIs.

Following the final review, the Milestone 4 Report will be vetted and submitted.

Dates: 24 Feb, 2024 - 2 Mar, 2024

- **Sprint 5 Scrum meetings minutes/details:**

The team will collaborate to design and document comprehensive test cases for each API endpoint, ensuring functionality, performance, and reliability. Each test case will include inputs, expected outputs, actual results, and success/failure validation.

Unit testing will be performed to verify the quality and robustness of the system. Any identified issues will be addressed through debugging and re-testing.

After the final validation, the Milestone 5 Report will be vetted and submitted.

Dates: 3 Mar, 2024 - 12 Mar, 2024

- **Sprint 6 Scrum meetings minutes/details:**

The team will collaborate to integrate the frontend and backend components, ensuring seamless communication between the UI and the API endpoints. Each module will be tested to verify proper data flow and functionality.

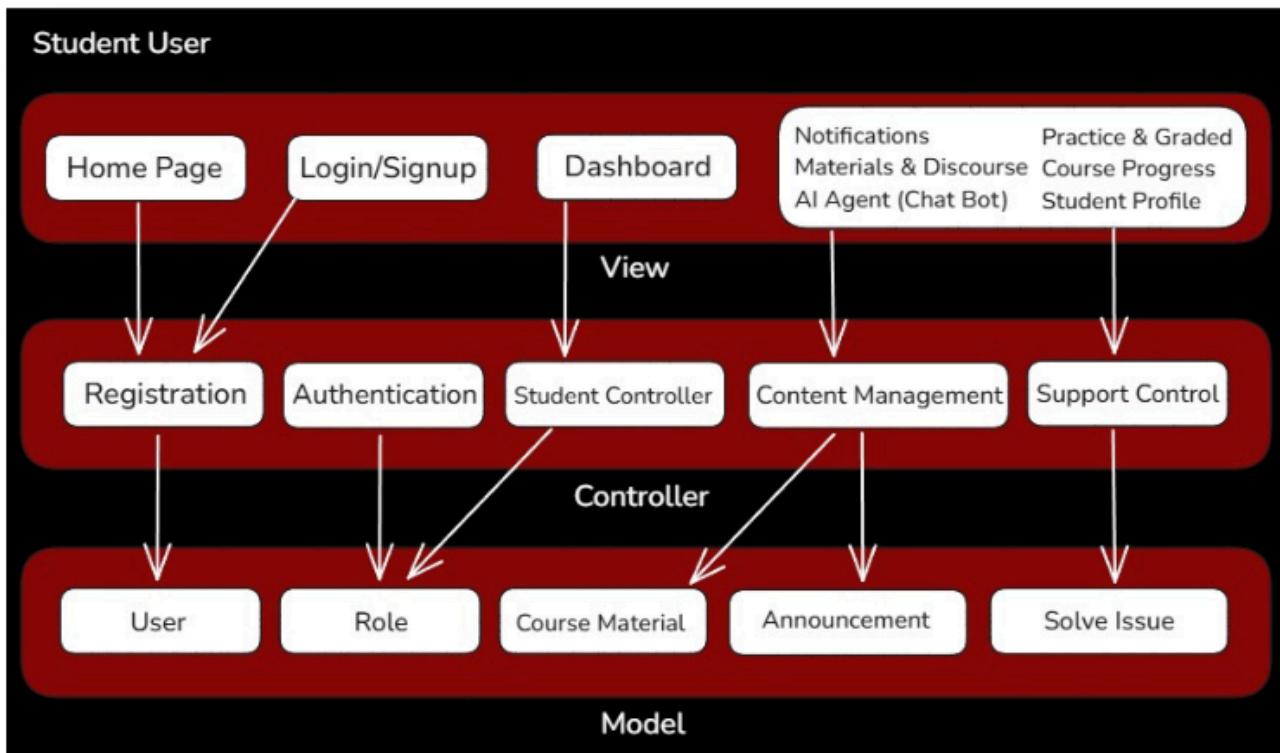
A discussion will be held to address any integration issues and optimize performance for a smooth user experience. Necessary debugging and refinements will be made to ensure stability.

Following successful integration, the Milestone 6 Report will be vetted and submitted.

Dates: 19 Mar, 2024 - 30 Mar, 2024

COMPONENT DESIGN

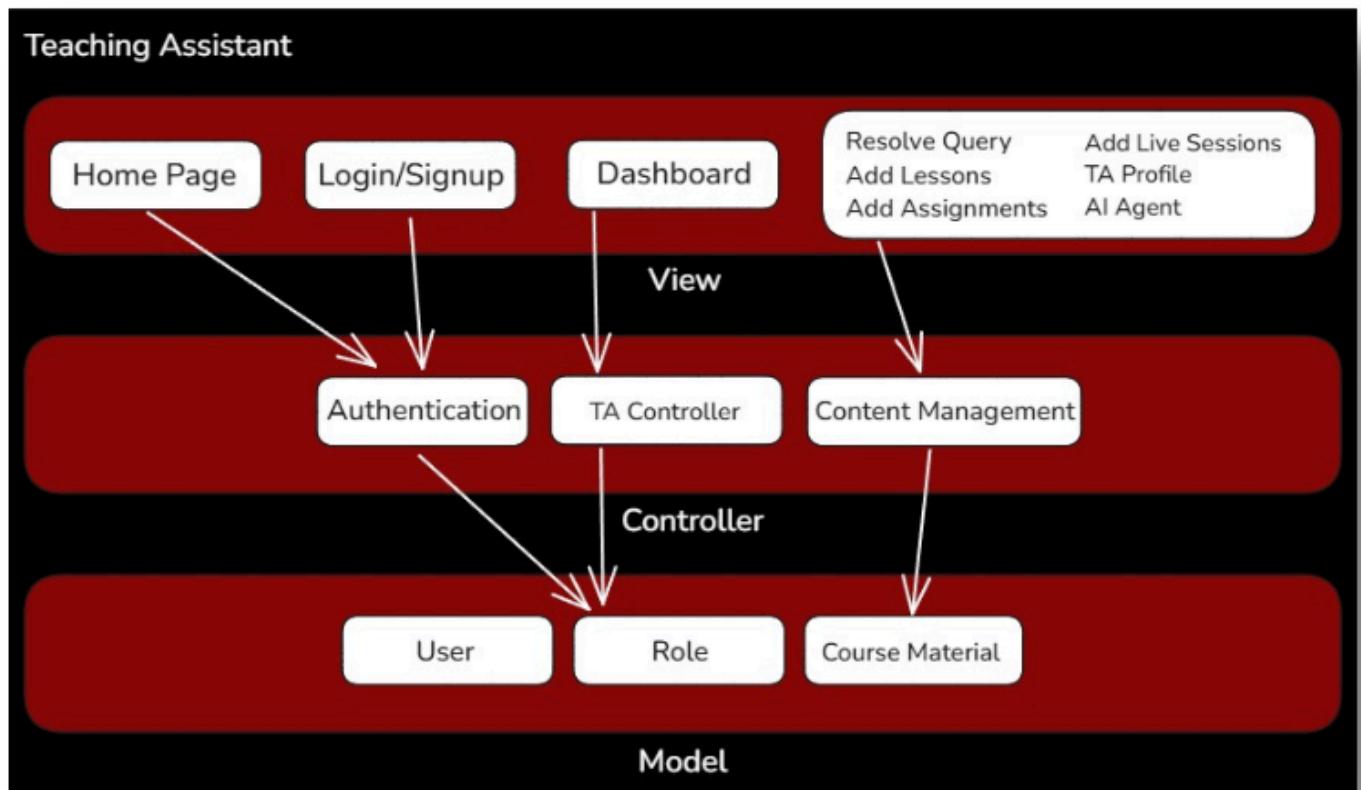
1. Student Component



- **Home Page:** Acts as the entry point, connecting to Registration and Authentication controllers for access management.
- **Login/Signup:** Handles user authentication via the Authentication Controller, interacting with the User and Role models.
- **Dashboard:** Displays student-related data, managed by the Student Controller and fetching information from Course Material and User models.
- **Registration & Authentication:** Manages user sign-ups and logins, ensuring security through the User and Role models.
- **Student Controller:** Oversees student interactions, retrieving study resources and tracking progress from the Course Material model.
- **Content Management:** Handles learning materials, discussions, and announcements, updating the Announcement model as needed.
- **Support Control:** Manages issue reporting and resolution by interacting with the Solve Issue model.

COMPONENT DESIGN

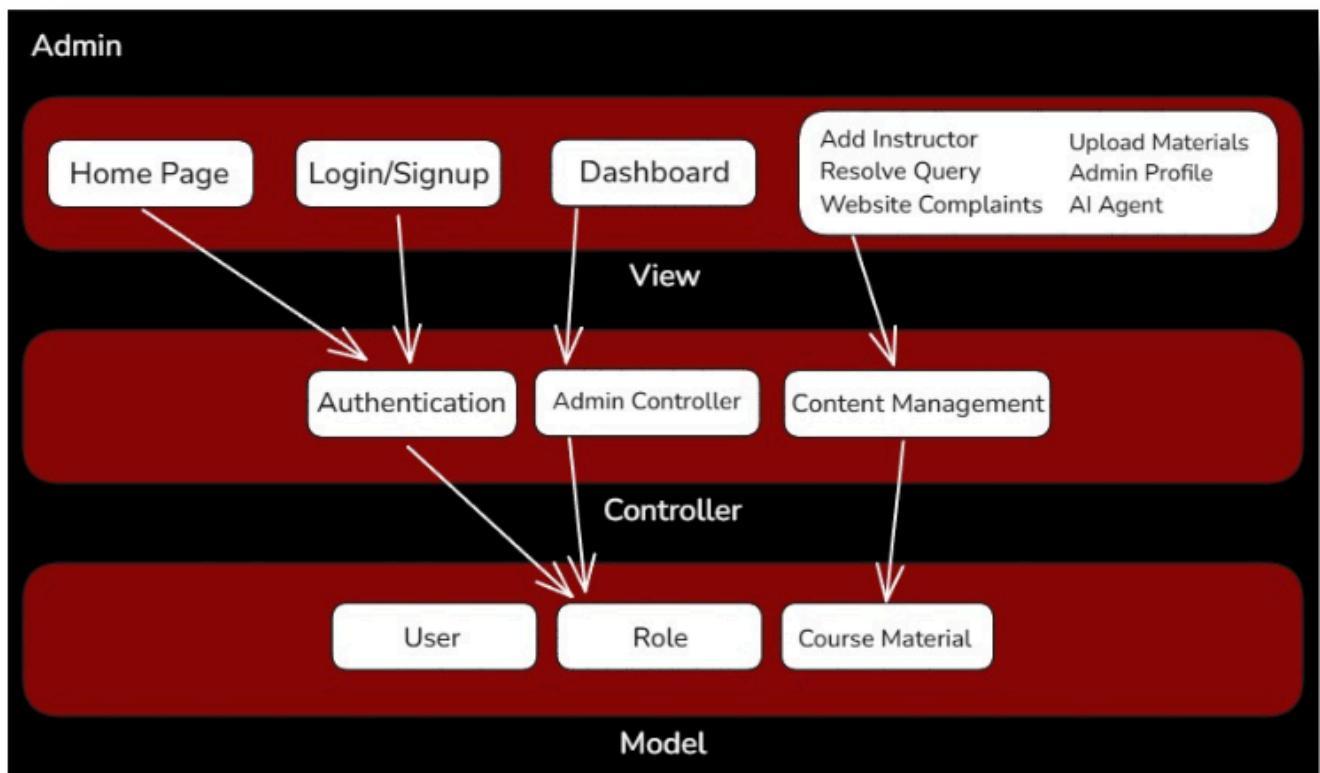
2. Teaching Assistant Component



- **Home Page:** Serves as the entry point, directing users to authentication and dashboard functionalities.
- **Login/Signup:** Manages TA authentication through the Authentication Controller, interacting with the User and Role models.
- **Dashboard:** Provides access to query resolution, lesson creation, assignments, live sessions, and TA profile via the TA Controller.
- **Authentication:** Ensures secure login and role-based access control using the User and Role models.
- **TA Controller:** Manages TA-specific actions such as resolving queries, adding lessons, and handling assignments.
- **Content Management:** Oversees course materials, assignments, and live sessions, interacting with the Course Material model.

COMPONENT DESIGN

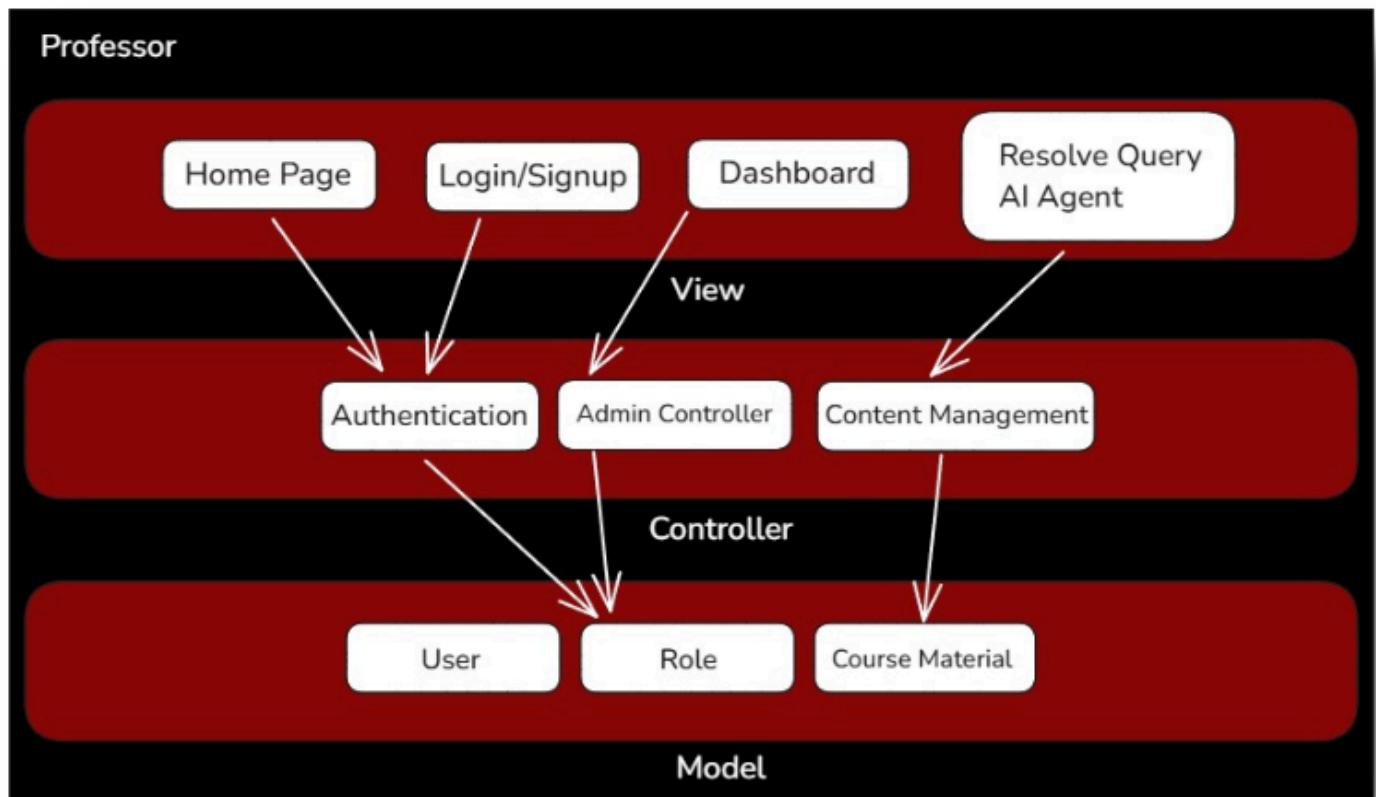
3. Admin Component



- **Home Page:** Entry point directing admins to authentication.
- **Login/Signup:** Manages secure admin authentication and access control.
- **Dashboard:** Provides an overview of admin functions like user management and AI interactions.
- **Authentication:** Ensures secure login and role-based access.
- **Admin Controller:** Handles instructor management, queries, and complaints.
- **Content Management:** Manages course materials and updates.
- **User Model:** Stores admin data and credentials.
- **Role Model:** Defines admin permissions and access levels.
- **Course Material Model:** Stores and retrieves uploaded materials.

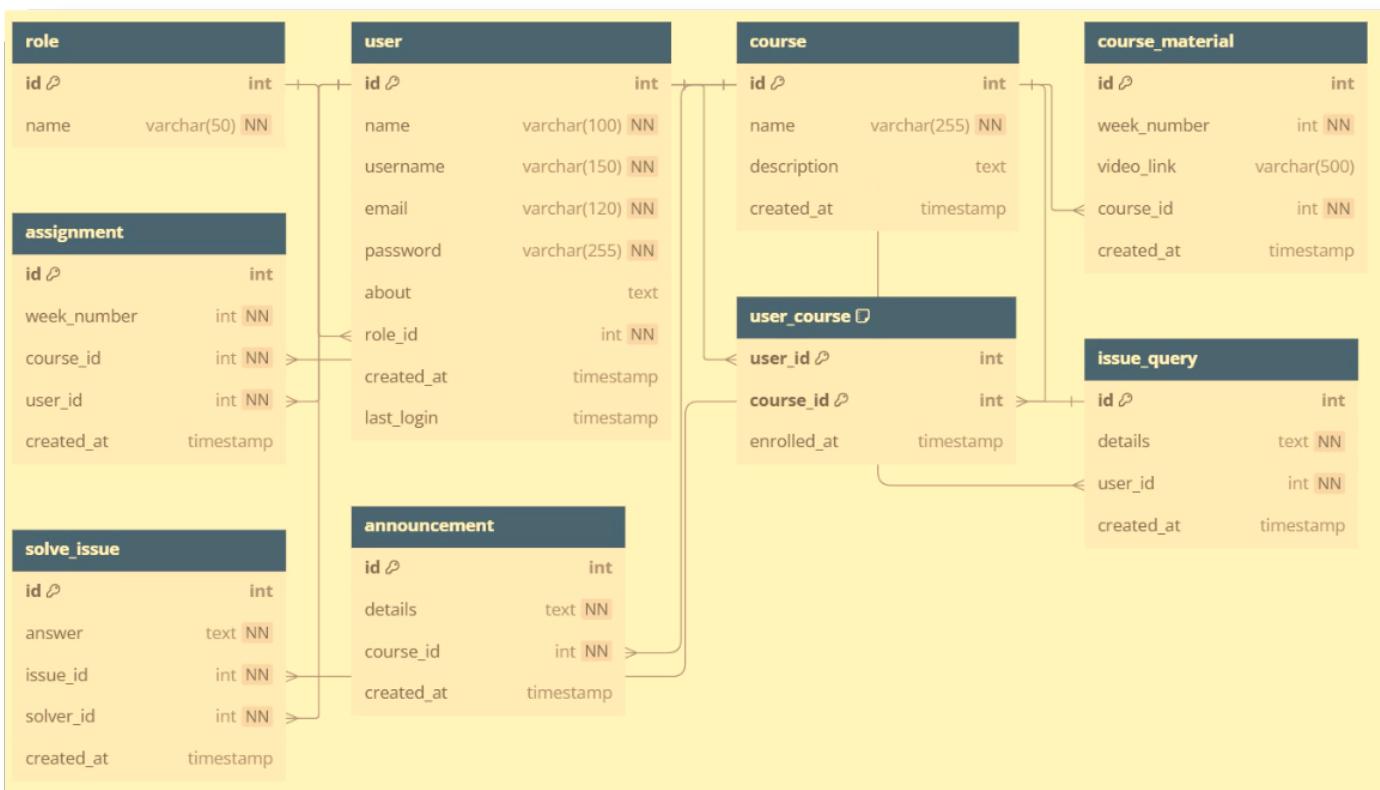
COMPONENT DESIGN

4. Professor Component



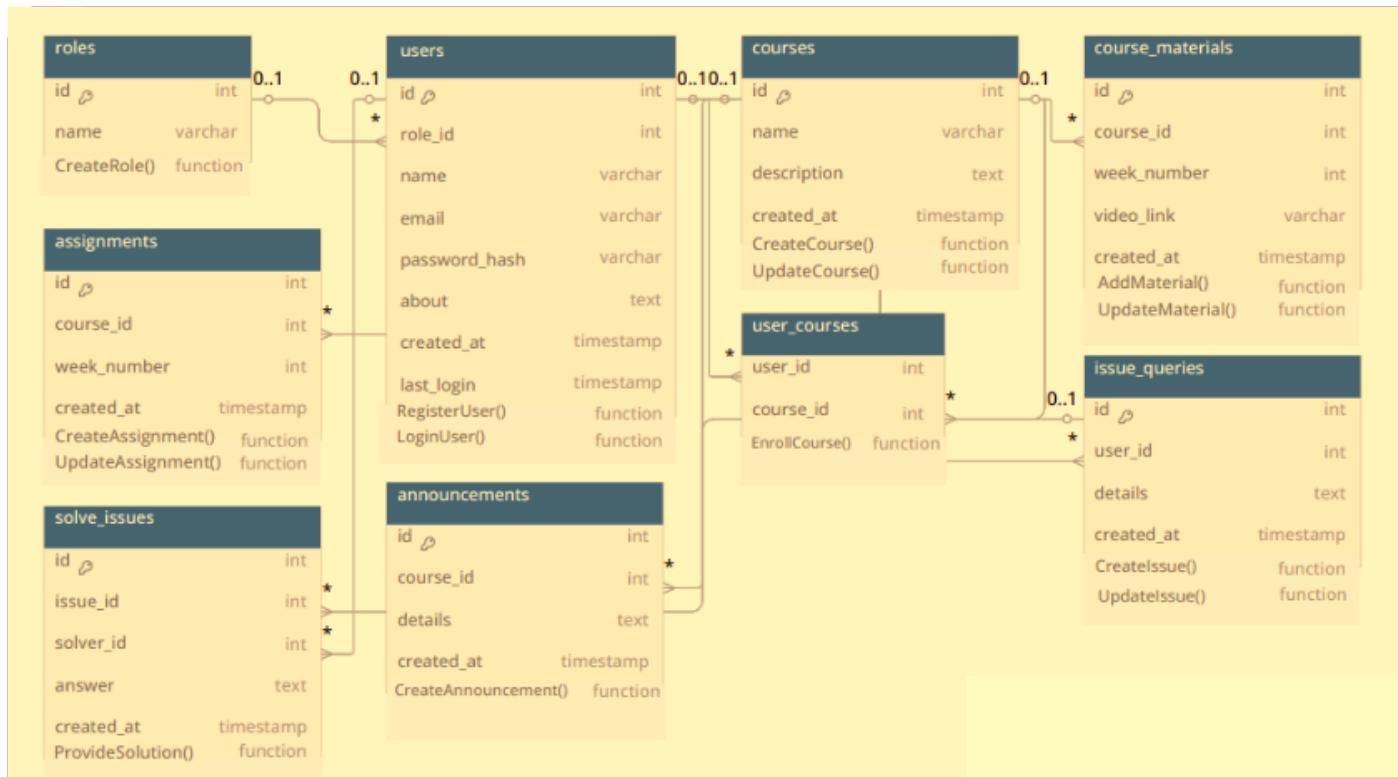
- **Home Page:** Entry point directing professors to authentication.
- **Login/Signup:** Manages secure professor authentication and access control.
- **Dashboard:** Provides an overview of teaching resources and interactions.
- **Resolve Query:** Allows professors to address student inquiries.
- **AI Agent:** Assists with automated responses and support.
- **Authentication:** Ensures secure login and role-based access.
- **Admin Controller:** Manages professor-related administrative tasks.
- **Content Management:** Handles course materials and academic resources.
- **User Model:** Stores professor data and credentials.
- **Role Model:** Defines permissions and teaching access levels.
- **Course Material Model:** Stores and retrieves learning materials.

DB Schema Design



This database schema is designed for an AI-driven academic guidance system, managing users, courses, and interactions between them. The **role** table defines different user roles, while the **user** table stores user details, including login credentials and role associations. The **course** table holds course information, and **course_material** links videos and materials to specific weeks. The **assignment** table tracks assignments submitted by users for each course. The **announcement** table stores course-related updates. The **user_course** table records user enrollments. The **issue_query** and **solve_issue** tables enable students to raise academic queries and receive solutions. Relationships between tables ensure efficient tracking of progress, resources, and interactions, forming a structured foundation for an AI agent to provide academic guidance.

CLASS DIAGRAM



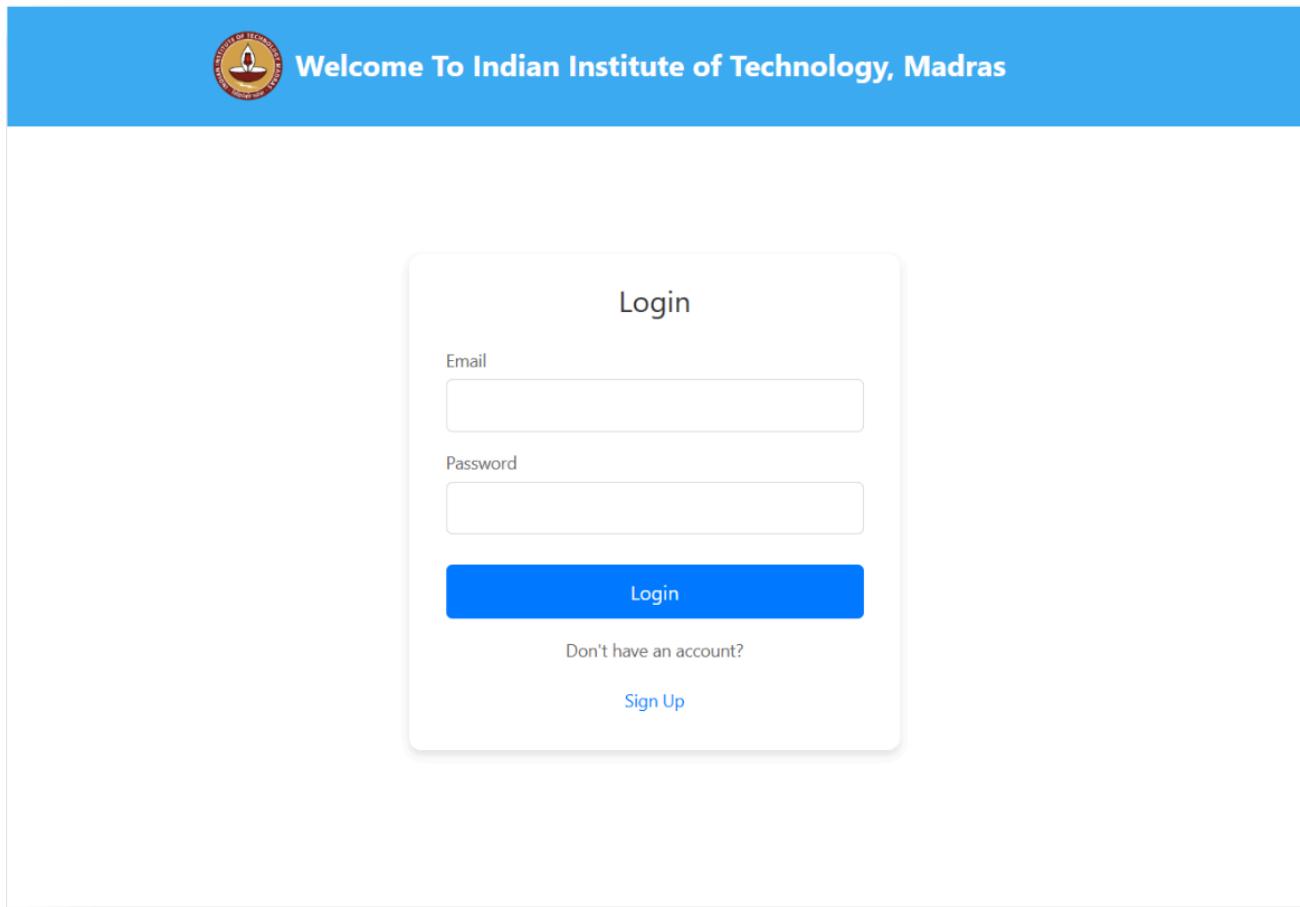
This class diagram represents the LMS database schema, including key entities such as **Users**, **Roles**, **Courses**, **Assignments**, **Course Materials**, **Issue Queries**, **Solve Issues**, and **Announcements**.

- **Users:** Manages user details and authentication.
- **Roles:** Defines permissions for different user types.
- **Courses:** Stores course details and enrollment.
- **Assignments:** Tracks assignment creation and updates.
- **Course Materials:** Manages learning resources.
- **Issue Queries:** Handles user-raised concerns.
- **Solve Issues:** Manages query resolution.
- **Announcements:** Stores course-related updates.

Entities are linked via foreign keys, ensuring structured data flow. Core functions like **RegisterUser()**, **CreateCourse()**, and **ProvideSolution()** enable smooth system operations.

UI PAGES

1. Login Page



The **Login Page** is a crucial entry point for users accessing the **Indian Institute of Technology, Madras** online platform. It ensures secure authentication by verifying user credentials before granting access to personalized dashboards.

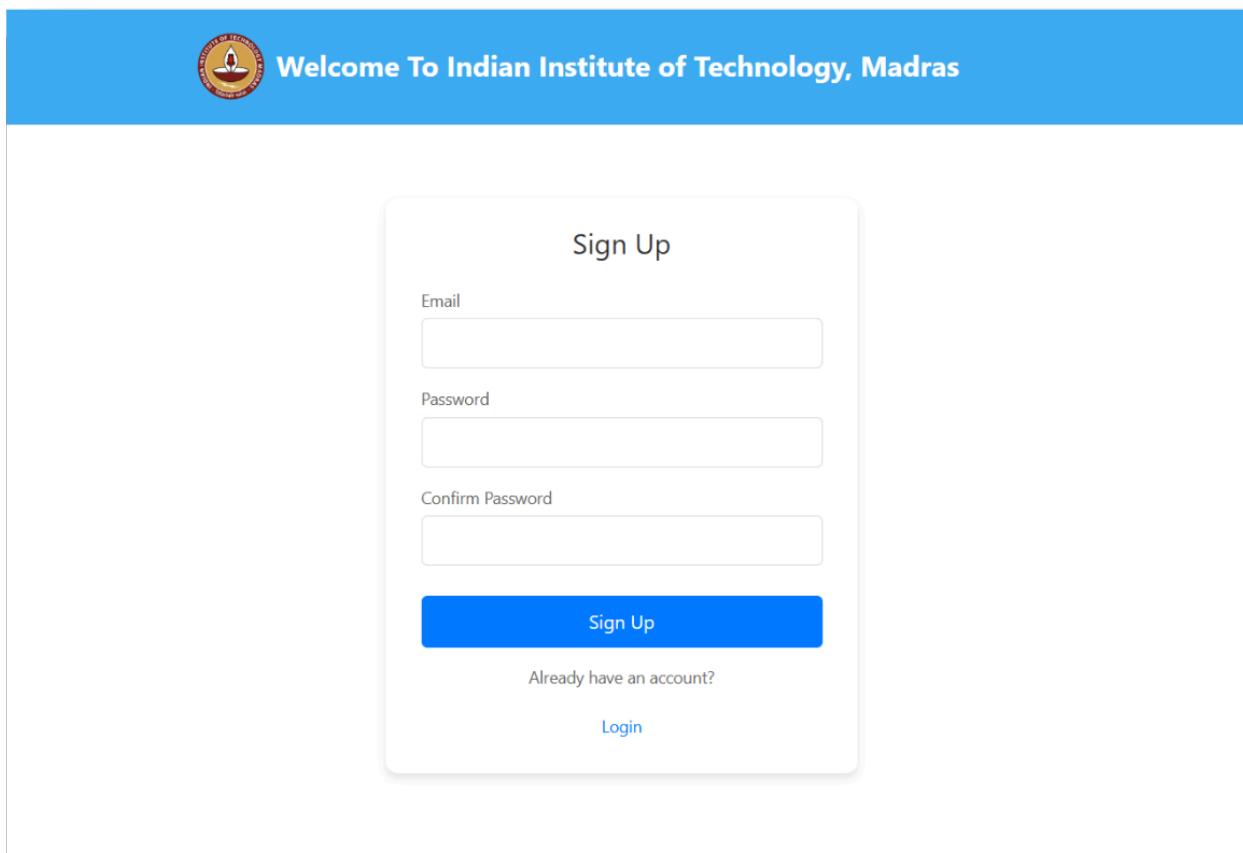
Key Features:

- **Institution Branding:** Displays the IIT Madras logo and welcome text.
- **User Authentication:** Requires Email and Password for secure login.
- **Call to Action:** A prominent Login button for user access.
- **New User Registration:** A Sign Up link for users without an existing account.

This page enhances usability and security, ensuring a seamless experience for students and faculty members.

UI PAGES

2. Sign Up Page



The **Sign-Up Page** allows new users to create an account on the **Indian Institute of Technology, Madras** online platform. It ensures secure user registration before accessing the system.

Key Features:

- **Institution Branding:** Displays the IIT Madras logo and welcome text.
- **User Registration:** Requires Email, Password, and Confirm Password fields.
- **Call to Action:** A prominent Sign-Up button to complete the registration.
- **Existing User Redirect:** A Login link for users who already have an account.

This page facilitates easy and secure user onboarding, ensuring a smooth registration process.

UI PAGES

3. Student Dashboard Page

The screenshot shows a student dashboard with a header containing a logo, "Student Dashboard", and navigation links for "Home", "Profile", and "Logout". Below the header, a section titled "My Current Courses" displays four course cards. Each card includes the course name, recent assignment scores, and a "Go to Courses" button.

Course Name	Recent Assignment Scores	Action
Software Engineering Course	Week 1 Assignment- 100 Week 2 Assignment- 92	Go to Courses
Modern Application Development - 2	Week 1 Assignment- 85 Week 2 Assignment- 78	Go to Courses
Business Data Management Course	Week 1 Assignment- 90 Week 2 Assignment- 88	Go to Courses
Machine Learning Practice Course	Week 1 Assignment- 95 Week 2 Assignment- 89	Go to Courses

The **Student Dashboard** provides an organized view of the enrolled courses and recent assignment scores. It serves as the central hub for students to manage their coursework efficiently.

Key Features:

- Institution Branding:** Displays the IIT Madras logo and a "Student Dashboard" heading.
- Course Overview:** Displays currently enrolled courses, Shows assignment scores for recent weeks, Provides a "Go to Courses" button for quick access.
- Navigation Menu:** Home - Redirects to the homepage, Profile - Allows students to view/edit their profile details, Logout - Logs out the student from the system.

This dashboard ensures easy access to coursework, streamlined navigation, and performance tracking for students.

UI PAGES

4. Profile Page

The screenshot shows a student dashboard with a header bar in orange. On the left is a circular logo with a yellow emblem. Next to it is the text "Student Dashboard". On the right side of the header are three buttons: "Home", "Profile", and "Logout". Below the header is a large white rectangular area containing a user profile. At the top of this area is a black placeholder icon for a profile picture. Below the icon, the name "Atharva Sarbhukan" is displayed in bold black text. Underneath the name, there is a line of smaller text providing academic details: "Degree: B.S. Degree in Data Science and Applications", "Branch: Data Science and Programming", and "College: Indian Institute of Technology, Madras". Further down, under the heading "Selected Courses", is a list of four courses: "Modern Application Development - 1", "Modern Application Development - 2", "Machine Learning Practice", and "Business Data Management".

The **Profile Page** provides a summary of the student's academic details and enrolled courses. It acts as a personalized hub for users to view their information at a glance.

Key Features:

- **User Info:** Displays name, degree (B.S. Data Science), branch, and institution (IIT Madras).
- **Selected Courses:** Lists enrolled courses (e.g., Modern App Dev, ML Practice, Business Data Management).
- **Navigation:** Home, Profile, Logout for easy access.

UI PAGES

5. Course Page

The screenshot displays the 'Course Page' for a 'Software Engineering' course. The top navigation bar includes a logo, 'Course Page' text, and 'Home', 'Profile', 'Logout' buttons. On the left, a sidebar shows a collapsed 'Week 1' section with 'Lecture 1' (highlighted in green), 'Lecture 2' (selected), and 'Lecture 3'. Below it is a collapsed 'Week 2' section. The main content area features a video player titled 'Thinking of Software in terms of Components'. The video frame shows two men: one in a pink shirt gesturing while speaking, and another in a striped shirt listening. A red play button is visible in the bottom right of the video frame. The video player has a 'Share' icon in the top right corner. At the bottom of the video frame, there's a banner with the text 'Watch on YouTube T Madras S Degree'.

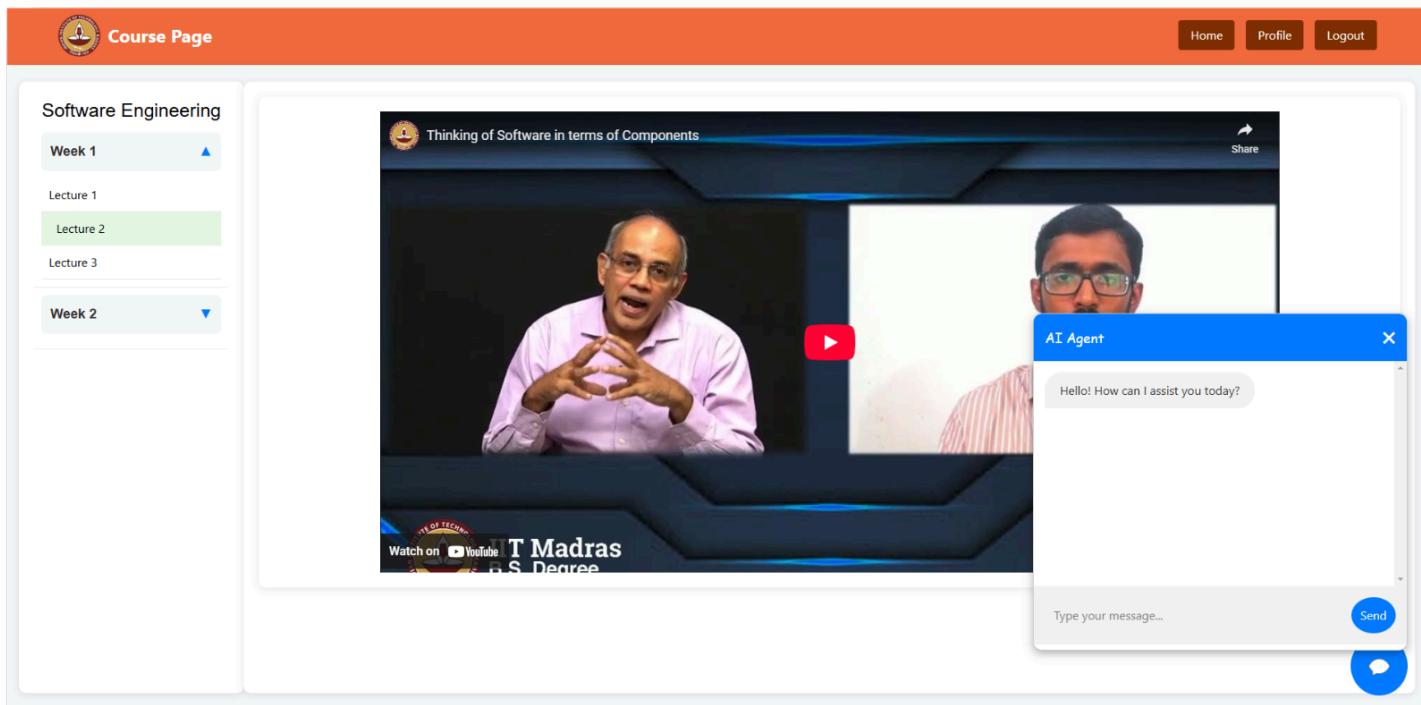
The **course page** provides an interactive platform for students to access lectures and learning materials efficiently.

Key Features:

- Course Structure:** The left sidebar displays the course title "Software Engineering" with collapsible sections for weeks and lectures. Users can navigate between weeks and lectures easily.
- Lecture Video:** The main section features an embedded YouTube lecture titled "Thinking of Software in Terms of Components" with two instructors. Users can play, pause, or expand the video.
- User Interface:** Sidebar Navigation for switching lectures, Top Navigation Bar with Home, Profile, and Logout buttons, Chat Support Button for assistance.

UI PAGES

6. Course ChatBot Page



This Course ChatBot page provides a structured layout for easy navigation and learning.

Key Features:

- **Course Navigation:** Sidebar with collapsible sections for weeks and lectures, Users can switch between topics easily.
- **Lecture Video Player:** Embedded YouTube lecture for seamless viewing, Playback and fullscreen options available.
- **User Interface Enhancements:** Top Navigation: Home, Profile, Logout buttons.
- **AI Chatbot:** Provides instant assistance.

Frontend Source Code Link :-

<https://drive.google.com/drive/folders/1WimCKGDVJy6ICqCFLTYRmSMxEqXf0lVD>