Student Relationship Manager

HRITHICK SUNDAR J – 22MCA15

Under the Guidance of

Dr. S. Kannan, M.Sc.(Phy).,M.Sc.(CS)., M.Phil., Ph.D.,

Table of contents

What is an SRM ?

Student Relationship Manager that manages the engagement of students with the institution

01

About the SRM

Student Relationship Management (SRM) system, developed for NextSkill Technologies, by me that offers a comprehensive solution for managing student-related information within educational institutions.

Designed to cater to the needs of various members of the institution such as

- **HR personnel,**
- **Staff**, and
- **Students**.

This SRM serves as a centralized platform for efficient interaction, access, and management of various student data, courses, attendance, and user roles.

Why the SRM is needed?

SRM is needed for various engagement purposes within the institution

02

Need for SRM ?

- There is no existing system to automate the student's affairs with the institution
- Student's daily objectives, attendance and fees must be manually managed
- Increased expense of Human Resource
- Student's progress and Efficiency cannot be calculated and maintained
- No proper official interaction platform has been defined in the institution.

Common problems

Attendance

Staff and Students have to sign the date and enter their attendance in a notebook

Students data

Data of the enrolled students are entered in Excel Sheets

Fee Tracking

Fees due deadlines are impossible to track simultaneously

Interaction

No official medium for communication inside the institution

Requirements Gathered

Attendance

Attendance must be automated

Tasks

Students must have timed assignments for evaluation

Reviews

The task submissions are reviewed by the staff

Course

Each course has a particular structure for the students

Enroll

The students data must be enrolled and managed efficiently

Chat Forum

A chat forum to discuss the course topics

How to create the SRM ?

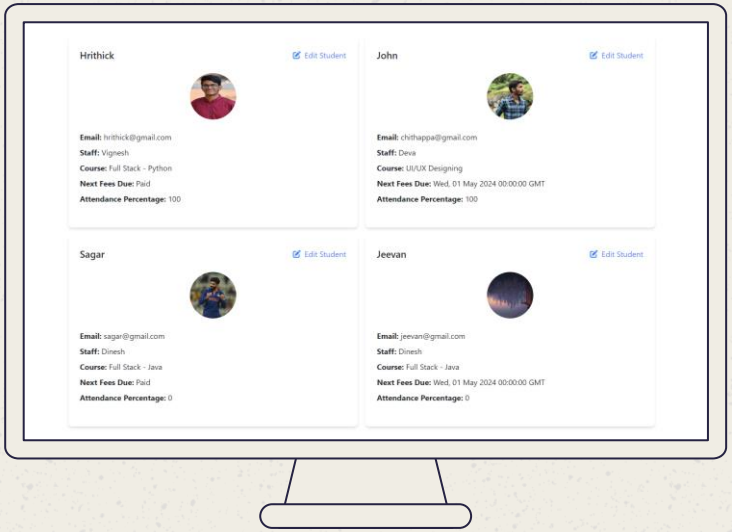
Approach towards the requirement satisfaction and creating the SRM

03

Approach

The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out in these plans; discussion has been made regarding the equipment, resources and how to test activities.

The plan is to automate various engagements within the institution and to combine all these in a single software while making it update on regular basis.



Primary Users

HR Admin

Manages the engagement and interactions between the students, staff and the institution

Staff

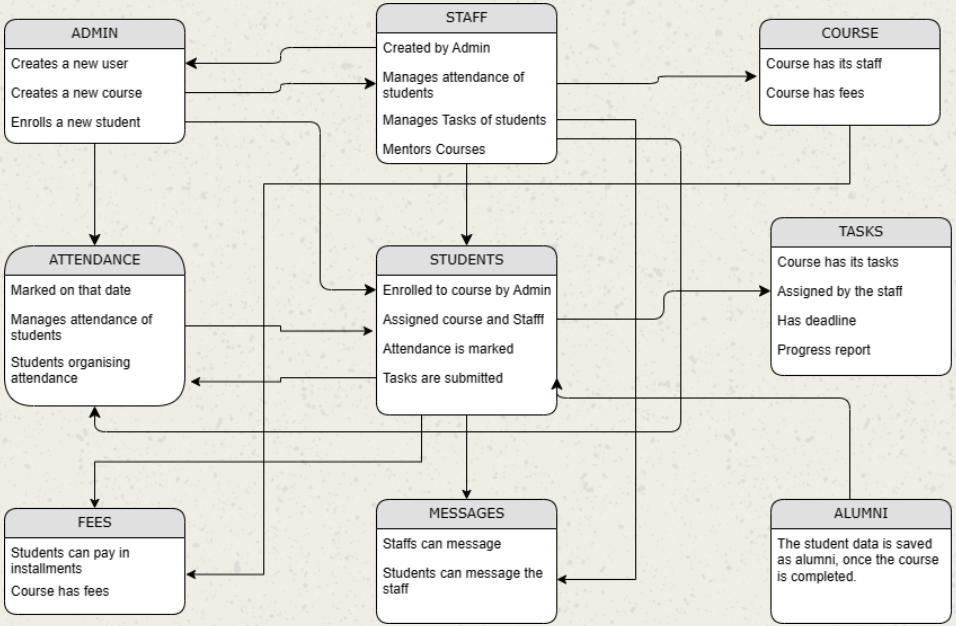
Staff has their associated students that are managed with personal care and efforts

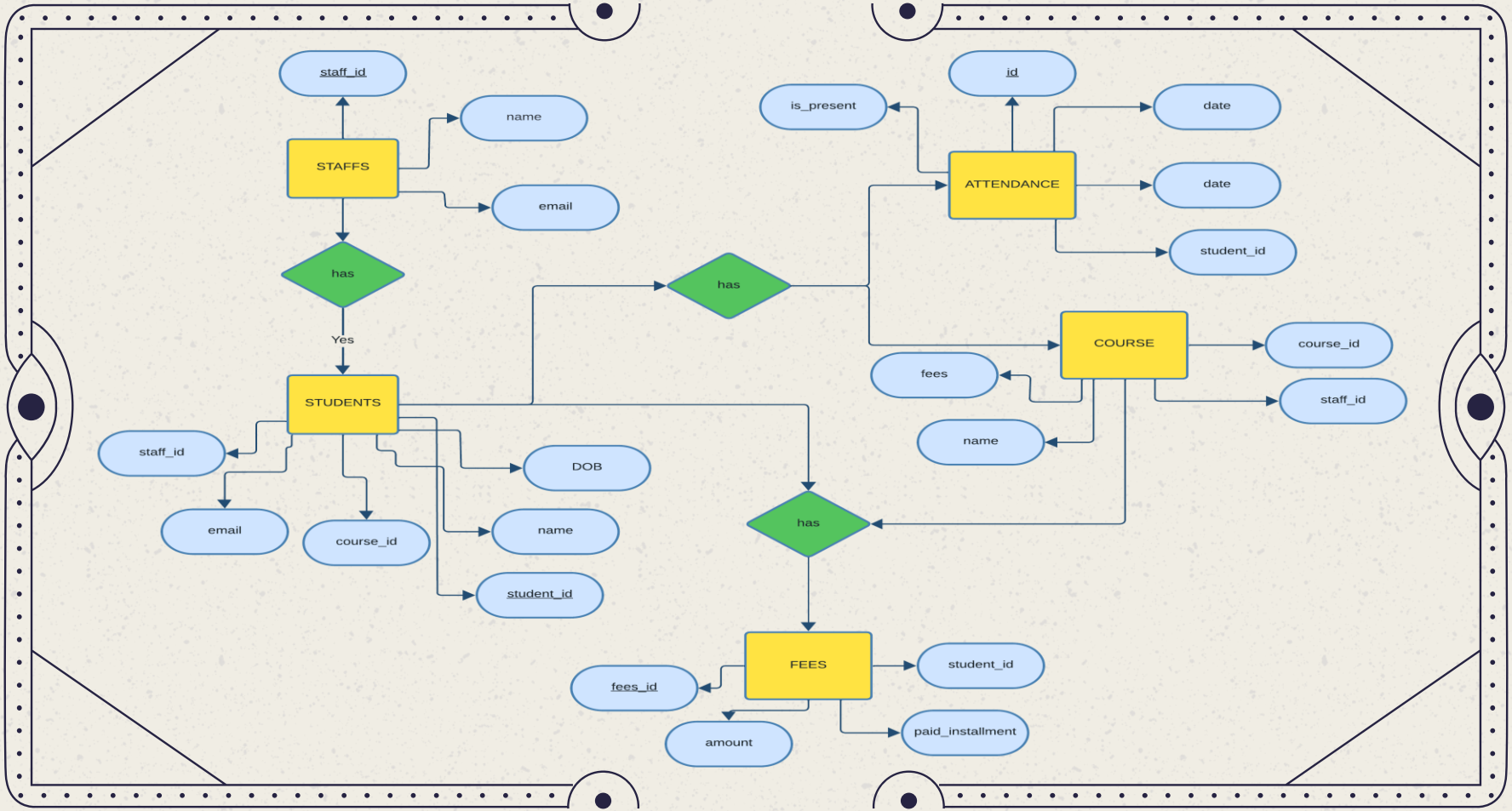
Students

The students have various profiles and can see their engagement with the institution

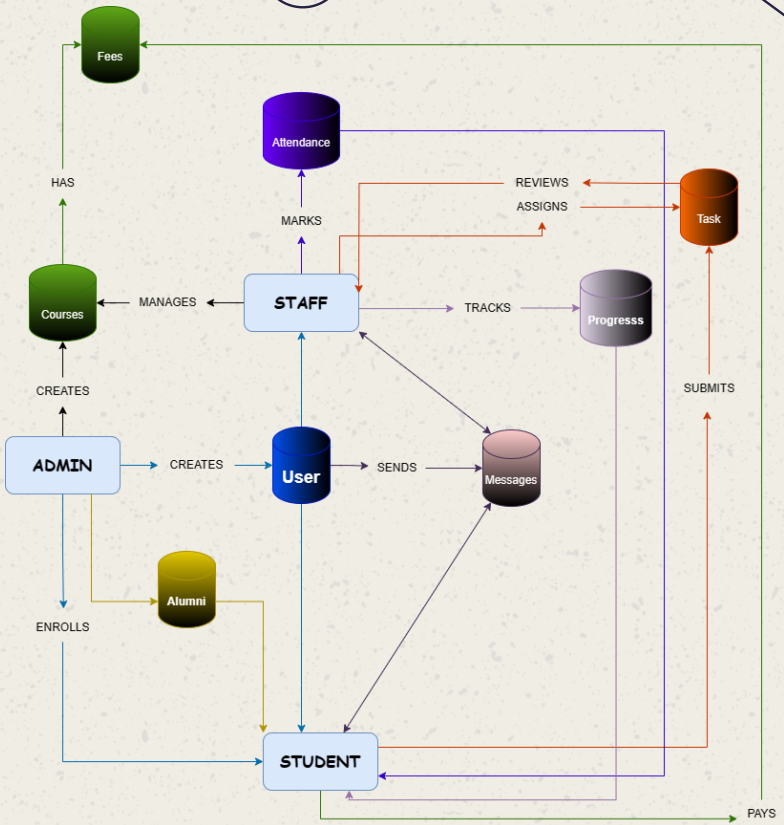
Data Models

1. Attendance model
2. Alumni model
3. Courses model
4. Fees model
5. Staffs model
6. Students model
7. Messages model
8. Queries model
9. Tasks model
10. User model




staff_id
name
STAFFS
email
has
Yes
STUDENTS
staff_id
email
course_id
DOB
name
student_id
has
id
is_present
date
ATTENDANCE
date
student_id
has
COURSE
course_id
fees
staff_id
name
has
FEES
student_id
fees_id
amount
paid_installment

In our Data Flow Diagram, we can see the data objects entities defined in ER diagram to have been accessed by various users and can be implemented in various processes to meet the requirements obtained from the client. The data flow between Users and their various roles such as tasks assignment, task submission are clearly seen in the DFD on the left.



Technological background

Approach towards the requirement satisfaction and creating the SRM technologically

04

Technologies Adapted

Frontend

For frontend, I used

- **ReactJS** for making it responsive without frequent requests and callbacks.
- **Javascript** functions for rendering the data obtained
- **Node JS** for frontend server setup
- **Tailwind CSS** for styling the pages

Technologies Adapted

Backend

For Backend, I used

- **Python** for making functions and logic implementation
- **Flask APIs** for rendering the data obtained through backend server
- **SQLite3** for database lightweight setup

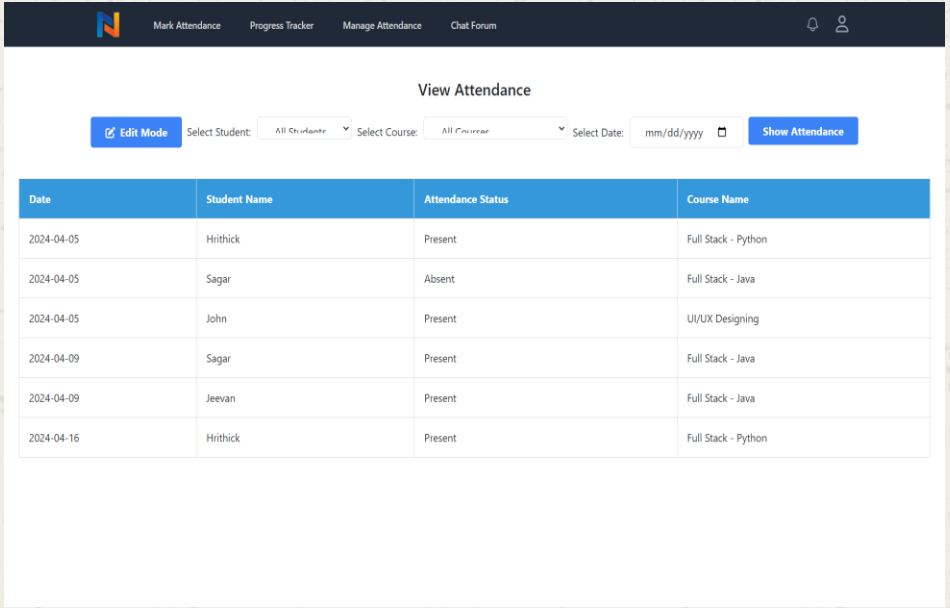
Features Implemented

Approach towards the requirement satisfaction and creating the SRM featuristically

05

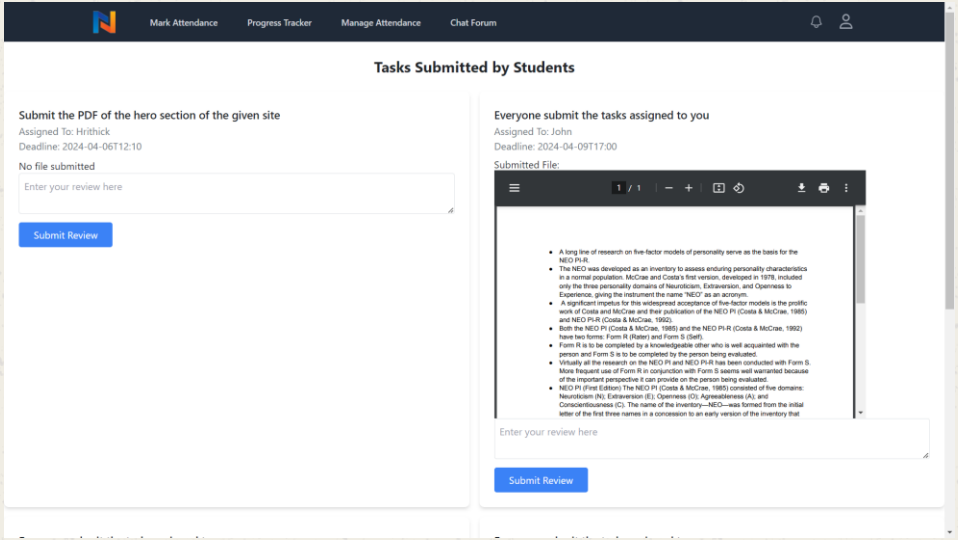
Attendance

- Attendance is automated for both the staffs and well as the students
- Admin can manage staff attendance
- Staffs can manage Student's attendance



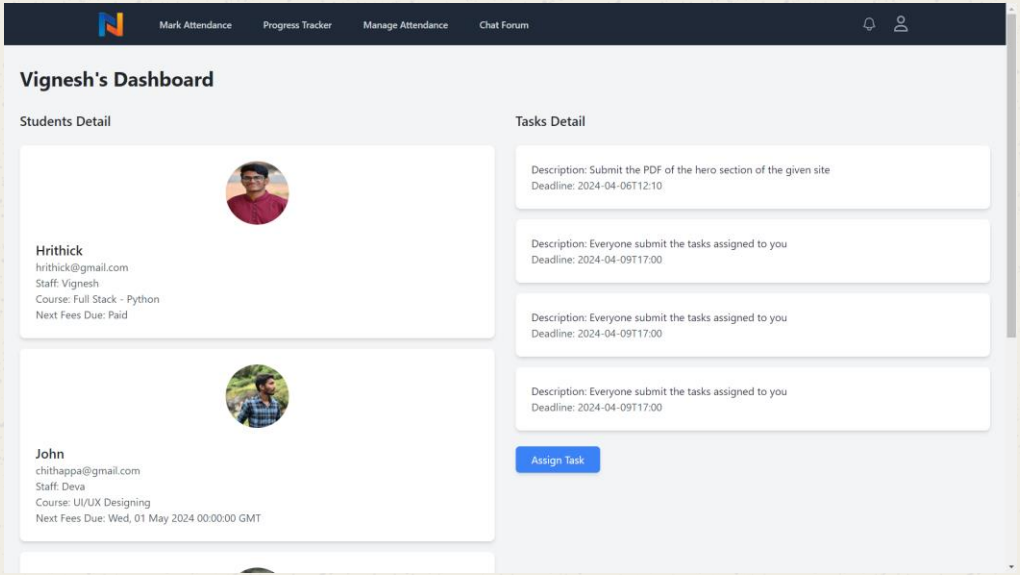
Tasks

- Tasks are assigned for each of the students and the deadline is allotted.
- Students can submit the task files
- Staffs can manage Student’s files and make reviews on them



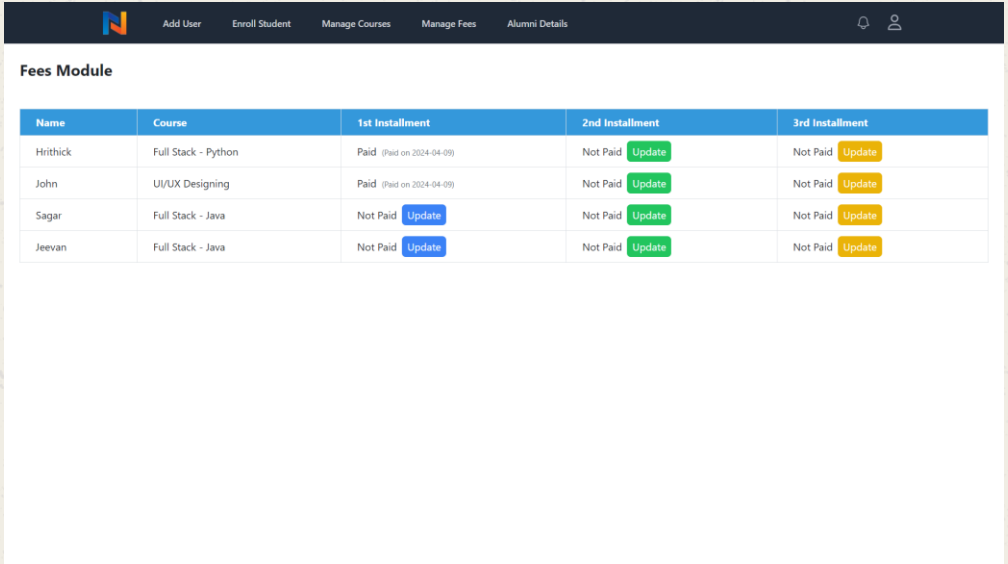
Profile and Dashboards

- Students can manage their profiles
- Staffs and Admins can view their dashboard with the students to be managed
- The students details are well displayed for ease of reference



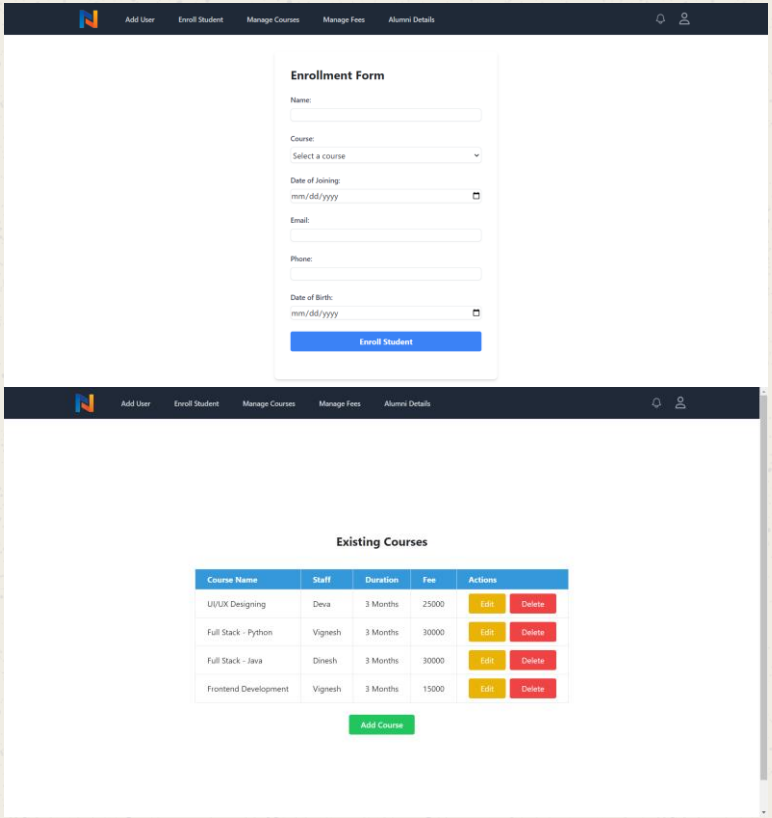
Fees Management

- Students can manage their payments and fee deadlines
- Admin can view the details of the fees due based on the installments
- The students details and their installments are well displayed for ease of reference



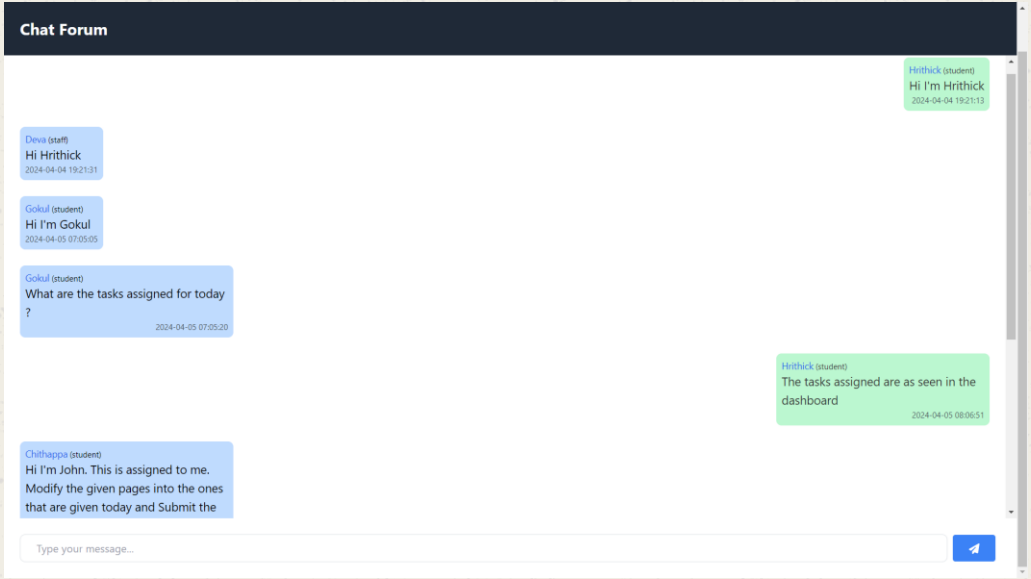
Student and Courses

- Once a Course is created and managed by the Admin, it gets assigned to a staff
- Once a student is enrolled on a particular course, they get allotted to that staff automatically and their fee dues are updated.



Discussion Forum

- An official discussion platform is integrated with this portal
- Students and staff can discuss the topics on this platform



Conclusion

Improvements to be made to this SRM to make it more feasible

06

Future Enhancements

- **Integration with Learning Management Systems (LMS):** Integrating our SRM system with popular LMS platforms such as Moodle or Canvas would provide a seamless experience for managing both administrative and academic aspects of student life..
- **Advanced Analytics and Reporting:** Enhancing the reporting capabilities of our SRM system with advanced analytics features would enable administrators to gain deeper insights into student performance, attendance trends, course effectiveness, and resource utilization.

Books

- **HTML & CSS**: The Complete Reference, Fifth Edition – Thomas A. Powell
- **React Js**: The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React. Js – Robin Wieruch
- **Python Flask**: Flask Web Development - Miguel Grinberg

Documentations

- Flask Documentation: Welcome to Flask — Flask Documentation (3.0.x) (palletsprojects.com)
- React JS Documentation: React
- Tailwind CSS: Documentation - Tailwind CSS
- JavaScript: JavaScript | MDN (mozilla.org)
- HTML- CSS: HTML5 (w3.org)
- SqLite: SQLite Documentation

Thank You !

Hrithick Sundar J - 22MCA15,
MCA – II Year,
Department of Computer Applications,
Madurai Kamaraj University.