

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

01

Introduction

What is an SRM ?
Why does the institution
need an SRM ?

02

Requirements

Requirements of an SRM
to the institution.

03

Methodology

Approach towards the
problem

04

Technology

You Implementation of
the technologies decided

05

Analysis

How the derived
solution works and helps

06

Conclusions

Improvements and future
feasibilities.

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 ?

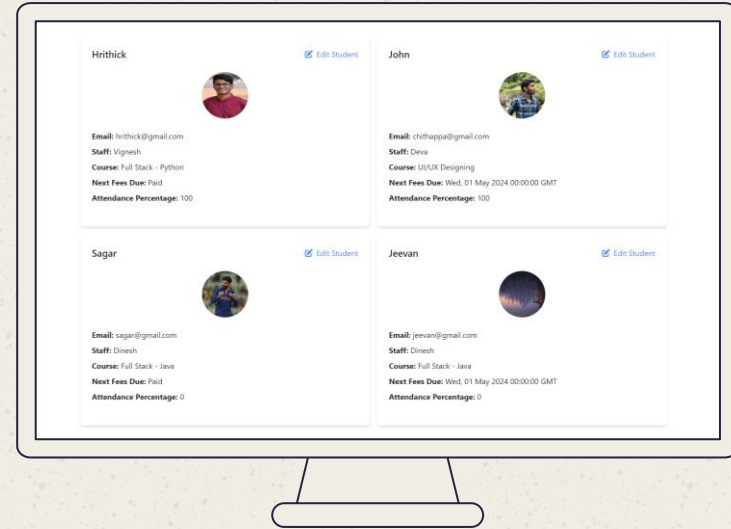
Approach towards the requirement
satisfaction and creating the SRM

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

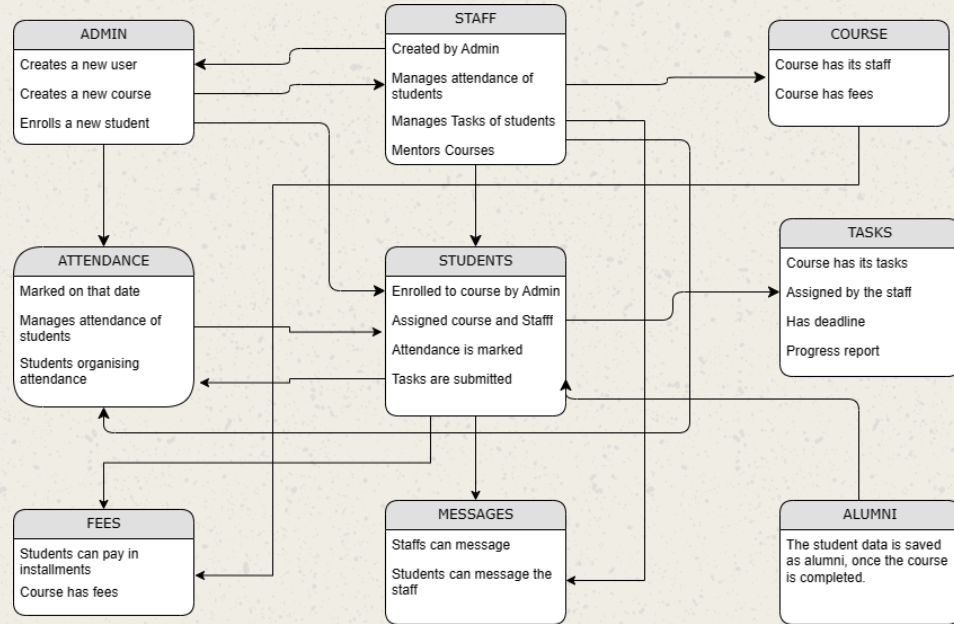


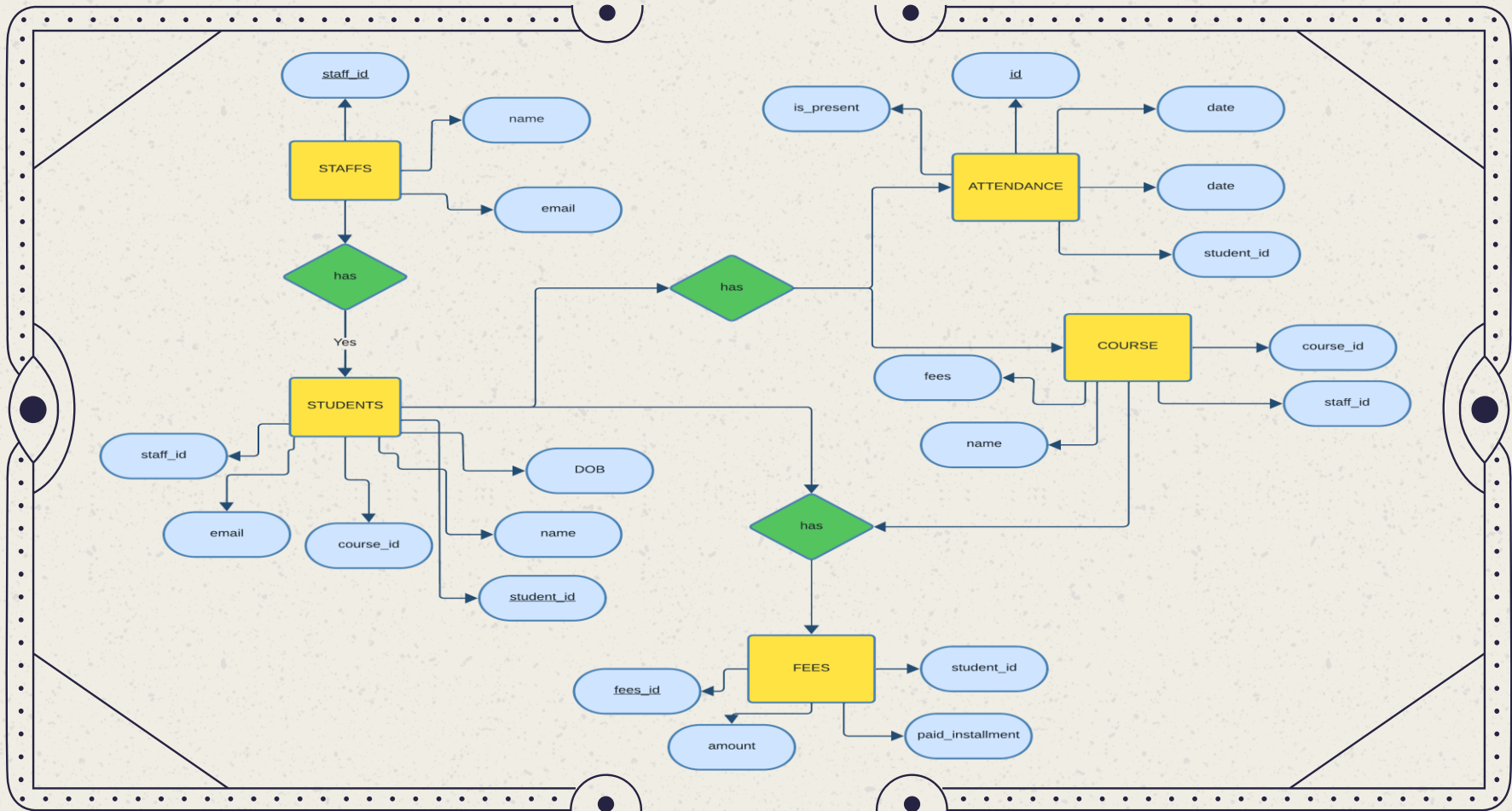
Students

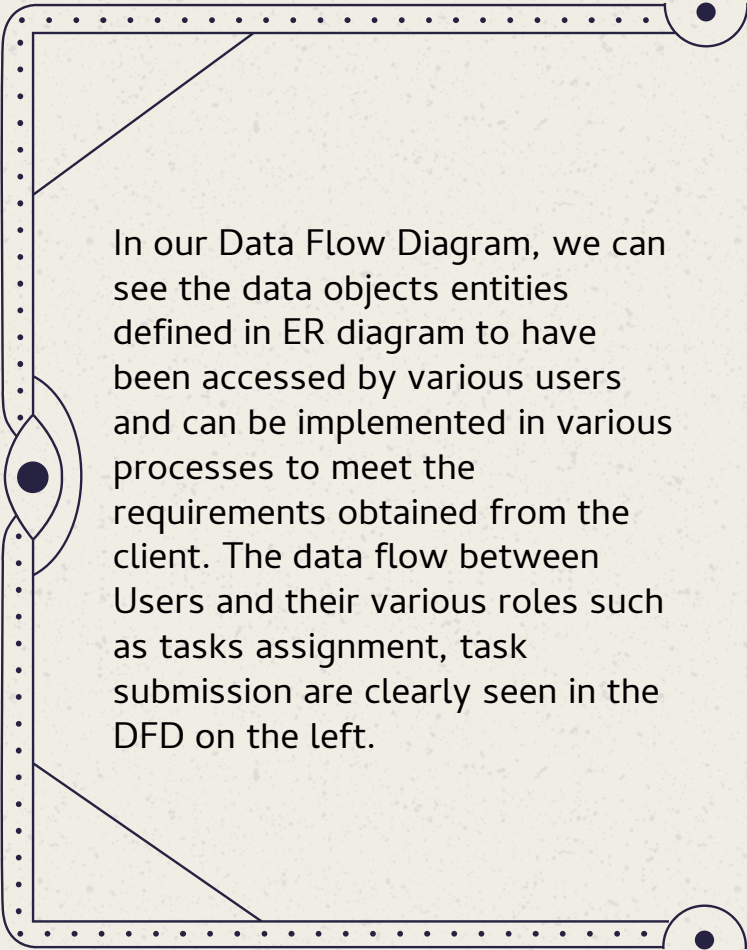
The students have various profiles and can see their engagement with the institution

Data Models

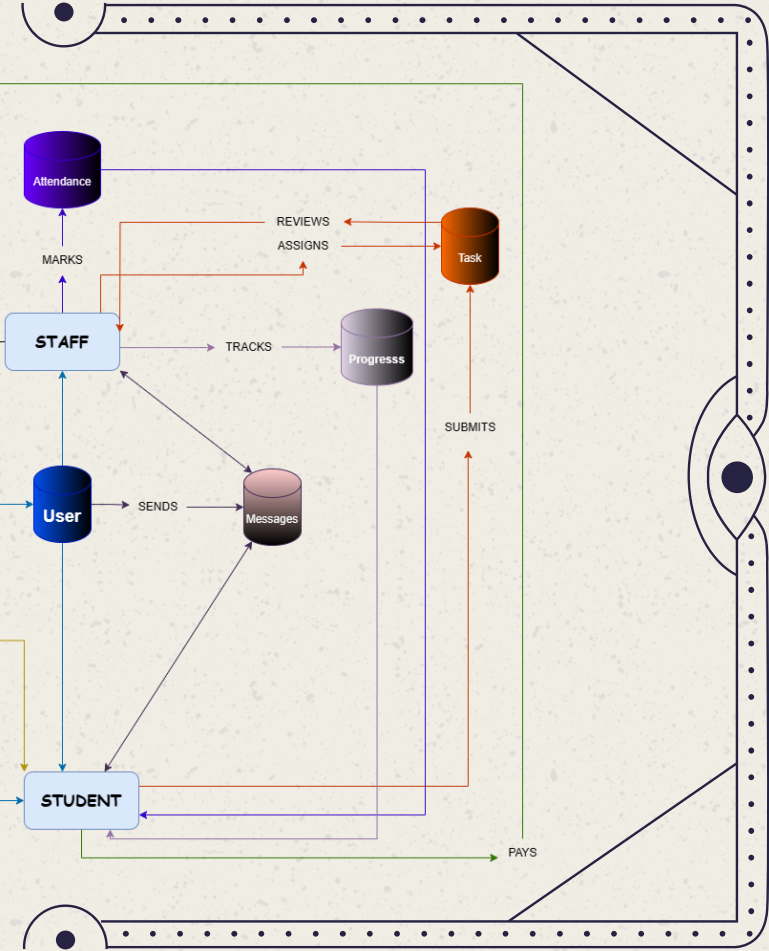
1. Attendance model
2. Alumni model
3. Courses model
4. Fees model
5. Staffs model
6. Students model
7. Messages model
8. Queries model
9. Tasks model
10. User model







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

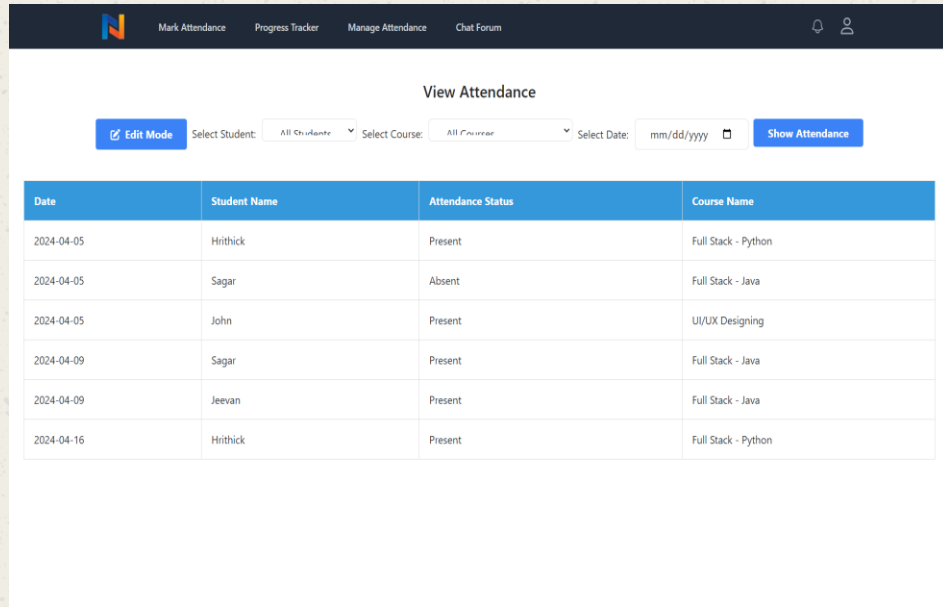
Features Implemented

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



The screenshot shows a web application interface for viewing attendance. At the top, there is a dark navigation bar with a logo and links for 'Mark Attendance', 'Progress Tracker', 'Manage Attendance', and 'Chat Forum'. On the right side of the navigation bar are a bell icon and a user profile icon. Below the navigation bar, the main content area is titled 'View Attendance'. It features a blue 'Edit Mode' button with a pencil icon, followed by two dropdown menus for 'Select Student' (currently showing 'All Students') and 'Select Course' (currently showing 'All Courses'). To the right of these is a 'Select Date' field with a date input showing 'mm/dd/yyyy' and a calendar icon, followed by a blue 'Show Attendance' button. Below these controls is a table with four columns: 'Date', 'Student Name', 'Attendance Status', and 'Course Name'. The table contains six rows of attendance data.

Date	Student Name	Attendance Status	Course Name
2024-04-05	Hrithick	Present	Full Stack - Python
2024-04-05	Sagar	Absent	Full Stack - Java
2024-04-05	John	Present	UI/UX Designing
2024-04-09	Sagar	Present	Full Stack - Java
2024-04-09	Jeevan	Present	Full Stack - Java
2024-04-16	Hrithick	Present	Full Stack - Python

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

The screenshot displays a web application interface with a dark blue header containing navigation links: Mark Attendance, Progress Tracker, Manage Attendance, and Chat Forum. The main content area is titled "Tasks Submitted by Students" and is divided into two columns. The left column shows a task assignment for "Submit the PDF of the hero section of the given site" assigned to "Hrithick" with a deadline of "2024-04-06T12:10". It indicates "No file submitted" and provides a text input field for a review, followed by a "Submit Review" button. The right column shows a task assignment for "Everyone submit the tasks assigned to you" assigned to "John" with a deadline of "2024-04-09T17:00". It shows a "Submitted File" as a PDF document titled "1 / 1". The PDF content is visible, listing bullet points about the NEO PI-R inventory. Below the PDF, there is a text input field for a review and a "Submit Review" button.

Tasks Submitted by Students

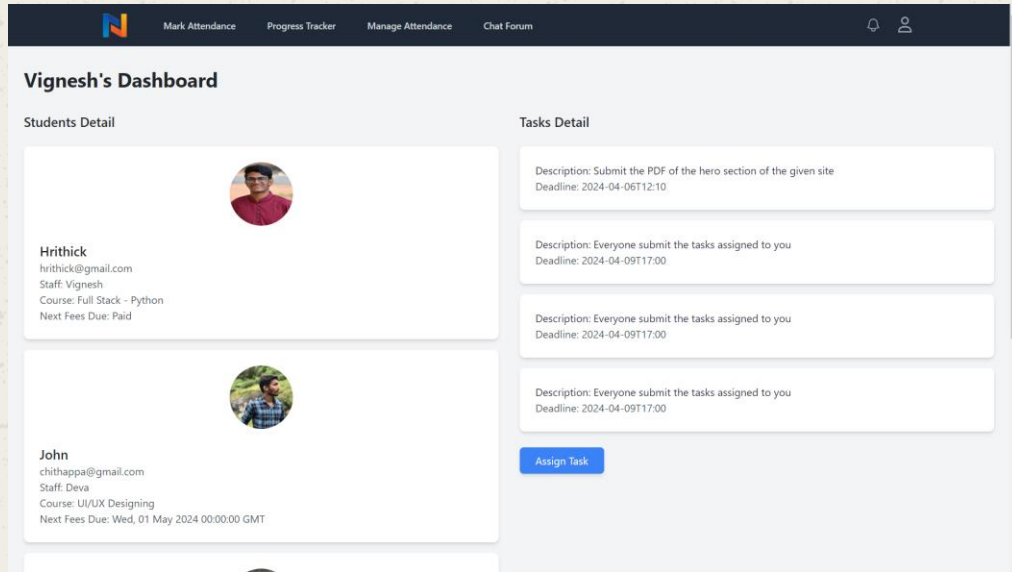
Submit the PDF of the hero section of the given site
Assigned To: Hrithick
Deadline: 2024-04-06T12:10
No file submitted
Enter your review here
Submit Review

Everyone submit the tasks assigned to you
Assigned To: John
Deadline: 2024-04-09T17:00
Submitted File:
1 / 1
Enter your review here
Submit Review

- A long line of research on five-factor models of personality serve as the basis for the NEO PI-R.
- The NEO was developed as an inventory to assess enduring personality characteristics in a normal population. McCrae and Costa's first version, developed in 1978, included only the three personality domains of Neuroticism, Extraversion, and Openness to Experience, giving the instrument the name "NEO" as an acronym.
- A significant impetus for the widespread acceptance of five-factor models is the prolific work of Costa and McCrae and their publication of the NEO PI (Costa & McCrae, 1988) and NEO PI-R (Costa & McCrae, 1992).
- Both the NEO PI (Costa & McCrae, 1988) and the NEO PI-R (Costa & McCrae, 1992) have two forms: Form R (Revised) and Form S (Self).
- Form R is to be completed by a knowledgeable other who is well acquainted with the person and Form S is to be completed by the person being evaluated.
- Virtually all the research on the NEO PI and NEO PI-R has been conducted with Form S. More frequent use of Form R in conjunction with Form S seems well warranted because of the important perspective it can provide on the person being evaluated.
- NEO PI (First Edition): The NEO PI (Costa & McCrae, 1988) consisted of five domains: Neuroticism (N), Extraversion (E), Openness (O), Agreeableness (A), and Conscientiousness (C). The name of the inventory—NEO—was formed from the initial letter of the first three names in a concession to an early version of the inventory that

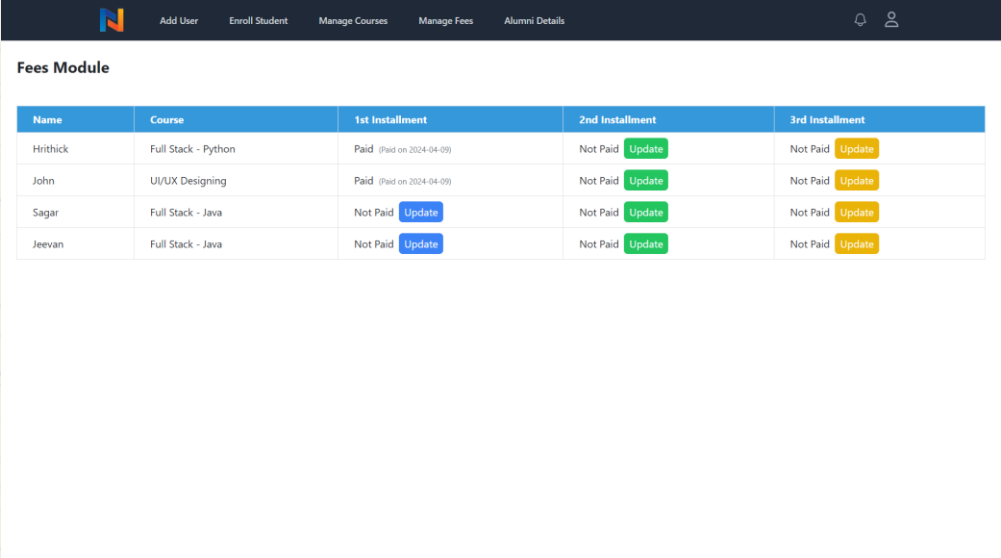
Profile and Dashboards

- 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



The screenshot displays a web application interface for 'Fees Management'. At the top, a dark navigation bar contains a logo and links for 'Add User', 'Enroll Student', 'Manage Courses', 'Manage Fees', and 'Alumni Details'. On the right of the navigation bar are a bell icon and a user profile icon labeled 'Do'. Below the navigation bar, the page title 'Fees Module' is shown. The main content area features a table with five columns: 'Name', 'Course', '1st Installment', '2nd Installment', and '3rd Installment'. The table lists four students: Hrithick, John, Sagar, and Jeevan. Each student's row shows their course and the status of their three installments, with 'Update' buttons for each installment that is not yet paid.

Name	Course	1st Installment	2nd Installment	3rd Installment
Hrithick	Full Stack - Python	Paid (Paid on 2024-04-09)	Not Paid Update	Not Paid Update
John	UI/UX Designing	Paid (Paid on 2024-04-09)	Not Paid Update	Not Paid Update
Sagar	Full Stack - Java	Not Paid Update	Not Paid Update	Not Paid Update
Jeevan	Full Stack - Java	Not Paid Update	Not Paid Update	Not Paid Update

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.

The screenshot displays a web application interface with a dark blue header containing navigation links: Add User, Enroll Student, Manage Courses, Manage Fees, and Alumni Details. On the right side of the header are a search icon and a user profile icon.

The main content area is divided into two sections. The top section, titled "Enrollment Form", contains the following fields:

- Name:
- Course:
- Date of Joining: with a calendar icon
- Email:
- Phone:
- Date of Birth: with a calendar icon

Below these fields is a blue button labeled "Enroll Student".

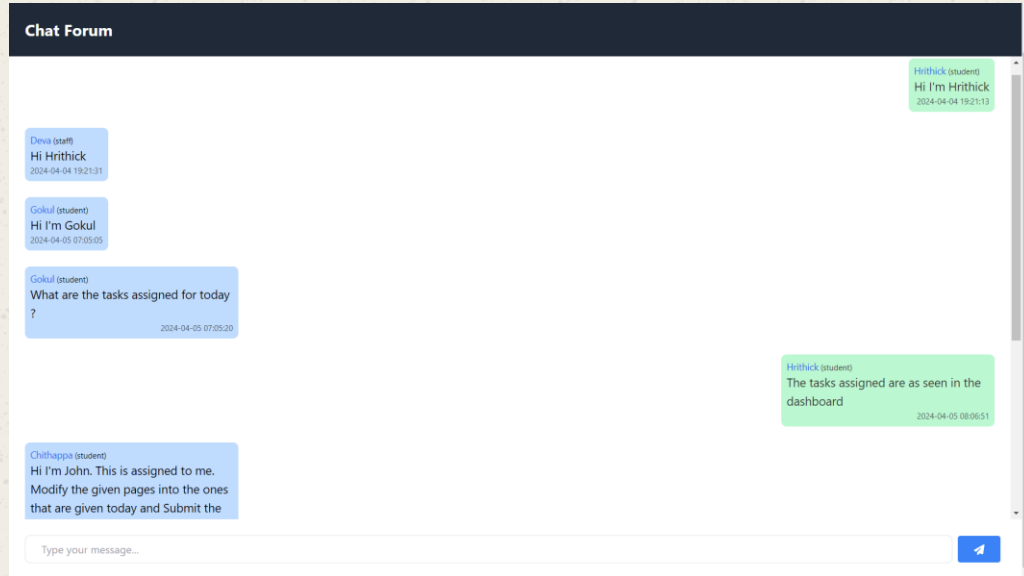
The bottom section, titled "Existing Courses", features a table with the following data:

Course Name	Staff	Duration	Fee	Actions
UI/UX Designing	Deva	3 Months	25000	<button>Edit</button> <button>Delete</button>
Full Stack - Python	Vignesh	3 Months	30000	<button>Edit</button> <button>Delete</button>
Full Stack - Java	Dinesh	3 Months	30000	<button>Edit</button> <button>Delete</button>
Frontend Development	Vignesh	3 Months	15000	<button>Edit</button> <button>Delete</button>

Below the table is a green button labeled "Add Course".

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.

