

A Software Engineering Project On

STUDENT LEAVE MANAGEMENT SYSTEM

as the partial fulfillment of

Semester - VI study

for the degree of

Bachelor of Science (Information Technology)
(Affiliated to Sarvajanik University)

during the academic year 2023-2024.

Developed By:

- 1) **Vansh Khalasi**
- 2) **Rudra Patel**
- 3) **Zalak Patel**

Guided By:

Prof.Shruti Revdiwala



Sarvajanik University

Shree Ramkrishna Institute of Computer Education and Applied Sciences

Behind P.T Science College, M.T.B. College Campus, Athwalines, Surat-395 001

Shree Ramkrishna Institute of Computer Education and Applied Sciences
Behind P.T. Science College, M.T.B College Campus, Athwalines, Surat-395 001.



CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE

This is to certify that Student name

- 1) Vansh Khalasi
- 2) Rudra Patel
- 3) Zalak Patel

Exam no

- SR21BSIT067**
SR21BSIT127
SR21BSIT130

*has / have successfully completed his / her / their software engineering project
work entitled*

STUDENT LEAVE MANAGEMENT SYSTEM

as the partial fulfillment of

Semester -VI study
for the degree of

Bachelor of Science (Information Technology)

*(Affiliated to Sarvajanik University)
during the academic year 2023-2024.*

Internal Project Guide

HOD
Dept. of Computer Science

Date : _____

Place : Surat

University Examination:

Examination No. : _____

Signature of External Examiners : _____

INDEX

SR.NO	CONTENTS	Page. No
1.	Introduction	1
	• Project Profile	
2.	System Introduction	3
	• System Information	
	• System Objective	
	• System Scope	
	• System Limitations	
	• Hardware & Software Requirements	
	• New Tools/Technologies Learned/Used	
3.	Requirement Analysis & Modeling	8
	• Expected Working of System	
	• System Diagrams	
	• Requirement Gathering Diagram	
	• Use Cases Diagram	
	• Activity Diagram	
	• Sequence Diagram	
4.	Design	14
	• Interface Design	
	• System Database	
	• System Architecture	
	• System Flowchart	
5.	Coding	31
	• System Code	
6.	Testing	39
	• System Testing	
7.	Project Deployment	46
8.	Future Enhancement	50
9.	Bibliography & References	52

1. Introduction

➤ Project Profile:-

- Project Title → Student Leave Management System
 - Organization → Shree Ramakrishna Institute Of Computer Education and Applied Sciences.
 - Framework → Visual Studio Code
 - Front -End tool → React JS , CSS
 - Back -End tool → MongoDB , Express JS , Node JS
 - Project Category → Leave Management System
 - Project Duration → 6 Month
 - Project Guided By → Mrs.Shruti Revdiwala
 - Developed By → [1] Vansh Khalasi
[2] Rudra Patel
[3] Zalak Patel

2. System Introduction

➤ System Information: -

- A Student Leave Management System is a digital platform or software solution that automates and facilitates the process of requesting, approving, and managing student leaves within an educational institution.
- Our Project Leave Management system is developed for students as well as faculty for managing leave.
- This system is useful for the college students, teaching and non-teaching staff.
- We have three modules in this system. First is the admin, second is the faculty, third is the students.
- Here members of the college can log in to this system and they can use features such as applying for leave requests, approving leaves of students or rejecting, leave tracking, and reporting functionalities.
- Student can apply for leave and keep track of their leave status.
- In this system admin have rights to add/update/delete student, staff member and add leave type.

➤ System Objective: -

- Admin can add/update/delete student, staff and add leave type.
- Student can apply leave and check their leave status.
- Faculty member can approve/reject student leave under their guidance or also apply own leave.
- Admin registers the users, Then only students and staff can log in.

➤ System Scope: -

- Login function accessed by student, faculty and admin.
- Student can apply leave and check their status.
- Register function accessed only by admin.
- Admin can only add/update/delete user and faculty.
- Faculty member approve/reject student leave.
- Faculty member (teaching and non-teaching staff) can also apply their leave and check their status.

➤ System Limitations:-

- This system is internet based, therefore it won't be operable without internet.
- User add by only admin.

➤ Hardware & Software Requirements:-

■ Hardware:-

- Processor : Intel i3 11th gen or above
- Ram : 8 GB or above
- Storage : 100 GB or above (SSD preferable)

■ Software:-

- Framework : Visual Studio Code (Version-1.88.1)
- Frontend : React JS , CSS
- Backend : MongoDB , Express JS , Node JS
- Oprating System : Windows 10 or above

➤ New Tools/Technologies Learned/Used:-

The MERN stack is a modern web development framework that leverages four key technologies:

1. **Node.js**: A JavaScript runtime environment that allows developers to execute JavaScript code outside of a web browser. It's known for its event-driven, non-blocking I/O model, making it efficient for building scalable and high-performance server-side applications.
2. **MongoDB**: A NoSQL database that stores data in a flexible, JSON-like format. MongoDB is designed for scalability and performance, and it's well-suited for handling large volumes of data and applications with evolving schemas.
3. **Express.js**: A minimalist web application framework for Node.js. Express.js simplifies the process of building web servers and handling HTTP requests, providing a robust set of features for developing server-side logic and APIs.
4. **React.js**: A JavaScript library for building user interfaces. React.js enables developers to create reusable UI components and build dynamic, interactive web applications. It follows a component-based architecture and utilizes a virtual DOM for efficient rendering, resulting in fast and responsive user experiences.

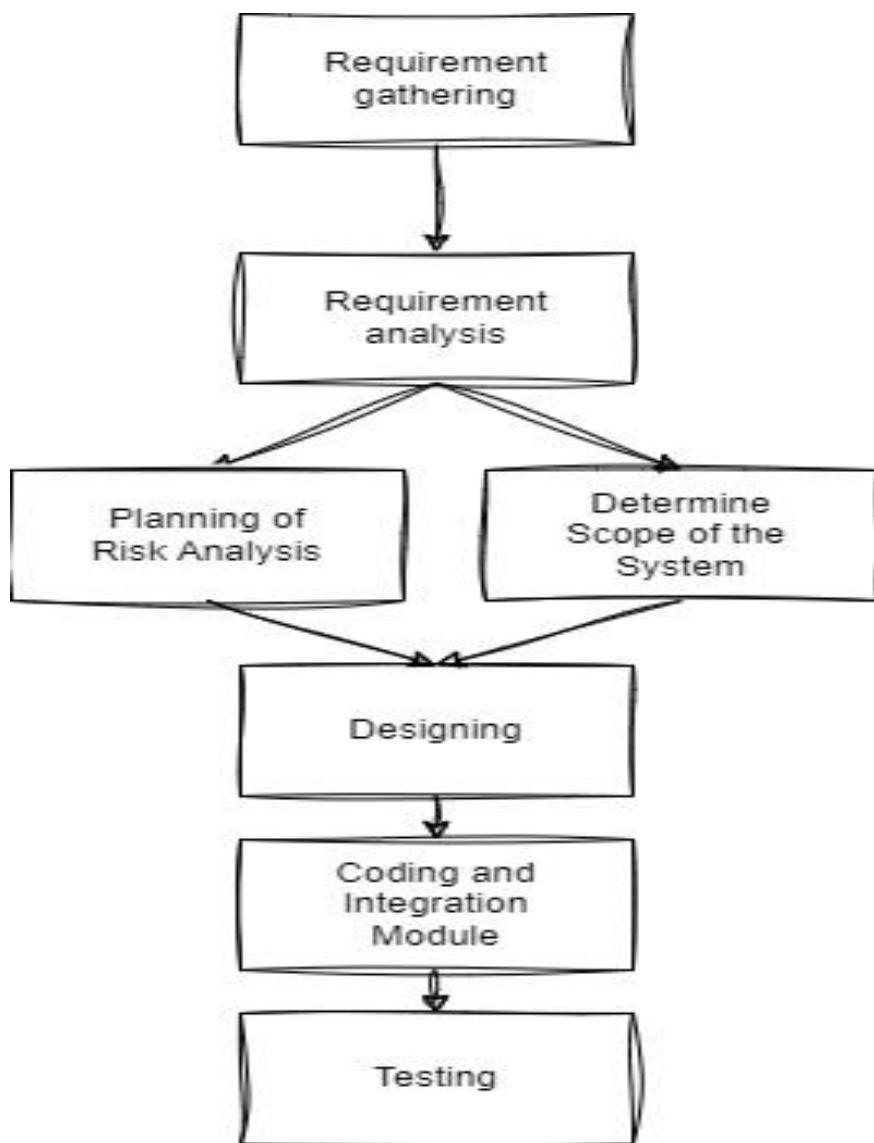
Together, the MERN stack allows developers to build full-stack JavaScript applications, from the backend to the frontend, using a unified language and set of technologies. It's widely used for building modern web applications that are scalable, efficient, and maintainable.

3. Requirement Analysis & Modeling

➤ Expected working of System:-

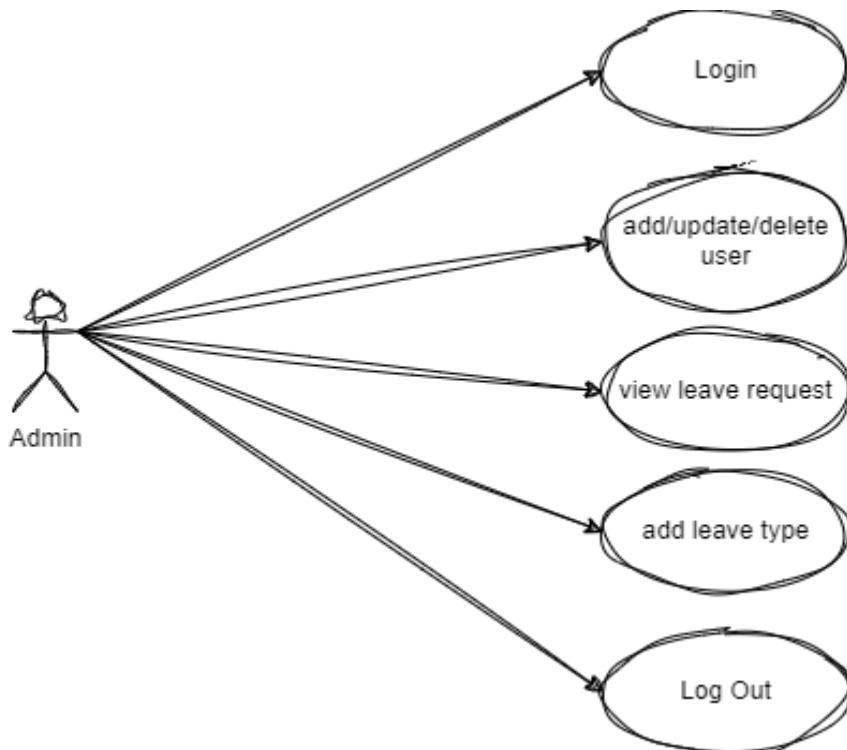
- **Admin Module:** Admin can login with user/email and password which is already stored in System database. Admin can add student, faculty member and view list of user and remove them if needed.
 - Add/Update/Delete Student And Faculty
 - View User List
 - View Leave Request List
 - Add new Leave Type
- **Faculty Module:** Faculty can login with username and password and approve/reject leave and they can also view their status.
 - Approve/Reject Student Leave
 - Status of Student Leave
 - View Leave Request List
 - Apply Leave
- **Student Module:** Student login with email and password which is provided by college.
 - Apply Leave
 - View Status of Leave
- **Login Module :** Admin, faculty member and student login in this system.

➤ Requirement Gathering Diagram:-

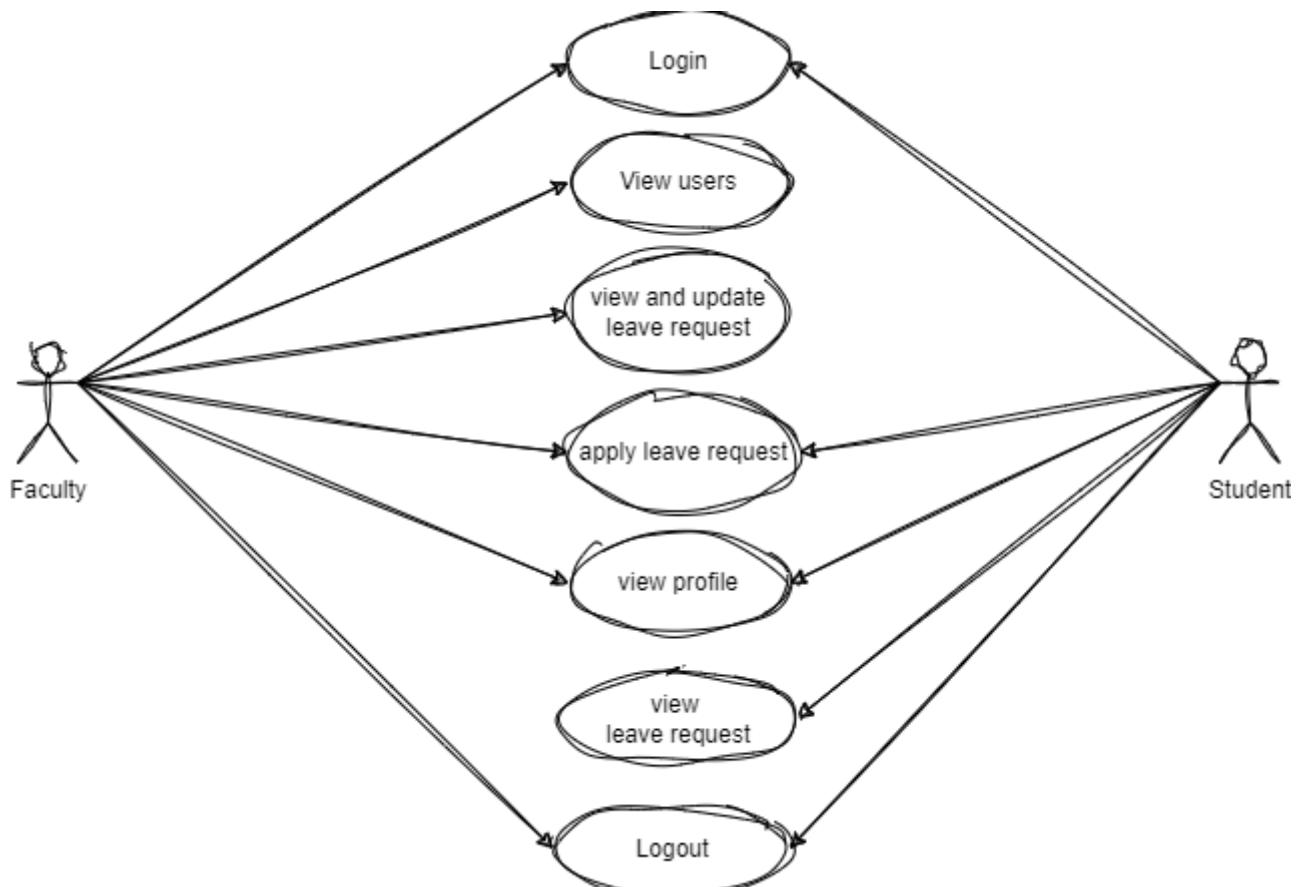


➤ **Use Cases Diagram:-**

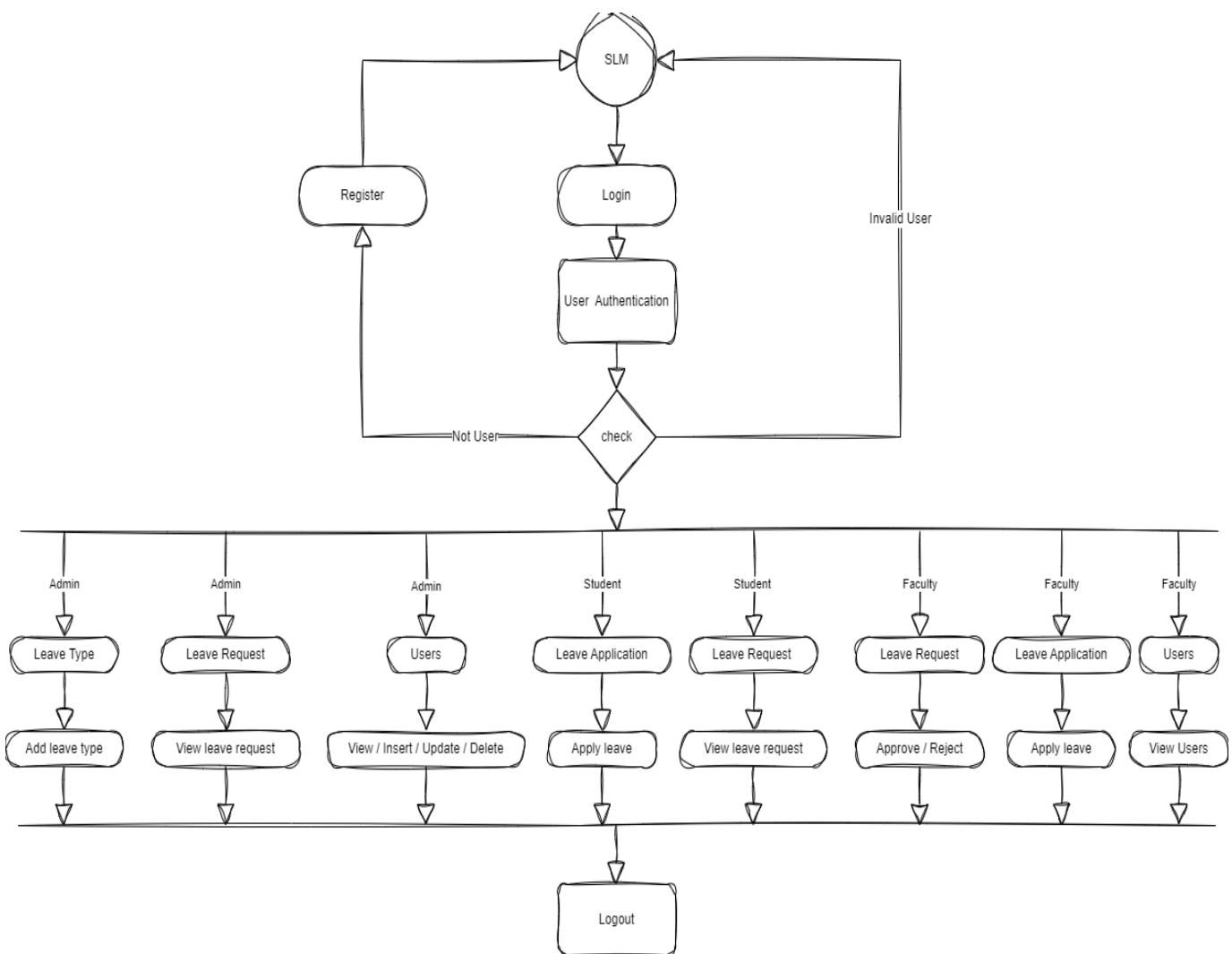
- **Admin**



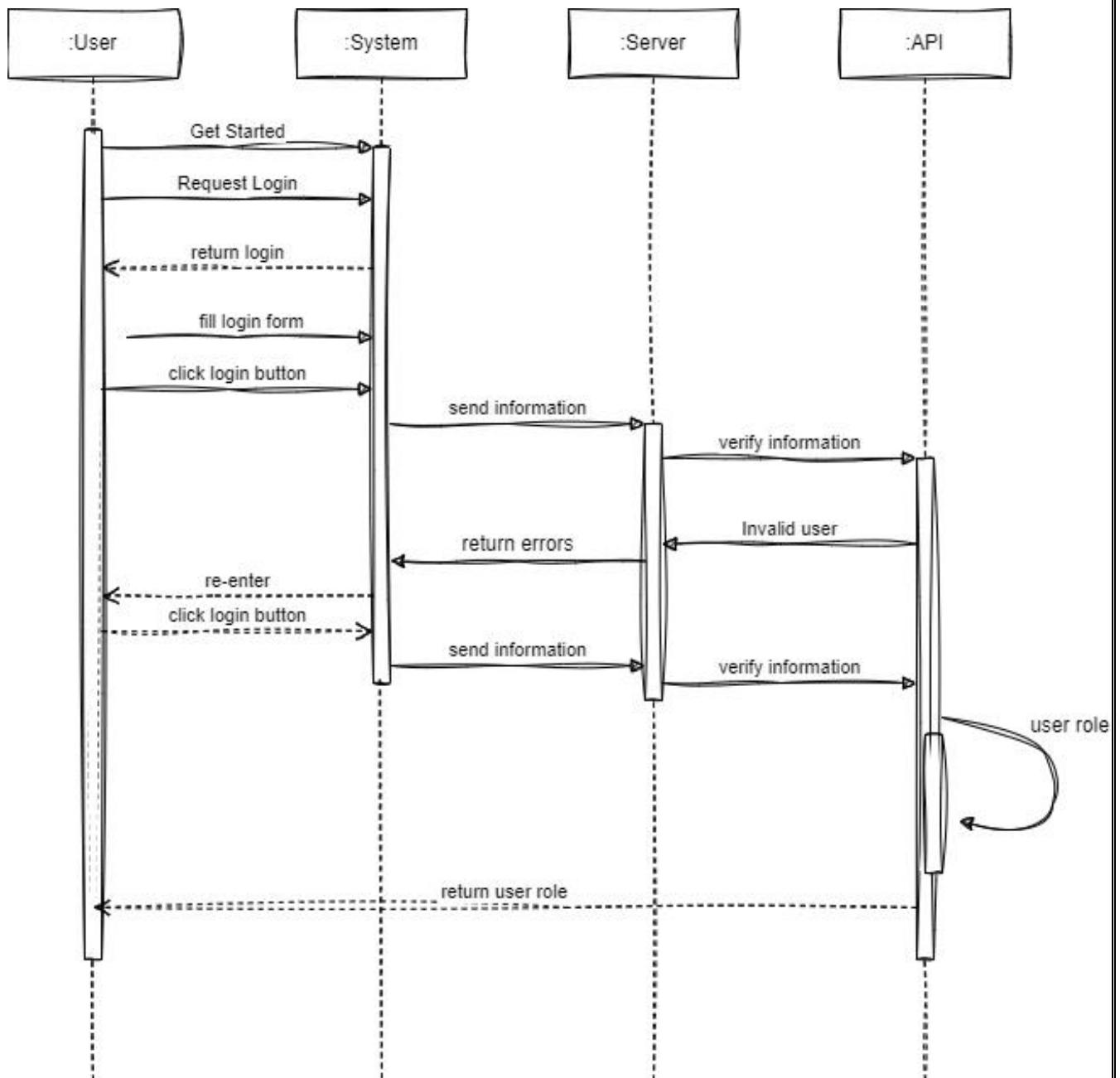
- **Student and Faculty**



➤ Activity Diagram:-



➤ Sequence Diagram:-



4. Design

➤ Interface Design:-

➤ STUDENT

Login

[Forgot password?](#)

Login

Student Leave Management Sys × holidays-list.2096766f918bdbd × +

localhost:3000/dashboard

WhatsApp Drive Gmail Classroom YouTube ChatGPT TradingView

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE

ZALAK PATEL

Approved Leaves 2

Rejected Leaves 1

Pending Leaves 0

UPCOMING HOLIDAYS LIST

- Independence Day 08-15
- Raksha Bandhan 08-19

[View Holiday List](#)

Types Of Leaves

Student Leave History

Name	Enrollment No	Leave Type	Leave Days	Attachment	Mentor	Status
Zalak Patel	SR21BSIT130	Medical	3	None	Rupal Snehkunj	Approved
Zalak Patel	SR21BSIT130	Vacation	2	None	Rupal Snehkunj	Approved
Zalak Patel	SR21BSIT130	Vacation	2	None	Rupal Snehkunj	Rejected

Type here to search

Student Leave Management Sy... localhost:3000/static/media/holidays-list.2096766f918bdbd4e4bb.pdf

WhatsApp Drive Gmail Classroom YouTube ChatGPT TradingView

holidays-list.2096766f918bdbd4e4bb.pdf 1 / 9 - 100% + ⌂ ⌂

Government of Gujarat
GENERAL ADMINISTRATION DEPARTMENT
Sachivalaya, Gandhinagar,
Dated the 20th November, 2023
Notification
PART – I
PUBLIC HOLIDAYS, 2024

No. GS/27/2023/JSR/102023/930/GH : The Government of Gujarat is pleased to declare the following days to be Public Holidays for State Government Offices during the year 2024.

Sr. No	Name of General Holidays	Date	Saka Era	Days of Week
1.	Republic Day	26 th January, 2024	06, Magh, 1945	Friday
2.	Maha Shivratri (Maha vad-13)	8 th March, 2024	18, Phalgun, 1945	Friday
3.	Holi 2 nd Day - Dhuleti	25 th March, 2024	05, Chaitra, 1946	Monday
4.	Good Friday	29 th March, 2024	09, Chaitra, 1946	Friday
5.	Cheti Chand	10 th April, 2024	21, Chaitra, 1946	Wednesday
6.	Ramjan-Eid (Eid-Ul-Fitr) (1 st Shawwal)	11 th April, 2024	22, Chaitra, 1946	Thursday
7.	Shree Ram Navmi	17 th April, 2024	28, Chaitra, 1946	Wednesday
8.	Bhagwan Shree Parshuram Jayanti (Vaishakh sud-3)	10 th May, 2024	20, Vaishakha, 1946	Friday
9.	Eid-Ul-Adha (Bakri Eid)	17 th June, 2024	27, Jyestha, 1946	Monday
10.	Muharram (Ashoora)	17 th July, 2024	26, Ashadh, 1946	Wednesday
11.	(1) Independence Day (2) Parsi New Year Day- Pateti (Parsi Shahenshahi)	15 th August, 2024	24, Shravan, 1946	Thursday
12.	Raksha Bandhan	19 th August, 2024	28, Shravan, 1946	Monday
13.	Janmashtami (Shravan Vad-8)	26 th August, 2024	04, Bhadrapad, 1946	Monday
14.	Samvatvarsi(Chaturthi Paksha)	07 th September, 2024	16, Bhadrapad, 1946	Saturday
15.	Eid-e-Meeladunnabi (Prophet Mohammad's Birthday) (bara vafat)	16 th September, 2024	25, Bhadrapad, 1946	Monday
16.	Mahatma Gandhi Jayanti	25 th October, 2024	14, Kartik, 1946	Wednesday

Type here to search 22-May-24

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Profile

Name	Enrollment No
Zalak Patel	SR21BSIT130
Email	Phone Number
zalakpatel1551@gmail.com	9978627164
Batch	Mentor
TYCS-B	Rupal Vaghela

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Request For Leave

First Name:

Enrollment No:

Div:

Year:

Mentor:

Details Of Leave

Leave Start:

Leave End:

Leave Type:

Browse Document:

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Year:

Mentor:

Details Of Leave

Leave Start:

Leave End:

Leave Type:

Browse Document:

Description:

➤ MODERATOR

Login

👤

🔒

[Forgot password?](#)

Login

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



- █ Dashboard
- █ Profile
- █ Leave Form
- █ Users List
- █ Leave Request List
- █ Logout

SHRUTI REVDIWALA

Approved Leaves 0

Rejected Leaves 0

Pending Leaves 0

UPCOMING HOLIDAYS LIST

- Independence Day 08-15
- Raksha Bandhan 08-19

View Holiday List

Student Leave History

There are no records to display

SRICEAS

18

Student Leave Management

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Student 3	Moderator 6	Admin 1
--------------	----------------	------------

Users List

Name	User	Authority	Email	Department
SRKI ADMIN	Admin		admin@gmail.com	
Chaulami Desai	Moderator	Principal	cd01@gmail.com	
Jayesh Pushtiwala	Moderator	H.O.D	jp01@gmail.com	CS
Rupal Vaghela	Moderator	Mentor	rs01@gmail.com	CS
Nidhi Vaniyawala	Moderator	Faculty	nv01@gmail.com	IT
Nirali Varu	Moderator	Faculty	nv02@gmail.com	IT
Shruti Revdiwala	Moderator	Faculty	sr01@gmail.com	IT
Zalak Patel	Student	Student	zalakpatel1551@gmail.com	CS

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Pending Requests 1	Approved Requests 5	Rejected Requests 3
-----------------------	------------------------	------------------------

Leaves Requests

Name	Leave Type	From	To	Mentor	Documents	Description	Status
Zalak Patel	Medical	5/22/2024	5/24/2024	Rupal Snehkunj	None	Medical	Approved
Rudra Patel	Vacation	5/22/2024	5/24/2024	Nirali Varu	None	Vaction	Approved
Vansh Khalasi	Others	5/21/2024	5/22/2024	Shruti Revdiwala	None	vansh	Rejected
Zalak Patel	Vacation	5/26/2024	5/28/2024	Rupal Snehkunj	None	a	Approved
Zalak Patel	Vacation	5/22/2024	5/23/2024	Rupal Snehkunj	None	b	Rejected
Rudra Patel	Medical	5/23/2024	5/25/2024	Nirali Varu	None	aa	Approve Reject

➤ ADMIN

Login

[Forgot password?](#)

Login

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE

- Dashboard
- @ Profile
- 📅 Leave Form
- 👤 Users List
- ☰ Leave Request List
- 📅 Add Leave Type
- ➡ Logout

SRKI ADMIN

Approved Leaves
5

Rejected Leaves
3

Pending Leaves
1

UPCOMING HOLIDAYS LIST

- Independence Day 08-15
- Raksha Bandhan 08-19

[View Holiday List](#)

Types Of Leaves

Medical	Casual
Vacation	NCC/NS/Sports

Student Leave History

Name	Enrollment No	Leave Type	Leave Days	Attachment	Mentor	Status
Zalak Patel	SR21BSIT130	Medical	3	None	Rupal Snehkunj	Approved
Rudra Patel	SR21BSIT127	Vacation	3	None	Nirali Varu	Approved
Vansh Khalasi	SR21BSIT067	Others	2	None	Shruti Revdiwala	Rejected
Zalak Patel	SR21BSIT130	Vacation	2	None	Rupal Snehkunj	Approved
Zalak Patel	SR21BSIT130	Vacation	2	None	Rupal Snehkunj	Rejected

Rows per page: 10 ▾ 1-6 of 6

SRICEAS

20

Student Leave Management

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Pending Requests
1
Approved Requests
5
Rejected Requests
3

Leaves Requests

Name	Leave Type	From	To	Mentor	Documents	Description	Status
Zalak Patel	Medical	5/22/2024	5/24/2024	Rupal Snehkunj	None	Medical	Approved
Rudra Patel	Vacation	5/22/2024	5/24/2024	Nirali Varu	None	Vaction	Approved
Vansh Khalasi	Others	5/21/2024	5/22/2024	Shruti Revdiwala	None	vansh	Rejected
Zalak Patel	Vacation	5/26/2024	5/28/2024	Rupal Snehkunj	None	a	Approved
Zalak Patel	Vacation	5/22/2024	5/23/2024	Rupal Snehkunj	None	b	Rejected
Rudra Patel	Medical	5/23/2024	5/25/2024	Nirali Varu	None	aa	Pending

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Student
3
Moderator
6
Admin
1

Users List

Name	User	Authority	Email	Department	Operation
SRKI ADMIN	Admin		admin@gmail.com		<button>Update</button> <button>Delete</button>
Chaulami Desai	Moderator	Principal	cd01@gmail.com		<button>Update</button> <button>Delete</button>
Jayesh Pushtiwala	Moderator	H.O.D	jp01@gmail.com	CS	<button>Update</button> <button>Delete</button>
Rupal Vaghela	Moderator	Mentor	rs01@gmail.com	CS	<button>Update</button> <button>Delete</button>
Nidhi Vaniyawala	Moderator	Faculty	nv01@gmail.com	IT	<button>Update</button> <button>Delete</button>
Nirali Varu	Moderator	Faculty	nv02@gmail.com	IT	<button>Update</button> <button>Delete</button>

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



- [Dashboard](#)
- [Profile](#)
- [Leave Form](#)
- [Users List](#)
- [Leave Request List](#)
- [Add Leave Type](#)
- [Logout](#)

Leave Types

Public Holiday

Leave Days:

Leave Name:

Authority: H.O.D Faculty Co-Ordinator Mentor

Select: Student Faculty

Create

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Student

- [Dashboard](#)
- [Profile](#)
- [Leave Form](#)
- [Users List](#)
- [Leave Request List](#)
- [Add Leave Type](#)
- [Logout](#)

Registration Form

First Name: <input type="text" value="First Name"/>	Last Name: <input type="text" value="Last Name"/>
Email: <input type="text" value="Email"/>	Phone Number: <input type="text" value="Phone Number"/>
Password: <input type="text" value="Password"/>	Confirm Password: <input type="text" value="Confirm password"/>
Enrollment No: <input type="text" value="Enrollment No"/>	Department: <input type="text" value="Select"/>
Course: <input type="text" value="Select"/>	Year: <input type="text" value="Select"/>
Division: <input type="text" value="Select"/>	Mentor: <input type="text" value="Select"/>
Authority: <input type="text" value="Select"/>	Register

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Moderator

Registration Form

First Name:	Last Name:
<input type="text" value="First Name"/>	<input type="text" value="Last Name"/>
Email:	Phone Number:
<input type="text" value="Email"/>	<input type="text" value="Phone Number"/>
Password:	Confirm Password:
<input type="text" value="Password"/>	<input type="text" value="Confirm password"/>
User:	Authority:
<input type="text" value="User"/>	<input type="text" value="Select"/>
Department:	Batch:
<input type="text" value="Select"/>	<input type="text" value="Select"/>
Mentor:	
<input type="text" value="Select"/>	

Register

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Moderator

Update Form

First Name:	Last Name:
<input type="text" value="Zalak"/>	<input type="text" value="Patel"/>
Enrollment No:	
<input type="text" value="SR21BSIT130"/>	
Phone Number:	
<input type="text" value="9978627164"/>	
Course:	
<input type="text" value="BSC"/>	
Department:	
<input type="text" value="CS"/>	
Year:	
<input type="text" value="TY"/>	
Division:	
<input type="text" value="B"/>	
Mentor:	
<input type="text" value="Rupal Vaghela (Mentor)"/>	

Update



SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE

Update Form

First Name: Shruti

Last Name: Revdiwala

Enrollment No: Enrollment No

Phone Number: 9151476188

Department: IT

Mentor: Jayesh Pushtiwala (H.O.D)

Update

- Dashboard
- Profile
- Leave Form
- Users List
- Leave Request List
- Add Leave Type
- Logout

➤ System Database:-

Table Name: user (Stores User Details)

Attribute	Data Type	Description
_id	ObjectId	Stores ID
type	String	Store user type
email	String	Stores email id data
firstName	String	Stores first name data
lastName	String	Stores last name data
password	String	Stores password data
phoneNumber	String	Stores phone number data
department	String	Store department data
batch	Array	Store batch data
userId	String	Store user id data
authority	String	Store authority of user
enrollno	String	Store enroll number data
course	String	Store course data
divison	String	Store division data
year	String	Store year data
mentor	String	Store mentor data

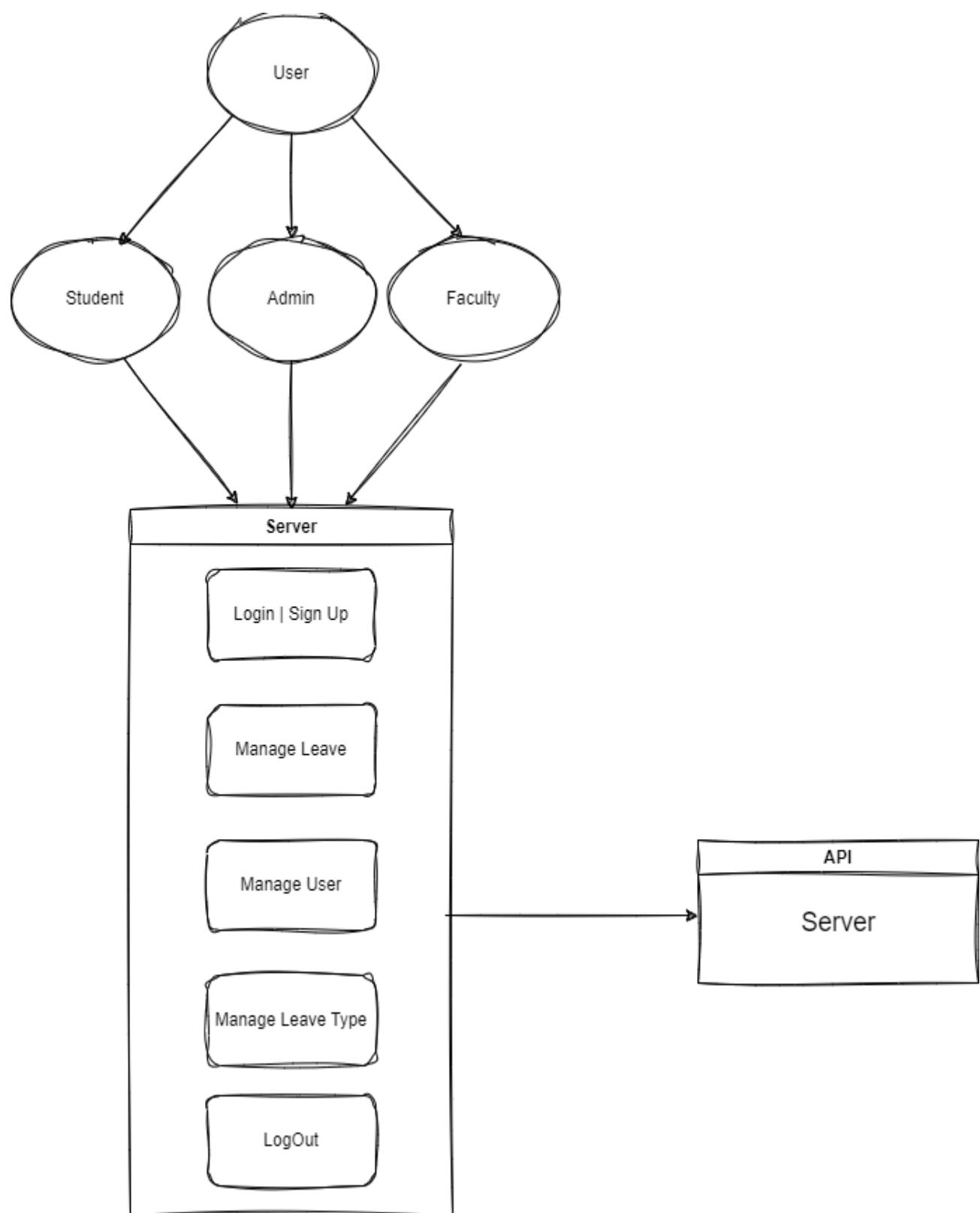
Table Name: leaves (Stores Leaves Types Details)

Attribute	Data Type	Description
_id	ObjectId	Stores ID
typeOfLeave	String	Stores type of leave data
numberOfLeave	Int32	Stores number of leave data
authority	Array	Stores authority data

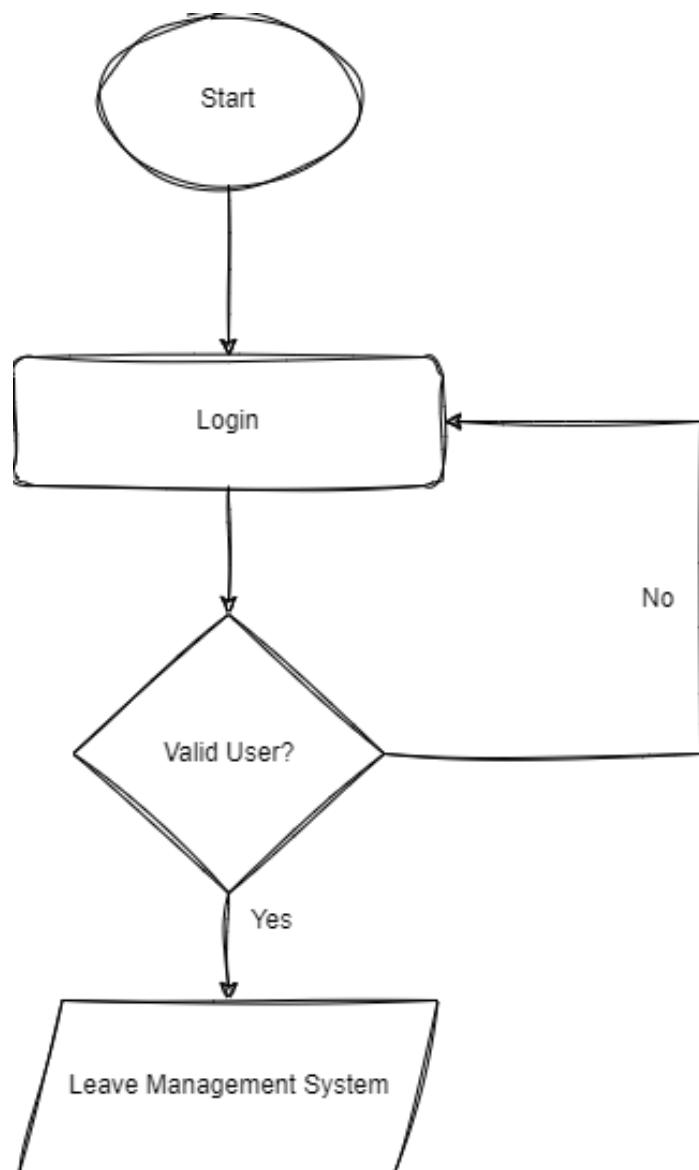
Table Name: leave.details (Stores Leave Details)

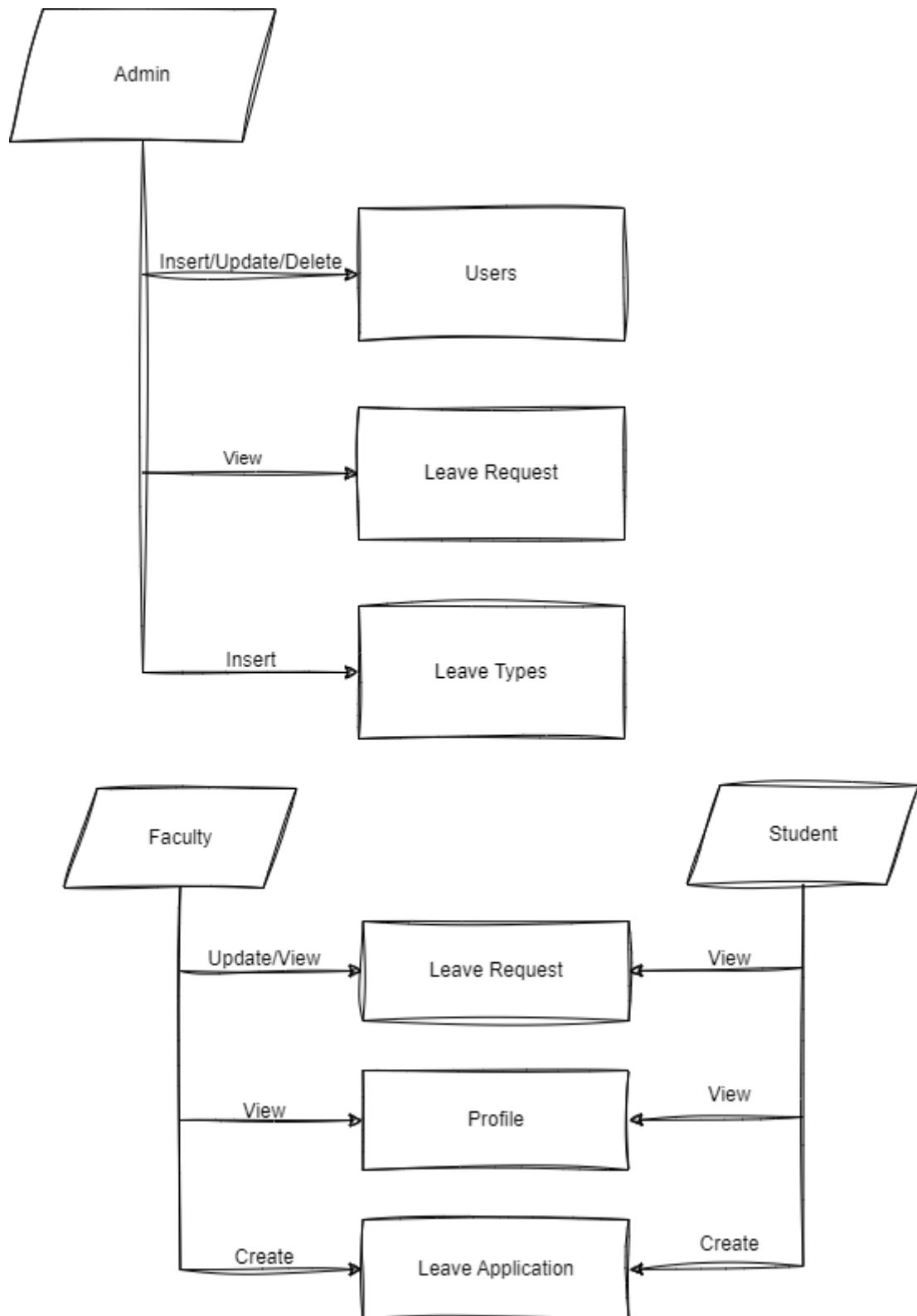
Attribute	Data Type	Description
_id	ObjectId	Stores ID
userInfo	Object	Stores user type,id and email
typeofLeave	String	Stores type of leave data
dateFrom	Date	Stores date from data
dateTo	Date	Stores date to data
ttlLeaves	Int32	Stores total leaves data
description	String	Stores description data
handleBy	String	Stores name of faculty member who approve or reject leave
CoOrdinatorStatus	String	Stores data of leave status only if leave type is ncc,nss or sports
status	String	Stores leave status data
attachment	Object	Stores file attachment data

➤ System Architecture:-



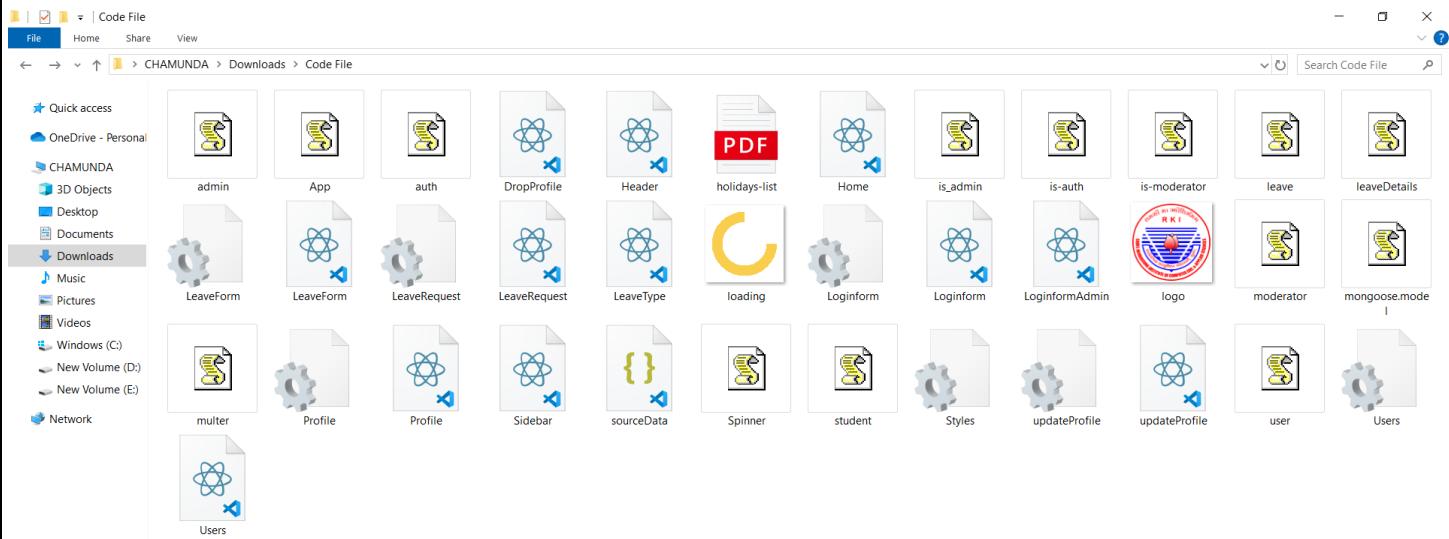
➤ System Flowchart:-





5. Coding

➤ System Code:-



37 items

Type here to search

File Edit Selection View Go Run Terminal Help ← → 🔍 SLM

EXPLORER

- SLM
 - client
 - src
 - Leaves
 - LeaveForm.css
 - LeaveForm.jsx
 - LeaveRequest.css
 - LeaveRequest.jsx
 - LeaveType.jsx
 - Login
 - Loginform.css
 - Loginform.jsx
 - LoginformAdmin.jsx
 - Profile
 - Profile.css
 - Profile.jsx
 - updateProfile.css
 - updateProfile.jsx
 - Register
 - Registerform.css
 - RegisterForm.jsx
 - RegisterModerator.jsx
 - App
 - App.js
 - App.test.js
 - auth.js
 - Dashboard.jsx
 - DataContext.js
 - index.css
 - index.js
 - logo.svg

updateProfile.jsx LeaveRequest.jsx LeaveForm.jsx Home.jsx Loginform.jsx RegisterForm.jsx App.js

```

client > src > App > App > render
1 import React, { useState } from "react";
2 import {
3   BrowserRouter as Router,
4   Routes,
5   Route,
6   Navigate,
7   Outlet,
8 } from "react-router-dom";
9 import Loginform from "./Login/Loginform";
10 import LoginformAdmin from "./Login/LoginformAdmin";
11 import RegisterForm from "./Register/RegisterForm";
12 import Dashboard from "./Dashboard";
13 import Profile from "./Profile/Profile";
14 import Users from "./Components/Users";
15 import { DataProvider } from "./DataContext";
16 import LeaveForm from "./Leaves/LeaveForm";
17 import LeaveRequest from "./Leaves/LeaveRequest";
18 import LeaveType from "./Leaves/LeaveType";
19 import { isLoggedIn } from "./auth";
20 import UpdateProfile from "./Profile/updateProfile";
21
22 class App extends React.Component {
23   constructor(props) {
24     super(props);
25     this.state = {
26       apiresponse: "",
27       openSidebarToggle: false,
28       loading: false,
29     };
30   }
31
32   render() {
33     return (
34       <DataProvider>
35         <React.Fragment>
36           <Router>
37             <Routes>

```

Ln 41, Col 74 Spaces: 2 UTF-8 LF {} JavaScript

Type here to search

```

File Edit Selection View Go Run Terminal Help < > ⚡ SLM
EXPLORER ... updateProfile.jsx LeaveRequest.jsx LeaveForm.jsx Home.jsx Loginform.jsx RegisterForm.jsx App.jsx
SLM
client
src
Leaves
LeaveForm.css
LeaveForm.jsx
LeaveRequest.css
LeaveRequest.jsx
LeaveType.jsx
Login
LoginForm.css
LoginForm.jsx
LoginFormAdmin.jsx
Profile
Profile.css
Profile.jsx
updateProfile.css
updateProfile.jsx
Register
Registerform.css
RegisterForm.jsx
RegisterModerator.jsx
App.css
App.js
App.test.js
auth.js
Dashboard.jsx
DataContext.js
index.css
index.js
logo.svg
> OUTLINE
> TIMELINE
Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript JSX 17:08 20-May-24
Type here to search

```

```

File Edit Selection View Go Run Terminal Help < > ⚡ SLM
EXPLORER ... updateProfile.jsx LeaveRequest.jsx LeaveForm.jsx Home.jsx Loginform.jsx RegisterForm.jsx App.jsx
SLM
client
src
Leaves
LeaveForm.css
LeaveForm.jsx
LeaveRequest.css
LeaveRequest.jsx
LeaveType.jsx
Login
LoginForm.css
LoginForm.jsx
LoginFormAdmin.jsx
Profile
Profile.css
Profile.jsx
updateProfile.css
updateProfile.jsx
Register
Registerform.css
RegisterForm.jsx
RegisterModerator.jsx
App.css
App.js
App.test.js
auth.js
Dashboard.jsx
DataContext.js
index.css
index.js
logo.svg
> OUTLINE
> TIMELINE
Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript JSX 17:08 20-May-24
Type here to search

```

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows the project structure. The root folder contains `backend`, `api`, `client`, `public`, and `src`. `src` contains `Components` (with files like `DropProfile.jsx`, `Header.jsx`, `Home.jsx`, `Sidebar.jsx`, `sourceData.json`, `Spinner.jsx`, `Styles.css`, `Users.css`, and `Users.jsx`), `Files`, `images`, `Leaves` (with files like `LeaveForm.css`, `LeaveForm.jsx`, `LeaveRequest.css`, and `LeaveRequest.jsx`), and `OUTLINE` and `TIMELINE` sections.
- Code Editor (Center):** Displays the `Home.jsx` file. The code is a functional component that imports various hooks and state management libraries from `react`, `chart.js`, `@mui`, and `axios`. It handles token retrieval from local storage, initializes state variables for leave counts, and uses `useEffect` to log the token payload and perform a console log. The code editor has syntax highlighting for JavaScript and JSX.
- Status Bar (Bottom):** Shows the current file as `Home.jsx`, line 1, column 1. It also includes icons for file operations, a search bar, and system status indicators like battery level and signal strength.

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows the project structure. The 'LeaveForm.css' file is currently selected.
- Editor (Center):** Displays the 'LeaveForm.jsx' component code in JSX. The code handles form submission to a local endpoint 'http://localhost:9000/user/leaveForm' using axios.
- Status Bar (Bottom Right):** Shows the current file is 'LeaveForm.jsx', line 1, column 1. It also displays the date as '20-May-24'.

```
import React, { useState, useEffect } from "react";
import Header from "../Components/Header";
import Sidebar from "../Components/sidebar";
import "./Leaves/LeaveForm.css";
import axios from "axios";
import Spinner from "../Components/Spinner";
import { ToastContainer, toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";

function LeaveForm(props) {
  const headers = { authorization: `Bearer ${localStorage.getItem('token')}` };
  const [isLoading, setLoading] = useState(false);
  const [data, setData] = useState({
    userName: "",
    userEnrollNo: "",
    Mentor: "",
    dateFrom: "",
    dateTo: "",
    typeOfLeave: "",
    attachment: "",
    description: "",
    userAuthority: "",
    division: ""
  });

  const unregister = async (e) => {
    e.preventDefault();
    const formData = new FormData();
    formData.append("dateFrom", data.dateFrom);
    formData.append("dateTo", data.dateTo);
    formData.append("typeOfLeave", data.typeOfLeave);
    formData.append("description", data.description);
    formData.append("attachment", data.attachment);
    try {
      await axios
        .post(`http://localhost:9000/user/leaveForm`, formData, {
          headers: {
            ...headers
          }
        })
        .then(res => {
          if (res.data.success) {
            toast.success(res.data.message);
            props.history.push("/leaves");
          } else {
            toast.error(res.data.message);
          }
        })
        .catch(error => {
          toast.error(error.message);
        });
    } catch (error) {
      toast.error(error.message);
    }
  };
}

export default LeaveForm;
```

File Edit Selection View Go Run Terminal Help ← → ⌘ SLM

EXPLORER updateProfile.jsx LeaveRequest.jsx Home.jsx Loginform.jsx RegisterForm.jsx App.jsx

client > src > Leaves > LeaveRequest.jsx > ...

```
1 import React, { useState, useEffect } from "react";
2 import {
3   Box,
4   VStack,
5   Text,
6   useDisclosure,
7   Modal,
8   ModalOverlay,
9   ModalContent,
10  ModalHeader,
11  ModalCloseButton,
12  ModalBody,
13  Table,
14  Head,
15  Tbody,
16  Input,
17  Button,
18  Tbody,
19  Tr,
20  Th,
21  Td,
22  useToast,
23 } from "@chakra-ui/react";
24 import Header from "../Components/Header";
25 import Sidebar from "../Components/Sidebar";
26 import axios from "axios";
27 import "../Leaves/LeaveRequest.css";
28 import spinner from "../components/spinner";
29 import { ToastContainer, toast } from "react-toastify";
30 import "react-toastify/dist/ReactToastify.css";

31 const LeaveRequest = () => {
32   const headers = { authorization: `Bearer ${localStorage.getItem(`token`)} ` };
33   const [loading, setLoading] = useState(true);
34   const [leaves, setLeaves] = useState([]);
35   const [count, setCount] = useState(0);
36   const pending = 0;
37 }
```

OUTLINE
TIMELINE

0 0 0 Type here to search

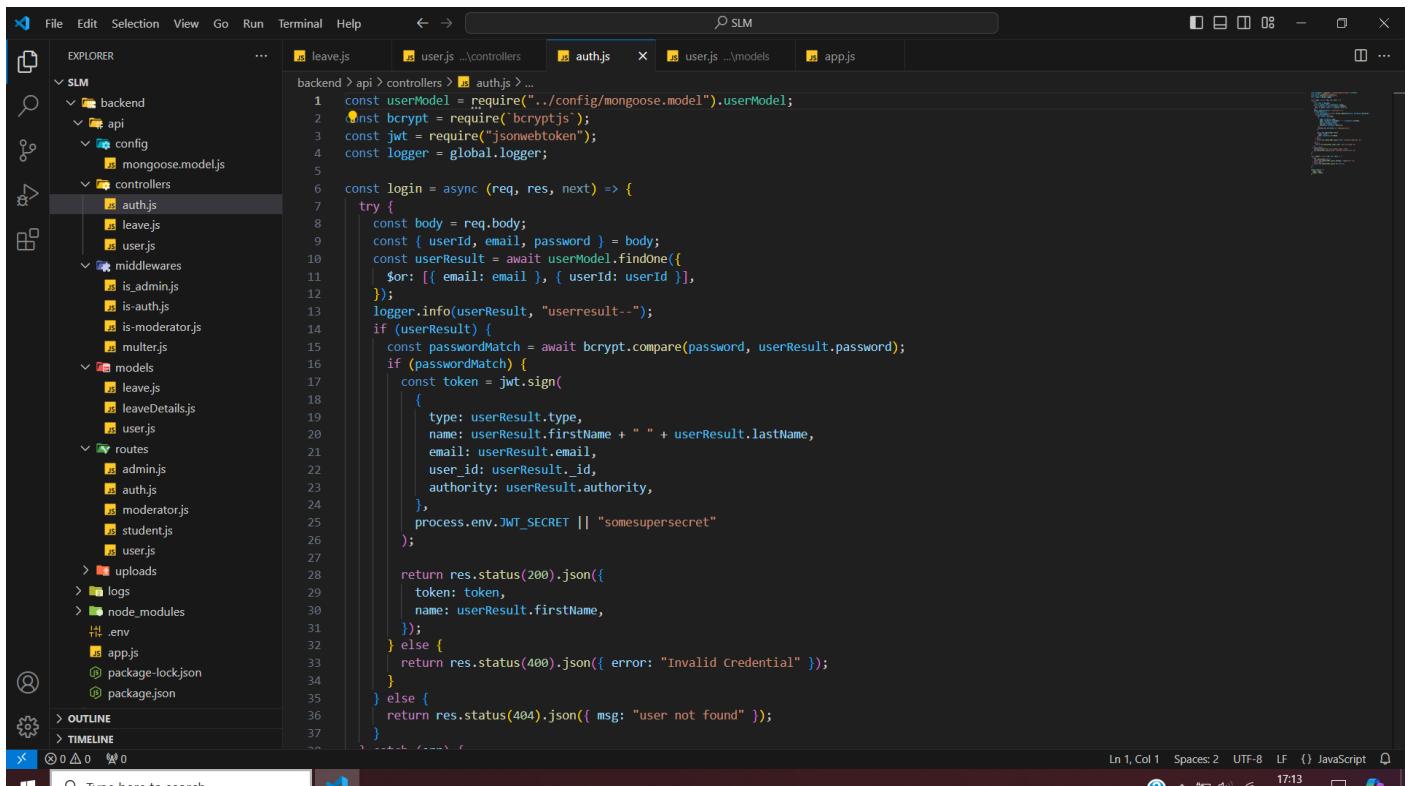
Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript JSX 17/08 20-May-24

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** SLM
- Left Sidebar (Explorer):**
 - SLM
 - client
 - src
 - Leaves
 - LeaveForm.css
 - LeaveForm.jsx
 - LeaveRequest.css
 - LeaveRequest.jsx
 - LeaveType.jsx
 - Login
 - Loginform.css
 - Loginform.jsx
 - LoginformAdmin.jsx
 - Profile
 - Profile.css
 - Profile.jsx
 - updateProfile.css
 - updateProfile.jsx
 - Register
 - Registerform.css
 - RegisterForm.jsx
 - RegisterModerator.jsx
 - App.css
 - App.js
 - App.test.js
 - auth.js
 - Dashboard.jsx
 - DataContext.js
 - index.css
 - index.js
 - logo.svg
 - OUTLINE
 - TIMELINE
- Central Area (Code Editor):** The file `updateProfile.jsx` is open, showing React code for updating a user profile. It imports various components and axios, and defines a state object for profile data and a mentor ID. It uses useEffect to fetch mentor data from a local host API.
- Bottom Status Bar:** Ln 1, Col 1, Spaces: 2, UTF-8, LF, JavaScript JSX, 17/08, 20-May-24, 100%

```
File Edit Selection View Go Run Terminal Help < > ⌘ SLM
EXPLORER ... leave.js user.js ...controllers auth.js user.js ...models app.js x
backend > app.js > ...
1 const express = require("express");
2 const cors = require("cors");
3 const cron = require("node-cron");
4 const mongoose = require("mongoose");
5 const app = express();
6 const log4js = require("log4js");
7 const logger = log4js.getLogger();
8 log4js.configure({
9   appenders: {
10     console: { type: "console" },
11     file: {
12       type: "file",
13       filename: "logs/app.log",
14       maxLogSize: 10485760,
15       backups: 3,
16       compress: true,
17     },
18   },
19   categories: {
20     default: { appenders: ["console", "file"], level: "info" },
21   },
22 });
23 global.logger = logger;
24
25 const studentRoutes = require("./api/routes/student");
26 const adminRoutes = require("./api/routes/admin");
27 const moderatorRoutes = require("./api/routes/moderator");
28 const authRoutes = require("./api/routes/auth");
29 const userRoutes = require("./api/routes/user");
30 // const resetLeave = require("./api/controllers/student"); //sefsefsefsefs
31 const MONGO_URL = `mongodb+srv://vansh1501:StudentLeave@cluster0.zzxw7zq.mongodb.net/sleave`;
32 const isAuthenticated = require("./api/middlewares/is-auth");
33
34 const (async () => {
35   try {
36     await app.listen(3001, () => {
37       console.log(`Server is running on port 3001`);
38     });
39   } catch (error) {
40     console.error(error);
41   }
42 })();
43
44 module.exports = app;
```

```
const mongoose = require("mongoose");
const schema = new mongoose.Schema({
  type: {
    type: String,
    required: true,
    enum: ["Student", "Admin", "Moderator"],
  },
  userId: { type: String, unique: true, required: false, sparse: true },
  email: { type: String, required: true, unique: true },
  authority: {
    type: String,
    enum: [
      "Admin",
      "Principal",
      "H.O.D",
      "Mentor",
      "Faculty",
      "Co-ordinator",
      "Student",
      "NCC Student",
      "NSS Student",
      "Sports Student",
    ],
  },
  firstName: { type: String, required: true },
  lastName: { type: String, required: true },
  enrollno: {
    type: String,
    unique: true,
    sparse: true,
    validate: {
      validator: function (v) {
        return /^[S][R][0-9]{9}$/.test(v);
      },
      message: (props) => `${props.value} is not a valid enroll number!`,
    },
  },
});
```



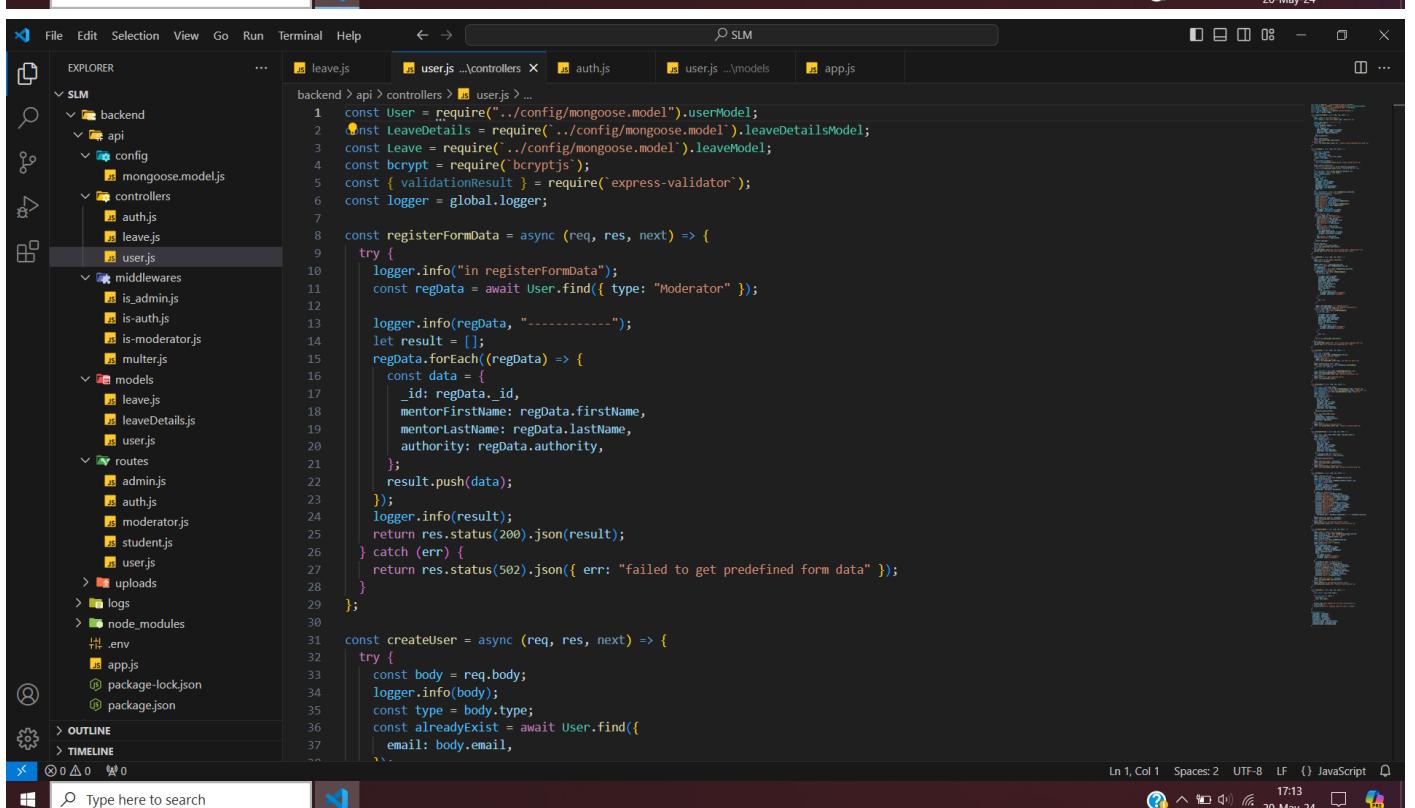
```

File Edit Selection View Go Run Terminal Help ← → ⌂ SLM
leave.js user.js ...controllers auth.js user.js ...models app.js
backend > api > controllers > auth.js ...
1 const userModel = require("../config/mongoose.model").userModel;
2 const bcrypt = require('bcryptjs');
3 const jwt = require("jsonwebtoken");
4 const logger = global.logger;
5
6 const login = async (req, res, next) => {
7   try {
8     const body = req.body;
9     const { userId, email, password } = body;
10    const userResult = await userModel.findOne([
11      $or: [{ email: email }, { userId: userId }],
12    ]);
13    logger.info(userResult, "userresult--");
14    if (userResult) {
15      const passwordMatch = await bcrypt.compare(password, userResult.password);
16      if (passwordMatch) {
17        const token = jwt.sign(
18          {
19            type: userResult.type,
20            name: userResult.firstName + " " + userResult.lastName,
21            email: userResult.email,
22            user_id: userResult._id,
23            authority: userResult.authority,
24          },
25          process.env.JWT_SECRET || "somesupersecret"
26        );
27
28        return res.status(200).json({
29          token: token,
30          name: userResult.firstName,
31        });
32      } else {
33        return res.status(400).json({ error: "Invalid Credential" });
34      }
35    } else {
36      return res.status(404).json({ msg: "user not found" });
37    }
38  }
39}

```

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

17:13 20-May-24



```

File Edit Selection View Go Run Terminal Help ← → ⌂ SLM
leave.js user.js ...controllers auth.js user.js ...models app.js
backend > api > controllers > user.js ...
1 const User = require("../config/mongoose.model").userModel;
2 const leaveDetails = require("../config/mongoose.model").leaveDetailsModel;
3 const Leave = require("../config/mongoose.model").leaveModel;
4 const bcrypt = require('bcryptjs');
5 const { validationResult } = require('express-validator');
6 const logger = global.logger;
7
8 const registerFormData = async (req, res, next) => {
9   try {
10     logger.info("in registerFormData");
11     const regData = await User.find({ type: "Moderator" });
12
13     logger.info(regData, "-----");
14     let result = [];
15     regData.forEach((regData) => {
16       const data = {
17         _id: regData.id,
18         mentorFirstName: regData.firstName,
19         mentorLastName: regData.lastName,
20         authority: regData.authority,
21       };
22       result.push(data);
23     });
24     logger.info(result);
25     return res.status(200).json(result);
26   } catch (err) {
27     return res.status(500).json({ err: "failed to get predefined form data" });
28   }
29 }
30
31 const createUser = async (req, res, next) => {
32   try {
33     const body = req.body;
34     logger.info(body);
35     const type = body.type;
36     const alreadyExist = await User.find({
37       email: body.email,
38     });
39     if (!alreadyExist.length) {
40       const user = new User({
41         type: type,
42         firstName: body.firstName,
43         lastName: body.lastName,
44         email: body.email,
45         password: bcrypt.hashSync(body.password, 10),
46         authority: type === "Student" ? "Student" : "Moderator",
47       });
48       await user.save();
49       res.status(201).json({ message: "User created successfully" });
50     } else {
51       res.status(400).json({ error: "Email already exists" });
52     }
53   } catch (err) {
54     logger.error(err.message);
55     res.status(500).json({ err: "Internal Server Error" });
56   }
57 }

```

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

17:13 20-May-24

```
const LeaveDetails = require('../config/mongoose.model').LeaveDetailsModel;
const Leave = require('../config/mongoose.model').LeaveModel;
const User = require('../config/mongoose.model').UserModel;
const fs = require('fs');
const mailer = require('nodemailer');
const path = require('path');
const { upload } = require('../middlewares/multer');

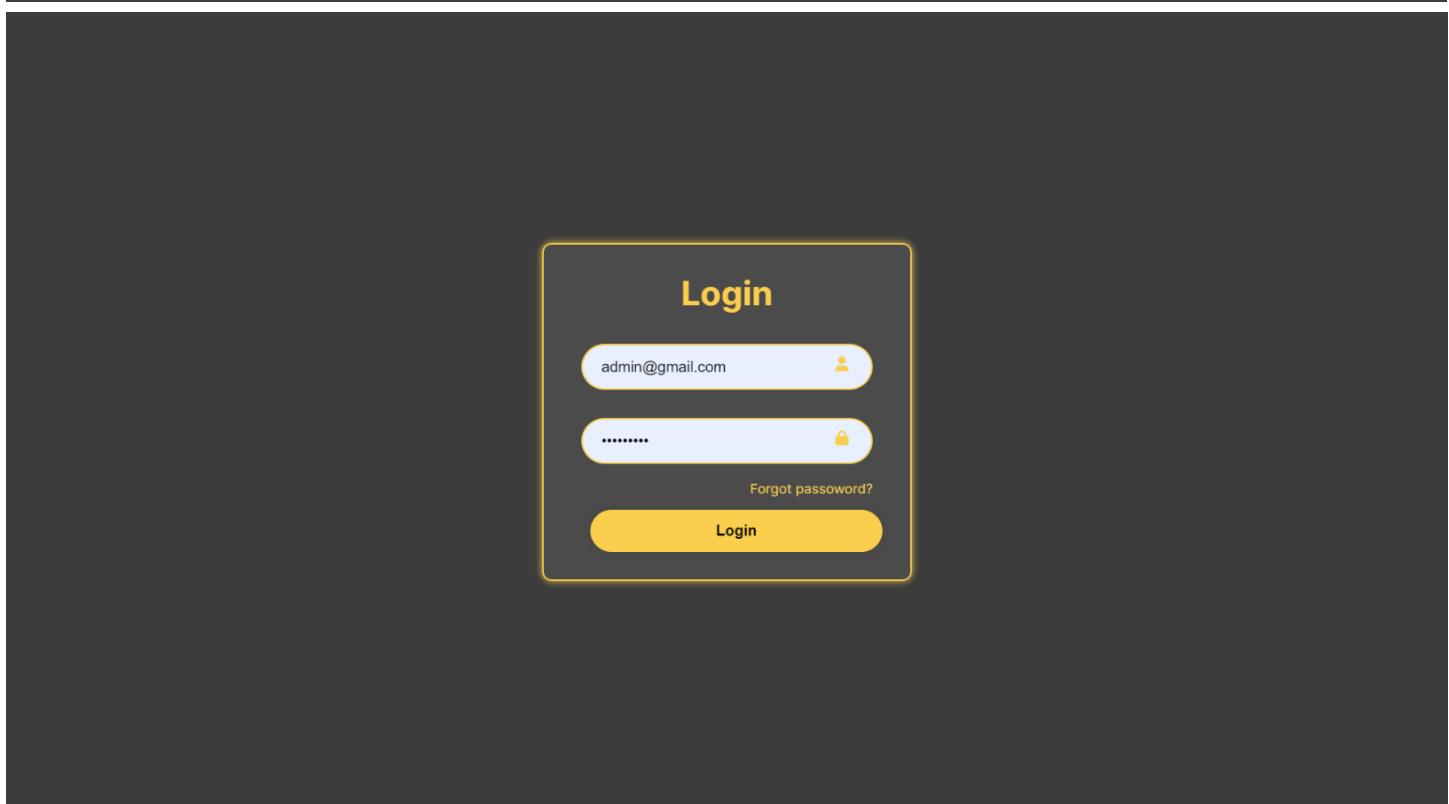
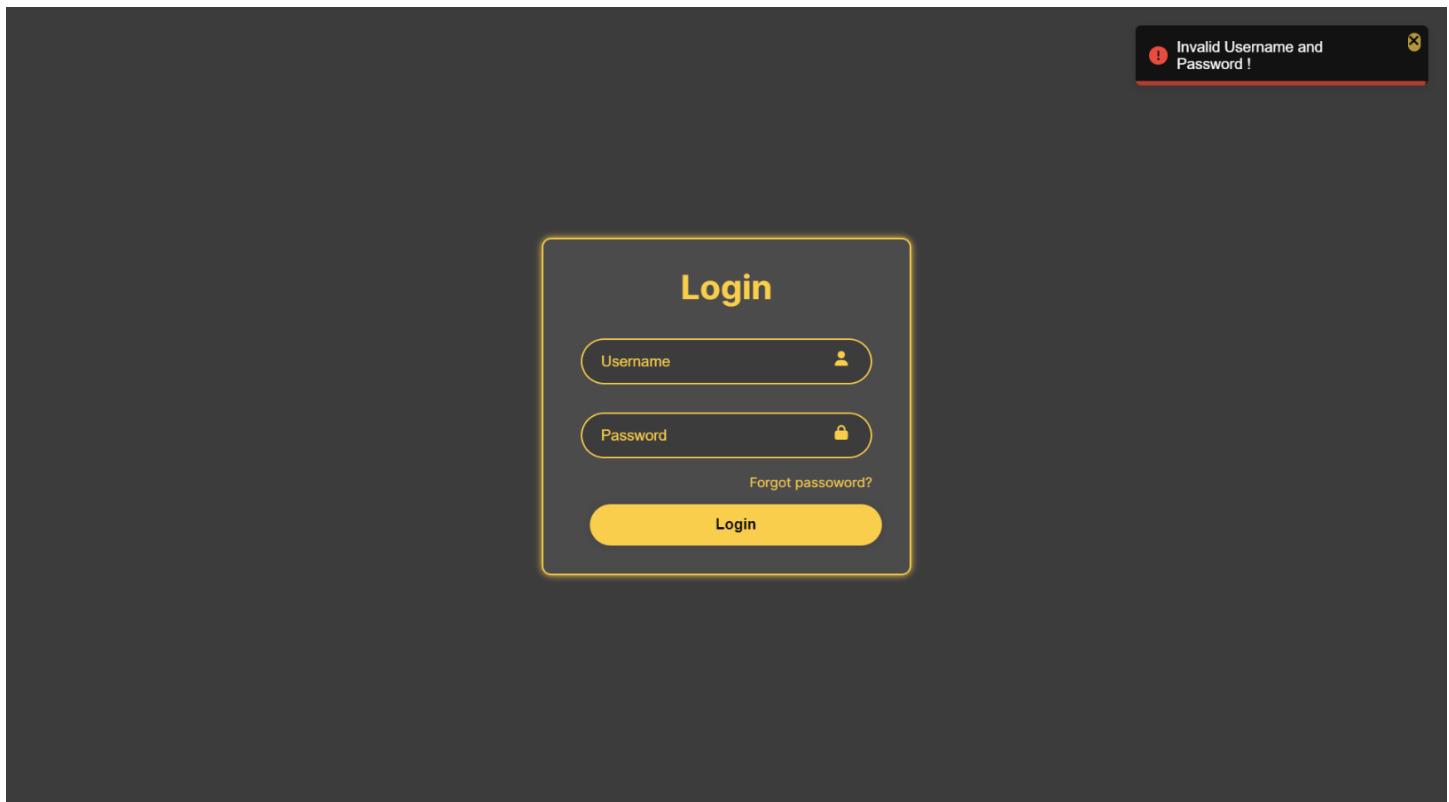
const createLeave = async (req, res, next) => {
  try {
    const body = req.body;
    logger.info(`Leavetype details`, body);
    const leave = new Leave({
      leaveName: body.leaveName,
      leaveDate: body.leaveDate,
      typeOfLeave: "Public Holiday",
      numberofLeave: body.leaveDays,
      authority: body.authority,
    });
    if (body.publicHoliday === true) {
      leave["LeaveName"] = body.leaveName;
      leave["LeaveDate"] = body.leaveDate;
      leave["typeOfLeave"] = "Public Holiday";
    } else if (body.publicHoliday === false) {
      leave["typeOfLeave"] = body.leaveName;
    }
    logger.info(leave);
    const result = await leave.save();

    res.status(200).json(result);
  } catch (err) {
    logger.info(err);
    res.status(400).json({ error: err + "error with leave upload" });
  }
};

const leavetypeDetails = async (req, res, next) => {
  try {
    const leavetype = await Leave.find({
      typeOfLeave: { $ne: "Public Holiday" },
    });
  } catch (err) {
    logger.info(err);
    res.status(400).json({ error: err + "error with leavetype" });
  }
};
```

6. Testing

➤ System Testing:-



SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Mentor:

Details Of Leave

Leave Start:

Leave End:

Leave Type:

Browse Document:

Description:

Leave Applied

Apply

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Mentor:

Details Of Leave

Leave Start:

Leave End:

Leave Type:

Browse Document:

Description:

Leave Limit Reached

Apply

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Pending Requests **2** Approved Requests **4** Rejected Requests **2**

Leaves Requests

Name	Leave Type	From	To	Mentor	Documents	Description	Status
Zalak Patel	Medical	5/22/2024	5/24/2024	Rupal Snehkunj	None	Medical	Approved
Rudra Patel	Vacation	5/22/2024	5/24/2024	Nirali Varu	None	Vaction	Approved
Vansh Khalasi	Others	5/21/2024	5/22/2024	Shruti Revdiwala	None	vansh	Rejected
Zalak Patel	Vacation	5/26/2024	5/28/2024	Rupal Snehkunj	None	a	<button>Approve</button> <button>Reject</button>
Zalak Patel	Vacation	5/22/2024	5/23/2024	Rupal Snehkunj	None	b	<button>Approve</button> <button>Reject</button>

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Leave Types

Public Holiday

Leave Days: Leave Name:

Authority: H.O.D. Faculty Co-Ordinator Mentor

Select: Student Faculty

Create

Leave Created Successfully

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Leave Created Successfully

Leave Types

Public Holiday

Leave Days: Leave Name:

Leave Days:

Leave Date:

Authority: H.O.D. Faculty Co-Ordinator Mentor

Select: Student Faculty

Create

- Dashboard**
- Profile**
- Leave Form**
- Users List**
- Leave Request List**
- Add Leave Type**
- Logout**

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE



Registered Successfully

Student

Registration Form

First Name:	Last Name:
<input type="text" value="Aryan"/>	<input type="text" value="Mavani"/>
Email:	Phone Number:
<input type="text" value="mavani@gmail.com"/>	<input type="text" value="5154848484"/>
Password:	Confirm Password:
<input type="password" value="*****"/>	<input type="password" value="*****"/>
Enrollment No:	Department:
<input type="text" value="SR21BSIT001"/>	<input type="text" value="MB"/>
Course:	Year:
<input type="text" value="BSC"/>	<input type="text" value="SY"/>
Division:	Mentor:
<input type="text" value="B"/>	<input type="text" value="Jayesh Pushtiwala (H.O.D.)"/>
Authority:	Register
<input type="text" value="Regular"/>	

- Dashboard**
- Profile**
- Leave Form**
- Users List**
- Leave Request List**
- Add Leave Type**
- Logout**



SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE

Moderator

Registration Form

First Name:

Last Name:

Email:

Phone Number:

Password:

Confirm Password:

User:

Authority:

Department:

Batch:

Mentor:



SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE

Updated Successfully

Update Form

First Name:

Last Name:

Enrollment No:

Phone Number:

Department:

Mentor:



- [Dashboard](#)
- [Profile](#)
- [Leave Form](#)
- [Users List](#)
- [Leave Request List](#)
- [Add Leave Type](#)
- [Logout](#)

SHREE RAMKRISHNA INSTITUTE OF COMPUTER EDUCATION AND APPLIED SCIENCE

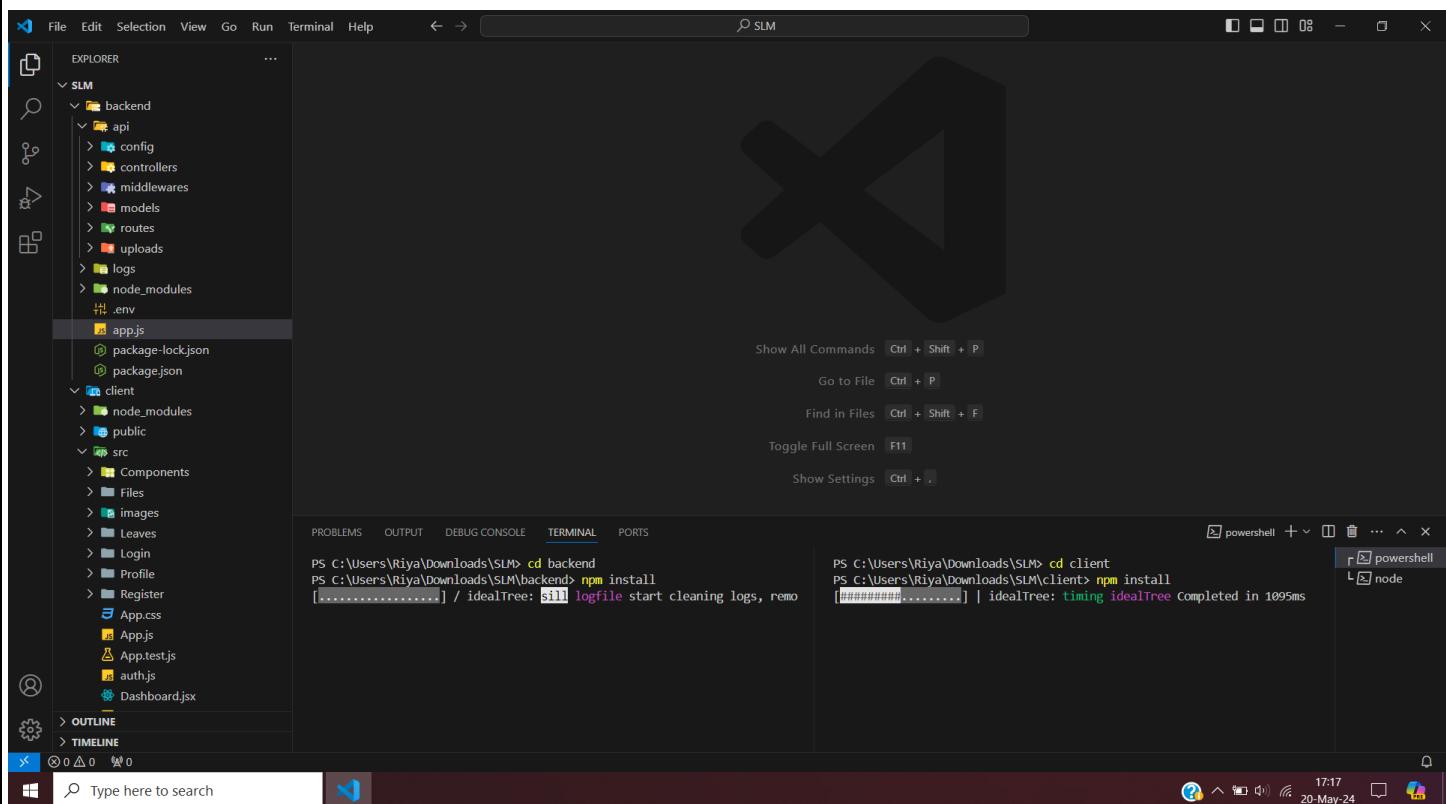
Name	Role	Designation	Email	Department	Action
Chaulami Desai	Moderator	Principal	cd01@gmail.com	CS	Update Delete
Jayesh Pushtiwala	Moderator	H.O.D	jp01@gmail.com	CS	Update Delete
Rupal Vaghela	Moderator	Mentor	rs01@gmail.com	CS	Update Delete
Nidhi Vaniyawala	Moderator	Faculty	nv01@gmail.com	IT	Update Delete
Nirali Varu	Moderator	Faculty	nv02@gmail.com	IT	Update Delete
Shruti Revdiwala	Moderator	Faculty	sr01@gmail.com	IT	Update Delete
Zalak Patel	Student	Student	zalakpatel1551@gmail.com	CS	Update Delete
Rudra Patel	Student	Student	prudra62@gmail.com	IT	Update Delete
Vansh Khalasi	Student	Student	vansh1501@gmail.com	IT	Update Delete
Aryan Mavani	Student	Student	mavani@gmail.com	MB	Update Delete

X
Deleted Successfully

7. Project Deployment

➤ Project Deployment:-

1. Open Visual Studio Code
2. Open Project
3. NPM INSTALL
4. NPM RUN BOTH
5. Run the Project



The screenshot shows the Visual Studio Code (VS Code) interface. The left sidebar (Explorer) displays the project structure of 'SLM' with the following files and folders:

- backend
- api
 - config
 - controllers
 - middlewares
 - models
 - routes
 - uploads
- logs
- node_modules
 - .env
- app.js
- package-lock.json
- package.json
- client
 - node_modules
 - public
- src
 - Components
 - Files
 - images
 - Leaves
 - Login
 - Profile
 - Register
 - App.css
 - App.js
 - App.test.js
 - auth.js
 - Dashboard.jsx

The 'TERMINAL' tab is active, showing the command line output:

```
PS C:\Users\Riya\Downloads\SLM> npm install
[#####.....] \ idealTree: timing idealTree Completed in 118ms
```

The status bar at the bottom right shows the date and time as '20-May-24 17:17'.

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface in dark mode. A large, semi-transparent watermark of a stylized 'X' is centered on the screen. The left sidebar contains the Explorer, Search, and Problems views. The Explorer view shows a project structure for 'SLM' with 'backend' and 'client' folders. The 'backend' folder contains 'api', 'config', 'controllers', 'middlewares', 'models', 'routes', 'uploads', 'logs', 'node_modules', '.env', 'app.js', 'package-lock.json', and 'package.json'. The 'client' folder contains 'node_modules', 'public', and 'src' which includes 'Components', 'Files', 'images', 'Leaves', 'Login', 'Profile', 'Register', 'App.css', 'App.jsx', 'App.test.jsx', 'auth.jsx', and 'Dashboard.jsx'. The bottom left also shows Outline and Timeline views. The main workspace has a terminal tab open with the following content:

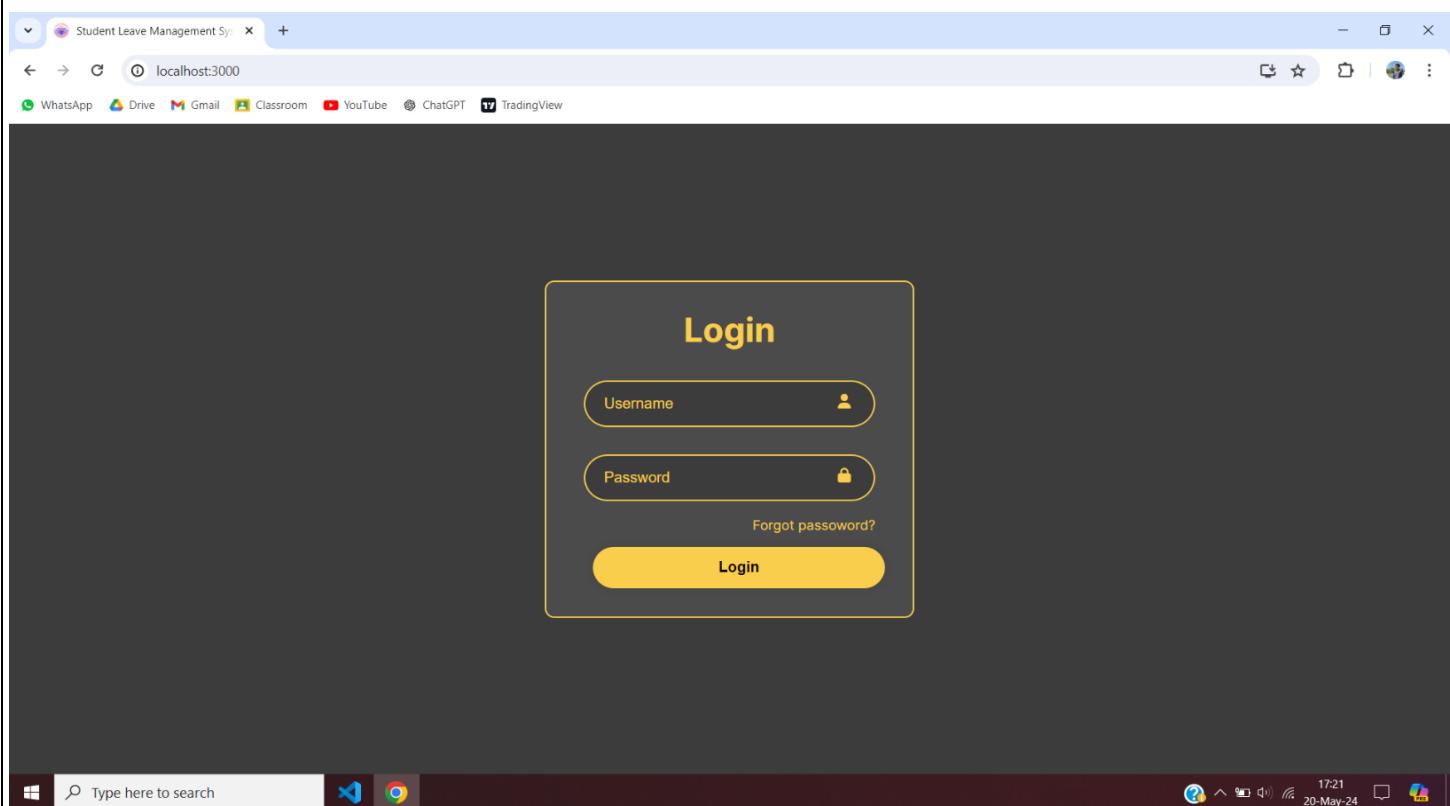
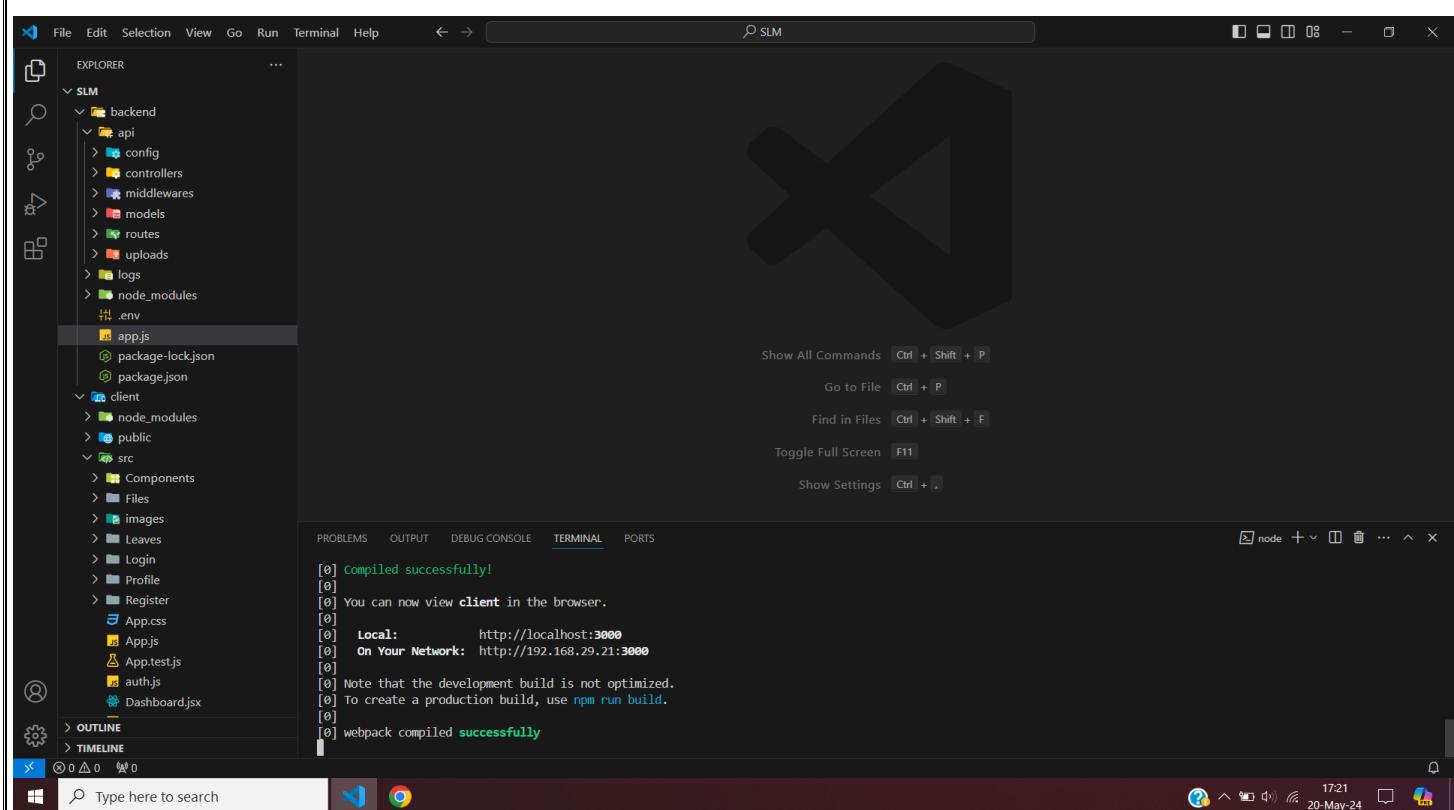
```
up to date, audited 88 packages in 2s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Riya\Downloads\SLM> npm run both

> both
> concurrently "npm run client" "npm run server"

```



8. Future Enhancement

➤ **Future Enhancement: -**

- More efficient design and user experience.
- Apply more efficient constraints.
- Automation for adding yearly public holidays instead of adding manually.
- Add module to change staff member/student password using Mobile No/Email OTP(Forgot Password).
- Add feature of login/signup using google.

9. Bibliography & References

➤ **Bibliography & References: -**

- [Google](#)
- [YouTube](#)
- [ChatGPT](#)
- [Stack Overflow](#)
- [Udemy](#)
- [GitHub](#)
- [Official Documentation](#)