VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“Jnana Sangama”, Belgaum -590014, Karnataka.

Full Stack Web Development Report
On

“ATTENDANCE TRACKER”

Submitted by

ROHAN S M (1WN24CS232)
ROMIT P(1WN24CS233)
ROOPESH S(1WN24CS234)
RUDRA (1WN24CS235)

Under the Guidance of

Dr. Priyanka C Hiremath
Assistant Professor

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sept 2025 – Jan 2026

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated to Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled **"Attendance Tracker"** carried out by Rohan S M(1WN24CS234) Romit P(1WN24CS233) Roopesh S(1WN24CS234) Rudra(1WN24CS235) of **3U** who are bonafide students of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2025-2026. The project report has been approved as it satisfies the academic requirements in respect of **Full Stack Web Development(23CS3AEFWD)** work prescribed for the said degree.

Signature of the Guide	Signature of the HOD
Dr. Priyanka C Hiremath	**Dr. Kavitha Sooda**
Assistant Professor	Prof & Head of Dept of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

External Viva

Name of the Examiner	Signature with date
1)	
2)	

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECALARATION

We,**Rohan**(1WN24CS232),**Romit**(1WN24CS233),**Roopesh**(1WN24CS234), **Rudra**(1WN24CS235) students of 3rd Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this Full Stack Web Development entitled **"Attendance Tracker"** has been carried out by us under the guidance of **Dr. Priyanka C Hiremath**, Assistant Professor, Department of CSE, BMS College of Engineering, Bangalore during the academic semester Sept 2025 - Jan 2026.

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

ROHAN S M (1WN24CS232)

ROMIT P(1WN24CS233)

ROOPESH S(1WN24CS234)

RUDRA (1WN24CS235)

TABLE OF CONTENTS

INTRODUCTION

Attendance management is one of the essential academic activities in any institute to keep track of students' participation and academic regularity. However, traditional methods of attendance are manual, time-consuming, and inefficient. With increased usage of Web technologies, simple, interactive digital attendance system is being felt. The **Attendance Tracker** project is a web application developed using modern frontend technologies with a user-friendly interface for managing and viewing attendance. The system focuses on ease of use, cleanliness of design, and role-based interaction for students and teachers.

1.1 Overview

The **Attendance Tracker** web application is a **frontend-based system** developed using **React** and the **Vite development tool**. The application provides **separate interfaces for Students and Teachers**, allowing each user to interact with the system according to their attendance-related requirements. based strictly on the project files, the key features of the application include **role-based login pages for Students and Teachers**, a **student interface for viewing profile and subject-related information**, and a **teacher dashboard for marking attendance**. The system also supports **reviewing previously recorded attendance details**, enabling better tracking and verification of attendance records. the user interface is designed with a **clean and simple layout**, making use of **reusable React components** to ensure consistency across different pages. Data handling within the application is managed using a local data file (`database.js`), which simulates structured attendance-related data for frontend operations. The project also follows a **responsive and well-organized layout structure**, ensuring better usability across different screen sizes. Overall, the application follows a **modular design approach**, where each feature is implemented as an independent React component. This design makes the system easy to understand, maintain, and modify, and also provides flexibility for future enhancements and expansion.

1.2 Motivation

Many learning institutions are using enterprise-level ERP solutions such as Contineo for managing attendance records. Though such solutions are efficient on a learning institution level, they can be complex, vendor-dependent solutions not geared towards learning. The reason for pursuing this project revolves around designing and building an easy, frontend-based system for tracking attendance that would help students and developers understand how an attendance management system works from within. Moreover, the system is a good platform to apply the latest trends in the frontend part, like component architecture, routing, and UI flow. It is also a scalable platform that can be upgraded in the future by incorporating other services such as backend services, authentication systems, and analytics capabilities.

SOFTWARE REQUIREMENTS

2.1 Hardware Requirements

• A personal computer or laptop with the following minimum specifications or higher:
  o Intel Core i3 processor or higher
  o 8 GB RAM (minimum 4 GB supported for basic execution)
  o 500 GB Hard Disk / SSD storage
  o Standard keyboard and mouse
• A stable internet connection (2 Mbps or higher) for development tools, libraries, and documentation access

2.2 Software Requirements

• Operating System
  – Windows 10 / 11, Linux (Ubuntu or similar distributions)

• Frontend Technologies
  – HTML5
  – CSS3
  – JavaScript (ES6)
  – React.js
  – Drawio

• Development Environment / IDE
  – Visual Studio Code

• Backend Technologies
  – Node.js

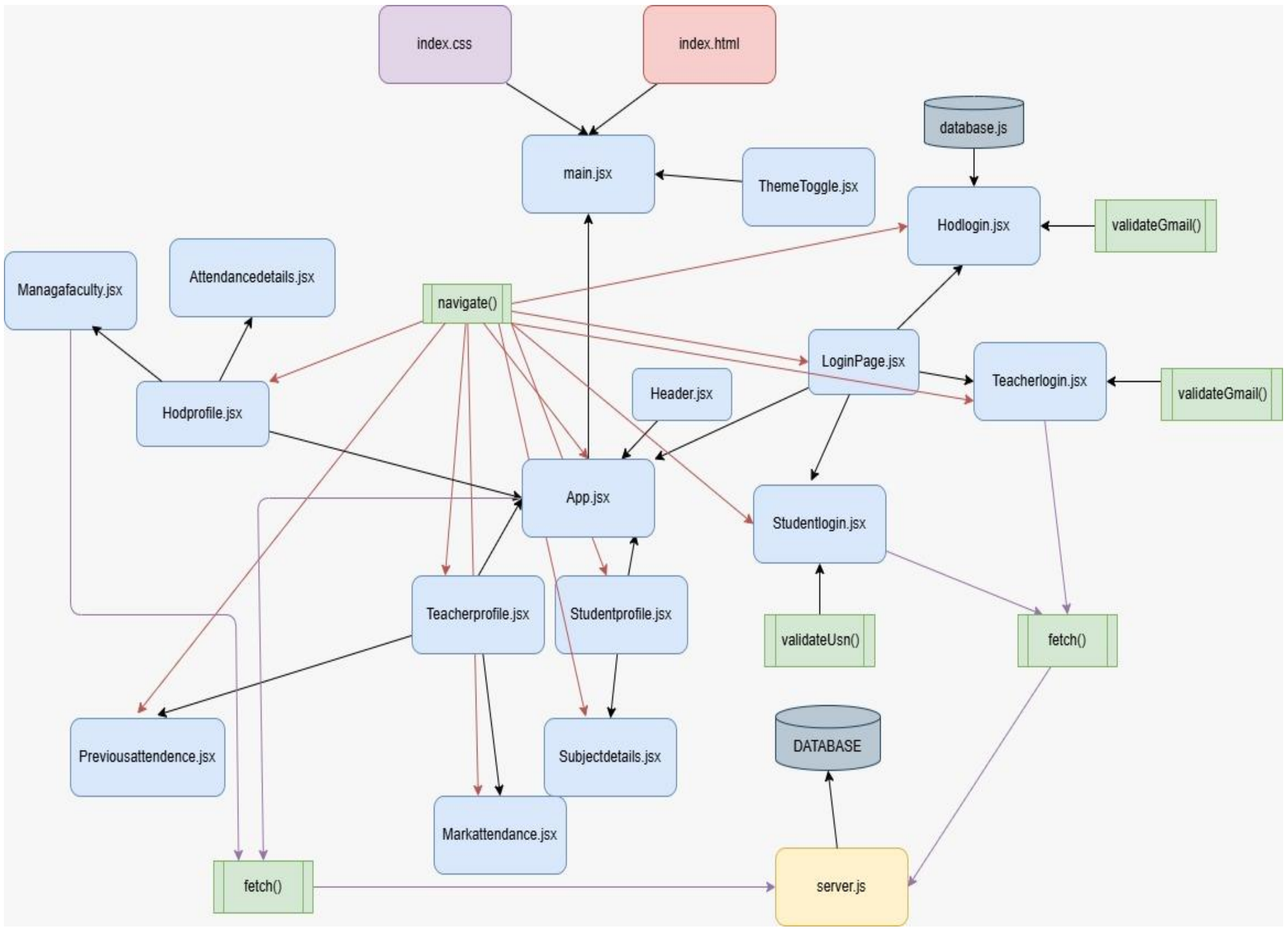
• Server Environment
  – Node.js Runtime

• Package Manager
  – npm (Node Package Manager)

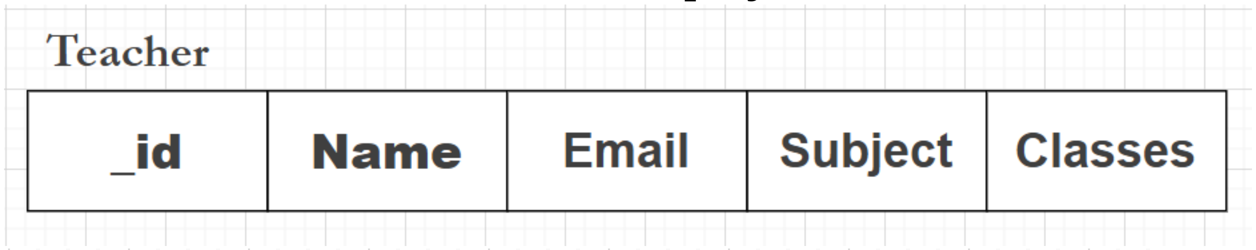
• Web Browser
  – Google Chrome / Mozilla Firefox / Microsoft Edge / Brave

• Version Control
  – Git

3. ER DIAGRAM OF THE PROJECT

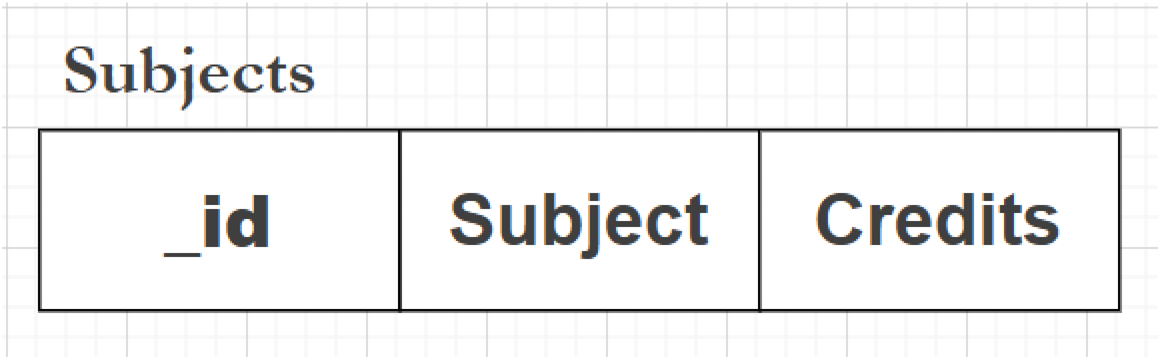


4. Schema of project



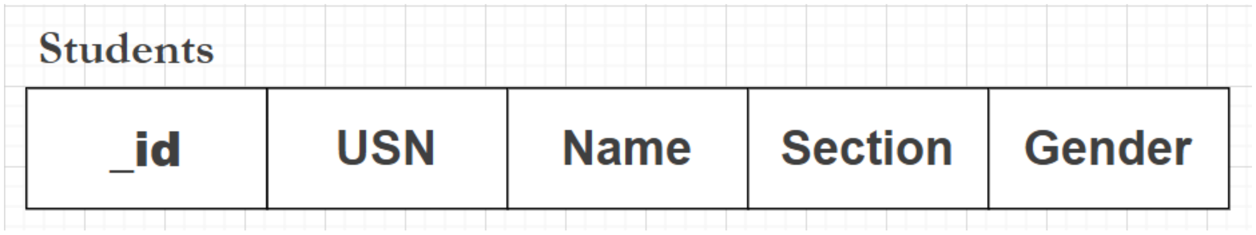
Teacher

_id	Name	Email	Subject	Classes



Subjects

_id	Subject	Credits



Students

_id	USN	Name	Section	Gender

USER INTERFACE DESIGN

There are 4 themes:

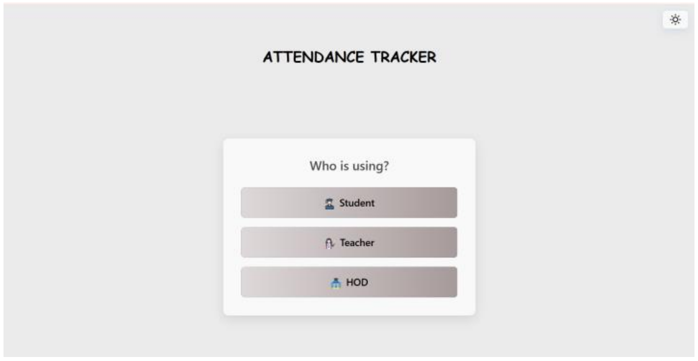


Fig 1. White theme: Black text with white background

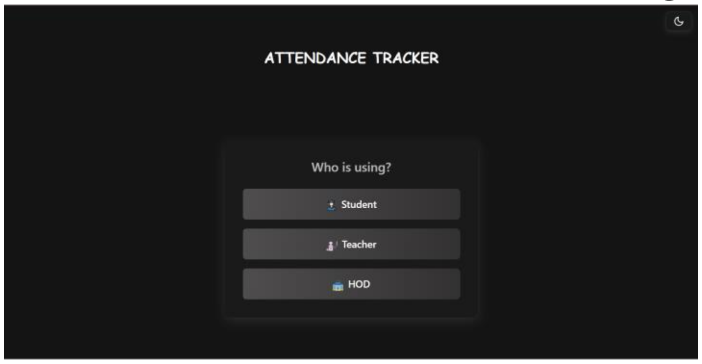


Fig 2. Black theme: White text with black background

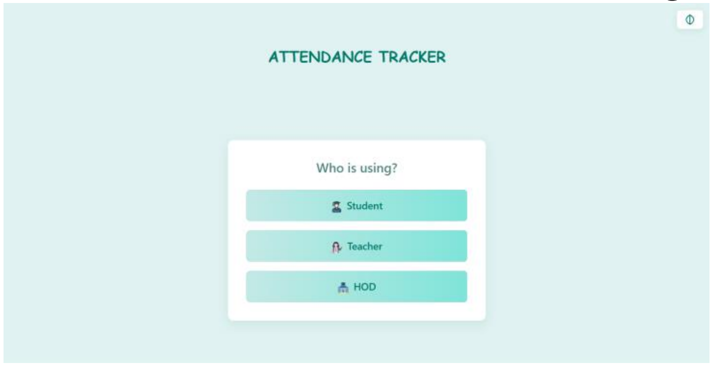


Fig 3. Green theme: Green text with light green background

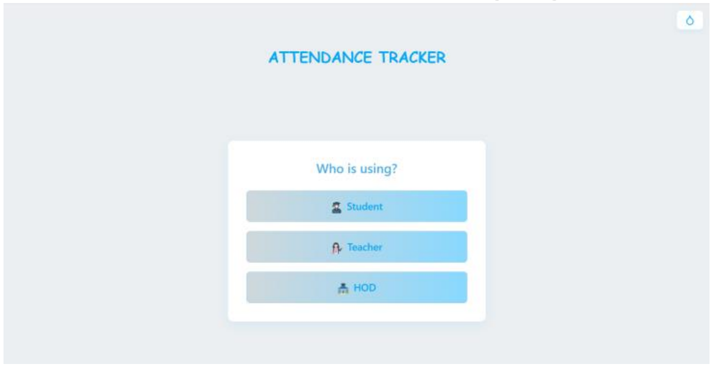
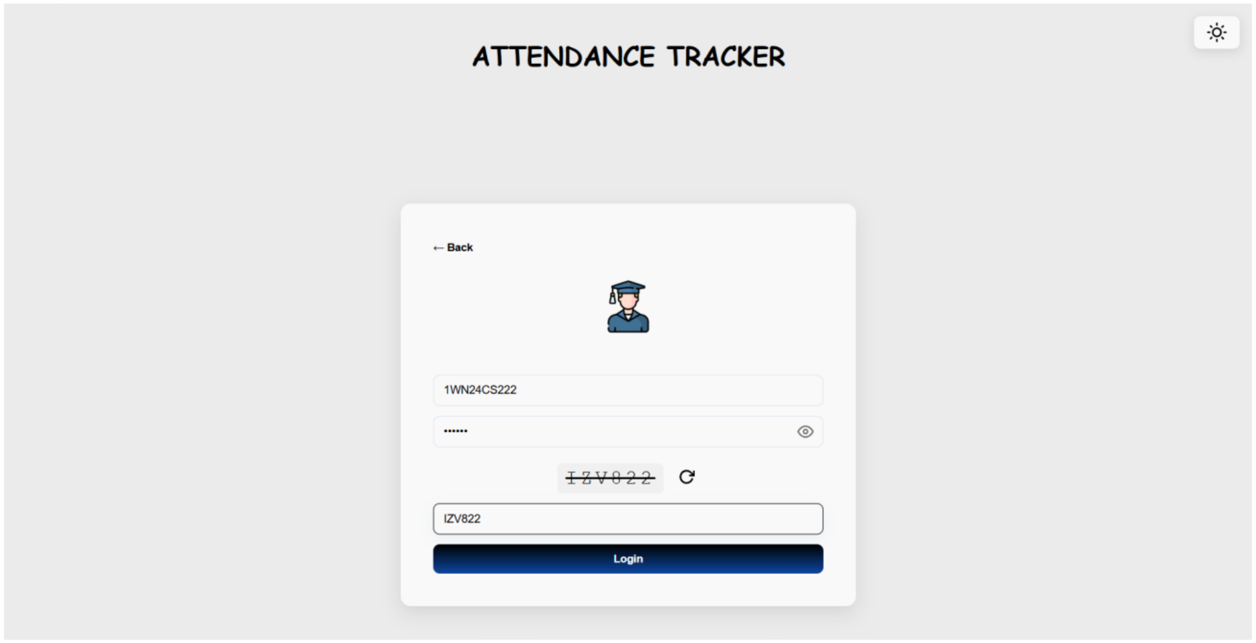


Fig 4. Blue theme: Blue text with grey background

Student view:

Student has to Log in to and can view their attendance in each of the seven subjects.


ATTENDANCE TRACKER

← Back

1WN24CS222

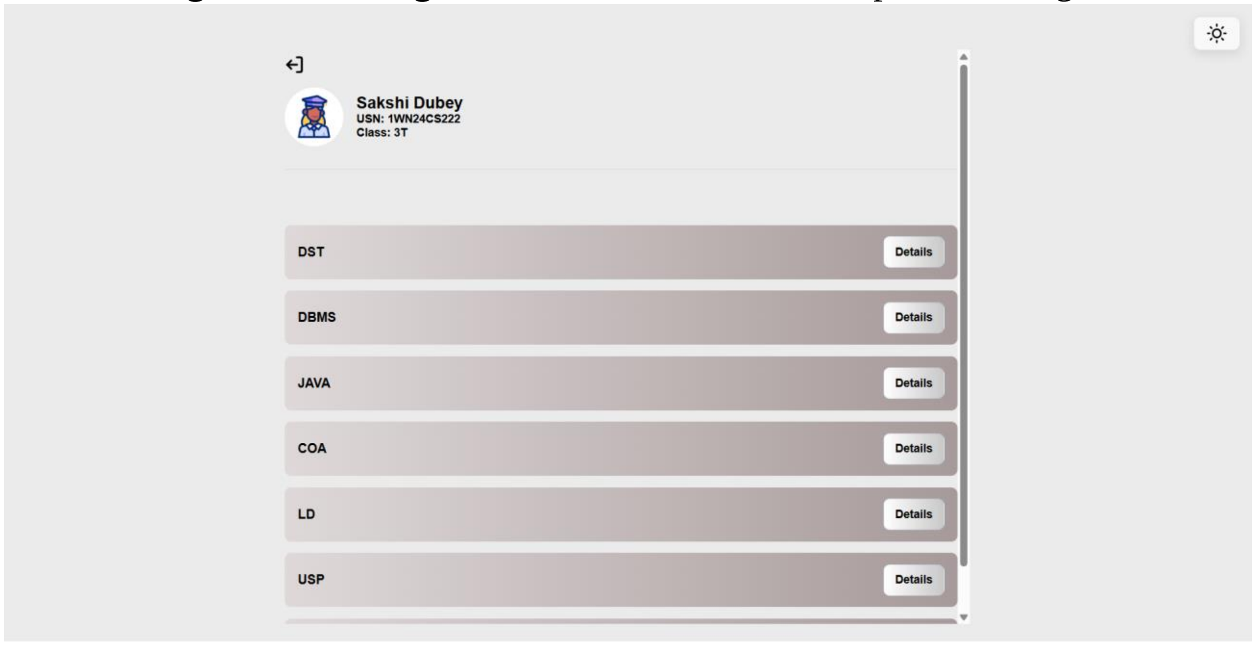
••••••

~~IZV822~~

IZV822

Login

Fig 5. Student login: Enter USN, Password, Captcha and login

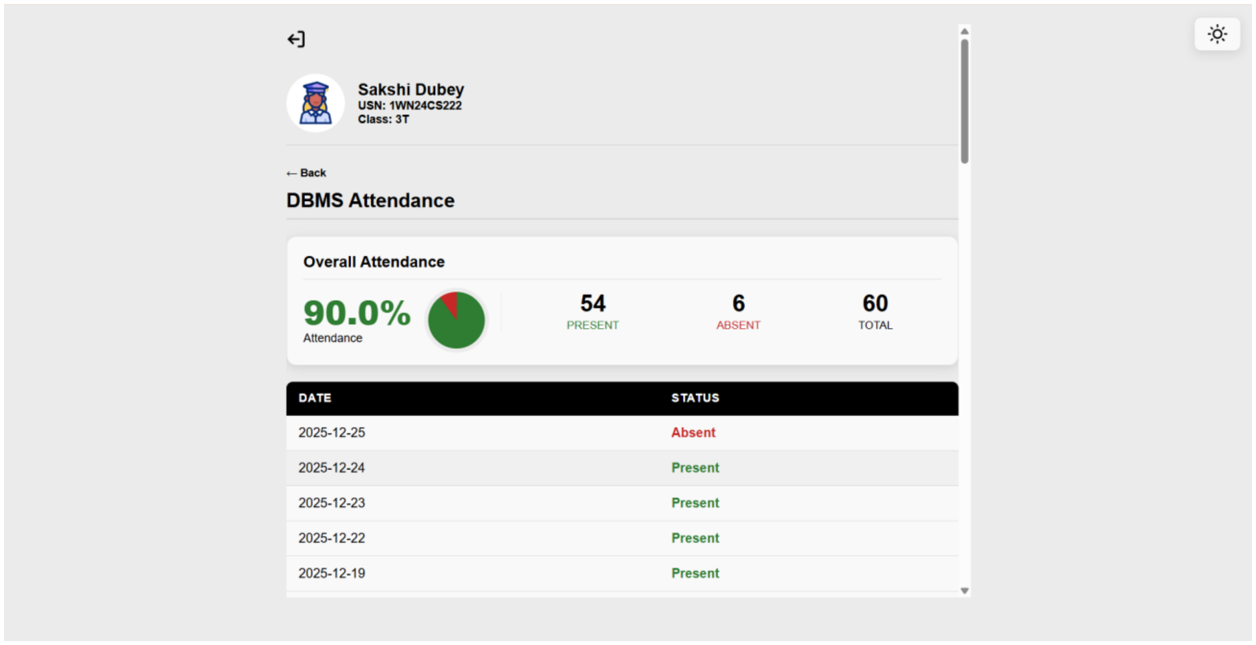

Sakshi Dubey

USN: 1WN24CS222

Class: 3T

DST	Details
DBMS	Details
JAVA	Details
COA	Details
LD	Details
USP	Details

Fig 6. Student profile: Profile is displayed with student name, USN, Class


Sakshi Dubey
USN: 1WN24CS222
Class: 3T

← Back

DBMS Attendance

Overall Attendance

90.0% Attendance

54 PRESENT

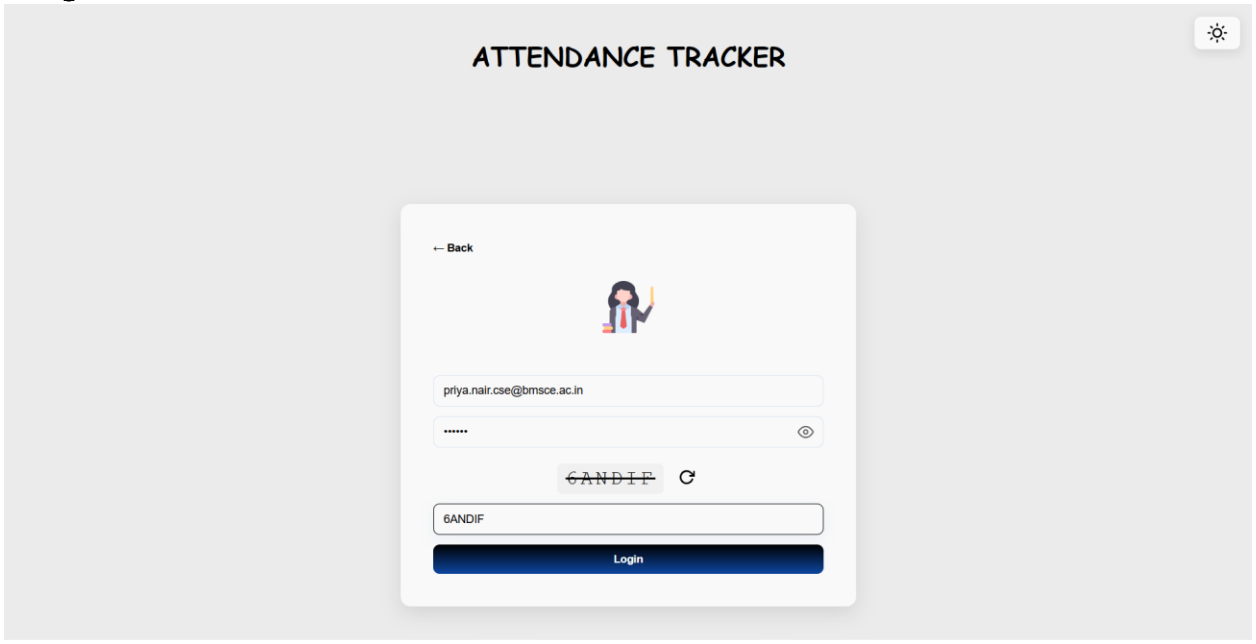
6 ABSENT

60 TOTAL

DATE	STATUS
2025-12-25	Absent
2025-12-24	Present
2025-12-23	Present
2025-12-22	Present
2025-12-19	Present

Fig 7. Attendance per subject: Attendance percentage and details for a each subject with pie-chart and attendance status

Teacher view: Teacher has to log in using college email ID and can mark attendance, update previous attendance of students coming under the classes assigned to them.


ATTENDANCE TRACKER

← Back

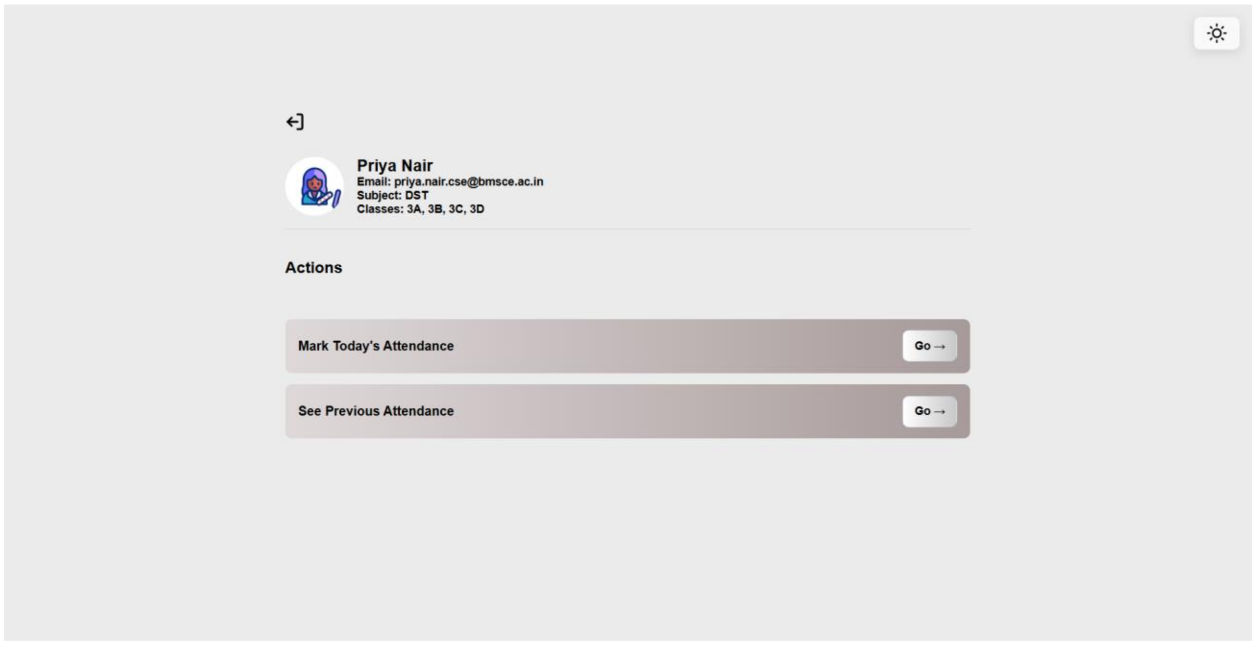
priya.nair.cse@bmsce.ac.in

••••••

6ANDIF

6ANDIF

Fig 8. Teacher login: Enter Email, Password, Captcha



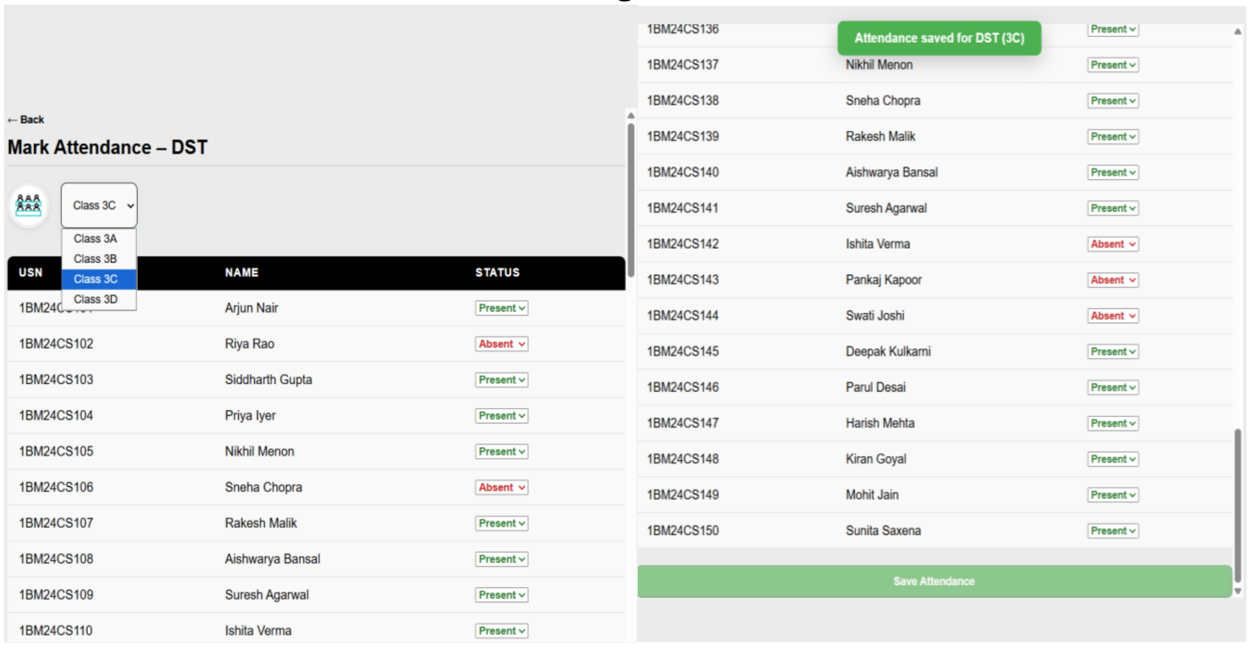
Priya Nair
Email: priya.nair.cse@bmsce.ac.in
Subject: DST
Classes: 3A, 3B, 3C, 3D

Actions

Mark Today's Attendance — Go →

See Previous Attendance — Go →

Fig 9. Teacher profile: Teacher profile has name, email, subject and classes assigned.



← Back

Mark Attendance – DST

Class 3C

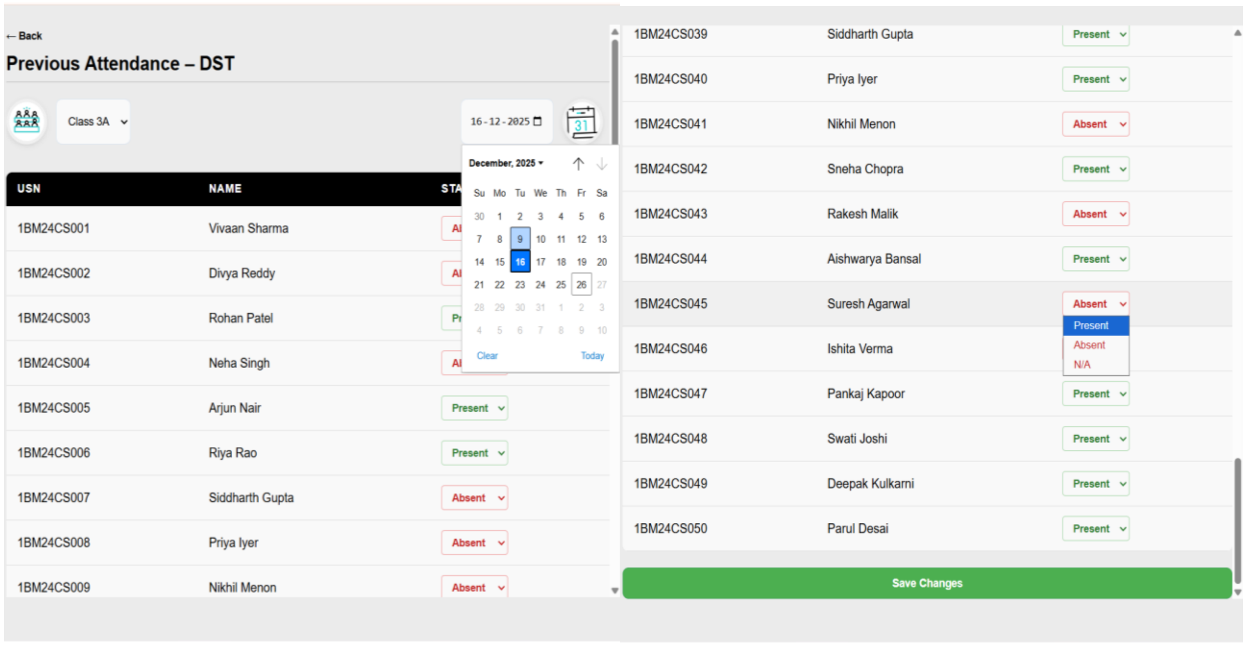
USN	NAME	STATUS
1BM24C	Arjun Nair	Present
1BM24CS102	Riya Rao	Absent
1BM24CS103	Siddharth Gupta	Present
1BM24CS104	Priya Iyer	Present
1BM24CS105	Nikhil Menon	Present
1BM24CS106	Sneha Chopra	Absent
1BM24CS107	Rakesh Malik	Present
1BM24CS108	Aishwarya Bansal	Present
1BM24CS109	Suresh Agarwal	Present
1BM24CS110	Ishita Verma	Present

Attendance saved for DST (3C)

1BM24CS136		Present
1BM24CS137	Nikhil Menon	Present
1BM24CS138	Sneha Chopra	Present
1BM24CS139	Rakesh Malik	Present
1BM24CS140	Aishwarya Bansal	Present
1BM24CS141	Suresh Agarwal	Present
1BM24CS142	Ishita Verma	Absent
1BM24CS143	Pankaj Kapoor	Absent
1BM24CS144	Swati Joshi	Absent
1BM24CS145	Deepak Kulkarni	Present
1BM24CS146	Parul Desai	Present
1BM24CS147	Harish Mehta	Present
1BM24CS148	Kiran Goyal	Present
1BM24CS149	Mohit Jain	Present
1BM24CS150	Sunita Saxena	Present

Save Attendance

Fig 10. Mark present attendance: Teacher has to select one of the four classes assigned to them and mark absentees for the day



← Back

Previous Attendance – DST

Class 3A

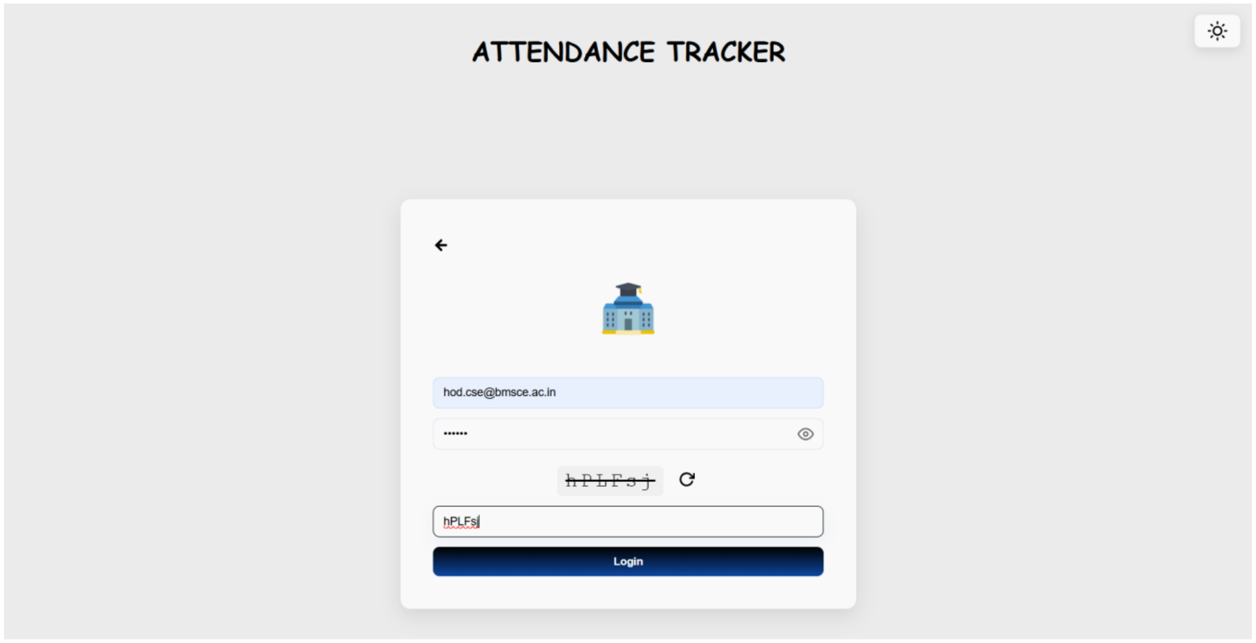
16 - 12 - 2025

USN	NAME	STA
1BM24CS001	Vivaan Sharma	Al
1BM24CS002	Divya Reddy	Al
1BM24CS003	Rohan Patel	Pr
1BM24CS004	Neha Singh	Al
1BM24CS005	Arjun Nair	Present
1BM24CS006	Riya Rao	Present
1BM24CS007	Siddharth Gupta	Absent
1BM24CS008	Priya Iyer	Absent
1BM24CS009	Nikhil Menon	Absent

1BM24CS039	Siddharth Gupta	Present
1BM24CS040	Priya Iyer	Present
1BM24CS041	Nikhil Menon	Absent
1BM24CS042	Sneha Chopra	Present
1BM24CS043	Rakesh Malik	Absent
1BM24CS044	Aishwarya Bansal	Present
1BM24CS045	Suresh Agarwal	Absent
1BM24CS046	Ishita Verma	
1BM24CS047	Pankaj Kapoor	Present
1BM24CS048	Swati Joshi	Present
1BM24CS049	Deepak Kulkarni	Present
1BM24CS050	Parul Desai	Present

Fig 11. Update previous attendance: Teacher has to select the class and date to review the attendance of students and update them when needed

HOD view: Head Of Dept has to login using college email ID and can assign classes to teachers and view attendance of all classes.



ATTENDANCE TRACKER

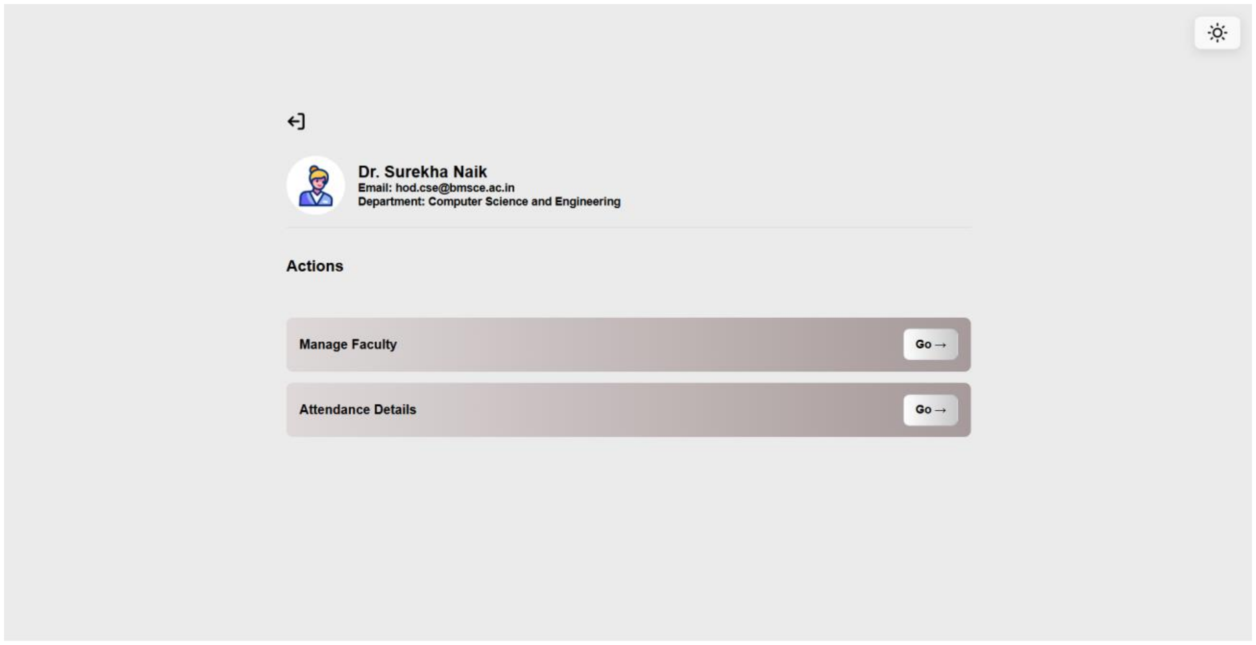
hod.cse@bmsce.ac.in

••••••

hPLFsj

hPLFsj

Fig 12. HOD login: Enter Email, Password, Captcha



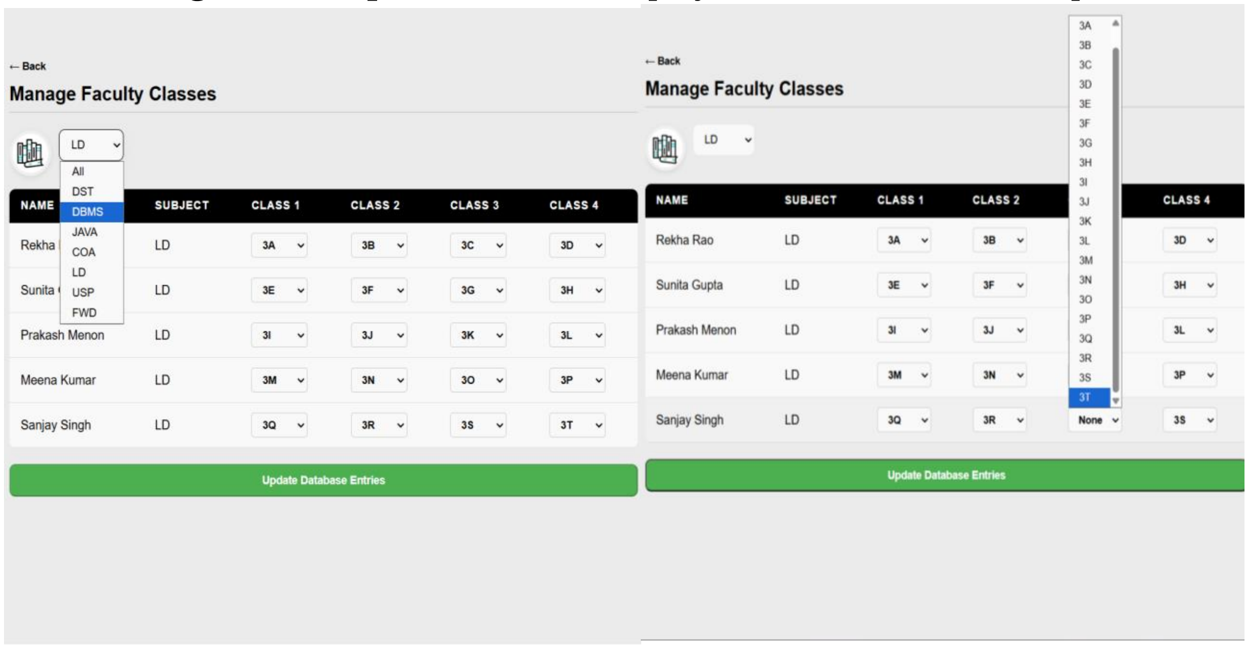
Dr. Surekha Naik
Email: hod.cse@bmsce.ac.in
Department: Computer Science and Engineering

Actions

Manage Faculty — Go →

Attendance Details — Go →

Fig 13. HOD profile: Profile displays HOD name, email, Dept



← Back

Manage Faculty Classes

LD

All
DST
DBMS
JAVA
COA
LD
USP
FWD

NAME	SUBJECT	CLASS 1	CLASS 2	CLASS 3	CLASS 4
Rekha	LD	3A	3B	3C	3D
Sunita	LD	3E	3F	3G	3H
Prakash Menon	LD	3I	3J	3K	3L
Meena Kumar	LD	3M	3N	3O	3P
Sanjay Singh	LD	3Q	3R	3S	3T

Update Database Entries

← Back

Manage Faculty Classes

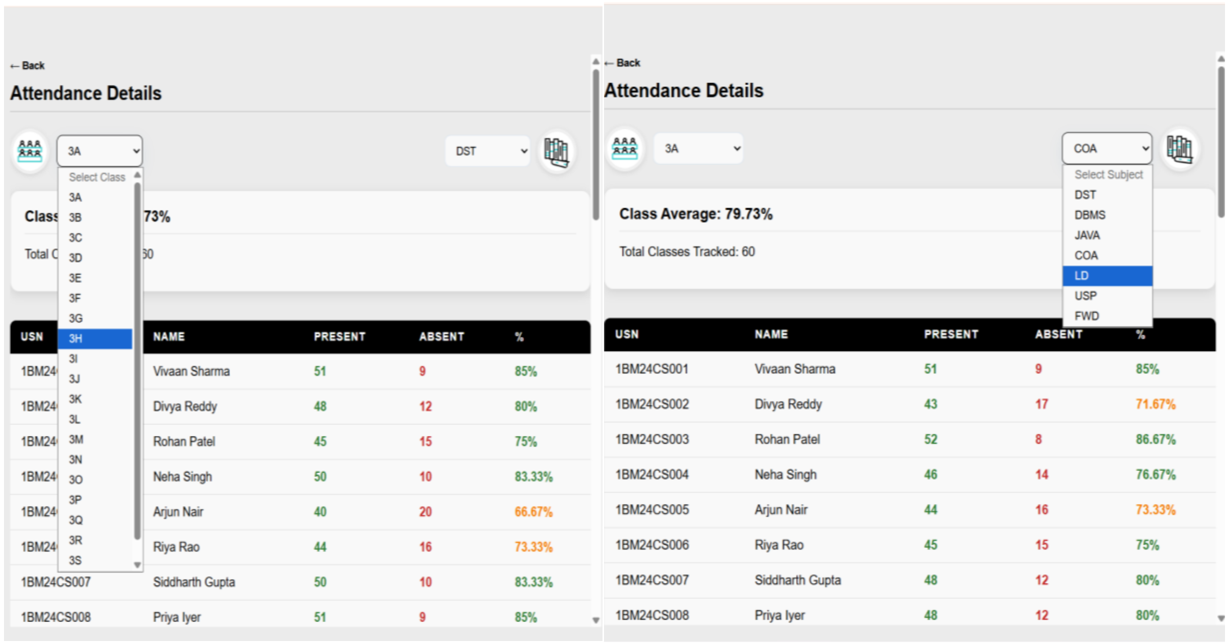
LD

3A
3B
3C
3D
3E
3F
3G
3H
3I
3J
3K
3L
3M
3N
3O
3P
3Q
3R
3S
3T

NAME	SUBJECT	CLASS 1	CLASS 2		CLASS 4
Rekha Rao	LD	3A	3B		3D
Sunita Gupta	LD	3E	3F		3H
Prakash Menon	LD	3I	3J		3L
Meena Kumar	LD	3M	3N		3P
Sanjay Singh	LD	3Q	3R	None	3S

Update Database Entries

Fig 14. Manage faculty: HOD selects subject and reassign the classes to the faculty



Attendance Details

3A DST

Class 73%

Total C 60

USN	NAME	PRESENT	ABSENT	%
1BM24	Vivaan Sharma	51	9	85%
1BM24	Divya Reddy	48	12	80%
1BM24	Rohan Patel	45	15	75%
1BM24	Neha Singh	50	10	83.33%
1BM24	Arjun Nair	40	20	66.67%
1BM24	Riya Rao	44	16	73.33%
1BM24CS007	Siddharth Gupta	50	10	83.33%
1BM24CS008	Priya Iyer	51	9	85%

Attendance Details

3A COA

Class Average: 79.73%

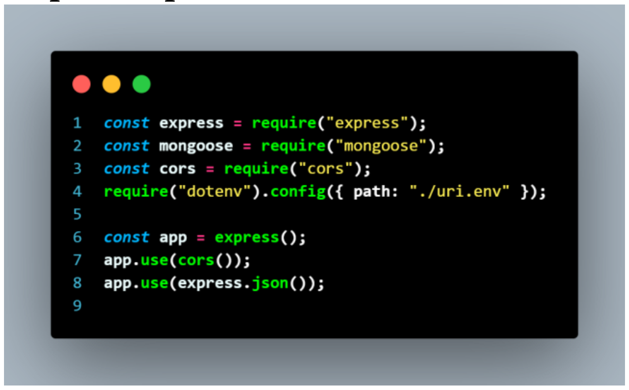
Total Classes Tracked: 60

USN	NAME	PRESENT	ABSENT	%
1BM24CS001	Vivaan Sharma	51	9	85%
1BM24CS002	Divya Reddy	43	17	71.67%
1BM24CS003	Rohan Patel	52	8	86.67%
1BM24CS004	Neha Singh	46	14	76.67%
1BM24CS005	Arjun Nair	44	16	73.33%
1BM24CS006	Riya Rao	45	15	75%
1BM24CS007	Siddharth Gupta	48	12	80%
1BM24CS008	Priya Iyer	48	12	80%

Fig 15. Attendance: Total attendance percentage of students from a class is displayed in correspondence to the subject. Students with attendance less than 75 percentage are highlighted.

BACKEND DESIGN

Import dependencies:

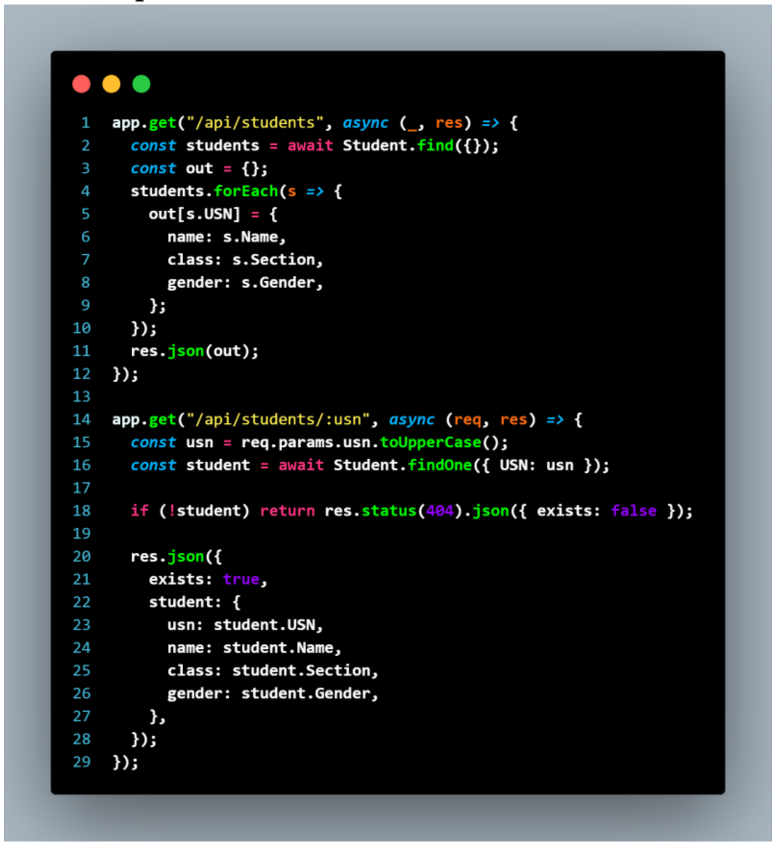


```javascript
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
require("dotenv").config({ path: "./uri.env" });

const app = express();
app.use(cors());
app.use(express.json());

```

Fig 16. Import express, mongoose, cors, dotenv packages using node package manager.

Routes: URL path that tells the server which piece of code to run when a request arrives.

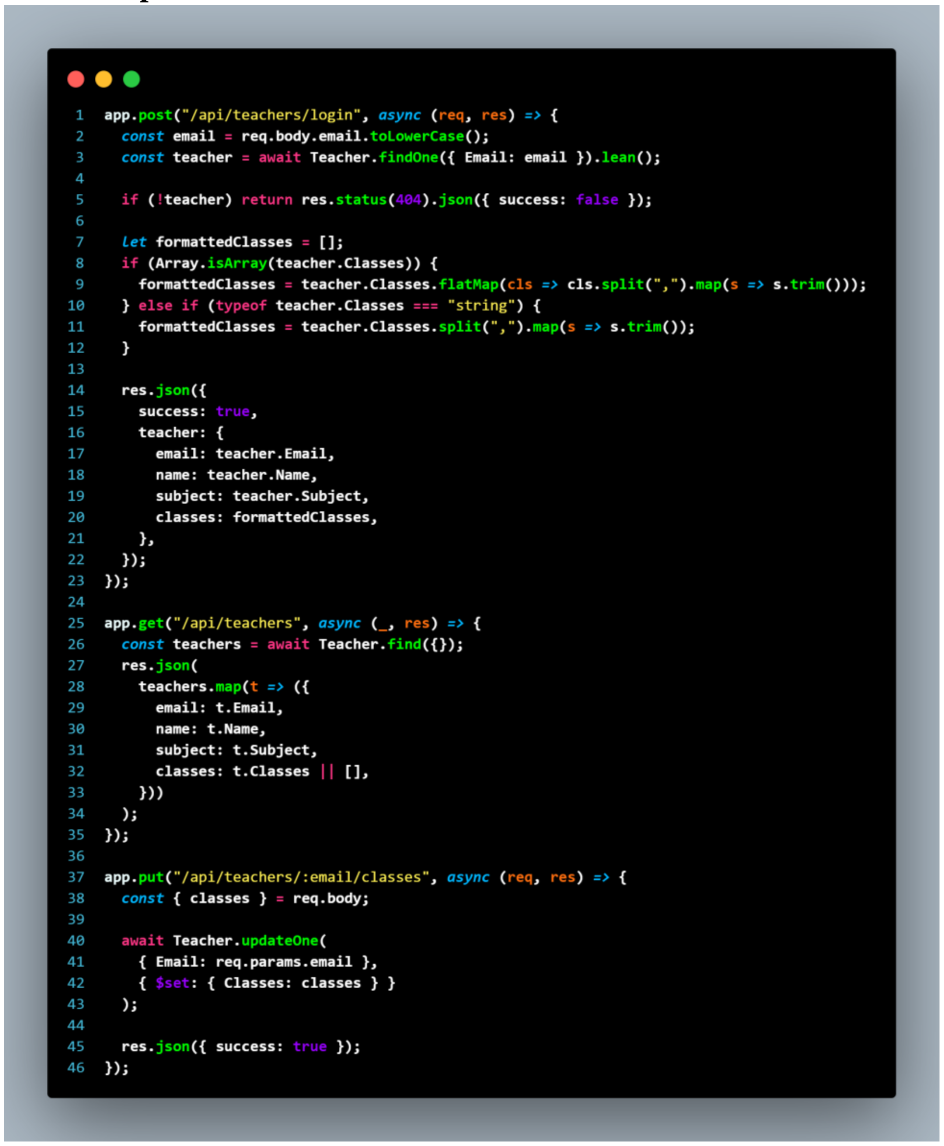
Student api:



```javascript
app.get("/api/students", async (_, res) => {
  const students = await Student.find({});
  const out = {};
  students.forEach(s => {
    out[s.USN] = {
      name: s.Name,
      class: s.Section,
      gender: s.Gender,
    };
  });
  res.json(out);
});

app.get("/api/students/:usn", async (req, res) => {
  const usn = req.params.usn.toUpperCase();
  const student = await Student.findOne({ USN: usn });

  if (!student) return res.status(404).json({ exists: false });

  res.json({
    exists: true,
    student: {
      usn: student.USN,
      name: student.Name,
      class: student.Section,
      gender: student.Gender,
    },
  });
});
```

Fig 17. These api can be used for accessing student object and validate USN.

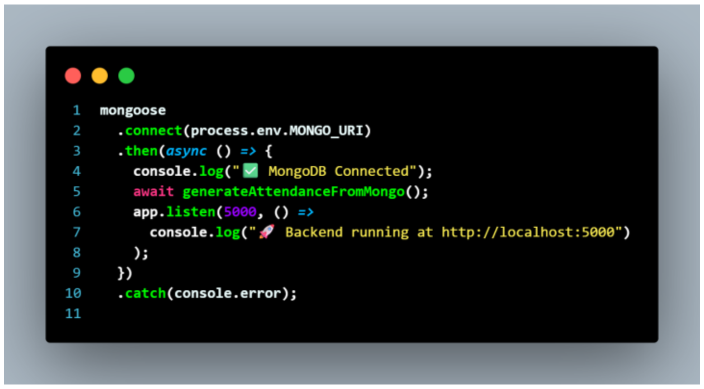
Teacher api:



```javascript
app.post("/api/teachers/login", async (req, res) => {
  const email = req.body.email.toLowerCase();
  const teacher = await Teacher.findOne({ Email: email }).lean();

  if (!teacher) return res.status(404).json({ success: false });

  let formattedClasses = [];
  if (Array.isArray(teacher.Classes)) {
    formattedClasses = teacher.Classes.flatMap(cls => cls.split(",").map(s => s.trim()));
  } else if (typeof teacher.Classes === "string") {
    formattedClasses = teacher.Classes.split(",").map(s => s.trim());
  }

  res.json({
    success: true,
    teacher: {
      email: teacher.Email,
      name: teacher.Name,
      subject: teacher.Subject,
      classes: formattedClasses,
    },
  });
});

app.get("/api/teachers", async (_, res) => {
  const teachers = await Teacher.find({});
  res.json(
    teachers.map(t => ({
      email: t.Email,
      name: t.Name,
      subject: t.Subject,
      classes: t.Classes || [],
    }))
  );
});

app.put("/api/teachers/:email/classes", async (req, res) => {
  const { classes } = req.body;

  await Teacher.updateOne(
    { Email: req.params.email },
    { $set: { Classes: classes } }
  );

  res.json({ success: true });
});
```

Fig 18. These apis can be used for accessing teacher object and validate emails.

Connect to database:



```javascript
mongoose
  .connect(process.env.MONGO_URI)
  .then(async () => {
    console.log("✅ MongoDB Connected");
    await generateAttendanceFromMongo();
    app.listen(5000, () =>
      console.log("🚀 Backend running at http://localhost:5000")
    );
  })
  .catch(console.error);
```

Fig 19. Connection to database started using .connect() method.

CONCLUSION AND FUTURE WORK

Conclusion

The Attendance Tracker project provides a systematic and interactive web application designed to simplify and streamline the attendance management process in academic environments. Developed as a React-based application, the system offers role-specific interfaces for students and teachers, ensuring clarity, ease of use, and organized access to attendance information. By adopting a component-based architecture, the application maintains modularity, reusability, and smooth navigation across different screens.

The project serves as a practical demonstration of applying modern frontend development concepts such as state management, routing, and responsive UI design to solve real-world academic challenges. The interactive nature of the system improves user engagement while reducing the complexity commonly associated with traditional attendance systems. Furthermore, the project highlights how a well-designed frontend application can enhance transparency and usability in attendance tracking.

Overall, the Attendance Tracker stands as an effective learning-oriented solution that bridges theoretical knowledge and practical implementation. It not only addresses the academic attendance management problem but also provides a strong foundation for future enhancements, including backend integration, analytics, and mobile support.

Future Work

In the future, the Attendance Tracker system can be enhanced by integrating a backend server and database to support real-time attendance management and permanent data storage. Secure authentication and authorization mechanisms can be implemented to ensure data security and role-based access control. Features such as automated attendance reports, graphical analytics, and notification alerts can be added to improve monitoring, transparency, and decision-making for students, teachers, and administrators.

The scope of the project can be further expanded by introducing mobile application support, enabling users to access attendance information anytime and anywhere. Advanced attendance methods such as QR code-based or biometric systems can be incorporated to increase accuracy and reduce manual effort. Cloud deployment can also be considered to improve scalability, reliability, and accessibility for large academic institutions.

What differentiates this project from existing ERP systems such as **Contineo** is its simplicity, flexibility, and learning-oriented design. While ERP systems are comprehensive, complex, and vendor-controlled, this project is lightweight and fully customizable, making it suitable for academic experimentation and rapid enhancement. The system focuses on clarity, usability, and understanding the internal working of attendance management applications rather than providing a fixed, enterprise-level solution.

REFERENCES

The following websites and online resources were referred to during the design and development of the Attendance Tracker project**.**

1. W3Schools – HTML, CSS, and JavaScript

W3Schools was used to understand the fundamentals of HTML structure, CSS styling, and JavaScript programming used in the user interface design.

Website:
https://www.w3schools.com

W3Schools Online Web Tutorials

2. JavaScript.info

This website was used to strengthen understanding of JavaScript logic, functions, arrays, and event handling.

Website:
https://javascript.info

3. GitHub

GitHub was used to explore sample React projects, understand folder structures, and learn best practices in frontend development.

Website:
https://github.com

4. React Official Documentation

The React documentation was referred to for implementing component-based architecture, JSX syntax, props, state handling, and event management used in the project.

Website:
https://react.dev

5. Node.js Official Documentation

Node.js documentation was referred to for understanding server-side JavaScript, backend architecture, and REST API concepts.

Website:
https://nodejs.org

6. MongoDB Compass Documentation

MongoDB Compass documentation was used to understand visual database management, data inspection, and query execution.

Website:
https://www.mongodb.com/products/tools/compass

7. REST API Concepts – MDN Web Docs

MDN Web Docs were referred to for understanding RESTful APIs, request–response cycles, and HTTP methods used in backend communication.

Website:
https://developer.mozilla.org/en-US/docs/Web/HTTP

9. Express.js Documentation

Express.js was used as a reference for building lightweight backend services and handling HTTP requests efficiently.

Website:
https://expressjs.com