SCTR's Pune Institute of Computer Technology Dhankawadi, Pune

A.Y. 2023-24

WADL MINI PROJECT REPORT ON

"PICT COMMUNITY"

Submitted By

Soham Sant– 33271
Himanshu Sangshetti– 33265
Sanika Rithe– 33263
Mihir Sawai– 33266
Aniket Saboo– 33264

Under the guidance of

Prof. Rachna Karnavat

DEPARTMENT OF INFORMATION TECHNOLOGY

ABSTRACT

The College Based Data Management System is a web application developed to streamline the management of academic data within a college setting. The system provides functionalities for both staff members, including teachers and Head of Department (HOD), as well as students.

Through this system, teachers and HODs can efficiently manage various aspects such as notes, attendance, internal marks, time schedules, and course offerings. Additionally, students can access their academic information, including notes, attendance records, and internal marks. They can also join or leave courses offered by the college.

Implemented using modern web development technologies such as React, TailwindCSS, NodeJS, ExpressJS, and MongoDB with Mongoose, the system offers a user-friendly interface with features like profile management, dark theme, and mobile responsiveness.

INTRODUCTION

In today's fast-paced educational environment, the effective management of academic data plays a crucial role in ensuring smooth functioning and enhanced productivity within educational institutions. With the increasing reliance on digital platforms, there arises a need for comprehensive web-based solutions to streamline the management of academic information.

The College Based Data Management System aims to address this need by providing a centralized platform for managing various aspects of academic data within a college. This system caters to the needs of both staff members, including teachers and HODs, as well as students, offering them a range of functionalities to facilitate efficient data management and access.

By leveraging modern web development technologies such as React, TailwindCSS, NodeJS, ExpressJS, and MongoDB with Mongoose, the system ensures a robust and scalable solution. Moreover, features like profile management, dark theme support, and mobile responsiveness enhance the user experience, making the system accessible across different devices and preferences.

In the subsequent sections of this report, we will delve into the details of the system architecture, functionalities, implementation technologies, and future enhancements. Additionally, we will discuss the challenges faced during the development process and the strategies employed to overcome them.

LITERATURE SURVEY

TOPIC	Key Insights and References
Educational Management Systems	- Adoption of technology in education improves management efficiency and student outcomes (Alavi and Leidner, 2001; Ramayah and Hwang, 2006)
Web Development Frameworks	- ReactJS offers benefits for building user-friendly and dynamic web interfaces (Meirelles et al., 2018)
Database Management Systems	- MongoDB provides scalability and flexibility for handling diverse data types (Redmond and Wilson, 2015)
Authentication and Authorization	- Various methods such as username/password authentication and role-based access control are relevant (Boyd et al., 2010)
User Interface Design and User Experience	- User-centered design principles are crucial for creating intuitive interfaces (Norman, 2013)
Mobile Responsive Design	- Responsive web design principles ensure optimal user experience across devices (Nielsen, 2016)
Dark Theme Implementation	- Dark themes enhance aesthetics and potentially reduce eye strain (Radigan, 2020)

By reviewing existing literature in these areas, we gain valuable insights into the theoretical foundations, practical considerations, and potential challenges associated with the development of a College Based Data Management System. These insights serve as a guide for designing and implementing a robust and effective system that meets the needs of stakeholders within educational institutions.

IMPLEMENTATION DETAILS

Web technologies used:

The College Based Data Management System leverages a combination of modern web technologies to ensure efficient development and seamless performance across different platforms. The primary web technologies utilized in the implementation include:

- **Client-side Technologies:**
  - **React:** React is employed for building the frontend user interface, enabling the creation of dynamic and interactive components.
  - **TailwindCSS:** TailwindCSS is utilized for styling the user interface components, providing a utility-first approach for rapid development and customization.
- **Server-side Technologies:**
  - **Node.js:** Node.js serves as the runtime environment for executing server-side JavaScript code, facilitating asynchronous and event-driven programming.
  - **Express.js:** Express.js is utilized as the web application framework for Node.js, simplifying the development of server-side logic and RESTful APIs.
- **Database:**
  - **MongoDB:** MongoDB, a NoSQL database, is employed for storing and managing the application's data. Its flexible document-oriented structure is suitable for handling various types of academic data.
- **ORM (Object-Relational Mapping):**
  - **Mongoose:** Mongoose is used as an ORM tool for Node.js and MongoDB, providing a straightforward interface for interacting with MongoDB databases and defining data models.

Frontend development

The frontend development of the College Based Data Management System focuses on creating a responsive and intuitive user interface using React and TailwindCSS. Key aspects of frontend development include:

- **Component-based Architecture:** React's component-based architecture is leveraged to modularize the user interface into reusable components, promoting code reusability and maintainability.
- **State Management:** React's state management capabilities are utilized to manage the application's state and handle dynamic data updates without page reloads.
- **Responsive Design:** TailwindCSS is employed to ensure a responsive design that adapts seamlessly to various screen sizes and devices, providing an optimal user experience across desktop and mobile platforms.
- **User Interaction:** React's event handling mechanisms are utilized to enable user interaction features such as form submissions, button clicks, and navigation.

Backend development

The backend development of the system involves implementing server-side logic and database interactions using Node.js, Express.js, and MongoDB. Key aspects of backend development include:

- **API Development:** Express.js is used to develop RESTful APIs that enable communication between the frontend and backend components of the application.
- **Routing:** Express.js routing is utilized to define endpoints for handling various HTTP requests, including GET, POST, PUT, and DELETE operations.
- **Middleware:** Express.js middleware functions are employed to perform tasks such as request processing, authentication, and error handling.
- **Database Integration:** Mongoose is utilized to establish connections with the MongoDB database, define data schemas, and perform CRUD operations (Create, Read, Update, Delete).

Integration

Integration involves combining the frontend and backend components of the College Based Data Management System to create a cohesive and functional application. Key integration points include:

- **API Integration:** Frontend components interact with the backend server through RESTful APIs, enabling seamless data exchange and communication.
- **Authentication and Authorization:** Integration of authentication and authorization mechanisms ensures secure access to system resources based on user roles and permissions.
- **Data Flow:** Data flows between the frontend React components and the backend Express.js server via HTTP requests and responses, facilitating data retrieval, storage, and manipulation.
- **Error Handling:** Integration includes implementing error handling mechanisms to gracefully handle exceptions, validation errors, and other unforeseen issues that may arise during system operation.

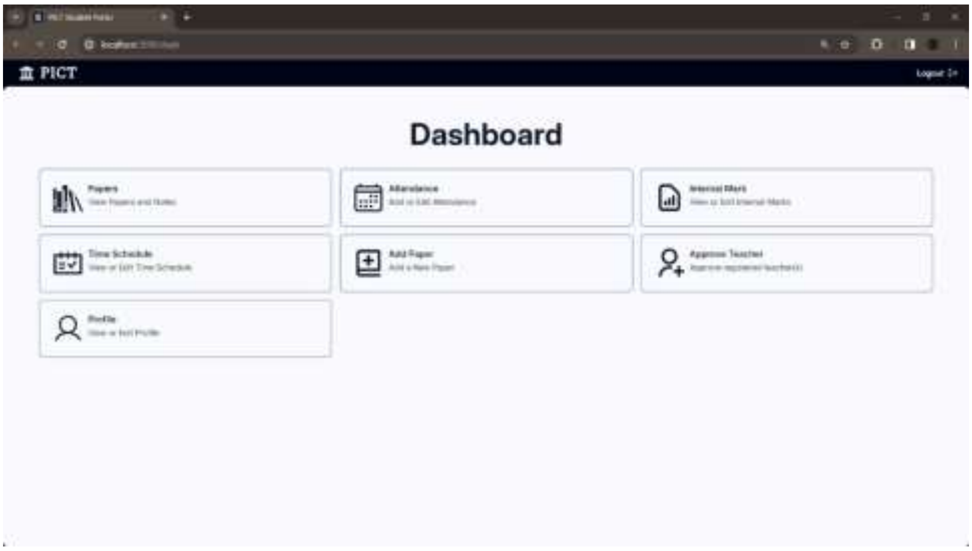
By effectively integrating the frontend and backend components while leveraging the chosen web technologies, the College Based Data Management System offers a robust and scalable solution for managing academic data within a college setting.

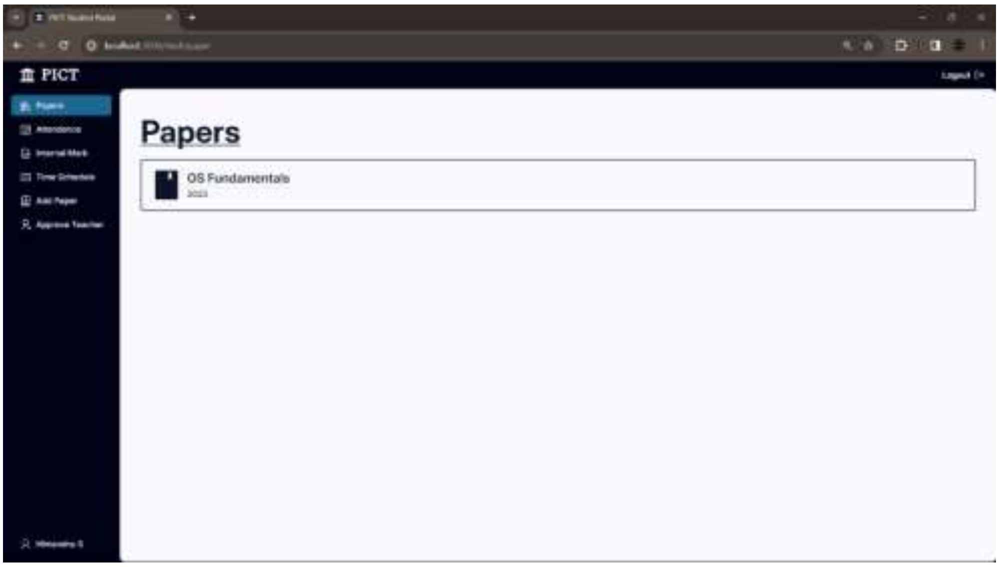
OUTPUT

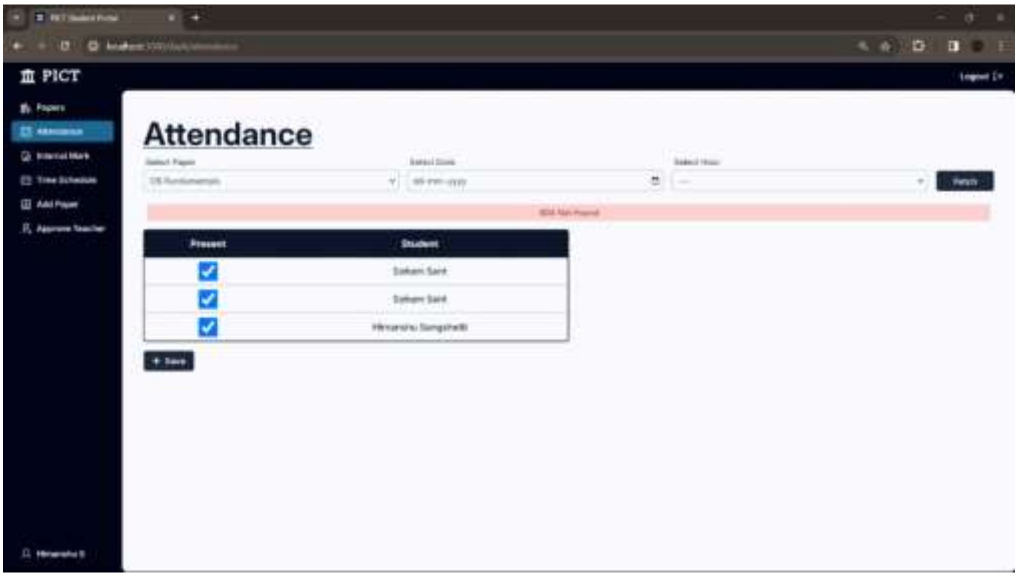
The output of the College Based Data Management System encompasses the user interface, functionality, and overall user experience provided by the application. Key components of the output include:

1. **User Interface (UI):**
   - The user interface of the system is designed to be intuitive, user-friendly, and visually appealing.
   - UI components are developed using React and styled using TailwindCSS, ensuring a modern and responsive design.
   - Navigation menus, buttons, forms, and other interactive elements are implemented to facilitate easy access to different functionalities.
2. **Functionality:**
   - The system provides various functionalities tailored to the needs of different user roles, including teachers, HODs, and students.
   - Teachers and HODs can add or edit notes, manage attendance, internal marks, time schedules, and course offerings.
   - HODs have additional privileges to approve new teachers and add new courses.
   - Students can view notes, attendance records, internal marks, and join or leave courses.
   - Authentication mechanisms ensure secure access to the system, with role-based authorization governing user permissions.
3. **User Experience (UX):**
   - The user experience of the system focuses on simplicity, efficiency, and accessibility.
   - Features such as profile management, dark theme support, and mobile responsiveness enhance the overall user experience.
   - Intuitive navigation, clear instructions, and error feedback mechanisms contribute to a seamless and enjoyable user experience.
4. **Data Management:**
   - Academic data such as notes, attendance records, internal marks, and course information are efficiently managed and stored in the MongoDB database.
   - Data integrity and security are maintained through proper validation, encryption, and access control mechanisms.
   - CRUD (Create, Read, Update, Delete) operations are implemented to enable seamless data manipulation by authorized users.

5. **System Performance:**
    - The system is designed to be scalable, robust, and performant, capable of handling concurrent user interactions and large volumes of data.
    - Backend server operations are optimized for efficiency, with asynchronous processing and caching mechanisms employed where applicable.
    - Monitoring and logging mechanisms are implemented to track system performance metrics and diagnose potential issues.
6. **Feedback and Improvement:**
    - Feedback mechanisms are incorporated to gather user feedback and suggestions for system improvement.
    - Continuous monitoring, testing, and iterative development practices ensure that the system evolves to meet changing user needs and technological advancements.







Attendance

Select Paper: OS Fundamentals

Select Date: dd-mm-yyyy

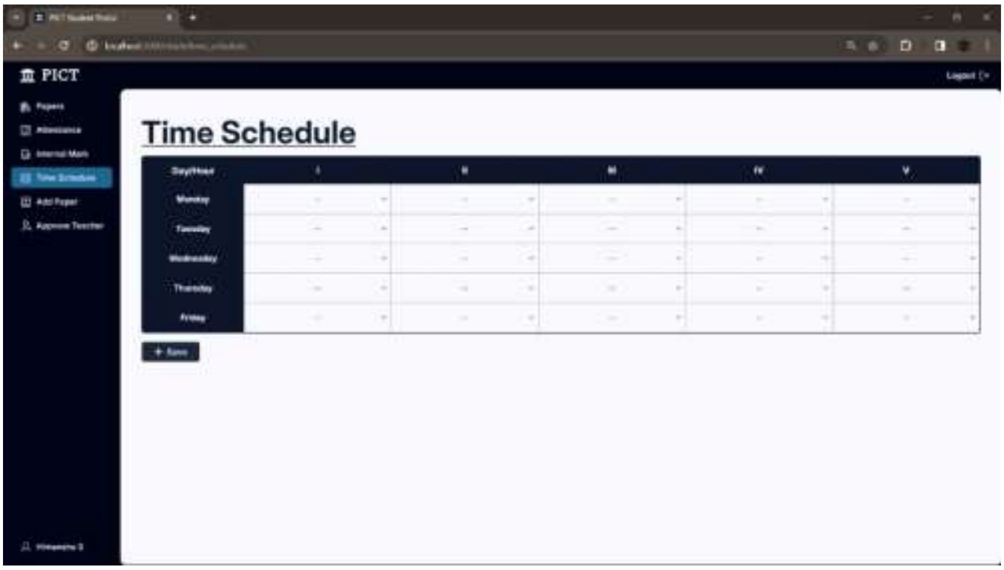
Select Hour: --

Fetch

404 Not Found

Present	Student
☑	Soham Sant
☑	Soham Sant
☑	Himanshu Sangshetti

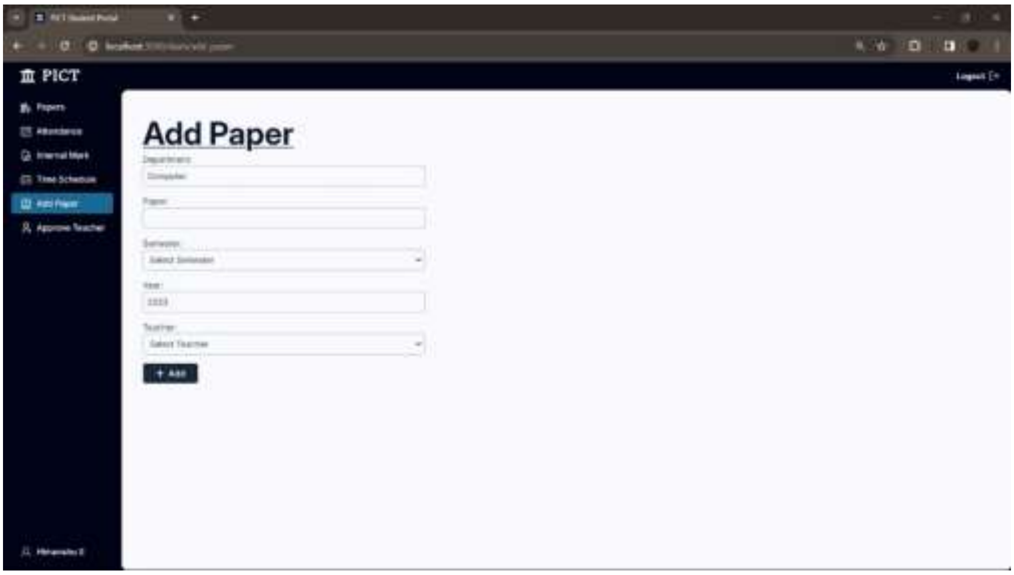
+ Save



Time Schedule

Day/Hour	I	II	III	IV	V
Monday	--	--	--	--	--
Tuesday	--	--	--	--	--
Wednesday	--	--	--	--	--
Thursday	--	--	--	--	--
Friday	--	--	--	--	--

+ Save



Add Paper

Department: Computer

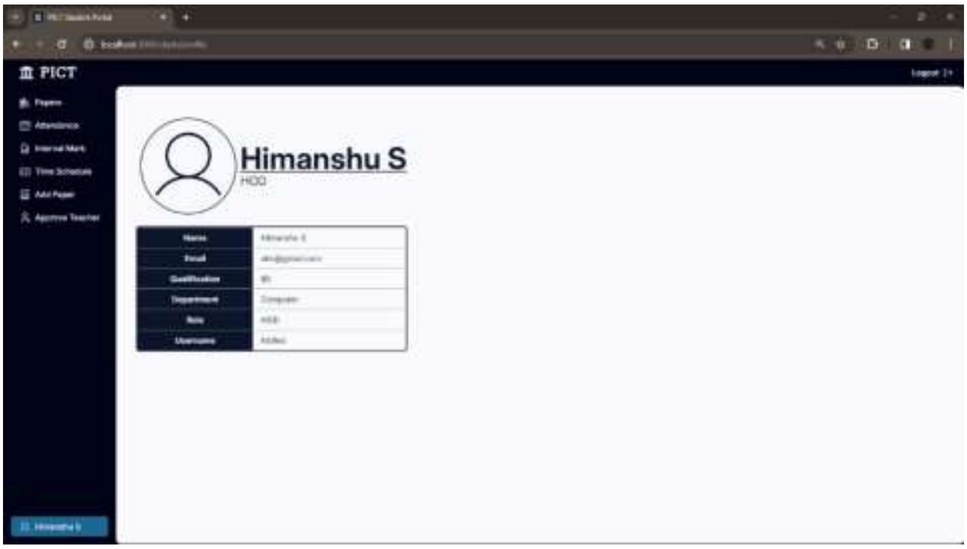
Paper:

Semester: Select Semester

Year: 2023

Teacher: Select Teacher

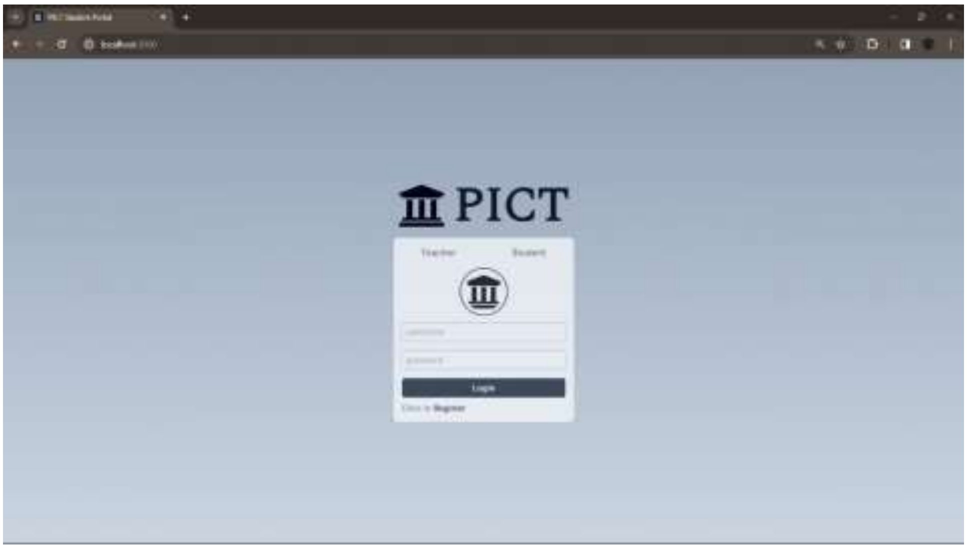
+ Add

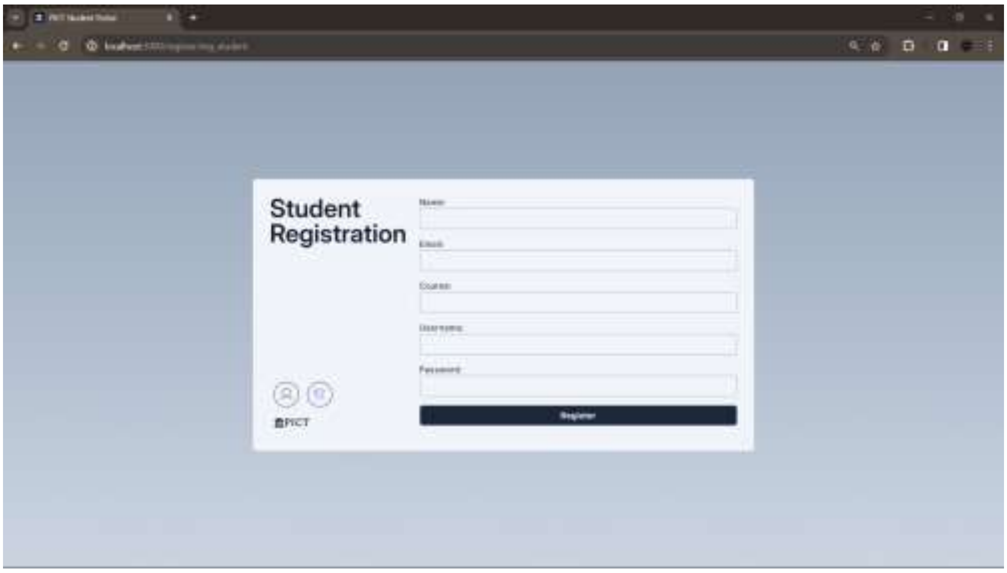


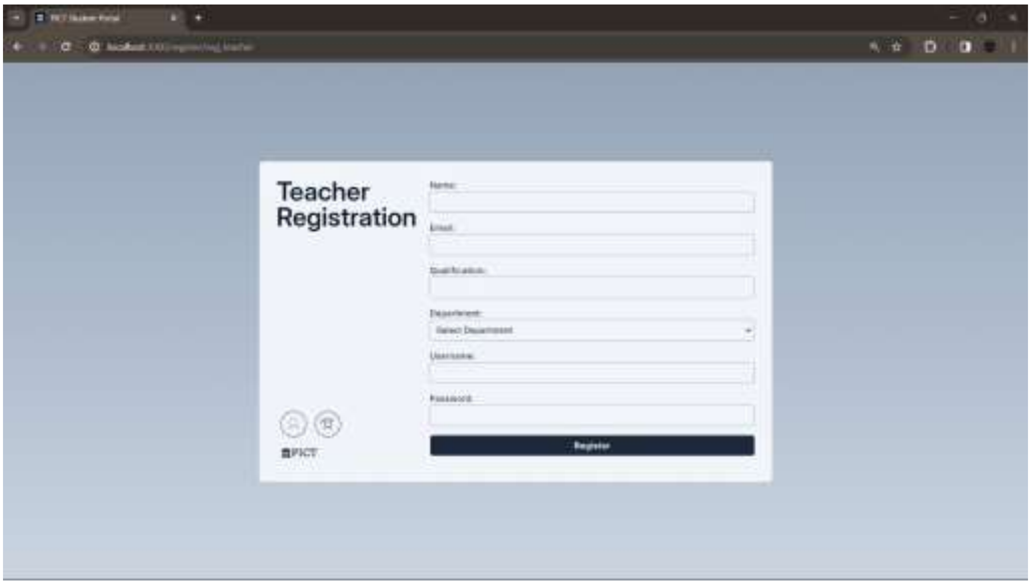
Himanshu S

HOD

Name	Himanshu S
Email	[illegible]
Qualification	[illegible]
Department	Computer
Role	HOD
Username	[illegible]







Teacher Registration

Name:

Email:

Qualification:

Department:
Select Department

Username:

Password:

CONCLUSION

The College Based Data Management System represents a significant advancement in the realm of educational technology, offering a comprehensive solution for managing academic data within a college setting. Through the integration of modern web technologies, robust backend architecture, and user-centric design principles, the system addresses the diverse needs of teachers, Head of Departments (HODs), and students alike.

The development and implementation of the system have yielded several key outcomes and insights:

1. **Efficiency and Streamlining:** By digitizing and centralizing academic data management processes, the system facilitates efficiency and streamlines administrative tasks for teachers and HODs. Tasks such as note creation, attendance tracking, and internal mark management are automated and simplified, saving time and resources.
2. **Enhanced Accessibility:** The system's user-friendly interface and mobile responsiveness ensure accessibility across different devices and platforms. Students can easily access their academic information and participate in course activities from anywhere, enhancing their learning experience.
3. **Security and Authorization:** Robust authentication and authorization mechanisms ensure that sensitive academic data is securely accessed and managed. Role-based access control ensures that users only have access to information and functionalities relevant to their roles, maintaining data integrity and confidentiality.
4. **Scalability and Future-Readiness:** The system's architecture is designed to be scalable and adaptable to future enhancements and expansions. The use of technologies such as React, Node.js, and MongoDB provides a solid foundation for accommodating future requirements and technological advancements.
5. **User Experience Focus:** Throughout the development process, a strong emphasis was placed on creating a positive user experience. Features such as profile management, dark theme support, and intuitive navigation contribute to an engaging and enjoyable user experience, fostering user satisfaction and adoption.

REFERENCES

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly, 25*(1), 107-136.

Boyd, C., Cain, J., & Hancock, J. (2010). Authenticating Users of Educational Web-Based Software Systems. *International Journal of Information and Communication Technology Education, 6*(3), 59-70.

Meirelles, D., Siqueira, F., & Galvão, M. (2018). ReactJS: An Overview. *International Journal of Computer Applications, 181*(38), 43-47.

Nielsen, J. (2016). Responsive Web Design. *Nielsen Norman Group*. Retrieved from https://www.nngroup.com/articles/responsive-web-design-definition/

Norman, D. A. (2013). The Design of Everyday Things. *Basic Books*.

Radigan, D. (2020). Dark Mode Design: The Complete Guide. *Medium*. Retrieved from https://uxdesign.cc/dark-mode-design-the-complete-guide-523ffabbde10

Ramayah, T., & Hwang, E. Y. (2006). An exploratory study of knowledge management practice in higher learning institutions. *Journal of Knowledge Management, 10*(3), 6-23.

Redmond, E., & Wilson, J. (2015). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. *Pragmatic Bookshelf*.

GitHub Link: https://github.com/Himanshu-Sangshetti/PICT-Student-Portal