International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Outcome-Based Education Management System Using MERN Stack

Pinninti Swarnalatha[1], Yandamuri Pradeep[2], Pilla Raghavendra Sai [3], Pitta Sai Jagadeesh[4]

[1]Pinninti Swarnalatha, Information Technology, GMRIT, Rajam, India
[2]Yandamuri Pradeep, Information Technology, GMRIT, Rajam, India
[3]Pilla Raghavendra Sai, Information Technology, GMRIT, Rajam, India
[4]Pitta Sai Jagadeesh, Information Technology, GMRIT, Rajam, India

A B S T R A C T :

In our country, many educational institutions have well-organized file management systems that depend on various approaches. A significant number of these institutions follow outcome-based education (OBE), which necessitates the monitoring and tracking of students from enrollment to graduation. Throughout this process, institutions offer multiple facilities, each of which is managed through physical file systems. Higher authorities monitor the status of these files daily. To address the challenges posed by traditional file management methods, we propose the development of a web application using the MERN stack (MongoDB, Express.js, React, and Node.js). This web application will streamline the management of student data and institutional files, ensuring efficient monitoring of student progress, maintaining records of the facilities provided, and automating several processes that were previously handled manually. The application will feature role-based access for different users, such as Head of Departments (HODs) and faculty members, ensuring that each user only has access to the data relevant to their responsibilities. By leveraging the MERN stack, we aim to provide a scalable, user-friendly, and efficient solution that will simplify the complexity of the file management system and improve administrative efficiency in institutions.

Keywords: MongoDB, Express.js, React.js, Node.js, File Management System, Real-Time Monitoring

1. INRODUCTION

In today's digital era, managing institutional files efficiently has become a significant challenge for educational institutions. Traditional paper-based file management systems often result in data loss, misplacement, and accessibility issues, making it difficult for faculty members and administrators to track student progress, maintain academic records, and manage institutional documents effectively. As educational institutions increasingly adopt Outcome-Based Education (OBE) methodologies, the need for a centralized and secure digital file management system has become more evident.

To address these challenges, this project proposes the development of a web-based file management system using the MERN stack, which includes MongoDB, Express.js, React, and Node.js. The system is designed to digitize and automate file handling processes, providing a centralized repository for student records, faculty files, and academic documents. By integrating modern web technologies, the platform ensures secure, role-based access control for HODs, faculty, and administrators, allowing them to upload, update, and retrieve documents seamlessly.

The proposed system leverages the MERN stack to create a scalable and efficient solution for handling institutional files. MongoDB, a NoSQL database, ensures structured and unstructured data storage with high availability. Express.js, a lightweight Node.js framework, facilitates backend API development for managing file operations. React.js is utilized for building an interactive and responsive user interface, enhancing user experience. Node.js serves as the server-side runtime, handling backend logic, authentication, and file processing efficiently. This technology stack enables real-time file access, seamless collaboration, and improved data security by implementing role-based permissions. The system allows HODs, faculty, and administrators to perform specific actions based on their access levels, preventing unauthorized modifications and ensuring data integrity. This dashboard provides role-based access, ensuring that users only have access to data and actions relevant to their responsibilities. Faculty members can manage student-related documents such as attendance sheets and marks, while HODs can oversee departmental records, track updates, and ensure the smooth flow of academic documentation. The system supports various file types spreadsheets making it highly adaptable to institutional needs.

The File Management System incorporates several essential features that streamline institutional file handling. Role-Based Access Control (RBAC) ensures that different user levels have appropriate permissions, preventing unauthorized access to sensitive files. The centralized document repository stores and categorizes files efficiently, allowing quick retrieval based on predefined criteria while supporting multiple file formats such as PDFs, Word documents, and images. Real-time collaboration and notifications enhance communication by enabling faculty members to upload, edit, and share academic files while keeping users informed about file updates and access permissions. Secure file storage and version control mechanisms safeguard sensitive academic data through encryption techniques while allowing users to restore previous versions if needed. The system is designed for scalability and high performance, built on a cloud-based architecture that ensures fast access and minimal downtime. MongoDB's distributed nature allows the handling of large datasets efficiently, making the platform suitable for expanding institutional needs.

2. LITERATURE SURVEY

The NBA accreditation plays a crucial role in enhancing the quality of education in Indian engineering colleges by ensuring that institutions adhere to established academic standards. However, the traditional accreditation process often faces challenges related to documentation management and application tracking. To address these inefficiencies, a MERN stack-based web portal is proposed, providing a centralized and streamlined platform for managing accreditation processes effectively. This system leverages MongoDB, a NoSQL database that enables flexible data storage in JSON format, making it suitable for real-time updates and scalability. Express.js serves as the web application framework for Node.js, simplifying server-side development and enabling efficient request handling. React.js is used on the frontend to ensure dynamic data rendering without requiring page refreshes, creating a seamless and interactive user experience. Node.js, as a server-side runtime environment, supports asynchronous programming, making it highly efficient in managing multiple user requests simultaneously. Despite its advantages, implementing this web portal may present challenges, particularly for institutions with limited technological infrastructure or digital literacy, as developing and maintaining such a system requires significant technical expertise and resources [1].

The Nostalgia Hub is designed as a user-friendly web platform that enables individuals to relive and share meaningful experiences from their past. By leveraging the MERN stack, which consists of MongoDB, Express.js, React.js, and Node.js, the platform ensures a dynamic and scalable environment while prioritizing user-centric design and privacy preservation. The technical foundation of the Nostalgia Hub allows for efficient data management, seamless interactions, and real-time updates, enhancing the overall user experience. The platform's advantages include fostering a strong emotional connection with users, maintaining a high level of privacy protection, and offering a well-structured interface for easy navigation. However, certain challenges exist, such as the reliance on technology for emotional experiences, potential data privacy concerns despite implemented safeguards, and the ongoing challenge of sustaining user engagement over time [2].

The project is designed to offer a seamless and interactive user experience by leveraging modern web technologies. HTML5, CSS, and Bootstrap ensure that the interface is well-structured, visually appealing, and fully responsive across different screen sizes and devices. Vue.js, being a progressive JavaScript framework, enhances the interactivity of the application by efficiently managing state and updating components in real-time. The Laravel Blade Template Engine further simplifies the development process by providing a clean and reusable approach to rendering dynamic web pages. On the backend, PHP is used as the server-side scripting language to handle business logic and application functionalities. MySQL serves as the relational database, efficiently storing and managing student and faculty data with the help of ACID compliance to maintain data integrity and consistency. The routing system ensures smooth navigation and efficient handling of user requests, making the application more robust and scalable. Additionally, API management is streamlined to allow seamless integration with external services or future expansions. Despite its many advantages, the system may have some limitations, particularly in real-time applications, where its performance might be slower compared to Node.js-based alternatives. However, with proper optimization techniques, caching strategies, and efficient database queries, the overall performance can be significantly improved, ensuring a smooth user experience [3].

The project leverages HTML, CSS, and JavaScript to design an intuitive user interface, ensuring a dynamic and interactive experience across different devices. These technologies provide a structured and visually appealing layout while enhancing user engagement and accessibility. Bootstrap plays a crucial role in improving responsiveness, offering pre-designed UI components and a flexible grid system that seamlessly adapts to various screen sizes and resolutions. On the backend, the ASP.NET Framework serves as the core architecture of the application, delivering robust security, efficient request handling, and seamless integration with Microsoft services. C# is used as the primary programming language for server-side development, facilitating efficient data processing, object-oriented programming, and integration with .NET libraries for enhanced functionality. While the system benefits from the stability of ASP.NET, using older technologies such as ASP.NET Web Forms may present maintainability challenges. Modern frameworks like ASP.NET Core offer better flexibility, improved performance, and long-term support, making them a preferred choice for future-proofing web applications. Upgrading to newer frameworks can enhance scalability, security, and efficiency, ensuring a more optimized and sustainable development approach [4].

3. PROPOSED SYSTEM

The Outcome-Based Education (OBE) Management System is designed to efficiently manage academic files and documents with role-based access control for different users, including administrators, HODs, faculty members, and students. This system ensures secure file management, efficient monitoring, and streamlined data access. The methodology adopted for this project follows a structured approach, including planning, design, development, testing, and deployment.

3.2. System Architecture

The system is developed using the MERN stack, comprising React with Vite for the frontend, Node.js with Express.js for the backend, and MongoDB as the database. The application follows a modular and scalable architecture to support authentication, authorization, and file management operations. The compete scenario has depicted in the above figure 3.1.

3.2.1 React:

React is a popular JavaScript library for building dynamic and interactive user interfaces, primarily for web applications. Developed and maintained by Facebook, it follows a component-based architecture, allowing developers to create reusable UI components that efficiently update and render when data changes. React uses a virtual DOM (Document Object Model) to optimize performance by updating only the necessary parts of the UI rather than reloading the entire page. It supports JSX (JavaScript XML), which allows developers to write HTML-like syntax within JavaScript. React is widely used in modern web development due to its flexibility, efficiency, and strong ecosystem, making it a preferred choice for single-page applications (SPAs) and complex frontend projects.

3.2.2 Node Js:

Node.js is an open-source, cross-platform runtime environment that allows developers to run JavaScript code outside the browser. Built on Chrome's V8 JavaScript engine, it enables fast and efficient execution of JavaScript on the server side. Node.js uses an event-driven, non-blocking I/O model, making it lightweight and ideal for building scalable network applications, such as web servers, APIs, and real-time applications. It has a rich ecosystem with npm (Node Package Manager), which provides access to thousands of libraries for various functionalities. Due to its high performance and ability to handle multiple concurrent requests, Node.js is widely used in backend development, particularly for building RESTful APIs, microservices, and full-stack applications when combined with frameworks like Express.js

3.2.3 Express Js:

Express.js is a minimal and flexible web application framework for Node.js, designed to simplify backend development by providing robust features for building web applications and APIs. It is lightweight, fast, and follows a middleware-based architecture, allowing developers to handle HTTP requests, routes, and responses efficiently. Express.js makes it easy to create RESTful APIs and dynamic web applications by providing built-in functions for routing, middleware integration, and request handling. It supports template engines, middleware customization, and seamless integration with databases like MongoDB and MySQL. Due to its simplicity and scalability, Express.js is widely used in full-stack development, especially in combination with front-end frameworks like React and Angular in the MERN and MEAN stacks.

3.2.4 MongoDB:

MongoDB is a popular open-source NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). Unlike traditional relational databases, it uses a document-oriented approach, allowing for dynamic and schema-less data storage, making it ideal for handling unstructured or semi-structured data. MongoDB is highly scalable and supports horizontal scaling with sharding, making it suitable for handling large datasets and high-traffic applications. It offers powerful querying capabilities with its rich set of operators and supports indexing for faster data retrieval. MongoDB integrates well with modern web development stacks, particularly the MERN (MongoDB, Express.js, React, Node.js) stack, making it a preferred choice for building real-time applications, content management systems, and cloud-based services.

The Steps involved are:

- Planning
- Requirement Analysis
- Design
- Web Application Development
- Testing

3.3 Planning and Requirement Analysis:

The initial phase of the project involved understanding the requirements of different users, including administrators, faculty, HODs, and students. A detailed analysis was conducted to identify the core functionalities needed for efficient management of the Outcome-Based Education (OBE) system. The primary focus was on role-based access, secure file management, and authentication mechanisms to ensure data integrity and confidentiality.

Stakeholder discussions were held to gather insights into how different users interact with the system. The project team conducted research on modern web technologies and database structures that could best support the application's scalability and performance. Various methodologies were considered, and the MERN (MongoDB, Express.js, React, Node.js) stack was chosen due to its efficiency, flexibility, and robust API handling.

3.4 System Design:

The system was structured into multiple modules based on user roles and functionalities to ensure a clear separation of concerns. The architecture followed a modular and scalable design, allowing efficient handling of different operations such as authentication, file management, and role-based access. The frontend was developed using React with Vite, enabling a fast and interactive user experience. The backend utilized Node.js with Express.js to handle API requests, authentication, and authorization efficiently. RESTful APIs were implemented to facilitate communication between the frontend and backend, ensuring smooth data transfer.

3.5 Implementation:

The implementation phase began with setting up the development environment, ensuring smooth integration between the frontend, backend, and database. The frontend was developed using React with Vite, which provided a fast and efficient build process, enabling rapid development and real-time hot module reloading. The component-based architecture of React allowed the UI to be modular, reusable, and scalable. Key libraries such as React Router were used to manage navigation, while Axios facilitated API communication with the backend. For the backend, Node.js with Express.js was used to create a lightweight yet powerful server. RESTful APIs were implemented to handle CRUD operations related to file management, authentication, and role-based access. Middleware functions were incorporated for input validation, error handling, and request logging, ensuring smooth request processing.

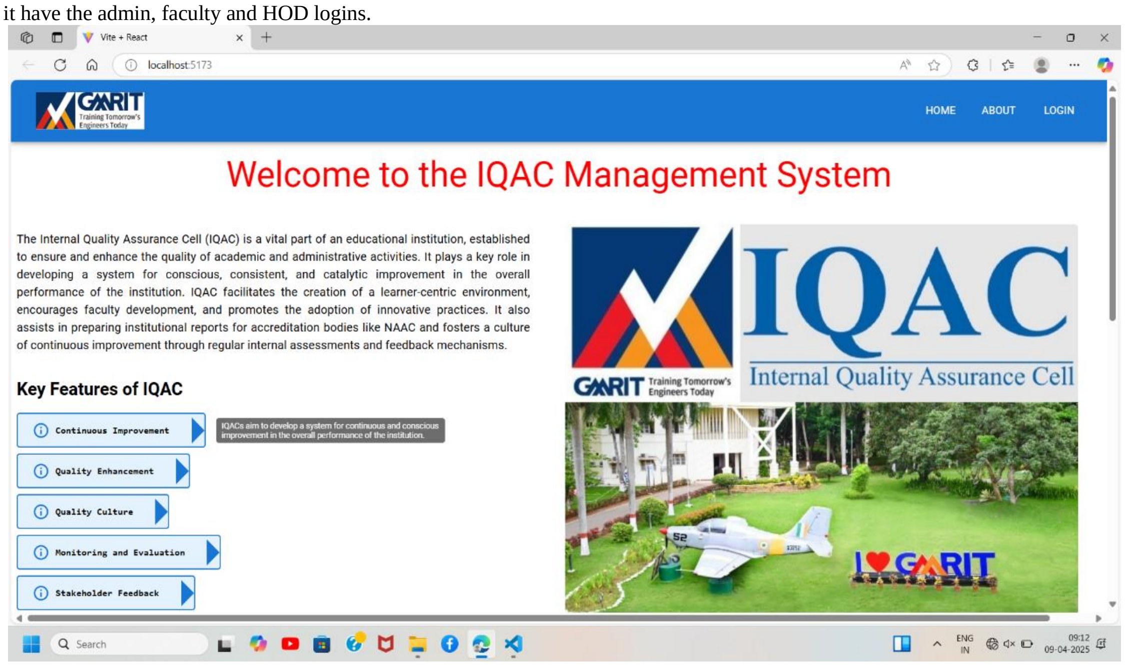
MongoDB served as the database, providing a NoSQL document-based storage structure that adapted well to dynamic project requirements. Collections were designed to store user details, role-based permissions, and file records efficiently. The database schema was kept flexible to accommodate future enhancements and scalability.

3.6 Testing and Debugging:

Comprehensive testing was conducted, including unit testing, integration testing, and user acceptance testing. Each module was tested for performance, security, and usability. Debugging was done iteratively to resolve issues before deployment.

4. RESULTS

Here is the result of the project outcome-based education management this project included with home page which have the information about the web application and it have the admin, faculty and HOD logins.



Welcome to the IQAC Management System

The Internal Quality Assurance Cell (IQAC) is a vital part of an educational institution, established to ensure and enhance the quality of academic and administrative activities. It plays a key role in developing a system for conscious, consistent, and catalytic improvement in the overall performance of the institution. IQAC facilitates the creation of a learner-centric environment, encourages faculty development, and promotes the adoption of innovative practices. It also assists in preparing institutional reports for accreditation bodies like NAAC and fosters a culture of continuous improvement through regular internal assessments and feedback mechanisms.

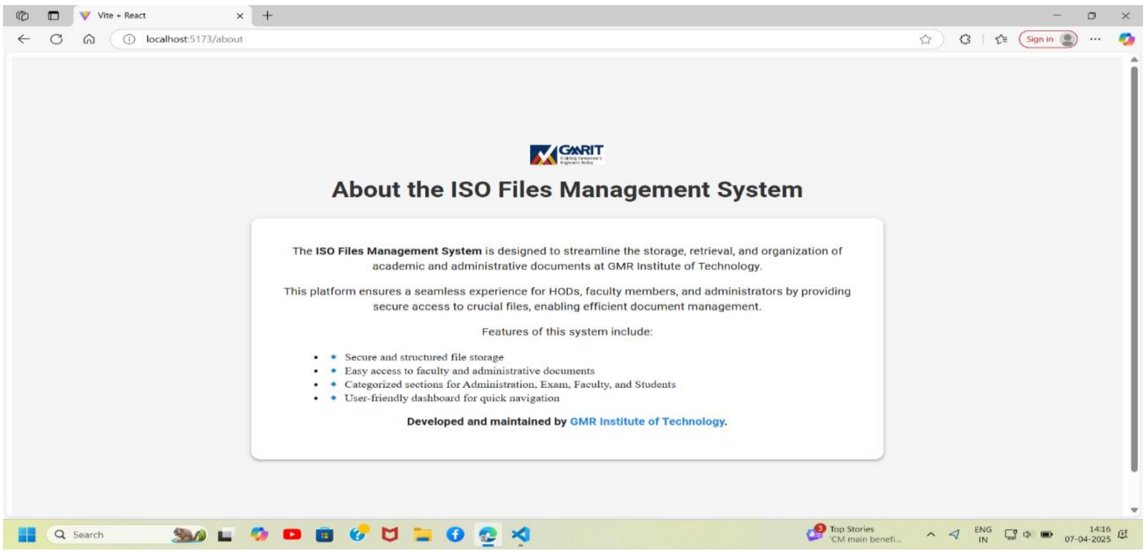
Key Features of IQAC

- Continuous Improvement
- Quality Enhancement
- Quality Culture
- Monitoring and Evaluation
- Stakeholder Feedback

IQACs aim to develop a system for continuous and conscious improvement in the overall performance of the institution.

IQAC
Internal Quality Assurance Cell
GMRIT Training Tomorrow's Engineers Today

Figure 4.1: Home Page



About the ISO Files Management System

The **ISO Files Management System** is designed to streamline the storage, retrieval, and organization of academic and administrative documents at GMR Institute of Technology.

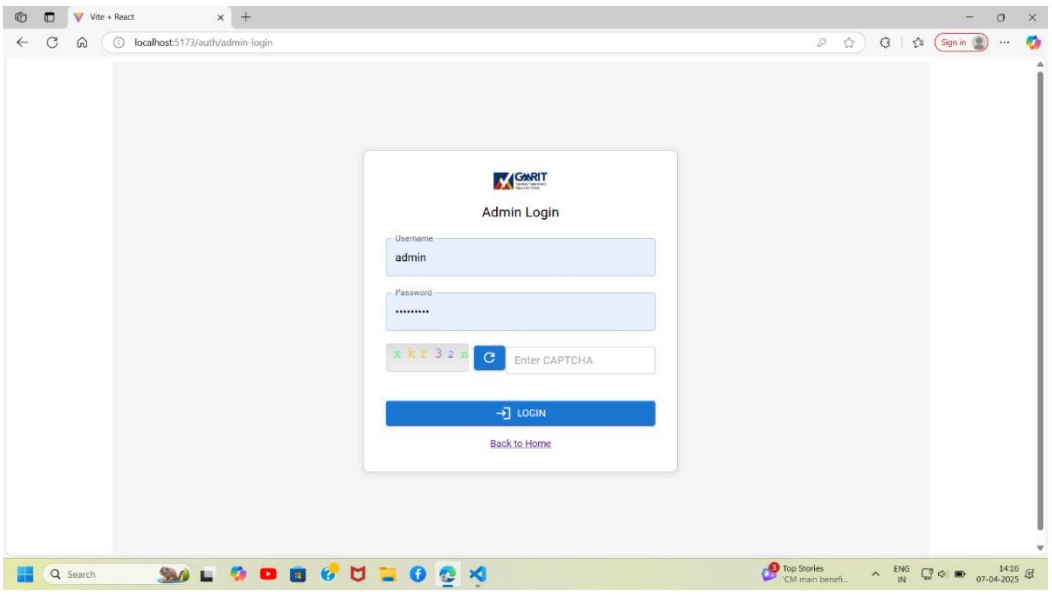
This platform ensures a seamless experience for HODs, faculty members, and administrators by providing secure access to crucial files, enabling efficient document management.

Features of this system include:

- Secure and structured file storage
- Easy access to faculty and administrative documents
- Categorized sections for Administration, Exam, Faculty, and Students
- User-friendly dashboard for quick navigation

Developed and maintained by GMR Institute of Technology.

Figure 4.2: About Page


Vite + React

localhost:5173/auth/admin-login

Admin Login

Username: admin

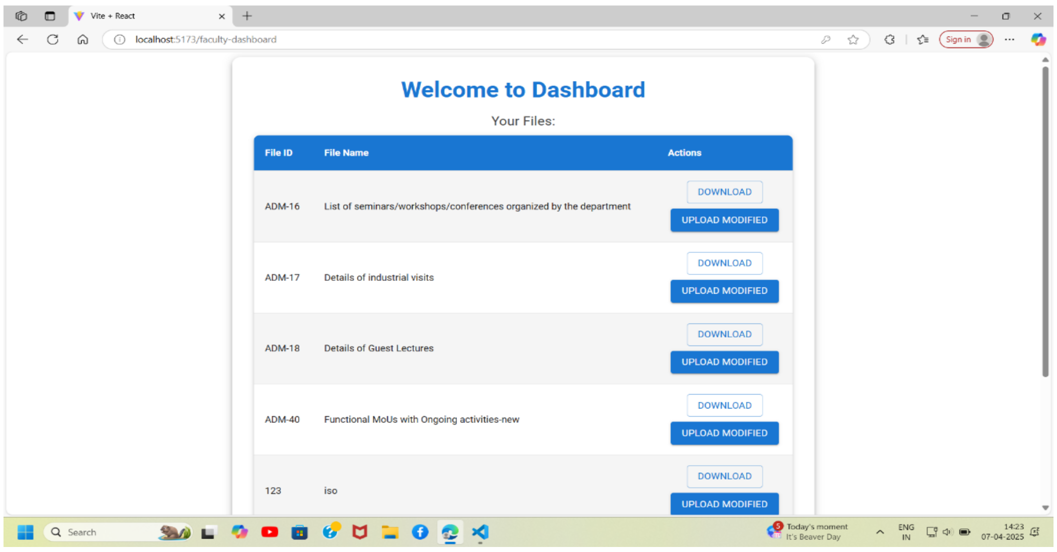
Password: ·········

Enter CAPTCHA

LOGIN

Back to Home

Figure 4.3: Admin Login Page


Vite + React

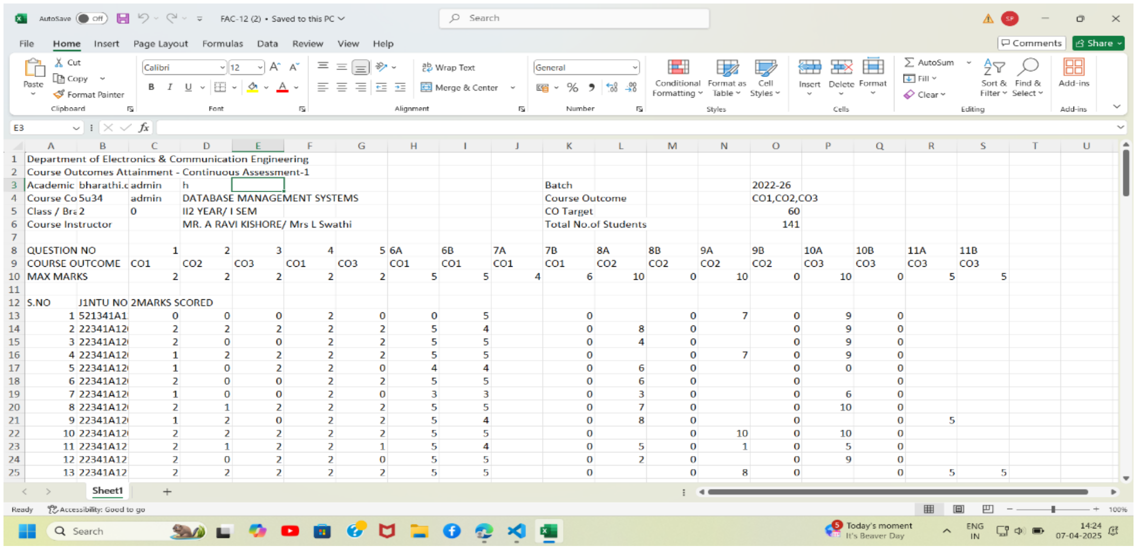
localhost:5173/faculty-dashboard

Welcome to Dashboard

Your Files:

File ID	File Name	Actions
ADM-16	List of seminars/workshops/conferences organized by the department	DOWNLOAD / UPLOAD MODIFIED
ADM-17	Details of industrial visits	DOWNLOAD / UPLOAD MODIFIED
ADM-18	Details of Guest Lectures	DOWNLOAD / UPLOAD MODIFIED
ADM-40	Functional MoUs with Ongoing activities-new	DOWNLOAD / UPLOAD MODIFIED
123	iso	DOWNLOAD / UPLOAD MODIFIED

Figure 4.12: Faculty Dashboard


FAC-12 (2) • Saved to this PC

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Department of Electronics & Communication Engineering																		
2	Course Outcomes Attainment - Continuous Assessment-1																		
3	Academic	bharathi.c	admin	h							Batch				2022-26				
4	Course Co	5u34	admin	DATABASE MANAGEMENT SYSTEMS							Course Outcome				CO1,CO2,CO3				
5	Class / Bra	2	0	II2 YEAR/ I SEM							CO Target				60				
6	Course Instructor			MR. A RAVI KISHORE/ Mrs L Swathi							Total No.of Students				141				
7																			
8	QUESTION NO		1	2	3	4	5	6A	6B	7A	7B	8A	8B	9A	9B	10A	10B	11A	11B
9	COURSE OUTCOME		CO1	CO2	CO3	CO1	CO3	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2	CO3	CO3	CO3	CO3
10	MAX MARKS		2	2	2	2	2	5	5	4	6	10	0	10	0	10	0	5	5
11																			
12	S.NO	J1NTU NO	2MARKS SCORED																
13	1	521341A1	0	0	0	2	0	0	5		0		0	7	0	9	0		
14	2	22341A12	2	2	2	2	2	5	4		0	8	0		0	9	0		
15	3	22341A12	2	0	0	2	2	5	5		0	4	0		0	9	0		
16	4	22341A12	1	2	2	2	2	5	5		0		0	7	0	9	0		
17	5	22341A12	1	0	2	2	0	4	4		0	6	0		0	0	0		
18	6	22341A12	2	0	0	2	2	5	5		0	6	0		0		0		
19	7	22341A12	1	0	0	2	0	3	3		0	3	0		0	6	0		
20	8	22341A12	2	1	2	2	2	5	5		0	7	0		0	10	0		
21	9	22341A12	1	2	0	2	2	5	4		0	8	0		0		0	5	
22	10	22341A12	2	2	2	2	2	5	5		0		0	10	0	10	0		
23	11	22341A12	2	1	2	2	1	5	4		0	5	0	1	0	5	0		
24	12	22341A12	2	0	2	2	0	5	5		0	2	0		0	9	0		
25	13	22341A12	2	2	2	2	2	5	5		0		0	8	0		0	5	5

Sheet1

Figure 4.14: Excel Sheet for Faculty Update

5. CONCLUSION

The Outcome-Based Education Management System Using MERN Stack presents an innovative and streamlined approach to managing academic records and monitoring student progress in educational institutions. By leveraging modern technologies - MongoDB, Express.js, React, and Node.js the system replaces outdated manual processes with a secure, role-based digital platform tailored to the needs of Outcome-Based Education (OBE). It enhances transparency, accountability, and collaboration through features like centralized file storage, version control, and real-time access, while also supporting sustainable and efficient administrative practices. Designed for scalability, the system holds significant potential for future enhancements such as AI-driven analytics, cloud integration, and automated report generation, making it a robust and future-ready solution for modern academic environments.

REFERENCES

[1]. Vinothini, C., et al. "NBA web portal: A Comprehensive Survey on NBA Accreditation and MERN Stack for the Purpose of Implementing a Portal." 2022 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2022.
[2]. Kansal, Anmol, and Neetu Mittal. "Nostalgia Hub Using MERN Stack: User-Centric Design and Privacy Preservation." 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE, 2024.
[3]. Chaudhary, Naren, and Neetu Mittal. "Leveraging Mongo DB for Efficient Data storage in MERN." 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE, 2024.
[4]. Masseroli, Marco, and Mario Marchente. "X-PAT: a multiplatform patient referral data management system for small healthcare institution requirements." IEEE Transactions on Information Technology in Biomedicine 12.4 (2008): 424-432.
[5]. Hayashi, Masaharu, et al. "A medical information management system using the semantic web technology." 2008 Fourth International Conference on Networked Computing and Advanced Information Management. Vol. 2. IEEE, 2008.
[6]. Jan, Saeed Ullah, Amin Ul Haq, and Saeed Ullah. "File Tracking System for Government Degree College (GDC) Wari Dir Upper, Khyber Pakhtunkhwa." Khyber Pakhtunkhwa (February 14, 2023) (2023).