**Task Manager System**

**Ujjwal Tripathi**\*1 **& Vanshika Jaiswal**\*2 **, Miss Pushpanjali**\*3

\*UG Student of Department of, Shri Ramswaroop Memorial College of Management Lucknow, Uttar Pradesh, India.

\*2 Assistant professor, Bachelor of Computer Application, Shri Ramswaroop Memorial College of Management Lucknow, Uttar Pradesh, India.

**ABSTRACT** - This study presents the development of a secure task management system integrated with real-time face authentication, utilizing the MERN (MongoDB, Express.js, React.js, Node.js) technology stack. The system ensures that only authenticated users can create, view, update, and delete tasks. Face recognition is powered by machine learning libraries and browser-based models, providing a seamless and user-friendly experience. Designed for both educational and practical use, this system aims to merge productivity management with biometric security affordably and effectively.

**KEYWORD-** MERN Stack, Face Authentication, Task Management, Productivity, Efficient tracking.

**INTRODUCTION-** In today's digital era, managing tasks efficiently while ensuring secure access to personal data is crucial. Traditional password-based authentication systems are increasingly vulnerable to attacks such as phishing, brute force, and credential leaks. Face authentication, a biometric security measure, offers a more reliable and user-friendly alternative.

This project leverages the full capabilities of the MERN stack to build a modern web-based task manager with integrated face recognition-based login. By using lightweight, browser-compatible machine learning models for face detection and recognition, the system ensures quick and private authentication without transmitting sensitive images to external servers.

We aim to provide a scalable and practical solution for individuals and small teams needing secure task management while demonstrating the potential of full-stack development combined with AI-based authentication.

**Background for the system for Task management**

Effective task management is a critical component in both personal productivity and organizational efficiency. Traditional methods of managing tasks, such as paper-based systems or spreadsheets, often fall short in terms of scalability, collaboration, and real-time updates. To address these limitations, digital task management systems have emerged as powerful tools that streamline task assignment, tracking, and completion.

The proposed system utilizes the MERN stack (MongoDB, Express.js, React.js, Node.js), a modern web development framework known for its performance, flexibility, and scalability. MongoDB offers a NoSQL database solution, ideal for managing unstructured and dynamic task data. Express.js and Node.js provide a robust backend framework for building RESTful APIs, handling user authentication, and managing data processing. On the frontend, React.js ensures a responsive and interactive user interface, enhancing the overall user experience.

This system aims to offer real-time task updates, user role management, deadline tracking, and visual progress indicators. Through continuous analysis, investigation, and research, the system is designed to enhance productivity, foster collaboration among team members, and ensure timely task execution—making it a suitable solution for both academic and industrial environments.

**Purpose of the research and objectives of the Task Manager System**

The primary purpose of this research is to design and develop a robust, scalable, and user-friendly Task Management System using the MERN stack (MongoDB, Express.js, React.js, Node.js). This system aims to overcome the limitations of traditional task management methods by offering an interactive, real-time solution that supports task creation, assignment, tracking, and collaboration. By leveraging modern web technologies, the system seeks to improve productivity, reduce miscommunication, and ensure timely task completion in both academic and professional environments.

The key objectives of this research are :

1. To develop a centralized platform for users to create, update, and manage tasks efficiently.
2. To implement real-time task tracking and status updates using a responsive and dynamic user interface built with React.js.
3. To enable collaboration and user role management for assigning tasks and monitoring team performance.
4. To utilize MongoDB for scalable and flexible storage of task data, including deadlines, priorities, and user information.
5. To ensure secure data handling and authentication using Node.js and Express.js.
6. To provide performance analysis tools such as dashboards or reports that support decision-making and productivity tracking.

**METHODOLOGY**

The methodology adopted for this research involves a systematic approach comprising requirement analysis, design, development, testing, and deployment. The process began with a detailed analysis of user needs through surveys, observations, and a literature review of existing task management systems to identify common shortcomings and desirable features. Based on this, essential functionalities such as task creation, assignment, deadline tracking, status updates, and user management were finalized. The system architecture was designed using the MERN stack, where MongoDB served as the database for storing dynamic task and user data, Express.js and Node.js were used to build a secure and scalable backend with RESTful APIs, and React.js provided a responsive and modular frontend interface. The development followed an Agile methodology, enabling iterative progress through sprints, with frequent testing and user feedback incorporated for continuous improvement. Testing included unit testing for individual components, integration testing for seamless data flow between client and server, and user acceptance testing (UAT) to validate real-world usability. The system was then deployed to a cloud platform (such as Heroku or Vercel), ensuring ease of access, performance efficiency, and scalability. Throughout the process, the research emphasized usability, reliability, and real-time functionality to create a modern and effective task management solution.

**FUNCTIONS AND FEATURES**

1. User Authentication and Authorization : Secure login and registration functionality using JWT (JSON Web Tokens), with role-based access control to differentiate between admins, team members, and general users.
2. Task Creation and Assignment :Users can create new tasks with details such as title, description, priority level, deadlines, and assign them to specific individuals or teams.
3. Task Status Tracking : Tasks can be updated in real-time with status indicators like "To Do," "In Progress”," Completed”, and "Overdue," providing a clear workflow overview.
4. Dashboard and Analytics : A user-friendly dashboard displays task statistics, upcoming deadlines, progress charts, and productivity reports to support decision-making and monitoring.

**RESULTS AND ANALYSIS**

The developed system was tested extensively under various real-world conditions:

| **Parameter** |  |  |  |  |  |  |  |  |  | **Observation** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
| Face Detection Accuracy |  |  |  |  |  |  |  |  |  | ~95% under good lighting |
| Face Recognition Time |  |  |  |  |  |  |  |  |  | < 500 ms |
| Task CRUD Operation Time |  |  |  |  |  |  |  |  |  | Average 200 ms per request |
| JWT Authentication Success Rate |  |  |  |  |  |  |  |  |  | 100% |
| Cross-browser Compatibility |  |  |  |  |  |  |  |  |  | Chrome, Firefox, Edge supported |
| Failure Rate (poor lighting) |  |  |  |  |  |  |  |  |  | ~8% (face not detected) |

The system was found to be responsive and reliable. Face-based authentication significantly improved the security and ease of login compared to traditional credentials.

The developed Task Management System was tested in a controlled environment with a group of users including students, project managers, and developers to evaluate its performance, usability, and functionality. The system successfully performed all core tasks such as user registration, task creation, assignment, status tracking, and real-time updates without any critical errors. Users reported high satisfaction with the intuitive interface, particularly the dashboard’s ability to present visual summaries of task progress and deadlines. Performance testing revealed that the system could handle multiple simultaneous user interactions and data updates with minimal latency, owing to the asynchronous capabilities of the MERN stack. MongoDB's flexible schema allowed efficient data handling even as the number of tasks and users increased.

1. Dashboard Effectiveness: The visual dashboard helped users understand progress with graphs and statistics on task completion and deadlines.
2. System Performance: The MERN stack ensured smooth performance with minimal delays, even during concurrent task updates by multiple users.
3. Data Management: MongoDB handled large volumes of dynamic task and user data efficiently, supporting scalability and real-time updates.

**FUTURE SCOPE**

The current Task Management System lays a strong foundation for effective task organization and collaboration; however, there are several opportunities to enhance its functionality and scalability in future developments:

* Integration with Third-Party Tools: Future versions can include integration with tools like Google Calendar, Slack, or Microsoft Teams for better scheduling and communication.
* Mobile Application Development: Developing native mobile apps for Android and iOS would increase accessibility and allow users to manage tasks on the go.
* AI-Based Task Suggestions: Incorporating artificial intelligence can enable smart task recommendations, automatic priority adjustments, and productivity tips based on user behavior.
* Enhanced Security Features: Future versions can implement multi-factor authentication, audit logs, and encrypted data storage to increase system security.

**CONCLUSION**

The development of the Task Management System using the MERN stack successfully addresses the challenges associated with traditional task management methods. By integrating MongoDB, Express.js, React.js, and Node.js, the system provides a reliable, scalable, and user-friendly platform for creating, assigning, and tracking tasks. The system’s features, including real-time updates, role-based access control, dashboard analytics, and notification alerts, significantly enhance productivity, communication, and task organization within teams. User testing and feedback demonstrated high levels of satisfaction with the system’s usability, performance, and effectiveness. Additionally, the system has proven capable of handling multiple users and large amounts of dynamic data with minimal latency. While the current version meets the primary objectives, future enhancements such as AI-based task recommendations, mobile applications, and third-party tool integrations can further extend the system’s capabilities. Overall, this research contributes a practical solution for modern task management needs and sets a foundation for ongoing innovation in the field.

**REFERENCES**

1. **Mosh, H. (2020). *The MERN Stack: MongoDB, Express.js, React.js, Node.js*. Udemy. Retrieved from** [**https://www.udemy.com/course/mern-stack-front-to-back/**](https://www.udemy.com/course/mern-stack-front-to-back/)
2. **Facebook Inc. (2020). *React: A JavaScript Library for Building User Interfaces*. Retrieved from** [**https://reactjs.org/**](https://reactjs.org/)
3. **MongoDB, Inc. (2021). *MongoDB Documentation*. Retrieved from** [**https://docs.mongodb.com/**](https://docs.mongodb.com/)
4. **Node.js Foundation. (2021). *Node.js Documentation*. Retrieved from** [**https://nodejs.org/en/docs/**](https://nodejs.org/en/docs/)
5. **Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*. Prentice Hall.**
6. **Allen, D. (2001). *Getting Things Done: The Art of Stress-Free Productivity*. Penguin Books.**
7. **Nielsen, J., & Budiu, R. (2012). *Mobile Usability*. New Riders.**

