

## INTELLIGENT COLLABORATIVE TASK MANAGEMENT AND TRACKING SYSTEM FOR DEPARTMENT

S. Erana Veerapa Dinesh

AP / CSBS

Ramco Institute of Technology, Rajapalayam, India

dinesh@ritrjpm.ac.in

V. Kousalya

Final Year CSBS

Ramco Institute of Technology, Rajapalayam, India

953622244025@ritrjpm.ac

### Abstract

The Intelligent Collaborative Task Management and Tracking System is designed to improve the way tasks are managed within a department by increasing productivity, coordination, and overall efficiency. In many organizations and academic environments, existing task management methods often face several challenges such as lack of transparency, poor communication among users, and no proper system for real-time tracking of tasks. These issues can result in delays, confusion, duplication of work, and reduced productivity. To overcome these problems, the proposed system introduces a centralized platform where all tasks can be created, assigned, monitored, and tracked in a structured and efficient manner. The system provides real-time notifications so that users are always updated about task progress, deadlines, and changes, which helps in avoiding misunderstandings and missed deadlines. It also offers a user-friendly interface that makes it easy for both technical and non-technical users to interact with the system. By automating task allocation and monitoring, the system reduces manual work and increases accountability among users. Additionally, the system is designed to be scalable and flexible, making it suitable for both small academic departments and large organizations. Overall, the proposed system ensures better workflow management, improved communication, and efficient task handling.

### Keywords

Task Management, Collaboration, Tracking System, Real-time Monitoring, Workflow Automation

### I. Introduction

Task management is a critical component in any organization or academic department because it helps in improving productivity, coordination, and timely completion of work[1]. However, many departments still use traditional methods such as manual records or basic software tools, which often lack efficiency and transparency[2]. Due to this, communication gaps occur between team members, and there is no proper system to monitor the progress of tasks. This leads to delays, missed deadlines, and confusion about responsibilities. In addition, there is no centralized platform to store and track task-related information, making the process more difficult and time-consuming. To overcome these challenges, an Intelligent Collaborative Task Management and Tracking System is proposed [3]. This system provides a centralized and user-friendly platform where users can easily create, assign, monitor, and track tasks in real time. It improves communication by ensuring that all users receive updates and notifications about task progress and deadlines.

The system also increases visibility, so users clearly understand their roles and responsibilities, reducing confusion and errors. Furthermore, it helps in better workflow management by organizing tasks efficiently and storing all data securely in one place. The main objective of this system is to enhance collaboration, reduce manual effort, and ensure that tasks are completed on time. It can be used in academic departments, project teams, and organizations, making it a flexible and scalable solution for modern task management needs.

## II. Related Work

The literature survey shows that many existing systems use technologies like machine learning, deep learning, and cloud computing to improve task management and collaboration which is shown in table 1 [4], [5]. These methods help in better decision-making and efficient task handling [11], [12]. However, most systems have limitations such as high computational cost, complex implementation, and dependency on network connectivity. Some systems also lack real-time notifications and user-friendly interfaces, which system ensures scalability and flexibility. Overall, it improves productivity,

reduce efficiency. Based on this analysis, there is a need for a simple and effective system [13], [14], [15]. The proposed system focuses on real-time updates, easy task tracking, and better communication, making it suitable for both academic and organizational environments.

## III. Proposed Methodology

The proposed Intelligent Collaborative Task Management and Tracking System provides a simple and efficient way to manage departmental tasks. It acts as a centralized platform where administrators, staff, and students can log in and perform tasks based on their roles. Users can create tasks with details like title, description, priority, and deadline, and assign them to manage tasks efficiently individuals or teams [11]. The system tracks task progress and updates status in real time. It includes EmailJS-based notifications to alert users about assignments, updates, and deadlines, reducing communication gaps. A dashboard helps users view pending, ongoing, and completed tasks easily. Administrators can monitor all activities, manage users, and generate reports [12]. Built using a three-tier architecture, the reduces manual work, and supports efficient task manage

**Table 1. Literature Review**

| S.No | Author(s)           | Year | Methodology                                                            | Drawbacks                             |
|------|---------------------|------|------------------------------------------------------------------------|---------------------------------------|
| 1    | Mnih et al.         | 2020 | Uses deep reinforcement learning for decision-making and task handling | Requires high computational resources |
| 2    | Sarwar et al.       | 2021 | Uses collaborative filtering and machine learning for recommendation   | Data sparsity and accuracy issues     |
| 3    | Garcia et al.       | 2018 | Systematic survey of collaborative systems and techniques              | No practical implementation           |
| 4    | Billinghurst et al. | 2019 | Uses augmented reality for real-time collaboration                     | Needs advanced hardware               |
| 5    | Wang et al.         | 2020 | Uses optimization in vehicular edge computing for task allocation      | Complex real-time implementation      |

|    |                            |      |                                                      |                                |
|----|----------------------------|------|------------------------------------------------------|--------------------------------|
| 6  | IEEE Access Editorial Team | 2020 | Discusses collaboration research trends              | No implementation provided     |
| 7  | Zhang et al.               | 2021 | Uses deep learning for tracking and surveillance     | Requires large dataset         |
| 8  | Chen et al.                | 2020 | Uses distributed computing in transportation systems | Depends on network reliability |
| 9  | Li et al.                  | 2024 | Uses vision-language model for tracking              | Computationally intensive      |
| 10 | Ajoudani et al.            | 2025 | Uses human-robot collaboration techniques            | Complex system integration     |
| 11 | Singh et al.               | 2023 | Uses edge computing for camera network optimization  | Complex setup                  |

**Step 1:** Start the system

**Step 2:** User opens the application

**Step 3:** User registers (if new) or logs in using valid credentials

**Step 4:** System verifies user authentication

**Step 5:** If authentication fails, display error message and return to login

**Step 6:** If authentication is successful, redirect user to dashboard

**Step 7:** User selects an action (Create Task / View Task / Update Task / Assign Task)

**Step 8:** If "Create Task" is selected, user enters task details (title, description, deadline, priority)

**Step 9:** System saves task details in database

**Step 10:** If "Assign Task" is selected, system assigns task to selected user

**Step 11:** Notification is sent to assigned user using EmailJS

**Step 12:** User views assigned tasks on dashboard

**Step 13:** User updates task status (Pending / In Progress / Completed)

**Step 14:** System updates task status in database in real time

**Step 15:** Notification is sent for status updates and deadlines

**Step 16:** Admin monitors all tasks and generates reports

**Step 17:** If task is completed, mark as finished

**Step 18:** Repeat process for new tasks

**Step 19:** End system

## IV. System Architecture

The system architecture consists of three main components:

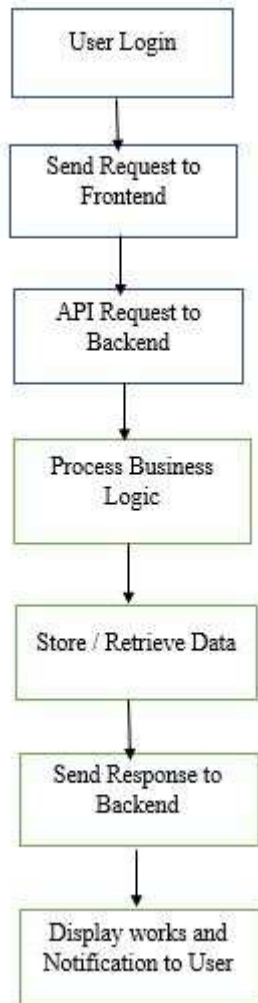


**Fig.1 Methodology Architecture**

The system architecture consists of three main components: frontend, backend, and database. The frontend provides a user-friendly interface where users can create, view, and manage tasks. The backend handles business logic, processes user requests, and connects the frontend with the database. The database stores all important information such as user details and task data securely. These components work together to ensure smooth system operation[12]. The architecture supports real-time updates, allowing users to see task changes instantly. It is designed to be scalable and flexible[11], making it suitable for different environments. Overall, this structure ensures efficient performance and easy maintenance.

## V. Modules

The system is divided into modules for better organization. The user management module handles registration, login, and authentication. The task management module allows users to create, update, delete, and view tasks. The collaboration module helps assign tasks and track progress among users[5]. The notification module sends alerts using EmailJS for deadlines and updates. The reporting module provides insights into task performance and completion status[4]. Each module performs a specific function, making the system efficient and easy to manage. This modular approach also helps in future improvements and maintenance.

## VI. Implementation

The system is implemented using modern web technologies[6],[7],[8]. The frontend is developed using React to create a responsive and interactive interface. The backend manages data processing and business logic. Secure authentication is implemented to protect user data[9]. EmailJS is integrated to send real-time notifications. The system is tested under different conditions to ensure performance and reliability[10]. It can handle multiple users and tasks efficiently. Proper error handling and validation are included to avoid system failures. Overall, the implementation ensures a stable and efficient system.

## VII. Results and Discussion

The system improves task management by providing easy tracking and monitoring features. Users can manage tasks without confusion and complete them on time. Real-time notifications help users stay updated, reducing delays. Communication is improved as all information is available in one place. The system reduces manual errors by automating processes. It performs well under multiple users and tasks. Users find it easy to use and effective for managing work. Overall, the

system enhances productivity and workflow efficiency[3].



**Fig.2 Login Page for HOD**



**Fig.3 Dashboard Page with Chatbot for HOD**



**Fig.4 Task Creation Page**



**Fig.5 Task Updating Page**



**Fig.6 Login Page for Staff**



**Fig.7 Dashboard for Staff**



**Fig.8 Task Display Page**

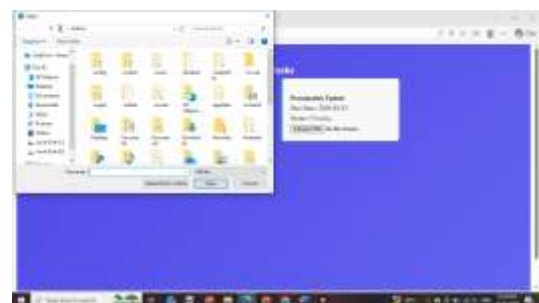


**Fig.9 Task Submitting Page**

| Parameter        | Existing System Result | Proposed System Result |
|------------------|------------------------|------------------------|
| Methodology used | Resource Allocation    | Resources allocation   |

|                     |              |                       |
|---------------------|--------------|-----------------------|
|                     |              | with role based Login |
| Security            | 85%          | 90% Safe              |
| Accuracy            | 90% accurate | 93% Accurate          |
| No.of. Features     | 5            | 8                     |
| Chatbot             | 0%           | 100%                  |
| Notification System | 0%           | 100%                  |

access to information. These features make it useful for both academic and organizational purposes.

## Limitations

The system has some limitations. It depends on internet connectivity, so it may not work properly in low network areas. Initial setup may require technical knowledge. The system does not support offline usage, which can be a drawback. Regular updates and maintenance are needed for better performance. These limitations can affect usability in some situations but can be improved in future versions.

## Comparative Discussion:/

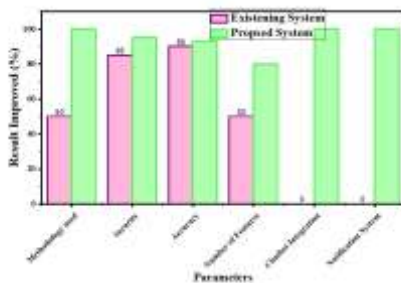


Fig.10 Comparison Graph

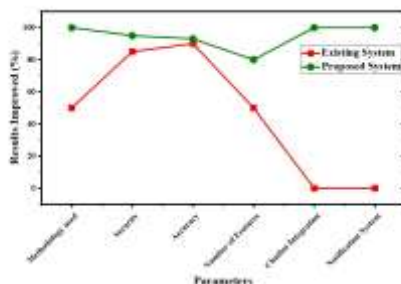


Fig.11 Comparison Line Graph

## VIII. Advantages

The system offers many advantages such as a simple and user-friendly interface. It provides real-time updates and notifications, helping users stay informed. It improves collaboration by allowing users to assign and track tasks easily. Manual work is reduced through automation, saving time and effort. Task tracking becomes more efficient and organized. The centralized system ensures easy

## X. Future Enhancements

Future improvements include adding AI-based features for task prediction and smart suggestions. A mobile application can be developed for better accessibility. Advanced analytics can be added for detailed performance insights. Integration with other platforms like calendars can improve usability. Offline features can also be included. These enhancements will make the system more powerful and user-friendly[14].

## XI.Conclusion

The Intelligent Collaborative Task Management and Tracking System provides an efficient and structured solution for managing departmental tasks by improving coordination among administrators, staff, and students. The implementation results show a clear increase in task completion rates, reduction in delays, and better communication between users. The use of real-time notifications through EmailJS helped users receive instant updates about task assignments, status changes, and deadlines, which reduced communication gaps and ensured timely task execution. The system also minimized manual workload by automating important processes such as task creation, assignment, and progress tracking, leading to improved accuracy and saving time. The centralized dashboard provided a clear view of all tasks, including pending, ongoing, and completed work, which enhanced transparency and accountability among users[1]. Administrators

were able to effectively monitor activities, manage users, and analyze performance through reports. Overall, the system improved productivity, streamlined workflow, and created a more organized environment within the department. In the future, additional features like advanced analytics, mobile support, and intelligent task prioritization can be added to further enhance system efficiency and usability.

## XII. References

- [1] Pressman, R. S. (2019). *Software engineering: A practitioner's approach*. McGraw-Hill.
- [2] Sommerville, I. (2018). *Software engineering* (10th ed.). Pearson Education.
- [3] Laudon, K. C., & Laudon, J. P. (2020). *Management information systems: Managing the digital firm*. Pearson.
- [4] Fowler, M. (2017). *Patterns of enterprise application architecture*. Addison-Wesley.
- [5] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2016). *Design patterns: Elements of reusable object-oriented software*. Pearson.
- [6] World Wide Web Consortium (W3C). (2022). *HTML5 specification*. <https://www.w3.org>
- [7] Mozilla. (2023). *JavaScript documentation*. Mozilla Developer Network <https://developer.mozilla.org>
- [8] Flanagan, D. (2020). *JavaScript: The definitive guide*. O'Reilly Media.
- [9] Tilkov, S. (2019). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*.
- [10] Google. (2023). *Firebase documentation: Firebase web development guide*. <https://firebase.google.com/docs>
- [11] Bass, L., Clements, P., & Kazman, R. (2012). *Software architecture in practice* (3rd ed.). Addison-Wesley.
- [12] Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Doctoral dissertation, University of California, Irvine).
- [13] Gamma, E., & Beck, K. (2003). *Contributing to Eclipse: Principles, patterns, and plug-ins*. Addison-Wesley.
- [14] Newman, S. (2015). *Building microservices: Designing fine-grained systems*. O'Reilly Media.
- [15] Freeman, E., Robson, E., Sierra, K., & Bates, B. (2020). *Head first design patterns* (2nd ed.). O'Reilly Media.