

**TrackMaven**

**A Minor Project Synopsis Submitted to**



**Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal  
Towards Partial Fulfillment for the Award of**

**Bachelor of Technology  
(Computer Science and Engineering)**

**Under the Supervision of  
Prof. Ritika Bhatt**

**Submitted By  
Adarsh Trivedi  
(0827CS223D02)**



**Department of Computer Science and Engineering  
Acropolis Institute of Technology & Research, Indore**

**Jan-June 2024**

## 1. Introduction of the Project

A task tracker project is akin to a digital conductor orchestrating the symphony of productivity, harmonizing the efforts of individuals or teams towards shared goals. It functions as a finely tuned instrument, allowing users to compose, delegate, and monitor tasks with precision. Through its ensemble of features such as task allocation, progress tracking, and deadline reminders, it ensures a seamless performance, minimizing discord and maximizing efficiency in the pursuit of project completion.

## 2. Objective

The objective of the task tracker project is to develop a comprehensive and user-friendly software solution that facilitates efficient task and project management for individuals and teams. By leveraging modern technologies like the MERN (MongoDB, Express.js, React, Node.js) stack, the project aims to address common challenges such as disorganization, poor communication, and inefficient workflow management. Key objectives include providing a visually intuitive interface for organizing tasks, enabling real-time collaboration and communication features, offering customizable workflows, implementing advanced automation capabilities, ensuring scalability and performance, and enhancing data visualization for better insights into project progress and performance metrics. Ultimately, the goal is to optimize productivity, streamline processes, and improve project outcomes.

## 3. Scope

The scope of the task tracker project encompasses creating a robust software solution for efficient task and project management. It includes features such as task organization, user management, collaboration tools, workflow customization, automation, reporting, integration, scalability, performance, user experience, security, and ongoing support. The project aims to streamline processes, improve productivity, enhance communication and collaboration, provide insights into project progress, ensure data security and privacy, and deliver a seamless user experience. This comprehensive scope addresses various aspects of task and project management to meet the diverse needs of individuals and teams across different industries and domains.

## 4. Study of Existing System

Include comparative case study of at least 5 existing systems with their merits and demerits. This will help in formulation of further objectives that could be addressed within the project.

Existing System 1: Trello

- Problems Addressed

Trello addresses task organization, collaboration, and progress tracking, facilitating efficient project management and team communication through visual boards.

- Advantages

Trello offers visual task management, easy collaboration, customizable workflows, integration with other tools, and accessibility across devices for teams.

- Disadvantages

Trello lacks advanced project management features, may feel overwhelming with extensive tasks, and requires manual effort for detailed reporting.

- Gaps Identified

Trello lacks built-in time tracking, resource allocation, Gantt charts, and comprehensive reporting features compared to more robust project management tools.

- Reference link : <https://trello.com/>

## Existing System 2 : Asana

- Problems Addressed :

Asana addresses task organization, team communication, project tracking, deadline management, and collaboration, facilitating efficient workflow management and productivity enhancement.

- Advantages :

Asana provides a structured system for organizing tasks, ensuring clarity and efficiency in workflow management.

The platform fosters seamless communication among team members, reducing misunderstandings and facilitating collaboration.

Asana allows for real-time tracking of project progress, enabling teams to stay on schedule and meet deadlines.

- Disadvantages

Asana's extensive features may overwhelm new users, requiring time and training to fully grasp its functionalities.

Compared to some other platforms, Asana may have fewer integrations with third-party tools, limiting workflow customization.

The learning curve for mastering Asana's advanced features may be steeper for some users, hindering initial adoption.

- Gaps Identified

Asana lacks advanced reporting, resource allocation, dependency management, budget tracking, and portfolio management features, which are crucial for comprehensive project management and optimization of workflows at scale.

- Reference link : <https://asana.com/>

## Existing System 3 : Jira

- Problems Addressed

Jira addresses the complexities of software development project management, including issue tracking, agile methodologies, collaboration, and workflow customization for teams.

- Advantages

Jira offers comprehensive issue tracking capabilities, allowing teams to efficiently identify, prioritize, and resolve issues throughout the development process.

The software provides extensive support for agile methodologies, enabling teams to implement Scrum, Kanban, or other agile frameworks seamlessly.

Jira allows teams to tailor workflows to match their specific processes, ensuring flexibility and alignment with organizational needs.

- Disadvantages

Jira's extensive features can lead to a steep learning curve, particularly for new users or teams unfamiliar with agile methodologies.

While customizable, the abundance of options may overwhelm users, potentially leading to inefficient workflows if not properly managed.

- Gaps Identified

Gaps in Jira include complexity for new users, limited non-software applications support, scalability challenges for large enterprises, reporting limitations, and potential shortcomings in dependency management, hindering comprehensive project management in diverse contexts.

- Reference link : <https://www.atlassian.com/software/jira>

#### Existing System 4 : Microsoft Planner

- Problems Addressed

Microsoft Planner addresses the need for simple task management and collaboration within Office 365, offering users an intuitive platform for organizing and tracking their work efficiently.

- Advantages

Microsoft Planner seamlessly integrates with other Office 365 applications, facilitating collaboration and productivity within the Microsoft ecosystem.

With its user-friendly interface similar to other Microsoft products, Planner offers easy adoption for users already familiar with Microsoft tools.

Planner enables users to organize tasks into boards, lists, and cards, providing a visual and intuitive way to manage projects and workflows.

- Disadvantages

Microsoft Planner lacks advanced project management features compared to dedicated project management tools, limiting its suitability for complex projects.

Planner does not offer robust features for managing task dependencies, making it challenging to visualize and track interdependent tasks.

- Gaps Identified

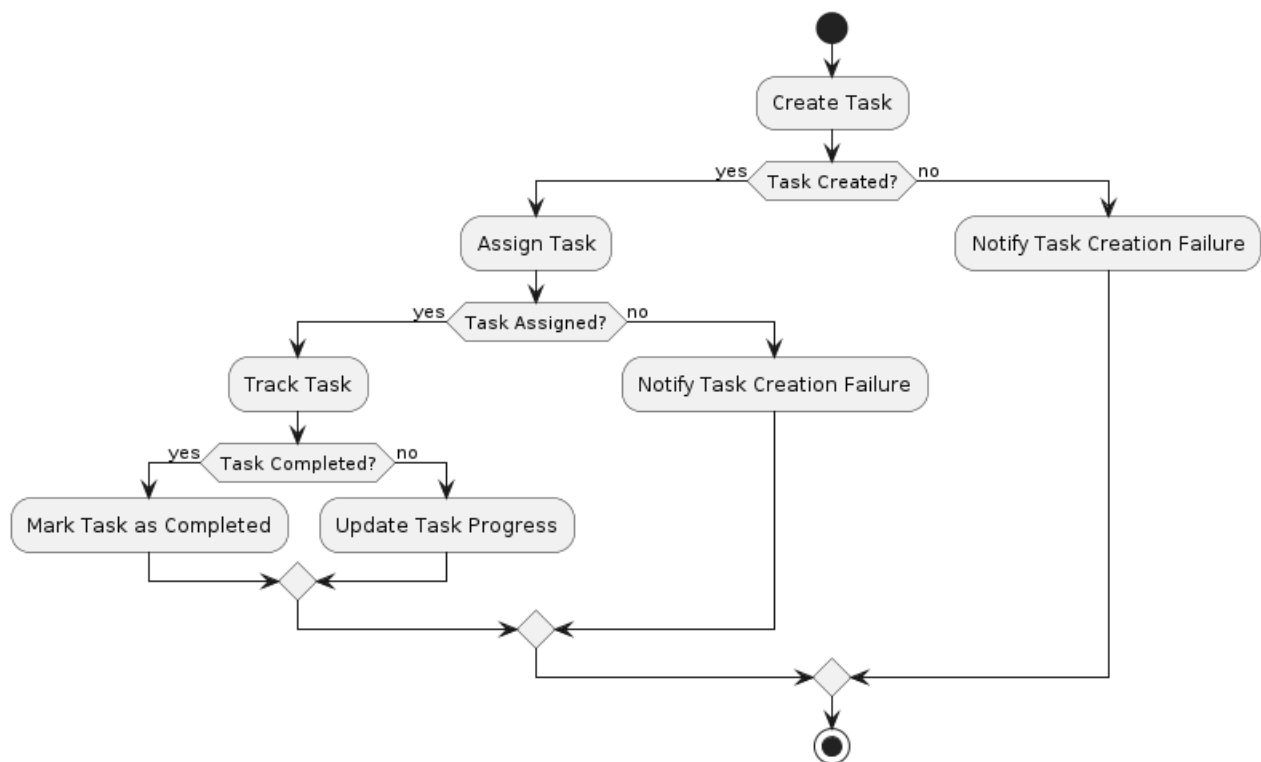
Gaps in Microsoft Planner include its lack of advanced project management features, such as comprehensive dependency management and reporting capabilities. Additionally, while

it integrates well with Office 365, its limited integration options with third-party tools and scalability challenges may hinder its suitability for larger or more complex projects and teams.

- Reference link : <https://tasks.office.com/>

## 5. Project Description

Introducing TrackMaven, your all-in-one solution for streamlined project management and enhanced team collaboration. With TrackMaven, you can create teams, assign tasks, track progress, and manage to-dos efficiently, all in one intuitive platform. Easily create teams for different projects or departments within your organization and assign tasks to team members with deadlines and priorities. Track the progress of tasks in real-time, ensuring deadlines are met and projects stay on schedule. With TrackMaven, you can create and manage to-do lists for individual tasks or projects, keeping everyone organized and focused on their objectives. Simplify project management, improve team productivity, and ensure that tasks are completed on time, every time, with TrackMaven.



## 6. Methodology/Planning of the Project work

### 1. Planning (1 week):

Define the project scope and objectives.

Identify essential features and functionalities.

Create a simplified project plan.

## 2. Design (1 week):

Develop basic wireframes for the user interface.

Plan a straightforward database schema for task data.

Outline the system's architecture.

## 3. Development (2-3 weeks):

Start with user registration and authentication.

Implement task creation, task assign and task management features.

Focus on the most essential functionalities while keeping the system minimal.

## 4. Testing (1-2 weeks):

Perform basic testing, including functionality and usability.

Address critical issues and ensure the platform is stable.

## 5. Deployment (1 week):

Set up hosting and server configurations.

Deploy the platform for limited testing with a small user group.

## 6. Final Adjustments (1 week):

Make any last-minute refinements based on feedback.

Ensure the platform is functional and user-friendly.

## 7. Documentation (1 week):

Create a brief user guide and project documentation.

Prepare a presentation for project evaluation.

## 8. Presentation and Submission (1 week):

Present your project to your college and submit all required documents.

## 7. Features

### 1. User Authentication and Authorization:

Allow users to create accounts and log in securely.

Implement role-based access control to manage user permissions.

### 2. Team Creation:

Enable users to create teams for different projects or departments.

Assign team members and set team leaders.

### 3. Task Management:

Create, edit, and delete tasks.

Assign tasks to team members.

Set due dates, priorities, and statuses for tasks.

### 4. Task Tracking:

Track the progress of tasks in real-time.

View task status, completion percentage, and deadlines.

### 5. To-Do Lists:

Create and manage to-do lists for individual tasks or projects.

Organize tasks by priority, due date, or status.

### 6. Notification System:

Send notifications for task assignments, updates, and approaching deadlines.

Allow users to customize notification preferences.

### 7. Dashboard and Reporting:

Provide a dashboard for users to view their tasks, to-do lists, and team activities.

Generate reports on task progress, team productivity, and project status.

### 8. Integration with Calendar and Email:

Sync tasks with users' calendars.

Send task reminders and updates via email.

### 9. Mobile Accessibility:

Develop a mobile app for users to access the task tracker tool on the go.

Ensure cross-platform compatibility for seamless user experience.

#### 10. Data Security and Backup:

Implement data encryption and secure authentication mechanisms.

Regularly backup user data to prevent data loss.

## 8. System architecture

The task tracker software will follow a client-server architecture, with a web-based client interface and a backend server responsible for handling user requests, data storage, and business logic. Here's an overview of the system architecture:

#### 1. Client Interface:

The client interface will be a web application accessible through a modern web browser.

Users will interact with the system through the client interface, accessing features such as task creation, team management, and reporting.

#### 2. Backend Server:

The backend server will be responsible for processing client requests, executing business logic, and managing data.

It will be developed using a scalable and robust framework such as Node.js, Django, or Spring Boot.

#### 3. Database:

The system will utilize a relational database management system (RDBMS) like MySQL, PostgreSQL, or Microsoft SQL Server to store user data, tasks, teams, and other relevant information.

The database will store user authentication details, task details, team information, and user preferences.

#### 4. API Layer:

An API layer will provide a structured interface for communication between the client interface and the backend server.

RESTful APIs will be developed to handle CRUD (Create, Read, Update, Delete) operations for tasks, teams, users, and other entities.

#### 5. Authentication and Authorization:

The system will implement user authentication and authorization using tokens or sessions.



Authentication services will ensure secure access to user accounts and data, while authorization services will control user permissions and access levels.

#### 6. Notification Service:

A notification service will be integrated to send real-time notifications to users for task assignments, updates, and deadlines.

Notifications will be delivered via email, in-app messages, or push notifications.

### **Interaction between Components:**

#### Client-Server Interaction:

The client interface will send HTTP requests to the backend server to perform various operations such as task creation, assignment, and retrieval.

The backend server will process these requests, execute the necessary business logic, and interact with the database to fetch or update data.

#### Database Interaction:

The backend server will interact with the database to store and retrieve user data, tasks, teams, and other relevant information.

Database queries will be optimized for performance and scalability to ensure efficient data retrieval and storage.

#### Authentication and Authorization:

User authentication requests will be handled by the backend server, which will verify user credentials against the database.

Authorization services will ensure that authenticated users have the necessary permissions to access specific features and data within the system.

#### Notification Service Integration:

The backend server will interact with the notification service to send real-time notifications to users.

Notifications will be triggered based on user actions such as task assignments, updates, and approaching deadlines.

## **9. User Interface (UI)**

The user interface of the task tracker software will be intuitive, user-friendly, and responsive, providing users with easy access to task management, team collaboration, and reporting features. Here's a description of the user interface design:

#### 1. Dashboard:

Upon logging in, users will be greeted with a personalized dashboard displaying

an overview of their tasks, to-do lists, and team activities.

The dashboard will include widgets for task status, upcoming deadlines, and project progress.

## 2. Task Management:

Users can create, edit, and delete tasks directly from the dashboard.

Each task will display relevant details such as task name, description, due date, priority, and status.

Tasks can be organized into categories such as "To-Do," "In Progress," and "Completed" for easy tracking.

## 3. Team Management:

Users can create and manage teams for different projects or departments.

Team members can be added or removed, and team roles can be assigned (e.g., team leader, member).

## 4. To-Do Lists:

Users can create and manage to-do lists for individual tasks or projects.

To-do lists can be organized by priority, due date, or status, helping users stay organized and focused on their objectives.

## 5. Reporting:

The reporting module will provide users with insights into task progress, team productivity, and project status.

## 6. Navigation:

The user interface will feature a clear and intuitive navigation menu for easy access to different sections of the application.

Navigation options may include "Dashboard," "Tasks," "Teams," "Reports," and "Settings."

# 10. Technology Stack

### Frontend:

**React.js:** React.js is a popular JavaScript library for building user interfaces. It allows you to create dynamic and interactive web pages efficiently.

### Backend:

**Node.js:** Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows you to easily build scalable network applications.

**Express.js:** Express.js is a web application framework for Node.js. It provides a robust set of features for web and mobile applications.

Database:

MongoDB: MongoDB is a widely used NoSQL database management system. It provides high performance, high availability, and easy scalability.

## 11. Testing Plan

Testing is a critical phase in ensuring the functionality and reliability of the Task tracker tool. Here's a testing plan that outlines the various aspects of testing: Certainly, for a minor college project, testing can be conducted in a straightforward manner to ensure functionality and reliability. Here's a simplified testing plan:

### 1. Unit Testing:

Frontend:

Test React components using React Testing Library or Jest.

Ensure proper rendering, state management, and user interactions.

Backend:

Test Node.js API endpoints using Jest or Mocha.

Verify request handling, data processing, and response formatting.

Database:

Test MongoDB database interactions using Jest or Mocha.

Verify data insertion, retrieval, updating, and deletion.

### 2. Integration Testing:

Test the interaction between frontend and backend components.

Ensure proper communication between client and server.

Verify data flow and consistency between frontend and database.

### 3. End-to-End (E2E) Testing:

Use tools like Cypress or Selenium for automated end-to-end testing.

Test the entire application flow from user interaction to database operations.

Verify user authentication, task management, team collaboration, and reporting functionalities.

### 4. User Acceptance Testing (UAT):

Conduct UAT with real users to ensure the application meets their requirements.

Gather feedback on usability, functionality, and performance.

Address any issues or concerns raised during UAT.

5. Performance Testing:

Measure application performance under various load conditions.

Use tools like JMeter or k6 to simulate concurrent user requests.

Identify and optimize performance bottlenecks in frontend, backend, and database components.

6. Security Testing:

Perform security testing to identify and address potential vulnerabilities.

Test for common security threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

Ensure proper user authentication, authorization, and data encryption.

7. Accessibility Testing:

Test the application for accessibility compliance using tools like Axe or Lighthouse.

Ensure the application is accessible to users with disabilities and meets WCAG standards.

8. Cross-Browser and Cross-Device Testing:

Test the application on different web browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).

Ensure consistent functionality and user experience across different platforms.

9. Regression Testing:

Conduct regression testing after each code change or new feature implementation.

Ensure existing functionality remains intact and unaffected by code updates.

Automate regression tests where possible to expedite the testing process.

10. Documentation:

Document test cases, test results, and any issues found during testing.

Provide detailed instructions for running tests and reproducing reported issues.

Update documentation as needed based on changes and improvements to the application.

## 12. Expected Outcome

The Task Tracker tool will provide users with a robust platform for efficient task management, team collaboration, and project tracking. Users will be able to create, assign, and track tasks effectively, ensuring timely completion of projects. The tool's intuitive user interface and comprehensive features will improve productivity, streamline project workflows, and enhance team communication. By promoting efficient task management and collaboration, the Task Tracker tool will benefit society by increasing workplace productivity, fostering better teamwork, and ensuring the timely delivery of projects and tasks.

## 13. Resources and Limitations

### Resources:

#### 1. Hardware Resources:

Development Machines: High-performance computers or laptops for development and testing purposes.

Server Infrastructure: Cloud-based servers for hosting the application and database.

#### 2. Software Resources:

Development Tools: Integrated Development Environments (IDEs) such as Visual Studio Code, Atom, or IntelliJ IDEA.

Frontend Technologies: React.js, HTML5, CSS3, JavaScript.

Backend Technologies: Node.js, Express.js, MongoDB.

Testing Tools: Jest, React Testing Library, Postman, Cypress.

Version Control: Git, GitHub/GitLab.

#### 3. Data Resources:

Sample Data: Dummy data for testing and development purposes.

Industry Data: Real-world task and project management data for testing and validation.

### Limitations:

#### 1. Scalability:

The project may face scalability challenges as the user base and data volume increase.

Scaling the application to accommodate a large number of concurrent users and data transactions may require additional resources and infrastructure.

#### 2. Performance:

Performance may be impacted under heavy load conditions.

Response times and application performance may degrade as the user base grows.

3. Security:

Ensuring data security and preventing unauthorized access will be a challenge.

Implementing robust security measures to protect user data and prevent security breaches is essential.

4. Integration:

Integrating the task tracker tool with existing systems or third-party applications may be complex.

Compatibility issues may arise when integrating with different operating systems, browsers, or devices.

5. User Adoption:

User adoption and acceptance may vary, and some users may resist transitioning to the new task tracker tool.

Proper training and support will be essential to encourage user adoption and ensure successful implementation.

6. Regulatory Compliance:

Ensuring compliance with data protection regulations such as GDPR or CCPA may pose challenges.

The project must adhere to relevant regulatory requirements to protect user privacy and data security.

## 14. Conclusion

In conclusion, the Task Tracker tool provides an efficient solution for task management, team collaboration, and project tracking. By leveraging modern technologies and intuitive user interfaces, the tool improves productivity, streamlines project workflows, and enhances team communication. With its comprehensive features and user-friendly design, the Task Tracker tool is poised to make a significant impact on workplace efficiency and project delivery.

## 15. References

[1] Atlassian, "Jira Software," in Atlassian, [Online]. Available: <https://www.atlassian.com/software/jira>.

[2] B. Trello, "Trello," in Trello, [Online]. Available: <https://trello.com/>.

[3] S. Asana, "Asana," in Asana, [Online]. Available: <https://asana.com/>.

[4] Doist, "Todoist," in Todoist, [Online]. Available: <https://todoist.com/>.