

12-B Status from UGC

**DEPARTMENT OF COMPUTER ENGINEERING AND
APPLICATIONS**

PROJECT

TaskPulse : A Task Visualization and Manager Tool

Team Members

Sneha Agarwal (2215800027)

Sahil Gupta (2215800021)

Vaishnavy Manglik (2215800031)

INDEX

S.NO.	TOPIC	PAGE NO.
1	Introduction	1
2	Survey of Existing Softwares	2
3	Need of Project	3
4	Objectives	4
5	Hardware/Software Requirement	5
6	Workflow of the Project	7
7	Methodology of the Project	8
8	Project Planning	9
9	References	11

Introduction

1.1 Overview

Task management is an integral aspect of personal and professional productivity. Whether managing personal goals, project timelines, or team workflows, efficient task tracking ensures better time management, prioritization, and execution.

With the rise of remote work and increasing workloads, individuals and teams require **intelligent, user-friendly, and interactive task management tools**. However, many existing applications offer **static, text-based lists** without AI-driven prioritization or **graphical representation** of progress.

Our proposed **TaskPulse: Task Manager & Visualizer** bridges these gaps by providing:

1. **AI-based prioritization** of tasks based on urgency and workload.
2. Graphical progress tracking via visual dashboards.
3. **Smart categorization** to automatically sort tasks.
4. Cross-platform accessibility for seamless task management.

1.2 Key Features and Benefits

Feature	Description	Benefit
AI-driven Task Prioritization	Tasks are ranked based on importance, deadlines, and workload.	Saves time and improves efficiency.
Visual Dashboards	Tasks are displayed using graphs and charts .	Enhances task tracking and decision-making.
Automated Categorization	Tasks are sorted into categories (Work, Personal, Urgent).	Reduces manual effort.
Notifications & Alerts	Users receive timely reminders for upcoming deadlines.	Prevents missed deadlines.
Multi-platform Support	Web and mobile compatibility.	Ensures flexibility.

Table Analysis:

This table summarizes how our **TaskPulse: Task Manager & Visualizer** differs from traditional task lists. While existing tools provide basic task tracking, they lack **AI-driven insights, categorization, and interactive analytics**, which our system offers.

Survey Of Existing Softwares

2.1 Comparison of Existing Solutions

Below is a **detailed feature comparison** between leading task management tools:

Feature	Trello	Asana	Microsoft To-Do	Google Tasks	Our System
Task Categorization	✓	✓	✓	✗	✓
AI-based Prioritization	✗	✗	✗	✗	✓
Visual Task Tracking	✗	✓	✗	✗	✓
Deadline Notifications	✓	✓	✓	✓	✓
Automated Task Sorting	✗	✗	✗	✗	✓

Table Analysis:

- **AI-based Prioritization** is **not present** in any major task manager today, making our system unique.
- **Automated task sorting** is an innovation that reduces manual categorization.
- **Interactive Visual Tracking** allows users to analyze their productivity, unlike static lists found in existing tools.

2.2 Patent & Copyright Research

Most patents related to task management cover basic features like task scheduling, reminders, and collaboration. However, none of them leverage AI-driven decision-making for task prioritization. Our project introduces machine learning-based prioritization and automated categorization, making it a novel contribution.

Need of Project

3.1 Challenges in Existing Systems

Challenges	Limitations of Existing Systems	Solution in Our Project
Lack of Visualization	Traditional task managers rely on static lists without visual elements.	Our system includes graphs, pie charts, and timelines.
Manual Task Sorting	Users must manually organize tasks.	AI automatically sorts and prioritizes tasks.
No Predictive Insights	Existing tools do not analyze task urgency dynamically.	Our system uses ML models to suggest priority tasks.

Table Analysis:

This table highlights major pain points users face with existing systems. Our AI-driven features provide automated categorization, prioritization, and visual analytics, solving these problems.

3.2 Project Impact:

- **Higher Productivity:** Reduces manual work.
- **Smarter Time Management:** AI-driven task prioritization improves efficiency.
- **Better Task Tracking:** Real-time dashboards enhance visibility.

Objectives

1. AI-Driven Task Prioritization and Automation

- Optimize task scheduling and prioritization using AI-based recommendations.
- Automate repetitive tasks to reduce manual workload and improve efficiency.
- Provide smart reminders and notifications to ensure timely task completion.

2. Seamless Agile Workflow Integration

- Enable Agile project management with sprint planning, backlog management, and Kanban boards.
- Facilitate iterative development by breaking down projects into manageable tasks.
- Provide real-time progress tracking for teams working in Agile environments.

3. Enhanced Collaboration and Communication

- Implement real-time collaboration tools such as in-app messaging, task comments, and file sharing.
- Ensure transparency by allowing all team members to track task updates and progress.
- Enable role-based access control to assign permissions and responsibilities efficiently.

4. User-Friendly and Data-Driven Insights

- Develop an intuitive, easy-to-use interface to enhance user experience and accessibility.
- Provide analytical dashboards and reports to track team performance and productivity.
- Ensure mobile and web accessibility for on-the-go task management.

Hardware/Software Requirement

5.1 Hardware Requirements

Component	Minimum Requirement	Recommended
Processor	Intel i5	Intel i7 or higher
RAM	8GB	16GB
Storage	50GB	100GB SSD
Display	1080p resolution	4K resolution

5.2 Software Requirements

Category	Technology
Frontend	React.js / Angular.js
Backend	Node.js / Django
Database	MongoDB / PostgreSQL
AI Integration	Python (Scikit-learn, TensorFlow)
Hosting	AWS / Firebase

1. Backend Development

- **Programming Language:** Python (Django/Flask) or Node.js (Express.js)
- **Database:** PostgreSQL or MongoDB for efficient data storage
- **Authentication & Security:** JWT (JSON Web Token) for secure user authentication
- **API Development:** RESTful APIs or GraphQL for communication between frontend and backend

2. Frontend Development

- **Programming Language:** JavaScript/TypeScript with React.js
- **State Management:** Redux or React Context API for efficient state handling
- **UI Framework:** Tailwind CSS, Material-UI, or Bootstrap for a responsive design
- **Progressive Web App (PWA):** Ensuring accessibility on both web and mobile

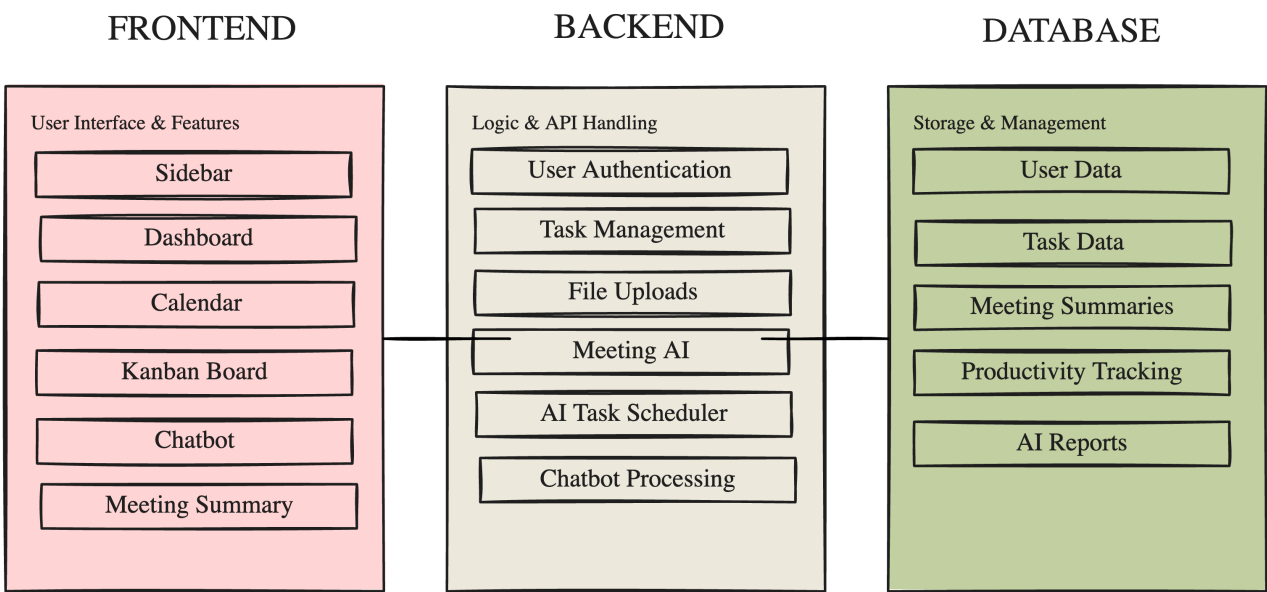
3. AI & Automation Integration

- **Machine Learning Frameworks:** TensorFlow or Scikit-learn for AI-driven task prioritization
- **Natural Language Processing (NLP):** GPT-based models for smart task recommendations
- **Task Automation Tools:** Selenium or Python scripts for automating recurring tasks

4. DevOps & Deployment

- **Version Control:** GitHub/GitLab for code management
- **Cloud Hosting:** AWS, Google Cloud, or Azure for scalable deployment
- **CI/CD Pipeline:** Jenkins, GitHub Actions, or Docker for continuous integration and deployment

WorkFlow of the Project



Methodology of the Project

TaskPulse is developed using the **Agile methodology**, ensuring an iterative, flexible, and efficient approach to project management and software development. The development process is divided into multiple **sprints**, each focusing on incremental improvements and refinements based on continuous feedback. Below is a detailed breakdown of the methodology followed in TaskPulse:

Phase 1: Requirement Analysis & Planning

- Identify user needs and define core functionalities of TaskPulse.
- Conduct stakeholder discussions to gather essential requirements.
- Establish sprint goals and define key milestones in the development process.

Phase 2: System Design & Architecture

- Design the overall system architecture, including frontend, backend, database, and AI-based automation.
- Define API endpoints for seamless communication between frontend and backend.
- Ensure scalability by selecting cloud-based deployment solutions.

Phase 3: Development (Iterative Sprints)

Each sprint lasts **2-4 weeks**, focusing on incremental improvements. The development includes:

- **Sprint 1:** Basic UI development and user authentication module implementation.
- **Sprint 2:** Core task management features such as task creation, assignment, and tracking.
- **Sprint 3:** AI-powered task prioritization and automation integration.
- **Sprint 4:** Performance optimization, UI enhancements, and bug fixes.

Phase 4: Testing & Quality Assurance

- Unit testing, integration testing, and system testing to identify and fix bugs.
- User acceptance testing (UAT) to ensure the platform meets user expectations.
- Security testing to protect user data and prevent vulnerabilities.

Phase 5: Deployment & Maintenance

- Deploy TaskPulse on cloud-based infrastructure for seamless access.
- Continuous monitoring using DevOps practices to ensure high availability.
- Regular updates based on user feedback and evolving requirements.

Project Planning

1. Project Timeline & Phases

The project is divided into six key phases, with each phase spanning a specific timeline to ensure structured development and timely completion.

Phase 1: Requirement Gathering & Analysis (Week 1-2)

- Conduct stakeholder interviews to identify core features.
- Define the problem statement and establish clear objectives.
- Research existing solutions to understand limitations and improvements.
- Prepare a Software Requirement Specification (SRS) document.

Phase 2: System Design & Prototyping (Week 3-4)

- Design the system architecture, including frontend, backend, and database.
- Develop wireframes and UI/UX mockups for TaskPulse.
- Finalize technology stack selection.
- Create a prototype demonstrating basic workflow.

Phase 3: Development & Sprint Execution (Week 5-12)

The development process is divided into multiple sprints, ensuring iterative progress:

- **Sprint 1 (Week 5-6):** User authentication, role management, and dashboard setup.
- **Sprint 2 (Week 7-8):** Task creation, assignment, and notification system.
- **Sprint 3 (Week 9-10):** AI-powered task prioritization and automation integration.
- **Sprint 4 (Week 11-12):** Advanced analytics, reporting, and UI refinements.

Phase 4: Testing & Debugging (Week 13-14)

- Perform **unit testing** for individual components.
- Conduct **integration testing** to ensure seamless communication between modules.
- Execute user acceptance testing (UAT) to verify system usability.
- Identify and resolve critical bugs before deployment.

Phase 5: Deployment & Go-Live (Week 15-16)

- Deploy TaskPulse on cloud infrastructure for accessibility and scalability.
- Perform final security and performance testing before release.
- Train users and stakeholders on the system's functionality.

2. Resource Allocation

To ensure efficient project execution, resources are assigned as follows:

- Frontend Development: 2 Developers (React.js/Angular)
- Backend Development: 2 Developers (Node.js/Python)
- Database Management: 1 Engineer (MySQL/PostgreSQL)
- AI/Automation Integration: 1 Data Scientist
- Testing & Quality Assurance: 1 QA Engineer
- Project Management: 1 Scrum Master

3. Risk Management & Contingency Plan

To mitigate potential risks, the following strategies are implemented:

- **Technical Risks:** Regular code reviews and automated testing.
- **Timeline Delays:** Agile sprints ensure adaptability to delays.
- **Security Risks:** Implementing industry-standard encryption and authentication mechanisms.
- **Scalability Issues:** Cloud deployment ensures smooth performance even with high traffic.

4. Gantt Chart Representation:

Phase	Task	Duration
Phase 1	Requirement Gathering & Research	Week 1-2
Phase 2	System Design & Prototyping	Week 3-4
Phase 3	Development	Week 5-12
Phase 4	Testing	Week 13-14
Phase 5	Deployment	Week 15-16

References

1. A. Blythe, D. McDonald, D. M. Russell, and M. J. Ackerman, "An Intelligent Personal Assistant for Task and Time Management," *AI Magazine*, vol. 23, no. 4, pp. 13–24, 2002.
2. N. Good and P. S. F. I. Group, "What a to-do: studies of task management towards the design of a personal task list manager," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2003, pp. 735–742.
3. J. Stasko, C. Görg, and Z. Liu, "Jigsaw: Supporting Investigative Analysis through Interactive Visualization," *Information Visualization*, vol. 7, no. 2, pp. 118–132, 2008.
4. S. O. Jolaoso, "TaskAmbient: A Study in Personal Task Management Visualization," M.S. thesis, Virginia Tech, Blacksburg, VA, USA, 2015.
5. T. Jadad-Garcia and A. R. Jadad, "The Foundations of Computational Management: A Systematic Approach to Task Automation for the Integration of Artificial Intelligence into Existing Workflows," *arXiv preprint arXiv:2402.05142*, 2024.