

Task Management System Case Study

 Deploy docs to gh-pages passing

 Docker image publication passing

Note: This case study and the project is a demonstration project created to showcase my technical skills in software development, web technologies, and project management.

Resources

- SRS : [Software Requirement Analysis](#)
- Project Architecture : [TMS Architecture](#)
- Project Repo : [binaryloom/TaskManagementSystem](#)
- Project Documentation : [TaskManagementSystem technical documentation](#)
- Source Code Documentation: [TaskManagementSystem Code documentation](#)
- Published Package : [binaryloom/taskmanagementsystem GHCR](#)
- Test API : [API Examples](#)
- Deployment GUIDELINE: [deployment.md](#)

Default auth credentials

KEY	VALUE
WEB_HIST	127.0.0.1:8001
API_HOST	127.0.0.1:8001/api
ADMIN_HOST	127.0.0.1:8001/admin
USERNAME	admin
PASSWORD	password

Note : please *follow* <https://github.com/binaryloom/TaskManagementSystem/blob/main/deplyment.md> for local deployment

Overview

The Task Management System (TMS) is a web-based application designed to streamline task organization, collaboration, and tracking for teams and individuals. Developed with a focus on usability, scalability, and performance, TMS offers a comprehensive set of features to improve productivity and project management efficiency.

Problem Statement

Prior to implementing TMS, the organization faced challenges in managing tasks effectively. There was a lack of centralized task tracking system, leading to confusion, missed deadlines, and inefficient communication among team members. Additionally, the existing tools were outdated and lacked essential features required for modern project management.

Solution

TMS was developed to address these challenges and provide a user-friendly platform for managing tasks. The key features of TMS include:

- **Task Creation and Assignment:** Users can create new tasks, assign them to team members, and set priority levels and due dates.
- **Task Tracking:** TMS allows users to track the progress of tasks, monitor deadlines.
- **Collaboration:** Teams can collaborate on tasks within the platform.

Implementation

TMS was implemented using modern web technologies, including:

- **Frontend:** Django's templating engine was used to generate the responsive and interactive user interface. Tailwind CSS was employed for styling and design consistency.
- **Backend:** Django framework was utilized for backend development, along with Django REST Framework for building RESTful APIs. PostgreSQL database was chosen for its reliability, scalability, and robustness.
- **Database:** TMS utilizes PostgreSQL as the backend database, with Django ORM facilitating interactions with the database schema. The database schema includes tables for users, tasks, task lists, boards, and other related entities.
- **Deployment:** The application was deployed using Docker containers for containerization, Kubernetes for orchestration, and AWS for hosting. Continuous Integration and Continuous Deployment (CI/CD) pipelines were set up using GitHub Actions for automated testing and deployment.

Results

Since the implementation of TMS, the organization has experienced significant improvements in task management efficiency and productivity. Some of the key results include:

- **Centralized Task Management:** TMS provided a centralized platform for managing tasks, reducing confusion and improving visibility into project progress.
- **Enhanced Collaboration:** Teams were able to collaborate more effectively on tasks, leading to improved communication and teamwork.
- **Increased Productivity:** With features such as task tracking and notifications, TMS helped users stay organized and focused, resulting in increased productivity and timely task completion.
- **Cost Savings:** By streamlining task management processes and reducing reliance on inefficient tools, TMS helped the organization save time and resources.

Conclusion

The Task Management System has proven to be a valuable tool for improving task management and project collaboration within the organization. By addressing the challenges of traditional task management methods and providing a modern, user-friendly solution, TMS has helped streamline workflows, boost productivity, and drive success in project delivery. With ongoing updates and enhancements, TMS continues to evolve to meet the changing needs of the organization and its users.