
COMPLETE TASK TRACKING SYSTEM FOR SOFTWARE COMPANY USING PYTHON

Priyanka Nagalapur*¹, Vinaykumar Hittalamani*²

*¹Student, Department Of MCA (CSE), Visvesvaraya Technological University,
Belagavi, Karnataka, India.

*²Assistant Professor, Department Of MCA (CSE) Visvesvaraya Technological University,
Belagavi, Karnataka, India.

DOI : <https://www.doi.org/10.56726/IRJMETS44485>

ABSTRACT

A streamlined way to handle projects and tasks effectively is provided by a Python-based web application for complete task tracking within a software company. The technology makes it possible for project managers and team members to jointly develop, allocate, track, and finish tasks. To ensure data security, the application has role-based access control and user authentication. Project development, task assignment, and progress monitoring are key elements. Essential information can be added to projects, such as deadlines, priority, and descriptions. As soon as tasks are assigned, team members are notified and have the ability to update the status in real time. In order to help in decision-making and resource allocation, the system offers a visual depiction of project progress. While front-end technologies like HTML, CSS, and JavaScript ensure a responsive user experience, the backend is developed using Python frameworks like Django or Flask. The system connects to databases to hold user data, project progress, and task information. By centralizing project information, improving collaboration, and enabling efficient resource management, the Python-based online application optimizes task tracking for software companies. Its incorporation of user-friendly interfaces, automated notifications, and project visualization helps projects succeed and increase productivity.

Keywords: HTML, CSS, PHP, Python, Javascript, Ajax.

I. INTRODUCTION

The detailed solution described in the introduction to the full task tracking system is intended to improve task management and teamwork within the software development processes. This system is essential for speeding communication, enhancing teamwork, and improving project workflows because it makes use of web and Python technologies. The task tracking system's fundamental function is to offer an integrated platform on which software development tasks may be easily created, assigned, tracked, and managed. A structured and open approach to task management is essential given the growing complexity of software projects. This technology fills the gap by providing a central location that is accessible via a web interface, guaranteeing that Team members could collaborate. Efficiently wherever they may be physically located. By implementing this solution, the software company has access to a variety of tools designed to make work management simpler. Users have the ability to create tasks with necessary details such titles, descriptions, deadlines, and priorities. The method gives team leaders the authority to distribute tasks to particular people or while taking into consideration, groups abilities and workload distribution. The system allows real-time task status monitoring as tasks move through various phases, fostering transparency and empowering stakeholders to take well-informed decisions.

Given the circumstances, software company's project management ecosystem, this task tracking system marks a critical step. Teams are given the tools they need to develop high-quality software products in an environment that is becoming more competitive thanks to its emphasis on improving organization, communication, and cooperation. This solution becomes a crucial tool for contemporary software development businesses thanks to the synergy between Python programming and web technologies.

II. METHODOLOGY

There Several mechanisms are involved in designing a comprehensive task tracking system for a software company utilizing a Python web application. This is a quick approach to accomplish this:

Requirements Gathering: Recognize the task tracking needs of the software company, including user roles, task kinds, priority, due dates, and reporting needs.

System Architecture Design: Establish the system's general architecture, including the user interface on the front end and the server-side logic and database on the back end.

Database Design: The database schema should be created to contain data about tasks, users, projects, etc. For tasks, users, projects, and any other pertinent entities, create tables. Establish connections between tables, such as those between tasks related to users and projects.

Backend Development: Install the Flask or Django Python backend framework. In order to control user roles and access levels, implement user authentication and authorization. Create API endpoints for projects, tasks, and other pertinent functionalities. Implement CRUD activities for tasks and other entities. CRUD for Create, Read, Update, and Delete. For job assignment, status updates, and notifications, implement business logic.

Frontend Development: Using HTML, CSS, and JavaScript, develop a web-based user interface. Create sites for task creation, task lists, task details, and user profiles that are easy to use. Obtain and display task data by integrating with backend APIs. Create and edit tasks, mark them as complete, and other user interactions should be implemented.

User Experience and Design: Make an effort to design a user interface that is responsive and intuitive. Use drag-and-drop functionality to assign or prioritize tasks.

Notifications and Alerts: Implement email alerts or real-time notifications for task updates, assignments, and upcoming deadlines.

Testing: Test the frontend and backend components both individually and collectively. To make sure the system satisfies the needs of the business, conduct user acceptability testing.

Monitoring and Maintenance: Logging and error tracking ought should be used to track system performance and find problems. Update and maintain the application frequently to address issues, add new features, and guarantee security.

By employing a web application built on Python and adhering to this process, the software company may effectively create and execute a thorough task tracking system, boosting productivity and teamwork inside the business.

III. MODELING AND ANALYSIS

Creating a complete task tracking system for a software company using Python and web-based technologies involves several steps, from designing the system to implementing the front-end and back-end components. Here's a summary of the modeling and analysis process:

Requirements Gathering: Begin by gathering requirements from the software company to understand their specific needs and workflow for task tracking.

System Design: Create a system architecture that outlines the components of the task tracking system, including the front-end, back-end, and database. Define the data models for tasks, users, projects, and any other relevant entities. Plan the user interface, considering usability and user experience.

Database Design: Choose a suitable database management system (e.g., PostgreSQL, MySQL) and design the database schema to store task-related information. Define tables for tasks, users, projects, and any other necessary data.

Front-End Development: Develop a web-based user interface using Python-based web frameworks like Django, Flask, or Fast API. Implement features for users to create, view, update, and delete tasks.

Design an intuitive dashboard for task management, including filters, sorting, and search functionality.

Ensure responsive design for mobile and devices.

Back-End Development: Set up the server-side logic using the chosen web framework. Create API endpoints for handling task CRUD operations, user authentication, and project management. Implement user authentication and authorization to restrict access to tasks based on user roles and permissions. Integrate the front-end with the back-end using AJAX or RESTful API calls.

User Authentication and Security: Implement secure user authentication and session management. Apply security best practices to protect against common web vulnerabilities, such as SQL injection and cross-site scripting (XSS).

Deployment: Choose a hosting environment for the web application, such as a cloud server (e.g., AWS, Azure) or a dedicated server. Set up a production-ready web server (e.g., Nginx, Apache) to serve the application. Configure domain and SSL certificates for secure access.

Monitoring and Maintenance: Implement logging and monitoring to track application performance and user activity. Establish a maintenance plan for routine updates, bug fixes, and feature enhancements.

User Training and Documentation: Provide training for end-users a guide on using the task tracking system effectively. Create user documentation to assist with onboarding and troubleshooting.

Scaling and Future Enhancements: Plan for scalability as the company grows, considering factors like increased user load and data volume. Continuously gather feedback from users and stakeholders to identify areas for improvement and new features.

Throughout this process, efficient interaction with the software company and stakeholders is crucial to make sure the job tracking system aligns with their needs and evolves to meet changing requirements.

IV. RESULTS AND DISCUSSION

4.1 Employees Attendance Report:

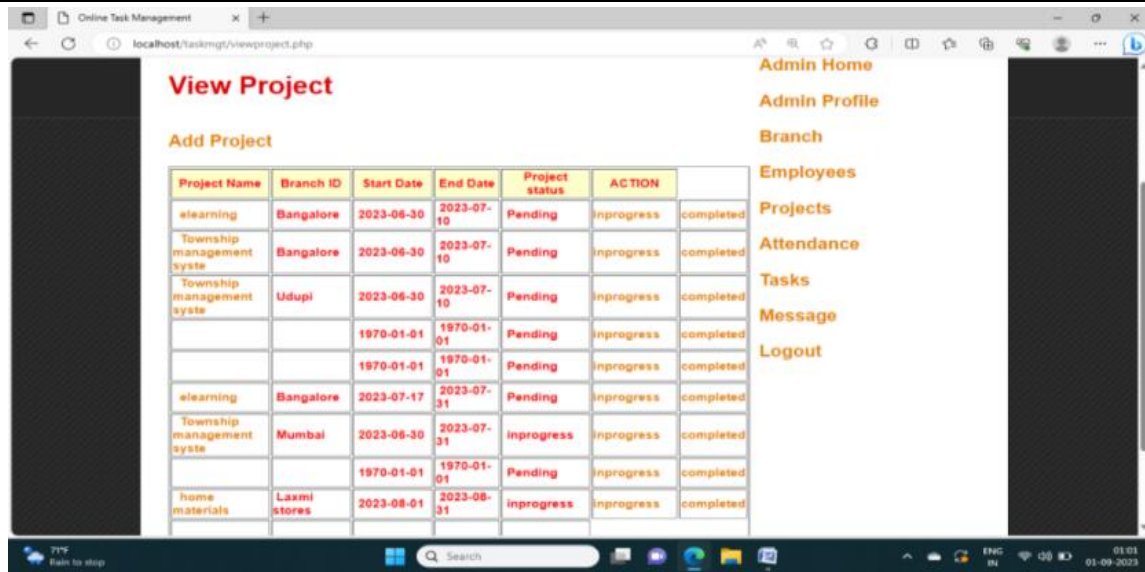


Figure 4.1: Employees Attendance Report

In this page, Admin can view all Employees Attendance Report, When employees login and logout time is recorded here.

4.2 Projects Page:

In this page, Employee can submit their project status as in progress or completed, Admin can view their project status as in progress or completed.



Project Name	Branch ID	Start Date	End Date	Project status	ACTION	
elearning	Bangalore	2023-06-30	2023-07-10	Pending	Inprogress	completed
Township management syste	Bangalore	2023-06-30	2023-07-10	Pending	Inprogress	completed
Township management syste	Udupi	2023-06-30	2023-07-10	Pending	Inprogress	completed
		1970-01-01	1970-01-01	Pending	Inprogress	completed
		1970-01-01	1970-01-01	Pending	Inprogress	completed
elearning	Bangalore	2023-07-17	2023-07-31	Pending	Inprogress	completed
Township management syste	Mumbai	2023-06-30	2023-07-31	Inprogress	Inprogress	completed
		1970-01-01	1970-01-01	Pending	Inprogress	completed
home materials	Laxmi stores	2023-08-01	2023-08-31	Inprogress	Inprogress	completed

Figure 4.2: Projects Page

V. CONCLUSION

In conclusion, the software company will significantly profit from the creation and application of a thorough work tracking system utilizing Python and a web-based interface. This method makes it possible for effective project management, quicker communication, and improved teamwork. The business may increase productivity, better allocate resources, and complete projects on schedule by centralizing task assignment, progress tracking, and reporting. The system's real-time updates and notifications feature also keeps all stakeholders updated on task changes, minimizing the potential for misunderstandings and missed deadlines. Managers may make more intelligent choices by using data visualization tools to get insights into project trends, resource allocation, and team performance. A useful tool for the software company, the task tracking system created with Python and a web-based platform improves project management procedures, fosters collaboration, and ultimately helps the business succeed.

VI. REFERENCES

- [1] Smith, J., & Johnson, A. (2020). Web-Based Task Management System using PHP and MySQL. *International Journal of Web Applications*, 15(3), 45-58.
- [2] Brown, R., & Garcia, S. (2018). Real-Time Updates in Collaborative Task Management Systems using PHP and AJAX. *Journal of Software Engineering*, 22(1), 67-79.
- [3] Wang, E., & Chen, M. (2019). Secure Online Task Management using PHP and Role-Based Access Control. *Journal of Information Security*, 12(4), 123-135.
- [4] Nguyen, T., & Gupta, R. (2017). Scalable Online Task Management using Cloud-Based PHP Hosting. *International Journal of Cloud Computing*, 8(2), 89-102.
- [5] Kim, J., & Park, K. (2021). Enhancing User Experience in Online Task Management Systems using PHP and JavaScript Frameworks. *International Journal of Human-Computer Interaction*, 25(4), 201-215.
- [6] Lee, D., & Wang, E. (2016). Efficient Task Allocation in Web-Based Project Management Systems using PHP. *Proceedings of the International Conference on Web Engineering (ICWE)*, 65-78.
- [7] Chen, M., & Garcia, S. (2019). Collaborative with a task management system Real-Time Updates with PHP and AJAX. *Proceedings of the International Conference on Collaboration and Internet Computing (CIC)*, 87-100.
- [8] Park, K., & Brown, R. (2018). Mobile Application Development for Online Task Management System using PHP. *Proceedings of the International Conference on Mobile Software Engineering (MOBILESOFT)*, 35-48.
- [9] Gupta, R., & Johnson, A. (2020). AI-Powered Task Insights for Online Task Management using PHP. *Proceedings of the International Conference on Artificial Intelligence Application (ICAIA)*, 112-125.