Project title

Tasktopia

Idea number (6)

Realized by :

Nagham Fakheraldine - **100570**
Sally Abi Harfouch - **100403**

Supervised by
Dr. Ali Ghorayeb

Team

Nagham Fakheraldine
100570
76616497
Naghamfakheraldine2002@gmail.com

Sally Abi Harfouch
100403
71690974
sallyharf@gmail.com

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr.Ali Ghorayeb for his guidance and support throughout the development of this project. We would also like to thank our classmates and family for their encouragement and support throughout this project.

Table of contents

TABLE OF FIGURES

SUMMARY

We were interested in developing a task management system to help individuals and teams organize and track tasks. The website typically includes features such as creating and assigning tasks, setting deadlines. In developing tasktopia, it is essential to prioritize user experience, as the success of the platform depends on its usability and functionality. The website is easy to navigate, with a clear and intuitive interface that allows users to quickly create and manage tasks.

Overall, tasktopia is a powerful tool for increasing productivity and managing workflow for individuals and teams alike. With careful planning, design, and development, a tasktopia can help streamline processes and improve overall efficiency.

Introduction

Tasktopia is a web-based application that helps individuals or teams to organize, manage, and track tasks from start to finish. The system enables admins to assign tasks, set deadlines, track progress, and communicate with team members. It is an essential tool for admins and team members.

In this project, we will be building a task management system using PHP-Laravel. Tasktopia will be built with modern design principles and will include features such as task creation/deletion/modification, assignment, and progress tracking, project management and user authentication,etc..

The system will allow admins to create and assign tasks to team members or individuals, set deadlines and priorities, track progress, and communicate with team members through description. It will also provide a dashboard view to see all tasks and projects at a glance, total number of tasks and various reports to track progress and productivity. Additionally, the system will have user authentication and role-based access control to ensure that users can only access the tasks assigned to them.

Overall, this task management system project will be a valuable tool for teams and individuals looking to improve their productivity and project management skills. By using modern web development technologies like Laravel, we can build a robust and scalable solution that can meet the needs of any organization.

Chapter 1. Preliminary Study

1. Introduction

During the preliminary study, a thorough analysis of the project goals, scope, and constraints is conducted. This includes examining the existing task management systems and identifying areas where improvements can be made, such as user experience.

Tasktopia enables users to assign tasks, set deadlines, track progress, and communicate with team members. It is an essential tool for admins and team members. The system will allow users to create and assign tasks to team members or individuals, set deadlines and track progress, and communicate with team members. It will also provide a dashboard view to see all tasks at a glance, total number of tasks and various reports to track progress and productivity. Additionally, the system will have user authentication and role-based access control to ensure that the team members can only access the tasks assigned to them.

2. Project objectives

1. Efficient task management: The primary objective of the task management system is to provide an efficient and intuitive way to manage tasks. This includes features such as task creation, assignment, tracking, and deadline setting.
2. Enhanced productivity: The task management system should help users increase their productivity, create recurring tasks, and set deadlines.
3. Cross-platform functionality: The task management system should be accessible across different platforms and devices, including desktop and mobile.
4. Collaboration and team management: The task management system should facilitate collaboration and team management by allowing admins to assign tasks to team members.
5. Enhanced Performance: To leverage Laravel's powerful features, such as routing, controllers, models, views, and migrations, to develop a robust and scalable application.

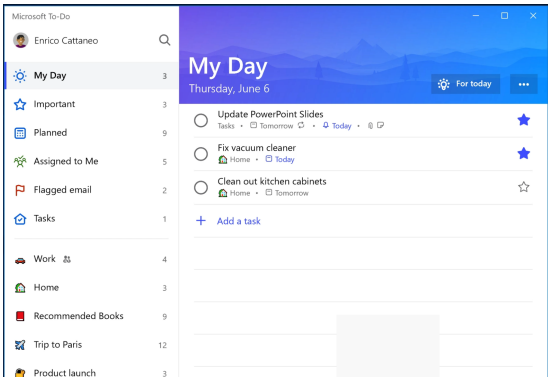
6. User Experience: To provide a smooth and user-friendly experience through an intuitive interface that enables users to easily navigate and access the various features of the application.

7. Security and data privacy: The task management system should be designed with security and data privacy in mind, ensuring that user data is protected and stored securely.

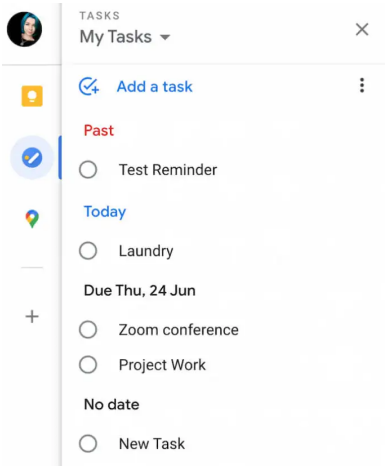
   Overall, the project aims to help individuals and teams manage their tasks efficiently, improve collaboration, increase productivity, and achieve better outcomes.

3. Similar applications

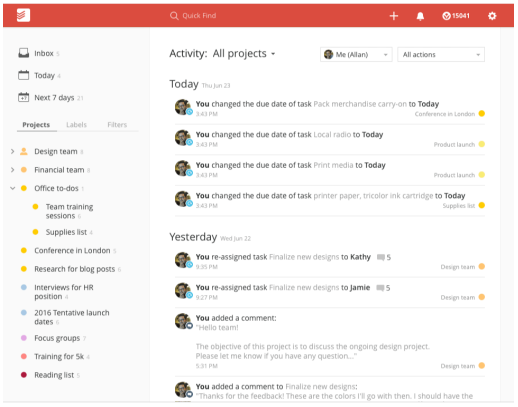
Microsoft To Do



Google Tasks



Todoist



4. Features

1. Task creation, editing, and deletion: Admins can create, edit and delete tasks as per their needs.
2. Task assignment: Tasks can be assigned to different team members.
3. Task status tracking: The status of tasks can be tracked and updated as "Started", "Pending", "In-Progress" and "Done".
4. Collaboration and team management: The task management system should facilitate collaboration and team management by allowing admins to assign tasks to team members.
5. Due date: Users can set due dates for tasks.
6. Data visualization tools: Users can view data, visualization of tasks and progress.
7. A responsive design: designed to work on different devices such as desktop computers, tablets, and smartphones. This ensures that users can take quizzes on any device they choose.
8. A user interface that is easy to navigate: This could include a simple menu system, clear and easy-to-read text, and intuitive controls.
9. User authentication: Users should be required to authenticate themselves before accessing the task management system. This can be done using techniques such as passwords.
10. Encryption: The task management system should use encryption techniques to protect sensitive information, such as user credentials or task details, when it is transmitted over the network or stored in the system.

4. Design

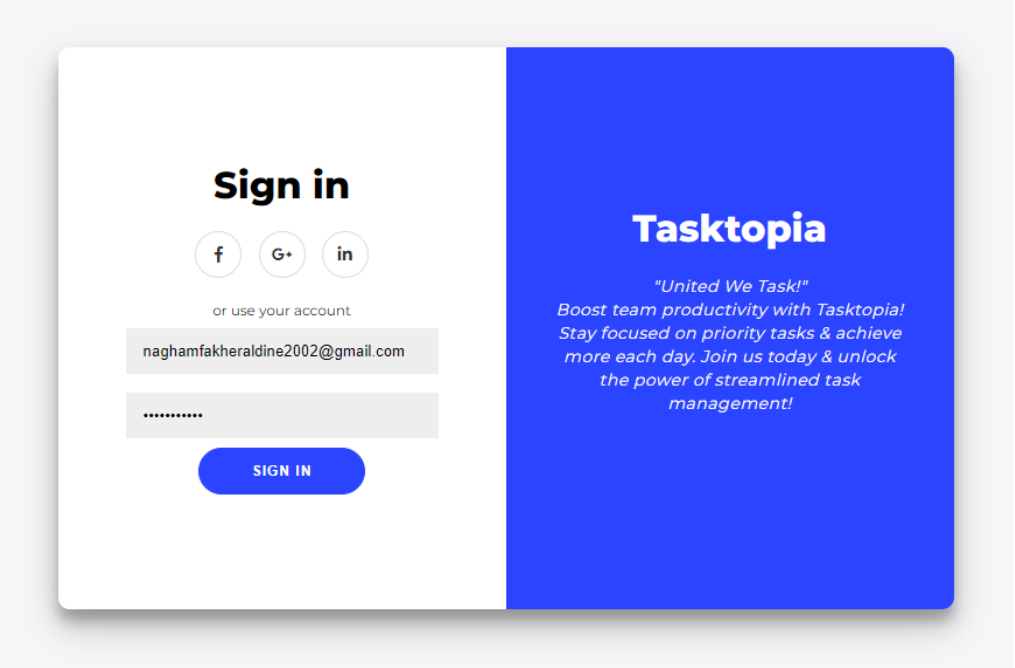


Figure.1. Login Page

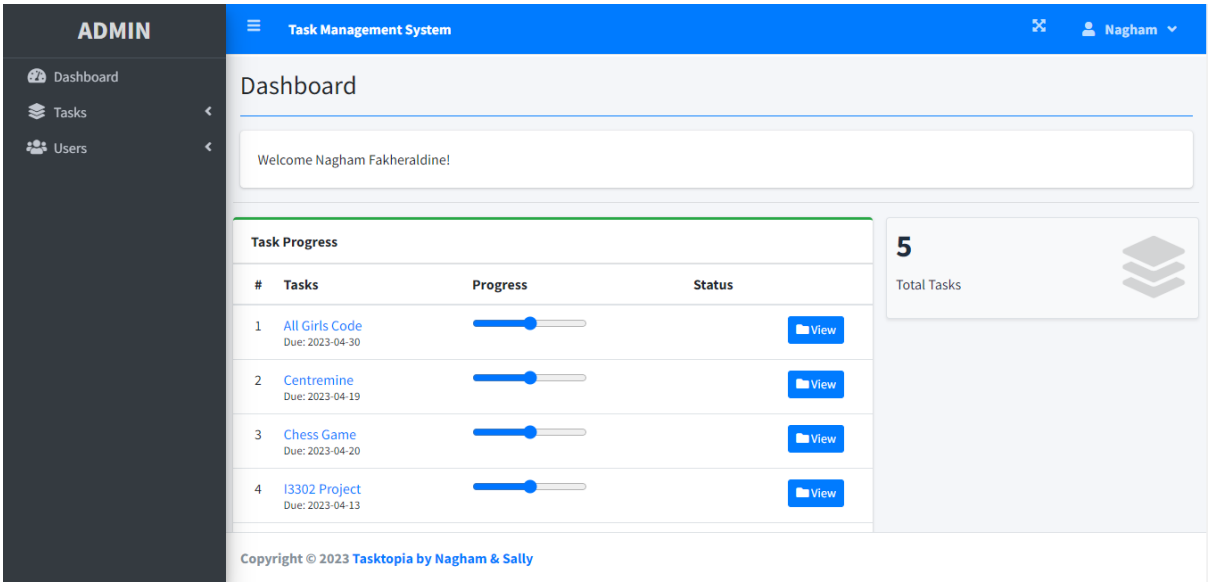


Figure.2. Admin's Home Page

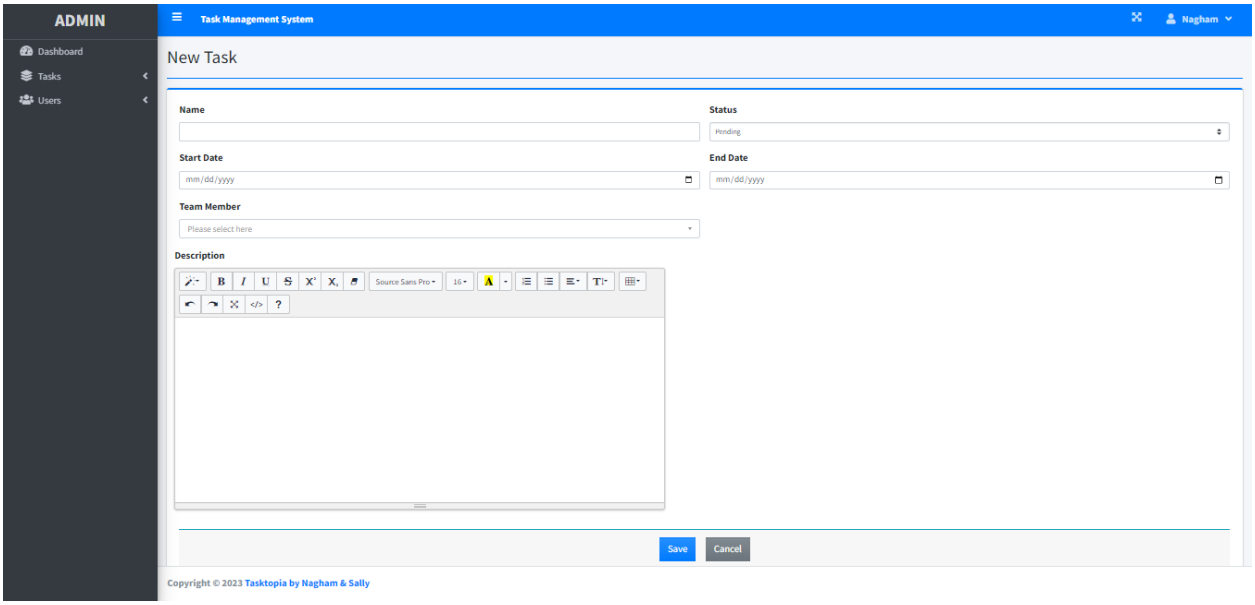


Figure.3. Admin's Task Creation Page

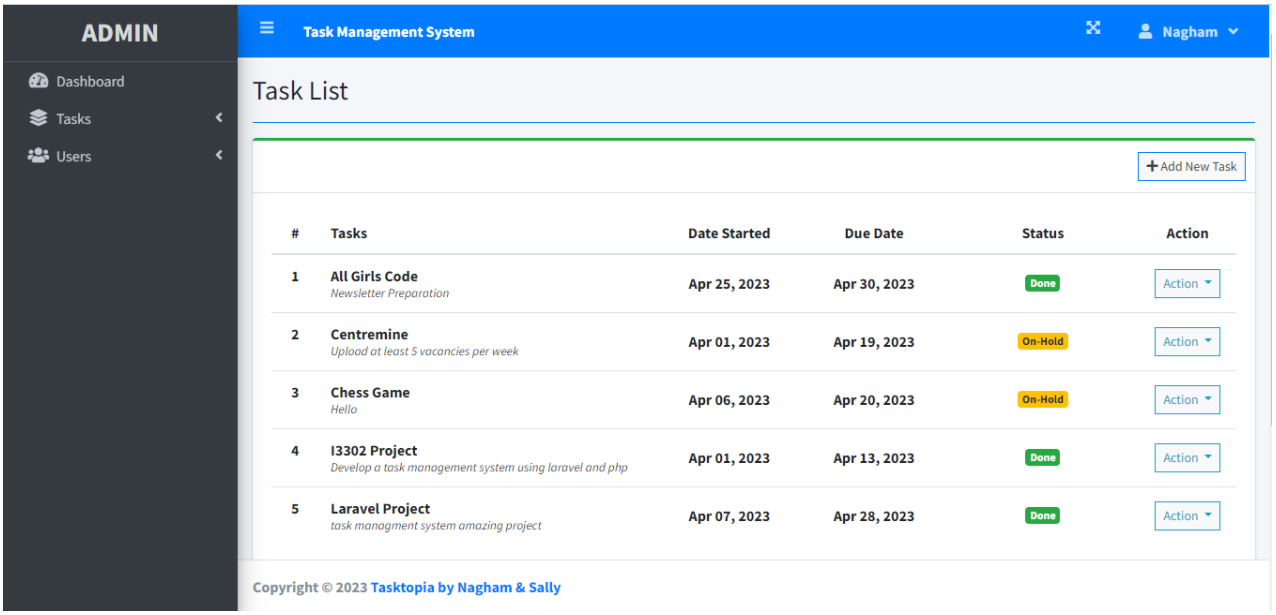


Figure.4. Admin's Task List Page

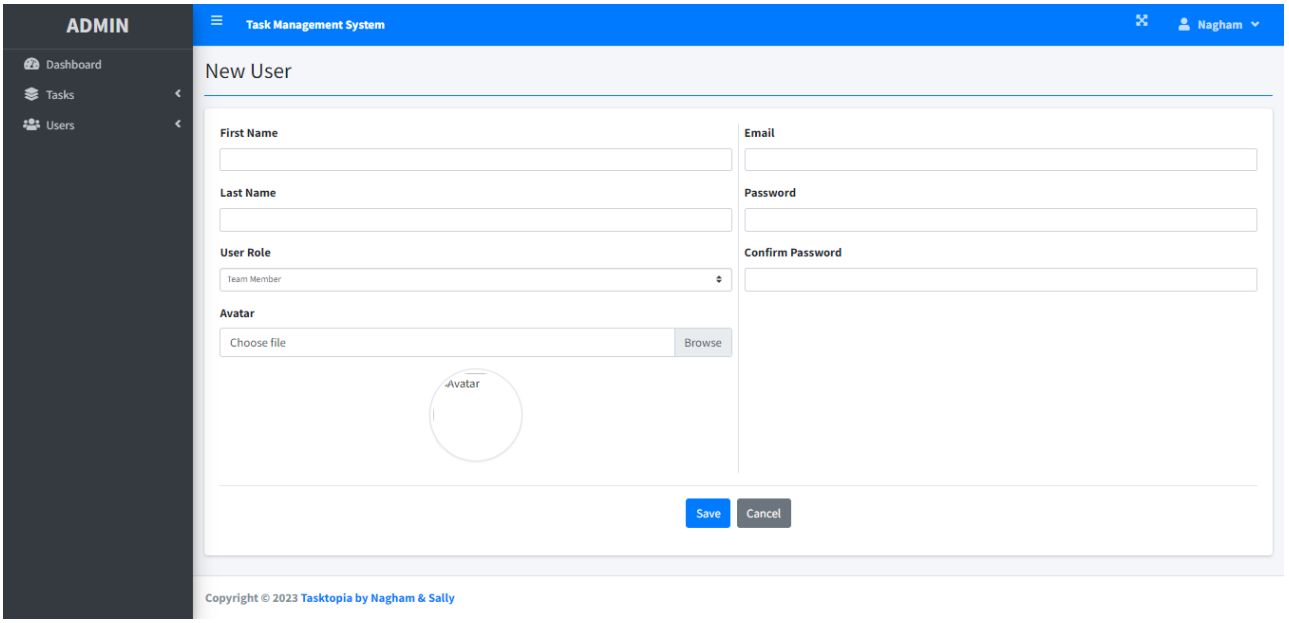


Figure.5. Admin's User Creation Page

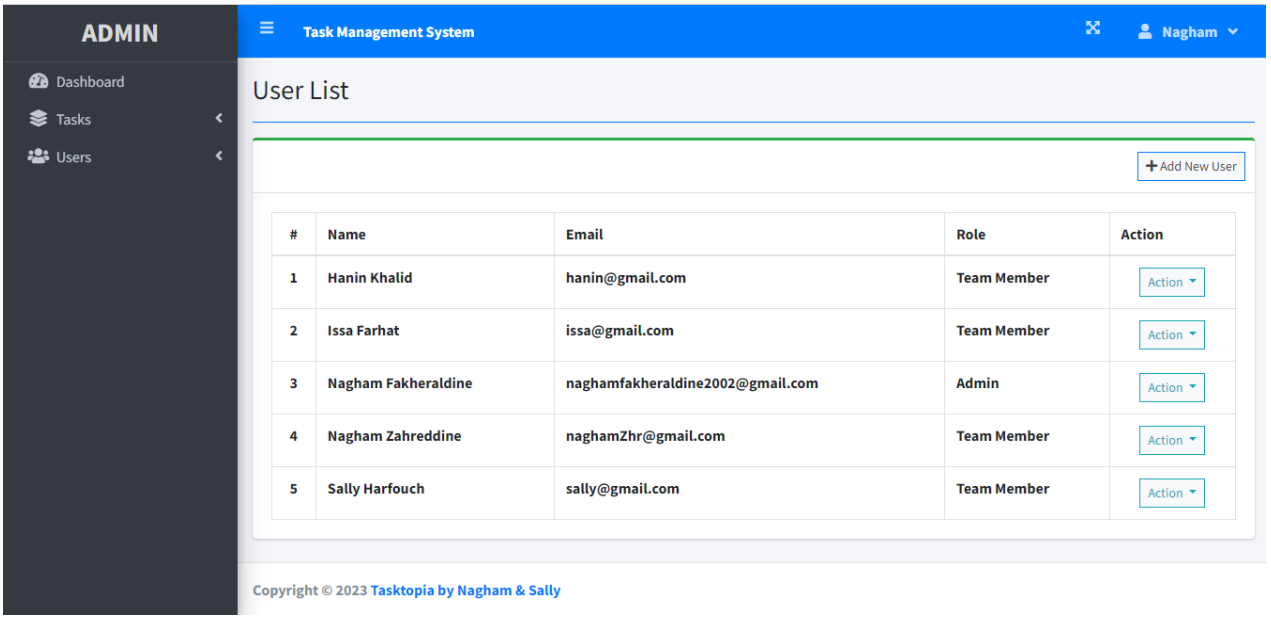


Figure.6. Admin's User List Page

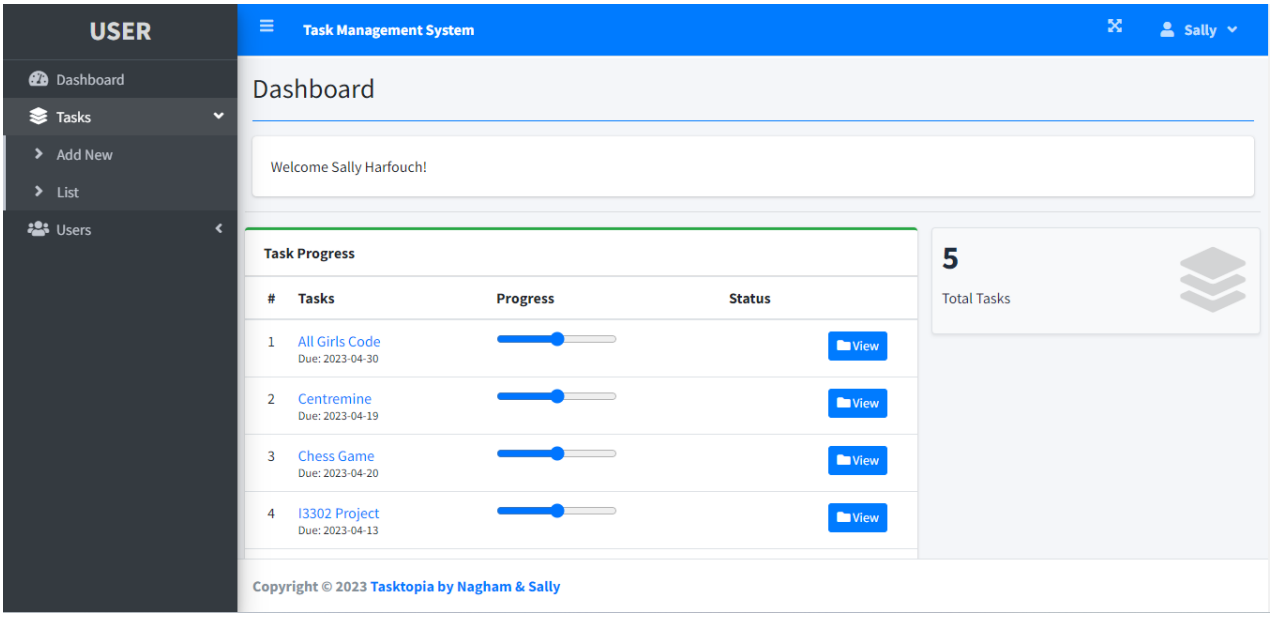


Figure.7. Team Members Home Page

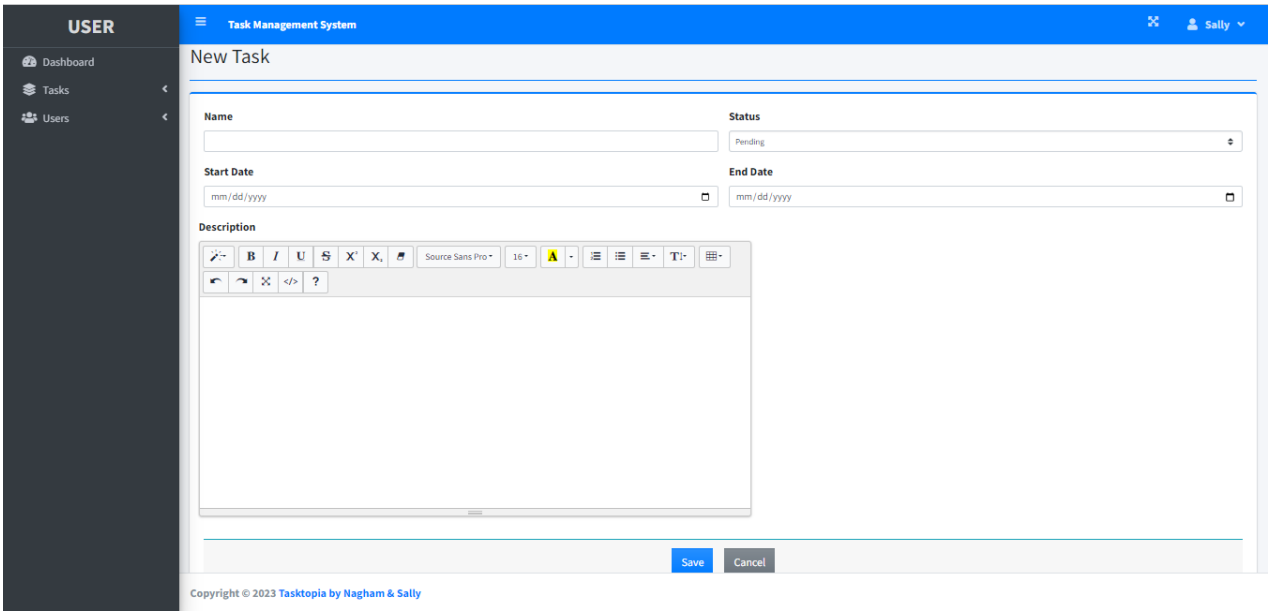


Figure.8. Team Member Task Creation Page

Chapter 2. Database

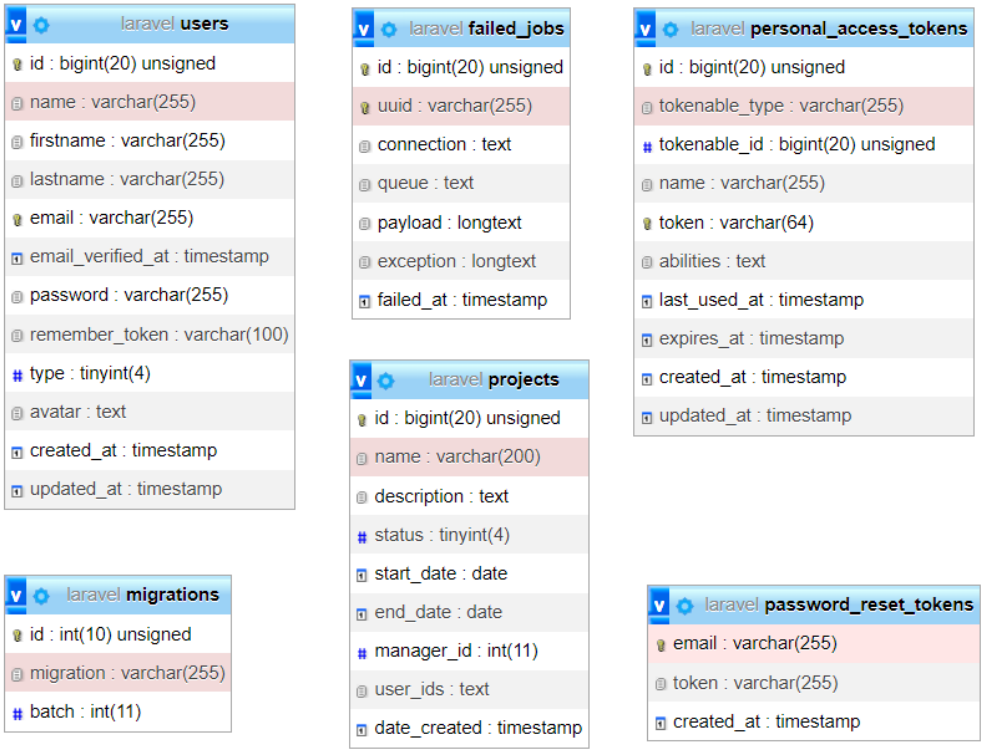
1. Introduction

The database plays a crucial role in storing and managing data related to tasks, admins and team members. In this chapter, we will discuss the design, implementation, and maintenance of the database for our task management system.

2. Implementation

To establish the database in Tasktopia, we utilized phpMyAdmin MySQL server as our chosen DBMS, with XAMPP serving as our management tool. Using Laravel's migration functionality, we generated the necessary tables, functions, procedures, and triggers within the database. Additionally, we created an installation file, **installation.php**, to automate the process of database creation and record import. Our database operations were performed using Laravel's CRUD (Create, Read, Update, Delete) operations.

3. ER diagram



4. Tables

In the previous ER diagram, we have 2 main tables: **users table** that stores users information like first name, last name, password, email, type(admin=1, team member=2), avatar, etc..

Secondly, the **projects table** that stores tasks information like id, name description, start date, end date, etc..

The projects table contains a column called user_ids which concatenates different user IDs for users who have been assigned the same task in the project.For this reason, projects table and users table doesn't have a direct link/relationship between each other.

While other tables are created by default from laravel to improve systems functionality and performance.

Chapter 3. Implementation

1. Introduction

This chapter provides an in-depth overview of the methodologies, technologies, and tools utilized to develop the task management system. It covers various technical aspects, including backend and frontend development, database design, security measures, performance optimization, and user interface design. This chapter aims to provide a comprehensive understanding of the implementation process, detailing the constraints, challenges, and solutions encountered during the development of tasktopia.

2. Technique

1. Requirements gathering: Define the requirements of the task management system, including features, functionality, and user interface.
2. Design: Design the database schema, user interface, and application architecture based on the requirements. Decide on the framework and programming language to be used.
3. Development environment setup: Install the necessary software tools and set up the development environment. This could include installing the Laravel framework, setting up the database management system, and configuring the web server.
4. Database design and implementation: Design the database schema based on the requirements and create the database tables, relationships, and constraints. Use Laravel migrations to automate the process of creating and updating the database schema.
5. User interface development: Develop the user interface for the task management system using HTML, CSS, and JavaScript. Use Laravel's Blade templating engine to create reusable templates and components.
6. Application logic development: Implement the application logic to handle user requests and interact with the database. Use Laravel's built-in features such as controllers(user, task, auth controller,etc.), models(user, task , system settings, etc.) , and routes to handle the application logic.

7. Testing: Test the application thoroughly to ensure that it meets the requirements and works as expected.

8. Maintenance: Maintain the application by fixing bugs, adding new features, and improving performance. Monitor the application and the server for security vulnerabilities.

3. Constraints

This chapter provides a detailed overview of the constraints that will be considered in the project, their implications, and the measures that will be taken to ensure that they are addressed and satisfied throughout the project's development and implementation.

1. Scalability constraint: The system must be scalable, able to accommodate an increasing number of users and tasks, without significant impact on performance.

2. Performance constraint: The system must be able to handle up to 100 concurrent users and tasks, with a maximum response time of 5 seconds and a maximum downtime of 30 minutes per week. A load testing and stress testing will be conducted on the system.

3. Compatibility constraint: The system must be compatible with the latest stable versions of major web browsers, operating systems, and devices, to ensure accessibility for all users. A compatibility test will be conducted on a sample of 5 different devices and browsers.

4. User interface constraint: The user interface must be simple and easy to use, requiring no more than 5 minutes of training for end-users. A usability test will be conducted on a sample of 5 end-users.

4. Security

1. User authentication: Implement a secure user authentication system, such as password hashing to prevent unauthorized access to user accounts.
2. Encryption: Use encryption to protect sensitive data, such as user passwords, when it is transmitted over the network or stored in the database.
3. Error handling: Implement appropriate error handling and logging mechanisms to prevent sensitive data from being leaked through error messages.
4. User education: Educate users on best security practices, such as choosing strong passwords and not sharing login credentials, to help prevent social engineering attacks.

Conclusion

Tasktopia is a powerful platform designed to assist individuals in managing their daily tasks effectively. It offers a centralized location for users to create, organize, and track their to-do lists, which ultimately helps them remain focused, increase productivity, and accomplish their objectives. This platform is not only beneficial for freelancers, small business owners, and large corporations, but also for anyone looking to streamline their work and reduce stress.

Task creation and assignment is a crucial feature of Tasktopia. It enables users to create tasks, assign them to team members, and set deadlines, ensuring that everyone on the team is aware of their responsibilities and can prioritize their work effectively. Moreover, Tasktopia provides progress tracking tools, enabling users to monitor the status of their tasks in real-time, and gain insight into their productivity and progress through visual representations of their data.

The development of a task management website demands a profound understanding of web development technologies and best practices, along with an in-depth analysis of user requirements and expectations. To be successful, a task management website must have an intuitive and user-friendly design, which is both visually appealing and functional. Furthermore, the website should be rigorously tested to ensure that it is reliable and secure.

In conclusion, Tasktopia is an essential tool that can help individuals and businesses increase their productivity and organization. With careful planning, thoughtful design, and thorough testing, Tasktopia can provide an effective solution for managing tasks in various contexts.