Task Management System – System Planning Document

Submitted by: M.F.P.Infaza

Date: 6th September 2025

1. Objective

The objective of this project is to evaluate the ability to plan, develop, document, and share a Laravel-based Task Management System using best practices, Git workflows, and clean code principles.

This project was developed as part of the **Laravel Developer Assessment (72 Hours)**. It demonstrates authentication, task CRUD operations, a dashboard with task grouping, and system documentation following Laravel best practices.

2. Introduction

The Task Management System is a Laravel-based web application designed to help users manage their daily tasks efficiently. The system supports multi-user registration, secure login, and provides task management with CRUD operations (Create, Read, Update, Delete).

Key features include:

- Add, edit, delete, and mark tasks as completed
- Task search and status filtering
- Dashboard with tasks grouped by status (Pending, In Progress, Completed)
- SweetAlert confirmation pop-ups for deletions and task completion
- Responsive UI using TailwindCSS

3. Technology Stack

Component	**Technology / Version**
Backend	Laravel 10.48.29
PHP	8.1.25 (XAMPP on Windows 10/11)
Database	MySQL 8.x / SQLite
Frontend	Blade Templates, TailwindCSS 3.x
Authentication	Laravel Breeze
JavaScript	Alpine.js
Node.js & NPM	node v22.14.0 , npm 11.5.2
Other Tools	Composer 2.8.11, SweetAlert 2
Version Control	Git v2.42.0 (Windows)

4. Database Schema

- **Users Table:**

Column	**Type**	**Key**	**Notes**
Id	BIGINT (PK)	Primary	Auto Increment (Primary Key)
name	VARCHAR(255)		User’s full name
email	VARCHAR(255)	Unique	Unique email address
password	VARCHAR(255)		Hashed password
created_at	TIMESTAMP		Managed by Laravel
updated_at	TIMESTAMP		Managed by Laravel

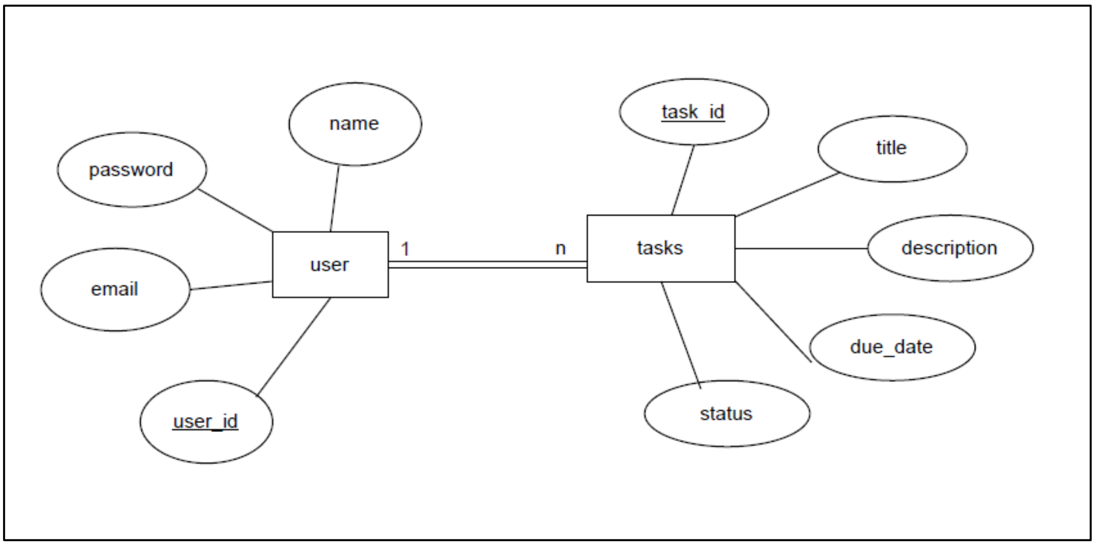
- **Tasks Table:**

Column	Type	Key	Notes
Id	BIGINT (PK)	Primary	Auto Increment (Primary Key)
user_id	BIGINT (FK → users.id)	Foreign	Cascade on delete
Title	VARCHAR(255)		Task title
description	TEXT	Nullable	Optional task description
due_date	DATE		Task due date
status	ENUM('pending','in_progress','completed')		Default: 'pending'
created_at	TIMESTAMP		Managed by Laravel
updated_at	TIMESTAMP		Managed by Laravel

4. ER Diagram

The ER diagram shows a 1-to-many relationship between Users and Tasks:

- One user can have multiple tasks
- Each task belongs to a single user



5. Application Flow Overview

The application begins with the **Home page**, which provides navigation options for users. A new user can click **Register** and submit the registration form. Upon successful registration, the system redirects them to the login page with a success message. After logging in, users are directed to the **Dashboard**, where tasks are displayed and grouped by status: **Pending, In Progress, and Completed**.

From the sidebar, users can access the **My Tasks** page to view all their tasks. Users can add a new task by entering a title, description, due date, and status. Existing tasks can be updated via the **Edit** button, or deleted with a **SweetAlert confirmation**. Tasks can also be marked as completed, again with a SweetAlert confirmation. Users can search tasks by keyword or filter them by status, with all tasks displayed in ascending order of due date by default.

6. Security & Best Practices

- Authentication and authorization are implemented using **Laravel Breeze**.
- Passwords are hashed using **Laravel's built-in hashing mechanism**.
- **CSRF protection** is enabled on all forms.
- Validation rules ensure **data integrity**, such as required fields and valid date formats.
- **Task ownership verification** ensures users can only edit or delete their own tasks.
- **SweetAlert confirmations** prevent accidental deletions or status changes.

7. Added Features / Enhancements

- **SweetAlert pop-ups** for task deletion and marking tasks as completed.
- **Search and status filter** functionality on the tasks page.
- **Responsive sidebar** with collapse/expand capability.
- **User dropdown menu** in the header with profile and logout options.
- **Task status badges** with color codes: Pending = Blue, In Progress = Yellow, Completed = Green.
- **Pagination** for tasks.
- **Optional future enhancements** ready, including task categories, reminders, file attachments, and export options.

8. Future Enhancements

- Add **task categories/labels** for better organization.
- Implement **notifications or reminders** for upcoming due tasks.
- Allow **file attachments** for tasks.
- Provide **export options** such as CSV or PDF reports.
- Enhance **mobile responsiveness** and user experience.

Conclusion.

This System Planning Document comprehensively represents the Laravel Task Management System, covering the database schema, ER diagram, technology stack, application flow, and implemented features. It demonstrates adherence to best practices in security, usability, and maintainability, and aligns fully with the requirements of the implemented web application. The document is ready for submission and provides a clear reference for future enhancements.