Task Management System Project Report

# Abstract

This project report outlines the development of a Task Management System using Spring Boot. The system is designed to manage users, tasks, and attachments efficiently. It includes features such as user authentication, task assignment, and file uploads. The project demonstrates the integration of RESTful APIs, entity relationships, and validation mechanisms to build a robust backend application.

# Final Outcome as a Learner

As a learner, this project helped me understand the core concepts of Spring Boot development, including controller-service-repository architecture, entity mapping, and RESTful API design. I gained hands-on experience in building scalable and maintainable backend systems, handling file uploads, and implementing validation and error handling.

# Project Creation: Why, What, How

## Why I Created This Project

The motivation behind creating this Task Management System was to build a practical application that solves real-world problems in organizing and tracking tasks. It also served as a learning platform to explore Spring Boot and backend development.

## What I Created

I developed a backend system that manages users, tasks, and attachments. The system includes features such as user registration, task assignment, file uploads, and data retrieval through RESTful endpoints.

## How I Created It

The project was built using Spring Boot. I structured the application into entities, repositories, services, and controllers. I used annotations like @Entity, @RestController, @Autowired, and @RequestMapping to define the behavior of each component. Validation was implemented using Jakarta annotations, and file uploads were handled using MultipartFile. The system was tested using Postman to verify each endpoint's functionality.

# Technologies and Tools Used

- Java 17

- Spring Boot

- Jakarta Validation

- JPA/Hibernate

- MySQL (or any relational database)

- Postman

- Maven

- IntelliJ IDEA or Eclipse

# Detailed Explanation of Modules

1. User Module: Handles user registration, authentication, and profile management. Includes endpoints for CRUD operations and search functionality.

2. Task Module: Manages task creation, assignment, status updates, and filtering by priority, category, and due dates.

3. Project Module: Enables creation and tracking of projects, including filtering by status, date range, and user roles.

4. Category Module: Organizes tasks into categories and provides task count per category.

5. Comment Module: Allows users to add comments to tasks for better communication.

6. Attachment Module: Supports file uploads linked to tasks, with metadata stored in the database.

7. Notification Module: Sends alerts and updates to users based on task and project activity.

8. User Role Module: Manages roles and permissions for users within the system.

# Testing and Validation

All REST APIs were tested using Postman. JUnit and Mockito were used for unit testing service and controller layers. Global exception handling was implemented to ensure consistent error responses.

# Conclusion

This project provided a comprehensive understanding of backend development using Spring Boot. It reinforced best practices in API design, validation, testing, and modular architecture. The system is scalable and can be extended with frontend integration or cloud deployment.