A Major Project Synopsis on

PlanIt: Modern To-Do List

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

2023-2025

By

Radhika Rathore

23FS20MCA00052

Under the guidance of

Dr. Monika Jyotiyana

Department of Computer Applications

Faculty of Science

Manipal University Jaipur

Jaipur, Rajasthan

2025

Introduction

PlanIt is a sophisticated web-based task management application engineered with Spring Boot and contemporary frontend technologies. The application stands out by incorporating gamification elements through streak tracking, AI-powered task prioritization, and an engaging user interface. Built for modern professionals and students, PlanIt transforms routine task management into an engaging and productive experience.

Development Environment

Development Tools

- IntelliJ IDEA Ultimate Edition
  - Spring Boot-specific features and tools.
  - Built-in database tools.
  - Git integration.
  - JUnit testing framework.
- Visual Studio Code [9]
  - Live Server extension for frontend development.
  - Prettier for code formatting.
  - HTML CSS Support extension.
- Version Control: Git with GitHub integration. [10]
- Database Management: MySQL Workbench 8.0.
- Build Automation: Maven 3.8+. [5]
- API Testing: Postman. [8]

Hardware

- MacBook Pro M1 512 GB
  - 8-core CPU for efficient compilation.
  - 8-core GPU for smooth animations.
  - 16 GB unified memory for seamless development.
  - 512 GB SSD for fast project loading and deployment.

Motivation

The digital task management landscape is saturated with basic to-do applications that fail to address the core challenges of sustained user engagement and effective task prioritization. Current solutions often lack:

- Meaningful user engagement mechanisms.
- Intelligent task prioritization systems.
- Gamification elements that encourage consistent usage.
- Cross-platform accessibility with offline capabilities. [7]
- Data-driven insights for productivity optimization.

Problem Statement

Manual task execution often lacks time management,which lowers the chances of completing tasks on time.PlanIt helps to resolve the problem by manging the priorities and sending timely reminders with deadlines and notifications to keep users on track.

Core Challenges

1. User Engagement
   a. Limited motivation to complete tasks.
   b. Poor task completion rates.
   c. Lack of reward systems.
2. Task Management
   a. Inefficient prioritization.
   b. Overwhelming task lists.
   c. Poor time management.

Solution Approach

Develop an intelligent task management system that addresses these challenges through:

- Attractive and responsive interface.
- Gamified streak system for consistent engagement.
- Smart notification system.
- Cross-device synchronization.
- Real-time progress tracking.
- ML-powered task prioritization.

Methodology

Backend Architecture

- Spring Boot 3.0 REST API
  - Layered architecture (Controller, Service, Repository). [2]
  - Spring Security with JWT authentication. [3]
  - Spring Data JPA for database operations.
  - Custom exception handling.

Database Design [4]

- MySQL 8.0 with the following schemas:
  - Users and Authentication.
  - Tasks and Categories.
  - Streaks and Analytics.
  - Settings and Preferences.

Frontend Development

The frontend is used to display the user interface of the application and handle user interactions.It allows users to add ,update,delete,and view their tasks in a visually organized and user friendly manner. [6]

Core Components

- **Home**: Main dashboard and task overview.
- **Preview**: Task preview and details view.
- **Task Management**: Core task operations interface.
- **Watchlist**: Priority and flagged tasks.
- **Analytics**: Progress and streak tracking.
- **Profile**: User settings and preferences.

User Portals

1. Admin Portal
   a. User management system.
   b. Access control.

    c. System configuration.
    d. User activity monitoring.
2. Manager Portal
    a. Task creation interface.
    b. Task template management.
    c. Schedule management.
    d. Performance tracking.
3. User Portal
    a. Task view and interaction.
    b. Progress tracking.
    c. Task completion workflow.

Technical Requirements

Functional Requirements

1. User Management
    a. Registration and authentication.
    b. Profile management.
    c. Preference settings.
    d. Activity history.
2. Task Management
    a. CRUD operations.
    b. Priority levels.
    c. Due dates and reminders.
3. Streak System
    a. Daily streak tracking.
    b. Achievement badges.
    c. Progress visualization.
4. Smart Features.
    a. AI-based task prioritization. [1]
    b. Smart reminders. [11]

Non-Functional Requirements

1. Performance
    a. Page load time < 2 seconds.

   b. API response time < 200ms.
2. Security
   a. HTTPS encryption.
   b. Input validation.
   c. Rate limiting.
   d. GDPR compliance.
3. Usability
   a. Mobile-first responsive design.
   b. Cross-browser compatibility.
   c. Offline functionality.
   d. Intuitive UI/UX.
   e. Accessibility compliance.

Expected Outcome

A production-ready web application delivering:

1. User Interface
   a. User Friendly , modern design.
   b. Attractive navigation.
   c. Dark/light theme support. [12]
   d. Responsive layouts.
   e. Interactive animations.
2. Core Features
   a. Seamless task management.
   b. Real-time updates.
   c. Smart prioritization.
   d. Streak tracking.
   e. Progress analytics.
3. Technical Achievement
   a. Scalable architecture.
   b. Robust security.
   c. High performance.
   d. Cross-platform support.

Future Enhancements

1. Collaboration Features
    a. Team workspaces.
    b. Real-time collaboration.

IX. Bibliography / References

[1] For AI-based Task Prioritization

Yadav, S., & Kumbhare, R. (2022). AI-based Task Prioritization and Scheduling for Efficient Time Management in Smart Applications. International Journal of Computer Applications, 184(32), 22-29.

[2] For Spring Boot framework

Spring Boot Reference Documentation. (n.d.). Retrieved from https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/.

[3] For Spring Security + JWT Authentication

Spring Security Reference. (n.d.). Retrieved from https://docs.spring.io/spring-security/reference/index.html.

[4] For MySQL Database Design

MySQL 8.0 Reference Manual. (n.d.). Retrieved from https://dev.mysql.com/doc/refman/8.0/en/.

[5] For Maven Build Tool

Apache Maven Project. (n.d.). Retrieved from https://maven.apache.org/guides/index.html.

[6] For Responsive Web Design

Marcotte, E. (2010). *Responsive Web Design*. A List Apart. Retrieved from https://alistapart.com/article/responsive-web-design/.

[7] For Cross-Platform Accessibility

Perry, D. (2021). Building Cross-Platform Web Apps with Progressive Web Apps. Medium Tech Publication..

[8] For API Testing with Postman

Postman Learning Center. (n.d.). *Introduction to Postman*. Retrieved from https://learning.postman.com/.

[9] For Visual Studio Code

Visual Studio Code Documentation. (n.d.). Retrieved from https://code.visualstudio.com/docs.

[10] For Git & GitHub

Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress. Retrieved from https://git-scm.com/book/en/v2.

[11] For Smart Notification System

Alok, A. & Priya, S. (2020). Smart Notification Systems Using Context-Aware Technology. International Journal of Computer Applications, 176(32), 10-14.

[12] For Light/Dark Theme Design Trends

UX Planet. (2020). *Dark Mode vs. Light Mode: UX Implications*. Retrieved from https://uxplanet.org/dark-mode-vs-light-mode-ux-implications-3b5fd27d15d4.