A Major Project Synopsis on

PlanIt: Modern To-Do List Web Application

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

MANIPAL UNIVERSITY
JAIPUR

Under the guidance of

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
    - Live Server extension for frontend development
    - Prettier for code formatting
    - HTML CSS Support extension
- Version Control: Git with GitHub integration
- Database Management: MySQL Workbench 8.0
- Build Automation: Maven 3.8+
- API Testing: Postman

Hardware

- MacBook Pro M1 512GB
    - 8-core CPU for efficient compilation
    - 8-core GPU for smooth animations
    - 16GB unified memory for seamless development
    - 512GB 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
- Data-driven insights for productivity optimization

Problem Statement

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:

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

Methodology

Backend Architecture

- Spring Boot 3.0 REST API
  - Layered architecture (Controller, Service, Repository)
  - Spring Security with JWT authentication
  - Spring Data JPA for database operations
  - Custom exception handling
  - Logging with SLF4J and Logback

Database Design

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

Frontend Development

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
   e. Support ticket management

   f. Subscription handling

2. **Manager Portal**
   a. Task creation interface
   b. Task template management
   c. Schedule management
   d. Team collaboration tools
   e. Performance tracking
   f. Task preview system
3. **User Portal**
   a. Task view and interaction
   b. Progress tracking
   c. Participation metrics
   d. Achievement system
   e. 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. Bulk actions
   c. Categories and tags
   d. Priority levels
   e. Due dates and reminders
   f. Recurring tasks
3. Streak System
   a. Daily streak tracking
   b. Achievement badges
   c. Progress visualization
   d. Milestone rewards
4. Smart Features

    a. AI-based task prioritization
    b. Workload optimization
    c. Productivity analytics
    d. Smart reminders

Non-Functional Requirements

1. Performance
    a. Page load time < 2 seconds
    b. API response time < 200ms
    c. 99.9% uptime guarantee
    d. Support for 10,000+ concurrent users
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. Clean, modern design
    b. Intuitive navigation
    c. Dark/light theme support
    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. Task delegation
   c. Shared projects
   d. Real-time collaboration
2. Integration Capabilities
   a. Calendar services (Google, Outlook)
   b. Cloud storage integration
   c. Email integration
   d. Third-party API support
3. Advanced Features
   a. Natural language processing for task input
   b. Advanced analytics dashboard
   c. Custom workflow automation
   d. Voice command support
4. Mobile Development
   a. Native iOS application
   b. Native Android application
   c. Cross-platform sync
   d. Push notifications
5. AI Enhancements
   a. Predictive task scheduling
   b. Personalized productivity insights
   c. Smart task categorization
   d. Workload optimization algorithms

IX. Bibliography / References

1. Spring Boot in Action – Craig Walls
2. Java Persistence with Hibernate – Christian Bauer, Gavin King
3. **Head First Design Patterns** – Eric Freeman & Elisabeth Freeman
4. Effective Java – Joshua Bloch
5. Official Spring Documentation – https://spring.io/docs
6. **Spring Security Reference Guide** – https://docs.spring.io/spring-security
7. **MySQL 8.0 Reference Manual** – https://dev.mysql.com/doc/refman/8.0/en/
8. Postman API Testing Documentation – https://learning.postman.com/