Software Requirement Specification

Project: KanbanFlow Task Manager (Full-Stack MERN Application)

1. Introduction

1.1 Purpose

The purpose of this Software Requirement Specification (SRS) is to define all functional and non-functional requirements for the KanbanFlow Task Manager, a full-stack MERN application built as an intermediate project for the internship program.

1.2 Scope

The system allows users to:

- Register and log in using secure authentication.
- Create, view, update, and delete tasks.
- Organize tasks into status categories (To-Do, In-Progress, Completed).
- Manage tasks through a kanban-style interface.
- Store data securely in MongoDB with an Express.js/Node.js backend.

2. Overall Description

2.1 User Roles

- Registered User: Can log in and manage tasks.
- Guest (Unauthenticated): Can only register/login.

2.2 System Features Overview

- JWT-based authentication
- Task CRUD (Create, Read, Update, Delete)
- Task status updates (move between columns)
- Protected API routes
- Responsive UI built with React
- RESTful API using Express.js

3. Functional Requirements (Core for SRS v1)

This section defines the core functional requirements of the *KanbanFlow Task Manager* system. These requirements describe what the system must do to support user authentication, task management, and task status updates.

3.1 User Authentication

3.1.1 User Registration

Description:
Users can create an account using name, email, and password.

Requirements:

- FR-1: System must validate unique email.
- FR-2: Password must be hashed using bcrypt.
- FR-3: System must store user details in MongoDB.
- FR-4: System must return a JWT token on successful registration.

3.1.2 User Login

Description:
Registered users can log in to access tasks.

Requirements:

- **FR-5: System must verify email and password.**
- **FR-6: System must return a valid JWT token on success.**
- **FR-7: Incorrect credentials must return error messages.**

3.1.3 Protected Routes

- **FR-8: All task-related routes must require a valid JWT.**
- **FR-9: Unauthorized access must return 401.**

3.2 Task Management (CRUD)

3.2.1 Create Task

Requirements:

- **FR-10: User can create a task with title, description, due date, and status.**
- **FR-11: Task must be linked to the authenticated user.**

3.2.2 Read Tasks

Requirements:

- **FR-12: User can fetch all tasks belonging to their account.**
- **FR-13: User must be able to fetch a single task by ID.**

3.2.3 Update Task

Requirements:

- **FR-14: User can edit title, description, and due date.**
- **FR-15: User can update task status (To-Do → In-Progress → Completed).**

3.2.4 Delete Task

Requirements:

- **FR-16: User can delete any of their tasks.**

- FR-17: System must ensure a user cannot delete another user’s task.

3.3 Task Status Updates (Kanban Flow)

Requirements:

- FR-18: Tasks must have a status field: "todo", "in-progress", "completed".
- FR-19: Frontend must allow drag-and-drop (optional for now; basic status button update is okay in SRS v1).
- FR-20: Backend must store updated status in MongoDB.

4. Non-Functional Requirements

4.1 Performance

- NFR-1: API should respond within 200–300ms on average.

4.2 Security

- NFR-2: Passwords must be stored using bcrypt hashing.
- NFR-3: JWT tokens must expire (e.g., after 1 hour).
- NFR-4: CORS properly configured for client-server communication.

4.3 usability

- NFR-5: UI must be responsive and mobile-friendly

5. Assumptions and Dependencies

- User must have internet access.
- MongoDB Atlas or local MongoDB instance must be available.
- Node.js and npm must be installed.