Smart SDLC AI-Enhanced Project Report

1. Introduction

- Project Title: Smart SDLC AI-Enhanced System

- Team Members:

  - Team Leader: Mohammad Anjum Bader

  - Team Member: Mohammad Yaseen

  - Team Member: Munjuluri Srikar Bharadwaj

  - Team Member: Navadeep Chowdary

2. Project Overview

- Purpose: To develop a Smart Software Development Life Cycle (SDLC) platform enhanced with AI capabilities to automate document generation, improve collaboration, and ensure efficient project tracking.

- Features:

  - AI-powered document generation (problem statements, solution architecture)

  - API integration and testing tools

  - Project planning with sprint tracking and velocity charts

  - User authentication and secure login

  - Dashboard for real-time updates

3. Architecture

- Frontend: Developed using React.js with modular component structure and state management via hooks.

- Backend: Built using Node.js and Express.js to manage routing, authentication, and API services.

- Database: MongoDB used for storing user data, project metadata, and document content.

4. Setup Instructions

- Prerequisites: Node.js, npm, MongoDB, Git

- Installation:
  1. Clone the repository: git clone <repo_url>
  2. Navigate to client and server directories
  3. Run `npm install` in both directories
  4. Set up `.env` files with necessary credentials
  5. Start MongoDB server

5. Folder Structure

- Client:
  - /components: React components (Forms, Dashboard, Auth)
  - /pages: Routing pages
  - /assets: Static files and styles
- Server:
  - /routes: Express routes
  - /controllers: Business logic handlers
  - /models: Mongoose schemas
  - /config: Environment configurations

6. Running the Application

- Frontend: Run `npm start` in the `client` directory
- Backend: Run `npm start` in the `server` directory

7. API Documentation

- /api/register [POST] ? Register a new user
- /api/login [POST] ? Authenticate user

- /api/generate [POST] ? Generate documents via AI
- /api/projects [GET] ? Retrieve project list

8. Authentication

- JWT-based authentication
- Tokens stored in localStorage for persistent login
- Middleware protects private routes

9. User Interface

- Features include:
  - Registration and Login forms
  - Interactive dashboard
  - Input forms for AI generation
  - Output display with download options

10. Testing

- Manual and functional testing for all modules
- Used Postman for API testing
- Basic unit testing with Jest (for frontend)

11. Screenshots or Demo

- Demo: Hosted on localhost or Replit
- Screenshots: Available in `/screenshots/` directory

12. Known Issues

- Token expiration not handled for all scenarios
- No third-party login integration yet
- PDF export styling might vary

13. Future Enhancements

- Add OAuth login (Google/Facebook)
- Role-based access control for admin/editor
- Analytics dashboard for activity tracking
- Expand AI support for more SDLC phases