■ Project Documentation

Project Title: SmartSDLC – AI-Enhanced Software Development Life Cycle
Course: Naan Mudhalvan Project
Prepared by:
Team members:
Ganesh . P
Harish . V
Harish Kumar . S
Nithishkumar . D
Kamalesh . A
Date: 12/09/2025

1. Introduction

Traditional SDLC provides a structured process for software development, but it often faces challenges like delays, high costs, human errors, and lack of adaptability. The proposed project, SmartSDLC, integrates Artificial Intelligence (AI) tools into each phase of SDLC to enhance efficiency, reduce risks, and improve decision-making. This AI-powered SDLC framework automates requirement gathering, optimizes design, improves code quality, enables intelligent testing, and supports predictive maintenance.

2. Objectives of the Project

- To enhance traditional SDLC with AI-powered automation and intelligence.
- To reduce development time and cost.
- To improve accuracy in requirement analysis and design decisions.
- To enable predictive testing and maintenance using AI.
- To develop a prototype AI-augmented framework for SmartSDLC.

3. Scope of the Project

- Applicable for small, medium, and large-scale projects.
- Supports Agile, Iterative, and Hybrid SDLC models with AI enhancements.
- Focuses on AI integration in Requirement Analysis, Design, Coding, Testing, Deployment, and Maintenance.
- Can be extended to DevOps + MLOps environments.

4. Phases of SmartSDLC (AI-Enhanced)

- AI-Enhanced Requirement Analysis - NLP for analyzing stakeholder inputs - Chatbots & AI assistants for requirements - Automated conflict detection
- AI-Driven Design - AI suggests architecture blueprints - Auto-generation of UML, ER models - Tech stack recommendations
- AI-Assisted Development - Code auto-generation - Bug prediction - Automated refactoring
- AI-Powered Testing - Test case generation - Predictive defect analysis - Intelligent regression testing
- AI-Optimized Deployment - Automated CI/CD pipelines - AI monitors server load - Adaptive deployment strategies

- AI-Based Maintenance - Predictive maintenance - AI chatbots for support - Self-healing monitoring systems

5. SmartSDLC Architecture

Layers of SmartSDLC: - Data Layer – Collects requirements, logs, and feedback. - AI Engine – Uses NLP, ML, and Predictive Analytics. - SDLC Workflow Layer – Requirement, Design, Coding, Testing, Deployment, Maintenance. - Automation Layer – CI/CD, Monitoring, Self-healing systems. - User Interface Layer – Dashboards for developers, testers, managers.

6. Advantages of SmartSDLC

- Faster Development – AI reduces manual work.
- Better Accuracy – Automated analysis & predictions.
- Cost-Effective – Optimized resource allocation.
- High Quality – Improved code and testing coverage.
- Scalability – Suitable for evolving requirements.

7. Challenges of SmartSDLC

- High initial cost for AI integration.
- Requires large datasets for accurate predictions.
- Dependency on AI tools may reduce human creativity.
- Security & Privacy issues with AI-driven automation.

8. Real-World Applications

- Healthcare Systems – Hospital management software.
- Banking & FinTech – Fraud detection.
- E-commerce Platforms – Personalization & predictive scaling.
- Smart Cities & IoT – Efficient deployment & predictive maintenance.

9. Case Study Example – Smart Library Management System

Requirement Gathering: AI chatbot collects needs. Design: AI generates ER diagram. Development: AI auto-generates CRUD in Python/Django. Testing: AI generates test cases. Deployment: AI selects optimized cloud infrastructure. Maintenance: AI predicts database scaling.

10. Conclusion

SmartSDLC demonstrates how AI can revolutionize software development by making the process faster, smarter, and more reliable. It combines the structured approach of SDLC with AI-driven automation, prediction, and optimization, enabling future-ready software engineering.

11. References

- Pressman, R. S., Software Engineering: A Practitioner’s Approach. - Sommerville, I., Software Engineering. - Research papers on AI in SDLC (IEEE, Springer). - Tools: ChatGPT, GitHub Copilot, Selenium AI, TensorFlow.