Project Report

## Title: AI-Enhanced Software Development Life Cycle (AI-SDLC)

MY TEAM FORMATION IN SMARTINTERNZ

Team ID: LTVIP2025TMID32245

Team Size: 4

Team Leader: Grandhe Prasanna Lakshmi

Team Member: D Bhashpika

Team Member: Dadi Suneel

Team Member: Chella Yuvaraj

1. INTRODUCTION
   1. Project Overview

The Software Development Life Cycle (SDLC) is a structured approach to building high quality software in a systematic and cost-effective way.

In this project, we design an AI-enhanced SDLC frameworks that incorporate AI- powered tools for requirements analysis, design optimization, automated code generation, predictive testing, and deployment monitoring.

* 1. Purpose

To demonstrate how AI can improve each phase of the SDLC.

To reduce human errors and accelerate development.

To provide predictive analytics for better project management.

To improve software through intelligent testing and continuous monitoring

2.IDEATION PHASE

2.1 Problem Statement

Traditional SDLC methods are prone to delays, communication gaps, and repetitive manual work. Common issues include

Misunderstanding requirements leading to costly changes.

Manual testing that misses hidden defects.

Time-Consuming code reviews.

Difficulty predicting project risks.

2.2 Empathy Map Canvas

Stake holders: purchasing the project and invest.

Developers – Want faster coding tools and reduced repetitive work.

Testers – Need better defect prediction and automated testing.

Project Manager – seek accurate progress tracking and risk alerts.

Clients – Expect quick delivery with high quality

2.3 Brainstorming

NLP – Based requirements extraction from client conversation.

Automated defect detection using ML algorithms.

Predictive analytics for project timelines

3.REQUIREMENT ANALYSIS

3.1 Customer journey map

1. Requirements gathering- AI converts conversation into structured specifications
2. Design stage- AI – suggests architecture patterns
3. Development- Ai-assisted coding and error detection
4. Testing – Automated test case generation and bug prediction
5. Deployment – AI monitors app health and user feedback

3.2 Solution Requirements

Hardware: Server with GPU (for AI models) developers

Software : Python, TensorFlow, github

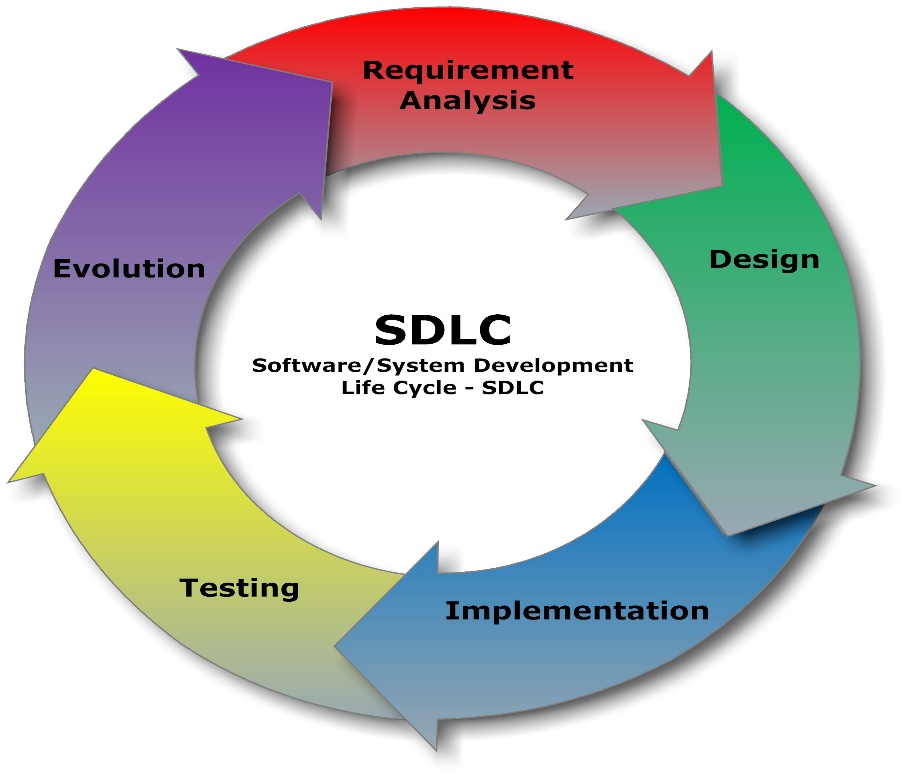
AI tools: ChatGPT, GitHub Copilot

3.3 Data Flow Diagram

Input : user requirements > AI parser > Requirement documents.

Processing : AI tools assist design, coding, testing, deployment

Output : High-quality, AI-assisted software product.



[This Photo](http://commons.wikimedia.org/wiki/File:SDLC_-_Software_Development_Life_Cycle.jpg) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

3.4 Technology Stack

Programming Language: python, Javascript

Frameworks: Django, React

Testing tools : Selenium, Test AI

Cloud services : AWS/GCP for hosting AI models

AI/ML Libraries : tensor flow, PYTorch

4 . PROJECT DESIGN

4.1 problem solution fit

The problem of delays and inefficiency in SDLC is solved by AI through automation,

Prediction, and intelligent assistance

4.2 Proposed Solution

An AI-Powered SDLC pipelines that:

Automates requirements documentation.

Generates code snippets

Predicts and fixes bugs early

4.3 Solution Architecture

Architecture Includes:

1. Requirements Analysis Modules (NLP-based)
2. AI-Driven Code Generater
3. Automated Testing Engine
4. Predictive Risk Analyzer
5. Monitoring Dashboards

5.PROJECT PLANNING & SCHEDULING

5.1 Project Planning

1. Requirement analysis NLP extraction from client docs

2. Development AI-assisted coding & code review

3. Deployment AI monitoring tools

1. FUNCTIONAL AND PERFORMANCE TESTING
   1. Performance Testing

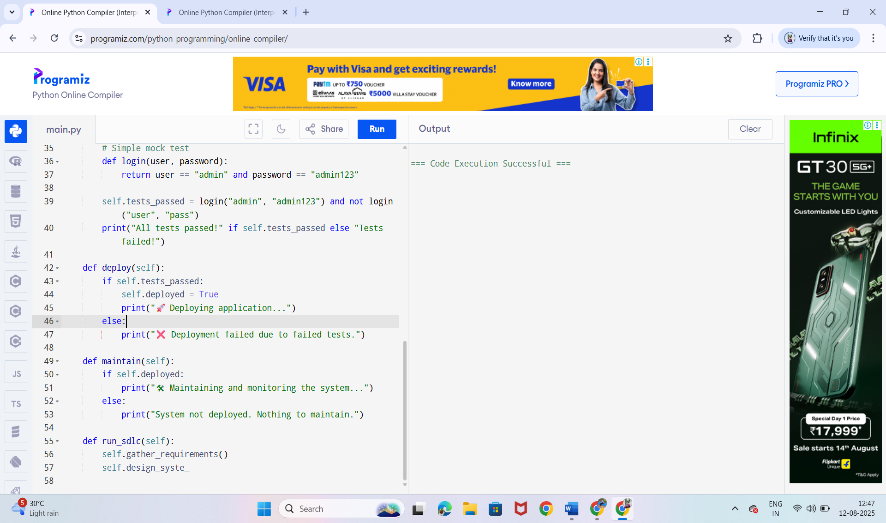
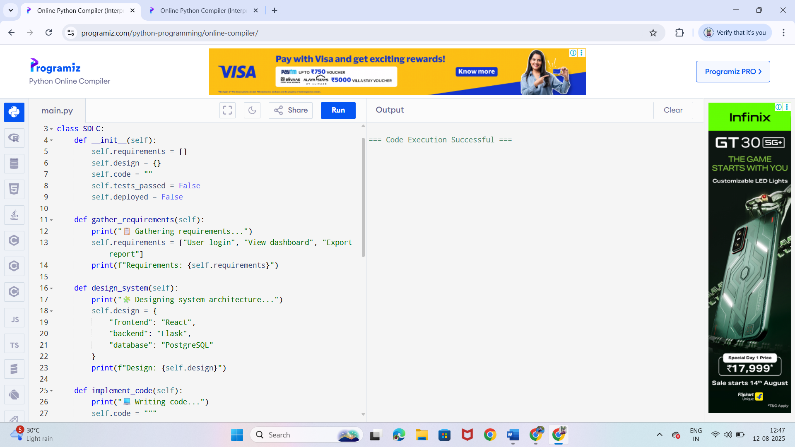
Comparison of traditional VS AI-enhanced SDLC:

Bugs detected earlier: 25% improvement

Reduced manual documentation work by 55%

7.RESULTS

7.1 Output screen shot



8.ADVANTAGES & DISADVANTAGES

Advantages

1. Faster development
2. Reduced manual errors
3. Better project risk prediction

Disadvantage

1. High initial setup cost
2. Dependency on AI tools
3. AI models bias if training data is poor

9. CONCLUSION

AI integration into SDLC significancy improves software development efficiency, quality, and delivery speed. This project demonstrates that AI-Enhanced SDLC can be a future standard for modern software engineering.

10. FUTURE SCOPE

1. AI ethics and bias reduction in SDLC

2. Fully autonomous software development agents

3.AI- driven real-time collaboration for remote teams

11.APPENDIX

Source code : GitHub link