# Software Development Life Cycle (SDLC) for a Blog Website

## Introduction

The Software Development Life Cycle (SDLC) is a structured framework that outlines the stages involved in developing software applications. By following SDLC principles, developers ensure that the final product meets user requirements, is delivered on time, and maintains high quality. This document details the SDLC approach for building a blog website.

## Problem Statement

In today’s digital age, many individuals and small businesses struggle to share their ideas, expertise, and updates with a broad audience due to the lack of an easy-to-use, customizable, and cost-effective blogging platform. Existing platforms either require complex technical knowledge or impose restrictions on customization. The goal of this project is to develop a simple, user-friendly, and scalable blog website that enables users to create, manage, and share their blogs efficiently. This solution aims to empower individuals with minimal technical expertise to showcase their content with ease.

## SDLC Phases

### 1. Planning

The planning phase focuses on defining the project's purpose and setting goals. Key steps in this phase include:

- Identifying the project’s purpose: A blog website where users can register, write, edit, and publish blog posts.

- Determining the target audience: Individuals, hobbyists, small businesses, and professional writers.

- Assigning roles among team members to ensure clear responsibility distribution (e.g., front-end developer, back-end developer, tester, and designer).

- Setting milestones to track progress (e.g., UI design completion in two weeks, implementing the login system in three weeks, etc.).

### 2. Requirement Analysis

Requirement analysis is crucial to define what features the project should deliver. Requirements are categorized as follows:

Functional Requirements:

- User authentication (signup, login, logout).

- Ability to create, edit, and delete blog posts

- Commenting system for user engagement.

- Responsive design for mobile and desktop users.

Non-Functional Requirements:

- Secure user authentication with encrypted passwords.

- Scalable architecture to handle increased website traffic.

- SEO-friendly URLs and metadata for better search engine visibility.

- Intuitive and visually appealing user interface to improve user experience.

### 3. Design

The design phase focuses on creating a clear layout and technical framework for the website.

Database Schema Design:

- Users Table

- Blog Posts Table

- Comments Table

Technology Stack:

- Frontend: HTML, CSS,

- Backend: Python (Flask)

- Database: MySQL for secure data management.

- Version Control: Git & GitHub for efficient code collaboration.