Hackathon Project Phases Template

# Project Title: AI BLOG GENERATOR USING LLAMA 2

# Team Name: SCRIPTSEERS

# Team Members:

* Gonabakthula Sruthi
* Bandila Martin Phillip
* Mulge Arpitha
* Kolnedi Mythili

# Phase-1: Brainstorming & Ideation

## Objective:

## To develop an AI-powered blog generator using LLama 2 that automates high-quality content creation, allowing users to generate well-structured, SEO-optimized blogs based on customizable preferences such as category, tone, and word count, thereby enhancing efficiency and creativity in content writing.

## Key Points:

1. **Problem Statement:**

Content creation is time-consuming and challenging, requiring originality, consistency, and SEO optimization. Traditional writing methods and existing AI tools often produce generic, low-quality, or unoptimized content that lacks customization.

Our AI Blog Generator using LLama 2 addresses these issues by:

* Generating high-quality, human-like blogs instantly.
* Allowing customization based on category, tone, and word count.
* Enhancing SEO for better search rankings.
* Reducing writing time and effort while maintaining creativity.

1. **Proposed Solution:**

Our AI Blog Generator using LLama 2 introduces key innovations to enhance content quality, personalization, and credibility. Unlike traditional AI tools that generate free-flowing text, our model creates well-structured blogs with clearly defined sections such as introductions, key points, and conclusions, ensuring better readability and logical flow.

1. **Target Users:**

Our AI Blog Generator using LLama 2 will benefit a wide range of users, especially those who create, manage, or optimize online content. Here’s who will gain the most:

* Content Creators and Bloggers
* Digital Marketers & SEO Specialists
* Journalists & News websites
* Students & Researchers

1. **Expected Outcome:**

The LLama 2 is a fully functional AI-powered tool that automates blog writing efficiently. It will generate high-quality, SEO-friendly, and customizable blog content, reducing manual effort and enhancing productivity.

# Phase-2: Requirement Analysis

## Objective:

* + Define technical and functional requirements.

## Key Points:

* **Technical Requirements:**

1. AI Model – Implement LLama 2 for content generation.

2. Backend – Use Python (Flask/Fast API) for processing AI requests.

3. Frontend – Develop a user-friendly UI with streamlit

4. Database – Store user inputs & generated content (PostgreSQL/MongoDB).

5. SEO & NLP Integration – Optimize content for readability and search engines.

6. Hosting & Deployment – Fast API backend and stream lit frontend

7. Security – Implement authentication, data encryption, and API protection.

* **Functional Requirements:**

1. User Input Processing – Accept topics, keywords, and preferences.

2. Content Generation – AI produces structured, high-quality blog posts.

3. Customization – Users can adjust tone, format, and length.

4. SEO Optimization – AI includes relevant keywords and headings.

5. Editing & Refinement – Users can modify generated content before publishing.

6. Multi-Format Support – Generate listicles, guides, news, and storytelling blogs.

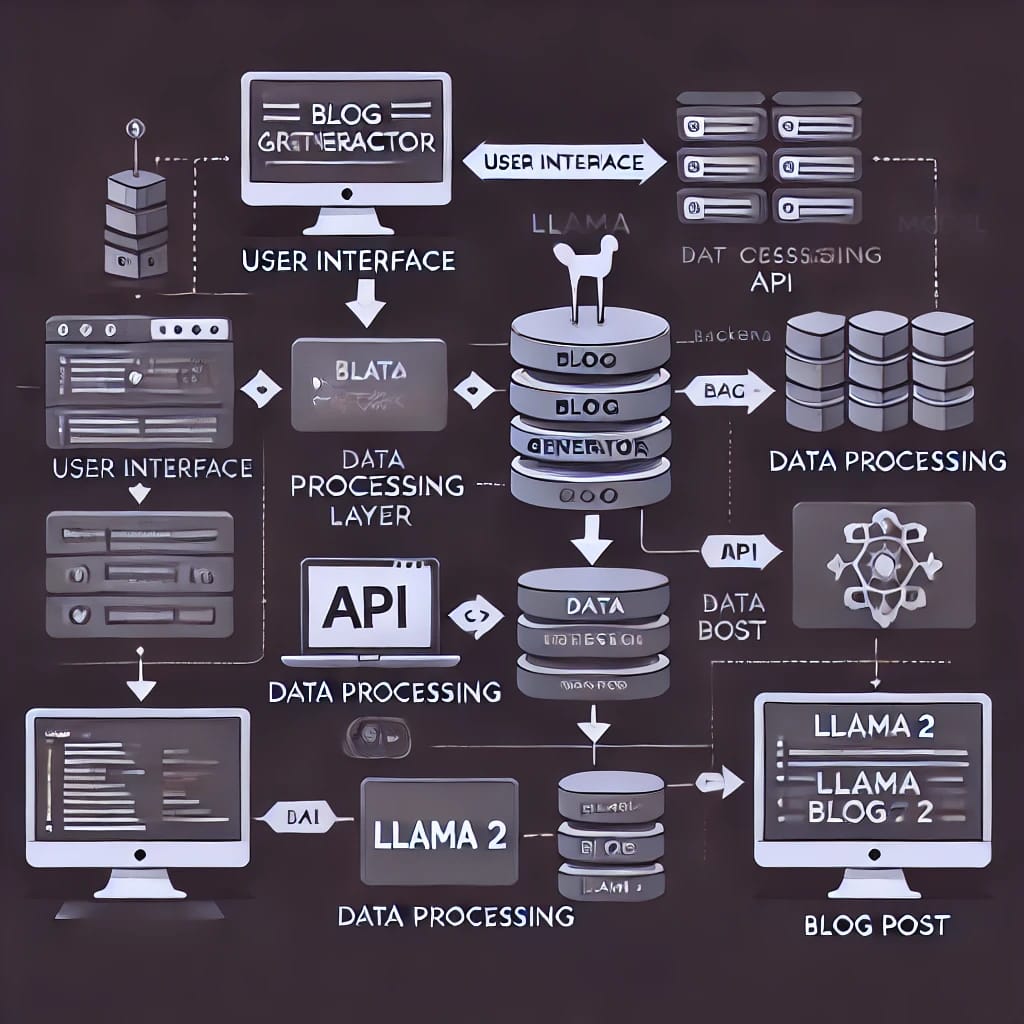
7. Export & Save Options – Allow users to download or share blogs.

* **Constraints & Challenges:**

1. Large amounts of generated content require efficient database management.
2. AI-generated content might lack deep creativity or specific brand voice customization.
3. Real-time content generation may experience delays, especially for long-form blogs.
4. AI might generate repetitive, generic, or inaccurate content that needs human refinement.
5. AI can generate incorrect information, requiring additional verification mechanisms.
6. Handling multiple users and high-volume content generation efficiently.

# Phase-3: Project Design

## Objective:



## Key Points:

1. **System Architecture:**

* Allows users to input topics, keywords, and preferences.
* Displays AI-generated blog content with editing options.
* Processes user inputs and sends requests to the AI model.
* Integrates SEO optimization and content refinement.
* By using llama2 it generates structured, high-quality blog content**.**

1. **User Flow:**

* user selects a topic based on their interests and requirements
* then user enters a topic based on their requirements, keywords or their ideas
* backend uses llama2 to a generate ai content
* This website processes the input data and displays the content in structured format in easy-to-read format

1. **UI/UX Considerations:** (If applicable, wireframe or basic layout)

# Phase-4: Project Planning (Agile Methodologies)

## Objective:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Duration** | **Key Objectives** | **Deliverables** |
| **Sprint 1: Project Initiation & Planning** | Day 1 | - Define project scope & requirements - Identify key features & tech stack - Set up Agile workflow & team roles | - Project roadmap - Tech stack selection - Initial wireframes |
| **Sprint 2: Frontend** | Day 1 | - Build UI for blog customization | - Fully functional UI for user input - Interactive blog generation page |
| **Sprint 3: Backend & AI Model Setup** | Day 2 | - Integrate LLama 2 for text generation - Set up API for AI responses - Implement user authentication | - Working AI-based text generation API - Secure user login system |
| **Sprint 4: Blog Generation & Optimization** | Day 2 | - Improve AI-generated content quality - Add SEO keyword optimization - Implement grammar & readability enhancements | - Enhanced AI-generated content - Basic SEO features |
| **Sprint 5:User Testing&Feedback** | Day 2 | - Conduct usability tests - Gather feedback from users - Identify bugs and performance issues | - Bug reports & fixes - UI/UX improvements |
| **Sprint 6: Final Enhancements & Deployment** | Day 2 | - Integrate additional features (plagiarism check, voice-to-text) - Optimize for scalability - Deploy on cloud hosting (AWS/GCP) | - Final AI Blog Generator version - Live deployment & documentation |

## Key Points:

1. **Sprint Planning with Priorities**

**Sprint 1 – Setup & Integration (Day 1)**

Set up the environment & install dependencies.

Integrate Google Gemini API.

Build a basic UI with input fields.

**Sprint 2 – Core Features & Debugging (Day 2)**

Implement search & comparison functionalities.

Debug API issues & handle errors in queries**.**

**Sprint 3 – Testing, Enhancements & Submission (Day 2)**

Test API responses, refine UI, & fix UI bugs.

Final demo preparation & deployment.

# Phase-5: Project Development

## Objective:

* + Code the project and integrate components.

## Key Points:

1. **Technology Stack Used:**

○ Frontend: Stream lit

○ Backend: AI Blog Generator using llama 2

○ Programming Language: Python

1. **Development Process:**

○ Implement API key authentication

○ Develop vehicle comparison and maintenance tips logic.

○ Optimize search queries for performance and relevance.

1. **Challenges & Fixes:**

○ Challenge: Delayed API response times.

Fix: Implement caching to store frequently queried results.

○ Challenge: Limited API calls per minute.

Fix: Optimize queries to fetch only necessary data

# Phase-6: Functional & Performance Testing

## Objective:

* + Ensure the project works as expected.

## Key Points:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case id | Category | Test scenario | Expected outcome | status | Tester |
| TC - 001 | Functional Testing | Query(Cricket) | Blog will be displayed | PASSED | Tester 1 |
| TC - 002 | Performance testing | API response time 500ms | API returns the results quickly | PASSED | Tester 2 |
| TC - 003 | Bug fixes and improvements | Fixed incorrect API responses | Data accuracy is improved | FIXED | Tester 3 |
| TC - 004 | Final validation | Ensure UI responsive across devices | UI works on mobile and desktop | PASSED | Tester 4 |
| TC - 005 | Deployment testing | Host the app using stream lit sharing | App is accessible online | DEPLOYED | Devops |

# Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link:**

**https://github.com/Sruthi-02-sw/blog-generator-using-ai/tree/main**

1. **Presentation**