Hackathon Project Phases Template that ensures students can complete it efficiently while covering all six phases. The template is structured to capture essential information without being time-consuming.

Hackathon Project Phases Template

Project Title:

Blog Generation Using LLaMA 2 and Streamlit

Team Name:

Hacktastics

Team Members:

- K.Vyshnavi
- K.Srivalli
- K.Sri Lakshmi
- Sk.Durre Shahvar
- K.Srikala

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI powered tool using LLaMA 2 and Streamlit that generates blogs automatically based on the inputs given by the user(content creators,researchers,professionals).

Key Points:

1. **Problem Statement:**
    - Nowadays, creating effective and widely reachable blogs has become a challenge for content creators, researchers, and professionals. Writing high-quality blogs manually is time-consuming and requires significant effort
    - This tool is useful for those who need to quickly generate high-quality written content.

2. **Proposed Solution:**

    - An AI powered blog generator using LLaMA 2 and Streamlit to provide blogs for content creators by specifying their topic,word count and target audience.
    - This application generates a detailed and technical blog post suitable for their requriments.

3. **Target Users:**

    - **Content Creators** – Helps bloggers, writers, and social media managers generate engaging blog posts efficiently.
    - **Researchers** – Assists in drafting articles, summaries, and reports based on specific topics.
    - **Marketers** – Generates SEO-optimized blog content for digital marketing and brand awareness.
    - **Students** – Aids in writing essays, research papers, and academic articles.

4. **Expected Outcome:**
    - A functional **AI-Blog generator** that provides high-quality,tailored blog content based on the user requirements.

Phase-2: Requirement Analysis

Objective:

- Define technical and functional requirements for the Blog generator Website.

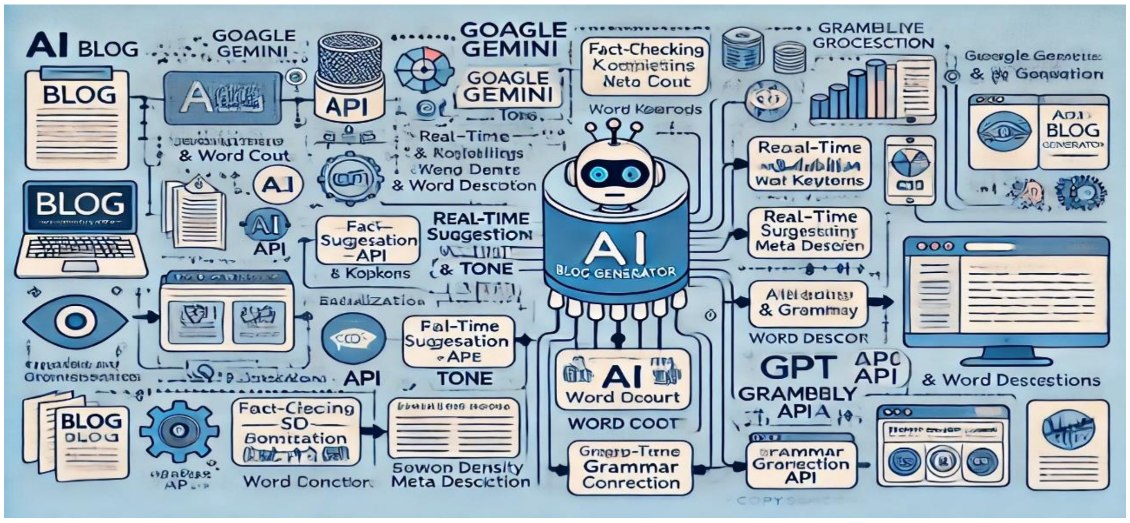
Key Points:

1. **Technical Requirements:**
    - Programming Language : Python
    - Frontend : Streamlit Web Framework
    - Backend : Hugging Face Transformers, PyTorch
2. **Functional Requirements:**
    - **Fetch relevant topic ideas and keywords from external APIs** for blog content generation.
    - **Display blog structure, content, and SEO tips** in an easy-to-use interface.
    - **Provide real-time content suggestions** based on input and SEO insights for optimization.
    - **Enable users to generate eco-friendly or niche blog topics** tailored to specific themes and keywords.
3. **Constraints & Challenges:**
    - **Generate blog content in real-time with minimal latency.**
    - **Optimize AI calls to handle high traffic without exceeding rate limits.**
    - **Ensure content quality with correct grammar, relevance, and clarity.**
    - **Provide an intuitive, responsive UI for seamless user experience.**

Phase-3: Project Design

Objective:

- Create the architecture and user flow.



Key Points:

1. System Architecture:

- User enters a **blog-related query** via the UI (e.g., topic, keywords, tone, and word count).
- The query is processed using **Google Gemini API** (or another AI model).
- The AI model **generates, enhances, and optimizes** the blog content.
- The frontend displays the **generated blog, SEO suggestions, and editing options**.

2. User Flow:

- **Step 1**: User enters a query (e.g., "Write a blog on AI trends in 2024").
- **Step 2**: The backend calls the **Gemini API** (or GPT model) to generate blog content.
- **Step 3**: The system **optimizes the content for readability, SEO, and grammar**.
- **Step 4**: The processed blog is displayed with options to **edit, export, or publish**.

3. UI/UX Considerations:

- **Minimalist, user-friendly interface** for seamless content creation.
- **Customization options** (tone, word limit, format).
- **SEO analysis tools** for keyword density and ranking improvements.
- **Dark & light mode** for better readability and user experience.
- **Export & publishing options** (WordPress, Medium, PDF, HTML).

Phase-4: Project Planning (Agile Methodologies)

Objective:

- Break down the tasks using Agile methodologies.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & API Integration	High	6 hours (Day 1)	End of Day 1	Member 1	Google API cloud Key, Python, Streamlit setup	AI model connected and operation
Sprint 1	Frontend UI Development	Medium	2 hours (Day 1)	End of Day 1	Member 2	API response format finalized	Basic UI with input fields for Blog topics
Sprint 2	AI Model Implementation & Keyword Customization	High	4 hours (Day 2)	Mid-Day 2	Member 1& 2	API response, UI elements ready	Blog topic generation with keyword customization
Sprint 2	Error Handling & Debugging	High	2 hours (Day 2)	Mid-Day 2	Member 1&4	API logs, UI components	Enhanced error messages and stability
Sprint 3	Testing & UI Enhancements	Medium	2 hours (Day 2)	Mid-Day 2	Member 2& 3	API model output, UI mockup	Polishes UI and improved responsibilities
Sprint 3	Final Presentation & Deployment	Low	1 hour (Day 2)	End of Day 2	Entire Team	Fully functional AI blog Generator	Final Demo-ready blog generator with working features

Key Points:

Sprint Planning for AI Blog Generator

Sprint 1 – Setup & Integration (Day 1)

- (High Priority) Set up the development environment & install dependencies.
- (High Priority) Integrate **Google Gemini API** for blog content generation.
- (Medium Priority) Build a **basic UI** with input fields for topic, keywords, and word count.

Sprint 2 – Core Features & Debugging (Day 2)

- (High Priority) Implement **blog generation & content enhancement modules**.
- (High Priority) Debug **API issues & handle errors in query processing**.
- (Medium Priority) Integrate **SEO optimization & readability improvements**.

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

- (Medium Priority) Test API responses, refine UI, and fix UI bugs.
- (Low Priority) Add **export & publishing options** (WordPress, PDF, HTML).
- (Low Priority) **Final demo preparation & deployment**.

Phase-5: Project Development

Objective:

- Implement core features of the AI Blog Generator application.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit (or React for a more dynamic UI)
    - **Backend:** Google Gemini API (or GPT-based model)
    - **Programming Language:** Python
    -
2. **Development Process:**
    - Implement **API key authentication** and **Google Gemini API integration**.
    - Develop **blog generation logic** with **SEO optimization** and **readability enhancements**.
    - Optimize **user queries for better blog quality and topic relevance**.
3. **Challenges & Fixes:**
    - **Challenge:** API response delay.
      Fix: Implement **caching** to store frequently requested topics.
    - **Challenge:** Limited API calls per minute.
      Fix: Optimize **query handling** to generate content efficiently with minimal API requests.

Phase-6: Functional & Performance Testing

Objective:

- Ensure the project works as expected.

Key Points:

1. ***Test Cases Executed***

    - Blog generation accuracy tested with various inputs.
    - SEO optimization and keyword placement verified
    - API response handling and UI functionality checked.

2. ***Bug Fixes & Improvements***

    - Fixed API timeout issues with caching.
    - Resolved formatting errors in generated content.

- Optimized query processing for better responses.

3. *Final Validation*

- The project generates AI-powered blogs effectively.
- User-friendly UI allows easy editing and customization.
- Efficient API handling ensures smooth performance.

4. *Deployment*

- Hosted on Streamlit or Flask backend.
- Integrated export functionality for multiple formats.
- Final demo link available if hosted.

Test Case ID	**Category**	**Test Scenario**	**Expected Outcome**	**Status**	**Tester**
TC-001	Functional Testing	Query "How to write an engaging blog post?"	AI should generate relevant tips for writing engaging blogs.	✓ Passed	Tester 1
TC-002	Functional Testing	Query "Top trending tech blogs in 2024	AI should suggest relevant trending topics.	✓ Passed	Tester 2
TC-003	Performance Testing	AI response time under 500ms	API should return blog topic suggestions quickly.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect blog topic responses.	AI-generated topics should be more accurate.	✓ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work smoothly on mobile and desktop.	✗ Failed - UI broken on mobile	Tester 2
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	🚀 Deployed	DevOps

Final Submission

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