

Hackathon Project Phases Template

Project Title:

CodeGenie using Llama

Team Name:

Silicon Sages

Team Members:

- Shaik Rafooz
 - A. Sashank
 - Nitish lingampalle
 - Mahesh Nanda
-

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI Assistant powered by Llama which would give the requested code snippets.

Key Points:

1. Problem Statement:

CodeGenie is an advanced project powered by CodeLlama, an AI model designed to streamline and enhance code generation. This innovative system simplifies the development process by providing accurate and efficient code snippets, comprehensive code explanations, and debugging support. CodeGenie can be utilized across various scenarios, offering robust solutions tailored to different developer needs

2.Solution :

CodeGenie allows developers to generate code effortlessly. For instance, a programmer can input a brief description of a desired function, and CodeGenie will provide the complete code snippet, including necessary libraries and comments. This feature enables developers to save time, reduce errors, and focus on more complex tasks by automating repetitive coding tasks.

Target Users:

- Programmers looking for automating certain parts of the code
- . Learners who are new to coding and require assistance.

3. Expected Outcome:

- a. A functional **CODE -ASSISTANT** that provides code snippets based on the request provided.
-

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the

Key Points:

1. Technical Requirements:

- Programming Language: **Python**
- Backend: **Llama**
- Frontend: **Streamlit Web Framework**

2. Functional Requirements:

- Ability to **generate code snippets** using codellama models.
- Display **the required code** with meaningful comments so user can understand

3. Constraints & Challenges:

- Generating code in the required language without errors
 - Making sure the code provided is working as intended
 - Providing a **smooth UI experience** with Streamlit.
-

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Model-1:



Model-2:



Key Points:

1. System Architecture:

- User enters the programming language code snippet required
- Query is processed using **Llama model**.
- AI model fetches and processes the data.
- The frontend displays **the code generated..**

2. User Flow:

- Step 1: User enters a query (e.g. "Programming language : Python , Task : Generate a code of printing n numbers ").
- Step 2: The backend **calls the llama** to retrieve vehicle data.
- Step 3: Next it processes the data and **displays the code** in an easy-to-read format.

3. UI/UX Considerations:

- **Minimalist, user-friendly interface** for seamless navigation.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & API Integration	● High	8 hours (Day 1)	End of Day 1	Rafooz,Sashank	Huggingface API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	● Medium	5 hours (Day 1)	End of Day 1	Mahesh Nanda, Nitish Lingampalle	API response format finalized	Basic UI with input fields
Sprint 2	Testing & UI Enhancements	● Medium	1.5 hours (Day 2)	Mid-Day 2	Mahesh ,Nitish	API response, UI layout completed	Responsive UI, better user experience
Sprint 2	Error Handling & Debugging	● High	2 hours (Day 2)	Beginning-Day 2	Rafooz, Sashank	API logs, UI inputs	Improved API stability

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

- (● High Priority) Set up the **environment** & install dependencies.
- (● High Priority) Integrate **HuggingFace API**

(🕒 Medium Priority) Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

(🕒 High Priority) Implement Llama model

(🕒 High Priority) Debug API issues & handle the model

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

(🕒 Medium Priority) Test API responses, refine UI, & fix UI bugs. (🕒 Low Priority) Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Implement core features of the CodeGenie

Key Points:

1. Technology Stack Used:

- **Frontend:** Streamlit
- **Backend:** HuggingFace API, Llama model.
- **Programming Language:** Python

2. Development Process:

- Implement **API key authentication** and **HuggingFace API integration** with **Llama model**.
- Develop **Code Generation logic**.
 - Also Create a streamlit page for easy access.

3. Challenges & Fixes:

- **Challenge:** Different models produce different results.
 - **We tried using a normal model**
- **Challenge:** API being too busy.

FIX: So we downloaded the Llama model and deployed it locally using streamlit.

- **Challenge:** Model size being too large

- **Fix:** We downloaded a smaller size model.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Python code for printing n numbers	Relevant Python code	☑ Passed	Rafooz
TC-002	Functional Testing	C code for printing n numbers	Code required for it	☑ Passed	Rafooz
TC-003	Functional Testing	Java code for implementing inheritance	Relevant code for inheritance	⚠ Needs Optimization	Sashank

Final Submission

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**