

**Hackathon Project Phases Template** for the **AutoSage App** project.

---

# **Hackathon Project Phases Template**

**Project Title:**

**Code Genie : AI-Powered Code Generation using Code Llama**

**Team Name:** STRAWHATS

**Team Leader :** Yada Varshith

**Team Members :** Rampaka Karthik

Soppari Naveen

Thanniru Mani Shashank Ramana

---

## **Phase-1: Brainstorming & Ideation**

**Objective:**

Code Genie allows developers to generate code effortlessly. For instance, a programmer can input a brief description of a desired function, and Code Genie 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.

**Key Points:**

**1. Problem Statement:**

Code Genie is an advanced project powered by Code Llama, 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. Code Genie can be utilized across various scenarios, offering robust solutions tailored to different developer needs.

**2. Proposed Solution :**

An AI-powered application using code Llama to generate code

Snippets, debug issues, and optimize existing code. The app provides context-aware suggestions and explanations.

### **3.Target Users:**

**Software developers looking to speed up coding tasks . Beginning Programmers needing assistance with syntax and logic. Teams working on rapid prototyping and development.**

### **4 .Expected Outcome:**

A functional AI-powered code generation tool that provides efficient and optimized code suggestions.

---

## **Phase-2: Requirement Analysis**

### **Objective:**

Define the technical and functional requirements for the Code Genie App.

### **Key Points:**

#### **1. Technical Requirements:**

**Programming Language : Python  
Backend : Code Llama API  
Frontend : Streamlit Web Framework  
Database : Not required initially (API-based queries)**

#### **2. Functional Requirements:**

#### **3. Constraints & Challenges:**

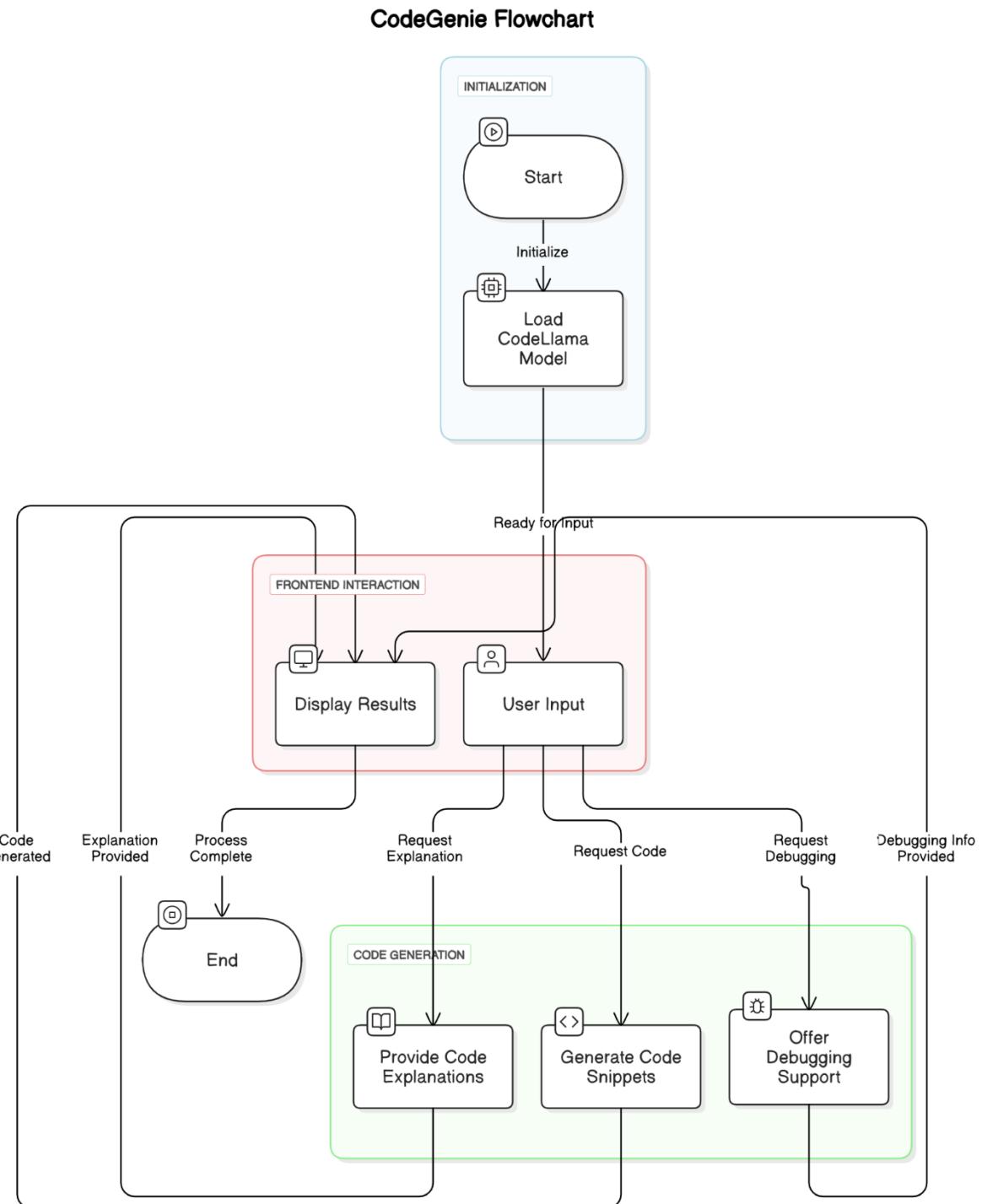
Ensuring accurate and efficient code generation. Handling API rate limits and optimizing API calls . Providing a user-friendly UI with syntax highlighting.

---

## **Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow of the application.



## **Key Points:**

1. **System Architecture:**
    - \* User inputs a programming-related query.
    - \* Query is processed using code Llama API.
    - \* AI model generates and processes the code.
    - \* The frontend displays generated code and recommendations.
  2. **User Flow:**
    - \* User enters a query (e.g., “Python function to sort a list”).
    - \* The backend calls the Code Llama API to retrieve the best code Snippet.
    - \* The app processes the response and displays the code with syntax highlighting
  3. **UI/UX Considerations:**
    - \* Clean, minimal UI for easy navigation.
    - \* Minimalist, user-friendly interface for seamless navigation.
    - \* Filters for price, mileage, and features.
    - \* Dark & light mode for better user experience.
- 

## **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	6 hours (Day 1)	End of Day 1	T.Mani shashank	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	● Medium	2 hours (Day 1)	End of Day 1	R.Karthik	API response format finalized	Basic UI with input fields
Sprint 2	Code Llama & Hugging Face	● High	3 hours (Day 2)	Mid-Day 2	Y.Varshith	API response, UI elements ready	Search functionality with filters

Sprint 2	Error Handling & Debugging	 High	1.5 hours (Day 2)	Mid-Day 2	Y.varshith & T.Mani shashank	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	 Medium	1.5 hours (Day 2)	Mid-Day 2	R.karthik & S.Naveen	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	 Low	1 hour (Day 2)	End of Day 2	Entire Team	Working prototype	Demo-ready project

## Sprint Planning with Priorities

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

- ( **High Priority**) Set up the environment & install dependencies.
- ( **High Priority**) Integrate Google Gemini API.
- ( **Medium Priority**) Build a basic UI with input fields.

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

- ( **High Priority**) Implement search & comparison functionalities.
- ( **High Priority**) Debug API issues & handle errors in queries.

### 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 AutoSage App.

### Key Points:

1. **Technology Stack Used:**

- **Frontend:** Streamlit
- **Backend:** Google Gemini Flash API
- **Programming Language:** Python

## 2. Development Process:

- Implement **API key authentication** and **Gemini API integration**.
- Develop **vehicle comparison and maintenance tips logic**.
- Optimize **search queries for performance and relevance**.

## 3. 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 that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	A user uses codegenie to get the required output	The user must be able to enter prompt	Should be improved	Naveen
TC-002	Functional Testing	Should generate text to text	Should generate relevant output	Not Fixed	Mani Shashank
TC-003	Performance Testing	API responds to the user prompt	API should return results quickly.	⚠ Needs Optimization	Varshith
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	<input checked="" type="checkbox"/> Fixed	Karthik
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	✗ Failed - UI broken on mobile	karthik
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**