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

Hackathon Project Phases Template

Project Title: A.I smart Resume genarator

AutoSage App Using Deep seek

Team Name: Micro Squad

(Provide your team’s name)

Team Members:

- Member 1 Y.Dayakar
- Member 2 R.Dinesh
- Member 3 K.Dorababu
- Member 4 J.Abhilash

Phase-1: Brainstorming & Ideation

Objective: Develop an A.I Resume smart genarator customized Resume for every Opportunity.

Key Points:

1. **Problem Statement: * Time -consuming-Manually creating a resume takes long time .**
2. **Lack of writing skills -many struggle to write impactful bullet points**

3. **Proposed Solution:**
 *Conver letter Genarator -Automatically generate a tailored cover letter that complements the resume.

* Multingual support -The ability to create resume multiple languages for global job applications

4. **Target Users:**
   ***University/collage students -Those applying for interships or their first full time job.**
   ***Freelance professionals -indipendent workers who need a resume that demonstrates.**
   ***A diverse range of project skill.**

Phase-2: Requirement Analysis

Objective:In requirement analysis, the objective is to clearly define and understand the needs, goals, and expectation Of stakeholders to ensure that the final product meets those criteria.

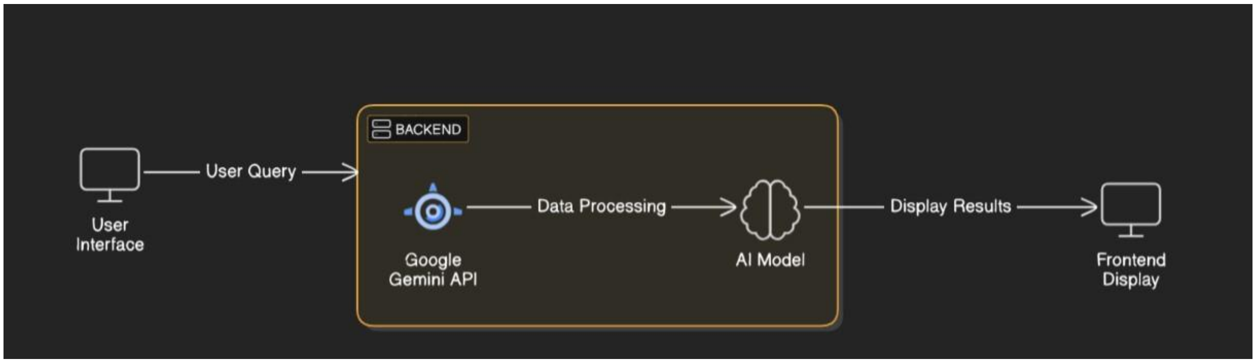
Key Points:

1. **Technical Requirements:**
   - ○ Programming Language: **HTML**
   - ○ Backend: **Deep seek**
   - ○ Frontend: **chatgpt**
   - ○ Database: **Not required initially (API-based queries)**
2. **Functional Requirements: Display specifications review and comparisons in an intuitive UI.**
3. **Allow users to search eco -friendly AI smart Resume genarator.**
4. **Constraints & Challenges:**
   - ○ Ensuring real-time updates from **Deep seek**
   - ○ Handling **API rate limits** and optimizing API calls.
   - ○ Providing a **smooth and user-friendly experience.**.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
   - User enters vehicle-related query via UI.
   - Query is processed using **Deep seek**.
   - AI model fetches and processes the data.
   - The frontend displays Mail ID and password

   User Flow:

   → Step 1: Start page/Landing page. Step 2: signup/Login registered account
   - Step 3: The app processes the data and **displays results** in an easy-to-read format.
2. **UI/UX Considerations:**
   - **Minimalist, user-friendly interface** for seamless navigation.
   - **Free of cost and open-source**
   - **Dark & light mode**

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	Dayakar	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	○ Medium	2 hours (Day 1)	End of Day 1	Dinesh	API response format finalized	Basic UI with input fields
Sprint 2	Vehicle Search & Comparison	● High	3 hours (Day 2)	Mid-Day 2	Dora Babu	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	● High	1.5 hours (Day 2)	Mid-Day 2	Dayakar and J Abhilash	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	○ Medium	1.5 hours (Day 2)	Mid-Day 2	Dinesh and Dora Babur	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:** chatgpt
   ○ **Backend:** Deep seek
   ○ **Programming Language:** HTML

2. **Development Process:**

   ○ Implement **API key authentication** and **with Deep seek**
   ○ 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	Query "Best budget cars under ₹10 lakh"	Relevant budget cars should be displayed.	☑ Passed	Tester 1
TC-002	Functional Testing	Query "Motorcycle maintenance tips for winter"	Seasonal tips should be provided.	☑ Passed	Tester 2

TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & 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**