

HackathonProjectPhasesTemplatefortheAutoSageAppproject.

HackathonProjectPhasesTemplate

ProjectTitle:AI Powered PDF Knowledge Assistant using Google PALM

AutoSageAppUsingGeminiFlash

TeamName: Warriors

(Provideyourteam'sname)

TeamMembers:

- K.Sai Prasanna
 - K.Karthika
 - K.Nireesha
 - K.Sri Sai Harini
-

Phase-1:Brainstorming&Ideation

Objective:

DevelopanAI-poweredvehicleexperttoolusingGeminiFlashtohelpuserscompareand analyze vehicle specifications, reviews, and eco-friendly options.

KeyPoints:

1. ProblemStatement:

- Manyusersstruggletofindreliable,up-to-dateinformationabouttwo-wheelers and four-wheelers before making a purchase decision.
- Usersalsoneedguidanceonvehiclemaintenanceandeco-friendlyvehicle choices.

2. **ProposedSolution:**

- AnAI-poweredapplicationusing**GeminiFlash**toprovidereal-timevehicle**specifications, reviews, and comparisons.**
- Theappoffers**maintenancetips**and**eco-friendlyvehicleinsights**basedon user preferences.

3. **TargetUsers:**

- **Vehiclebuyers**lookingforspecificationsandcomparisons.
- **Vehicleowners**needingseasonalmaintenancetips.
- **Eco-consciousconsumers**searchingforhybridandelectricvehicleoptions.

4. **ExpectedOutcome:**

- Afunctional**AI-poweredvehicleinformationapp**thatprovidesinsightsbased on real-time data and user queries.
-

Phase-2:RequirementAnalysis

Objective:

DefinethetechnicalandfunctionalrequirementsfortheAutoSageApp.

KeyPoints:

1. **TechnicalRequirements:**

- ProgrammingLanguage:**Python**
- Backend:**GoogleGeminiFlashAPI**
- Frontend:**StreamlitWebFramework**
- Database:**Notrequiredinitially(API-basedqueries)**

2. **FunctionalRequirements:**

- Abilityto**fetchvehicledetails**usingGeminiFlashAPI.
- Displays**specifications, reviews, and comparisons**inanintuitiveUI.
- Providereal-time**vehiclemaintenancetips**basedonseasons.
- Allowusersto**searcheco-friendlyvehicles**basedonemissionsandincentives.

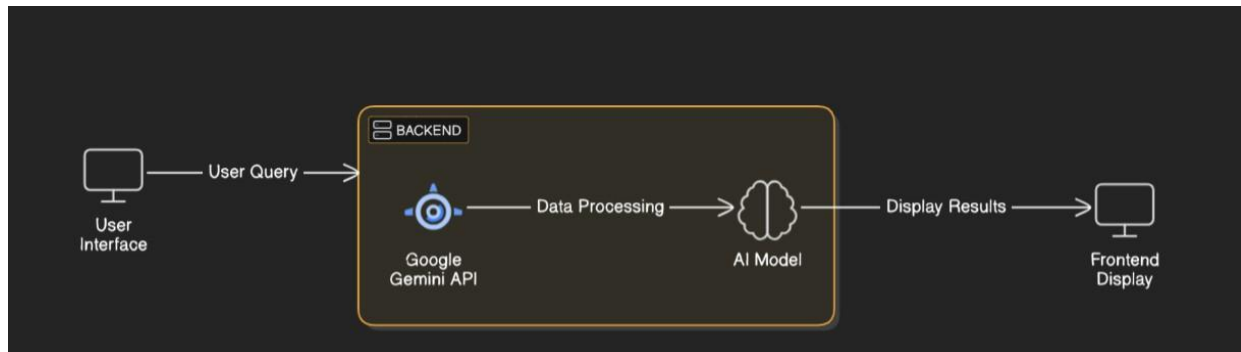
3. **Constraints&Challenges:**

- Ensuringreal-timeupdatesfrom**GeminiAPI.**
- Handling**APIratelimits**andoptimizingAPIcalls.
- Providinga**smoothUIexperience**withStreamlit.

Phase-3:ProjectDesign

Objective:

Develop the architecture and user flow of the application.



KeyPoints:

1. SystemArchitecture:

- User enters vehicle-related query via UI.
- Query is processed using **Google Gemini API**.
- AI model fetches and processes the data.
- The frontend displays **vehicle details, reviews, and comparisons**.

2. UserFlow:

- Step 1: User enters a query (e.g., "Best motor cycles under ₹1 lakh").
- Step 2: The backend **calls the Gemini Flash API** to retrieve vehicle data.
- Step 3: The app processes the data and **displays results** in an easy-to-read format.

3. UI/UX Considerations:

- **Minimalist, user-friendly interface** for seamless navigation.
 - **Filters for price, mileage, and features.**
 - **Dark & light mode** for better user experience.
-

Phase-4:ProjectPlanning(AgileMethodologies)

Objective:

Breakdown 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	Member 1	Google API Key, Python, Streamlit setup	API connection established & working
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
Sprint 2	Vehicle Search & Comparison	High	3 hours (Day 2)	Mid-Day 2	Member 1 & 2	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	High	1.5 hours (Day 2)	Mid-Day 2	Member 1 & 4	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	Medium	1.5 hours (Day 2)	Mid-Day 2	Member 2 & 3	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

Sprint1–Setup&Integration(Day1)

(High Priority) Setup the environment & install dependencies.

(High Priority) Integrate Google Gemini API.

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

Sprint2–CoreFeatures&Debugging(Day2)

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

Sprint3–Testing,Enhancements&Submission(Day2)

(Medium Priority) Test API responses, refine UI, & fix UI bugs.

(Low Priority) Final demo preparation & deployment.

Phase-5:ProjectDevelopment

Objective:

ImplementcorefeaturesoftheAutoSageApp.

KeyPoints:

- 1. **TechnologyStackUsed:**
 - **Frontend:**Streamlit
 - **Backend:**GoogleGeminiFlashAPI
 - **ProgrammingLanguage:**Python
- 2. **DevelopmentProcess:**
 - Implement**APIkeyauthentication**and**GeminiAPIintegration**.
 - Develop**vehiclecomparisonandmaintenancetipslogic**.
 - Optimize**searchqueriesforperformanceandrelevance**.
- 3. **Challenges&Fixes:**
 - **Challenge:**DelayedAPIresponsetimes.
Fix:Implement**caching**to store frequently queried results.
 - **Challenge:**LimitedAPIcallsperminute.
Fix:Optimizequeries to fetch **only necessary data**.

Phase-6:Functional&PerformanceTesting

Objective:

EnsurethattheAutoSageAppworksasexpected.

Test CaseID	Category	TestScenario	ExpectedOutcome	Status	Tester
TC-001	Functional Testing	Query"Bestbudgetcars under ₹10 lakh"	Relevantbudgetcars shouldbedisplayed.	✔Passed	Tester1
TC-002	Functional Testing	Query "Motorcycle maintenancetipsfor winter"	Seasonaltipsshould be provided.	✔Passed	Tester2

TC-003	Performance Testing	APIresponsetimeunder 500ms	APIshouldreturn results quickly.	⚠ Needs Optimization	Tester3
TC-004	Bug Fixes & Improvements	FixedincorrectAPI responses.	Dataaccuracyshould be improved.	✓ Fixed	Developer
TC-005	Final Validation	EnsureUIisresponsive across devices.	UIshouldworkon mobile&desktop.	✗ Failed - UI brokenonmobile	Tester2
TC-006	Deployment Testing	Hosttheappusing Streamlit Sharing	App should be accessibleonline.	📦 Deployed	DevOps

FinalSubmission

1. **ProjectReportBasedonthetemplates**
2. **DemoVideo(3-5Minutes)**
3. **GitHub/CodeRepositoryLink**
4. **Presentation**