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

Hackathon Project Phases Template

Project Title:

AI Study Planner

Team Name:

Hustlers

Team Members:

- B.Sathish Kumar
- A.Vijay
- K.Vishnu Vardhan
- G.Shiva Kumar
- S.Dilip Kumar

Phase-1: Brainstorming & Ideation

Objective:

An AI study planner can enhance academic success by creating personalized study paths tailored to individual learning styles and schedules. It can incorporate features like subject-specific study schedules, progress tracking, and adaptive learning techniques to optimize study efficiency.

Key Points:

1. **Problem Statement:**
    - The AI study planner should provide personalized, adaptive study schedules tailored to individual learning styles and academic goals.
    - It should offer a user-friendly interface, collaboration options, and ensure data privacy for a seamless user experience.

2. **Proposed Solution:**
    - The proposed AI study planner will generate personalized study plans using adaptive algorithms that adjust based on user performance and feedback.
    - It will feature progress tracking, resource recommendations, and time management tools, all within an intuitive interface.
3. **Target Users:**
    - The target users for the AI study planner include students at various educational levels, from high school to university, seeking to enhance their study efficiency and academic performance.
4. **Expected Outcome:**
    - The expected outcome of the AI study planner is improved academic performance and study efficiency for students through personalized and adaptive learning experiences

---

Phase-2: Requirement Analysis

Objective:

- Requirement analysis is a critical phase in the development of the AI Study Planner.

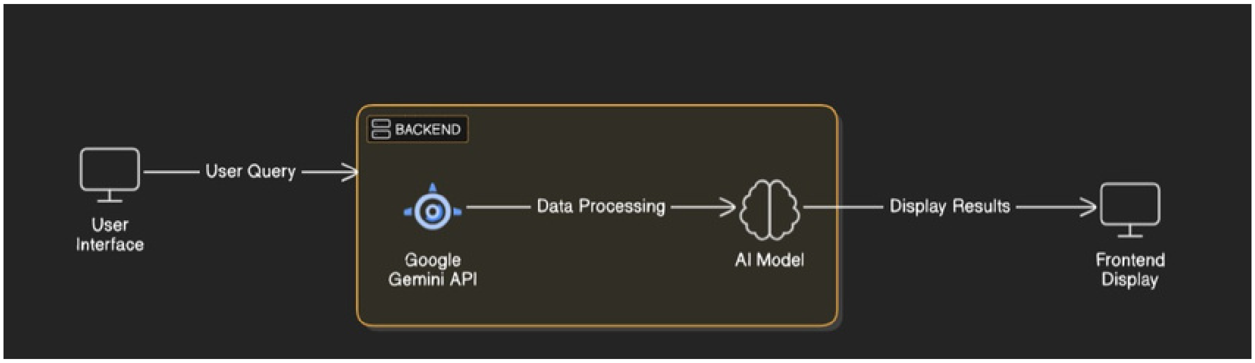
Key Points:

1. **Technical Requirements:**
    - The technical requirements for the AI Study Planner focus on the system's architecture, tools, technologies, and infrastructure needed to build a functional, scalable, and secure application.
    - Main requirement are Frontend, Backend, Database, AI/ML Components, Cloud Infrastructure ,...etc
2. **Functional Requirements:**
    - The planner should create customized study schedules based on each user’s needs and goals.
    - The planner should recommend helpful study materials like videos and articles.
3. **Constraints & Challenges:**
    - Developing an AI Study Planner involves addressing several constraints and challenges that can impact the design, development, and deployment of the system. Below is a detailed analysis of these constraints and challenges:

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
   - Client-Server Model: The application will follow a client-server architecture where the client (user interface) communicates with the server (backend) to process requests and manage data.

2. **User Interface Design:**
   - Dashboard: A central hub where users can view their study plans, progress, and upcoming tasks
   - progress bars and visual indicators for completed tasks.
   - Quick access to study materials and resources.
   - Notifications and reminders for upcoming deadlines.

3. **UI/UX Considerations:**
   - The success of the AI Study Planner heavily depends on its user interface (UI) and user experience (UX) design.
   - A well-designed UI/UX ensures that the application is intuitive, engaging, and accessible to all users.
   - 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	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

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	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**