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



Hackathon Project Phases Template

# Project Title:

**StudBud : AI Study Planner**

# Team Name:

LIMIT BREAKERS

# Team Members:

* PEDURI VENKATA SAIRAM
* KATAMONI VIJAY KUMAR



# Phase-1: Brainstorming & Ideation

## Objective:

AI-powered platform that creates personalized, adaptive study plans to help students optimize learning.

## Key Points:

1. **Problem Statement:**
   * Students often struggle with **time management, discipline, and finding the right study resources**.
   * Generic study plans fail to accommodate **individual learning speeds, subject difficulties, and personal schedules**.
2. **Proposed Solution:**
   * **Gemini Flash API** processes input and structures a study schedule**.**
   * AI adapts plans based on task completion and student feedback.
3. **Target Users:**
   * **Students (School & College)** Managing semester schedules, assignments, projects.
   * **Competitive Exam Aspirants** preparing for **Government Job Exams** – UPSC, SSC, Banking, RRB, etc.
   * **Working Professionals** – Balancing **skill development with jobs** (Python, AI, Web Dev).
4. **Expected Outcome:**
   * StudBud aims to **revolutionize study planning** by providing **AI-powered personalized study schedules** that help students optimize their learning experience efficiently.



# Phase-2: Requirement Analysis

## Objective:

**AI Study Plan** is an AI-powered platform that creates personalized, adaptive study plans to help students optimize learning.

## Key Points:

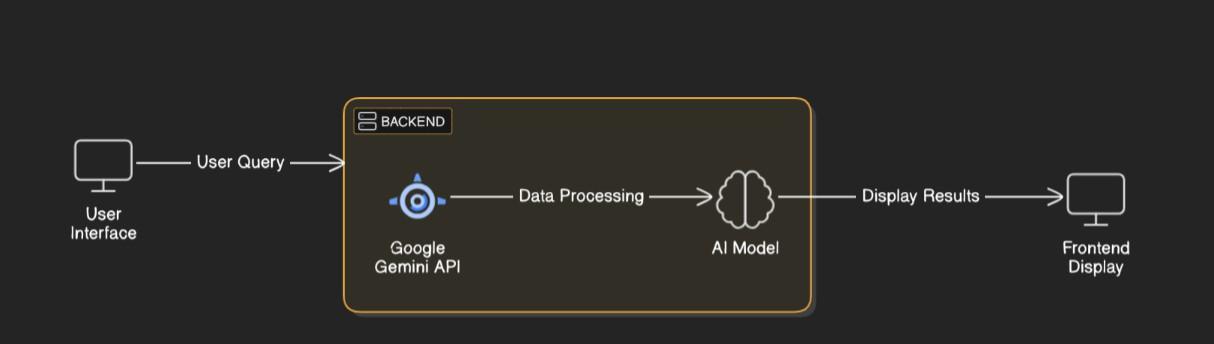
1. **Technical Requirements:**
   * Backend: **Google Gemini Flash API**
   * Frontend: **HTML, CSS, JavaScript, React**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   * Fetch personalized study schedules using Gemini Flash API.
   * Recommend learning resources such as books, videos, and online courses.
   * Allow users to track study progress and receive AI-driven feedback.
3. **Constraints & Challenges:**
   * Ensuring real-time study plan generation using Gemini Flash API.
   * Handling API rate limits and optimizing calls for efficient performance.
   * Integrating third-party APIs for additional resources (books, courses, quizzes).



# Phase-3: Project Design

## Objective:

Develop the architecture and user flow of the application.



## Key Points:

1. **System Architecture:**
   * User enters subjects and query via UI.
   * Query is processed using **Google Gemini API**.
   * AI model fetches and processes the data.
   * The frontend displays **schedules and timetable to prepare**.
2. **User Flow:**
   * Step 1: User enters a query (e.g., "FrontEnd within 6 months").
   * Step 2: The backend **calls the Gemini Flash API** to retrieve schedules and tasks with resources.
   * 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 duration, subjects, 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 | Member 1 | Google API Key, React, Node | 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 | Integration | 🔴 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&2 | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Member 1&2 | 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:** React
   * **Backend:** Google Gemini Flash API
   * **Programming Language:** JavaScript
2. **Development Process:**
   * Implement **API key authentication** and **Gemini API integration**.
   * Develop **schedules for the subjects provided in the particular duration**.
   * 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.

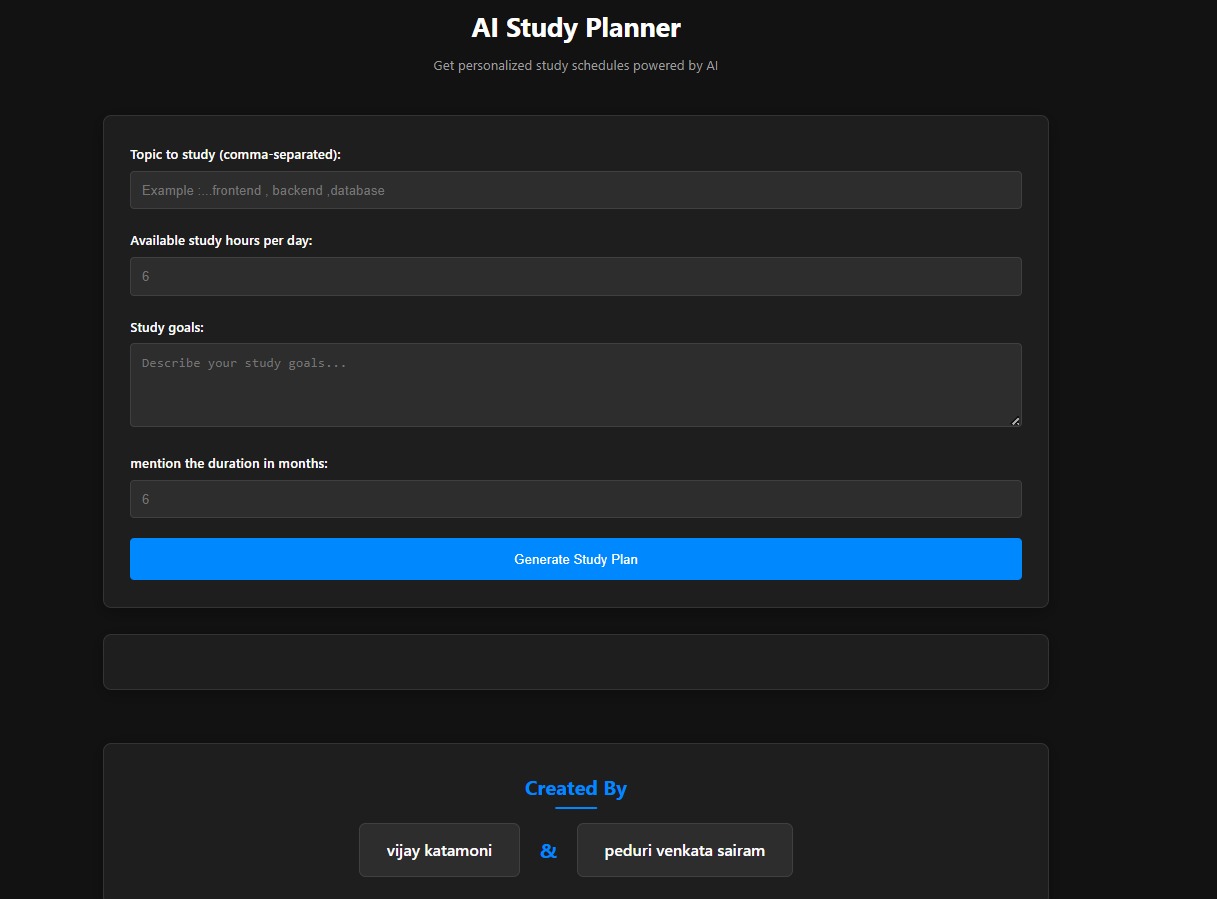
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| TC-001 | Functional Testing | Query "FrontEnd" | Tasks and Syllabus scheduled. | ✅ Passed | Tester 1 |
| TC-002 | Functional Testing | Query "Backend" | Provides Resources and practice arenas | ✅ 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 | Develop er |
| 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 | App should be accessible online. | 🚀 Deployed | DevOps |



# Final Submission

**UI :**

****

**github link :** <https://github.com/sairam-peduri/AI-StudBud>