**Hackathon Project Phases Template**  for the **StudBud AI study panner**

**Hackathon Project Phases Template**

**Project Title:**

StudBud AI study panner

**Team Name:**

Studybuddies

**Team Members:**

* M.Vasundhara
* M.Poojitha
* P.Tejaswini
* Fakiha Tabassum

**Phase-1: Brainstorming & Ideation**

**Objective:**

StudBud AI aims to revolutionize the way students learn by leveraging Generative AI to create personalized, adaptive, and efficient study plans. By incorporating AI-driven scheduling, intelligent content generation, and real-time tutoring, StudBud AI enhances learning outcomes, improves time management, and makes studying more engaging and effective.

**Key Points:**

1. **Problem Statement:**

Students struggle with inefficient study habits, lack of personalized learning, and poor time management.  
Traditional study methods do not adapt to individual strengths, weaknesses, or learning preferences.  
Limited access to real-time tutoring and structured study materials reduces engagement and retention.

1. **Proposed Solution:**

An AI-powered platform that creates personalized, adaptive study plans based on student progress and needs.  
Smart AI scheduling optimizes study sessions around deadlines, exams, andlearning capacity.  
AI generates quizzes, summaries, and flashcards while providing real-time tutoring and motivation tracking.

1. **Target Users:**

Students in high school, college, and competitive exams who need structured and efficient study plans.  
Self-learners and lifelong learners looking for AI-generated, interactive learning materials.  
Educators, institutions, and neurodiverse learners who benefit from customized AI learning experiences.

1. **Expected Outcome:**Enhanced **learning efficiency** through personalized AI-driven study plans and adaptive content.  
   Improved **time management** with AI-optimized schedules and real-time progress tracking.  
   Increased **student engagement** through interactive content, gamification, and 24/7 AI tutoring.

**Phase-2: Requirement Analysis**

**Objective:**

Define technical, functional, and operational needs to ensure StudBud AI’s scalability, personalization, and compliance.

**Key Points:**

1. **Technical Requirements:**

AI-driven personalized study plans based on real-time student progress.

Scalable cloud infrastructure to handle high user traffic efficiently.  
 AI-powered tutoring & content generation using NLP, GPT-4, and multimodal learning.  
 Secure data storage & privacy compliance (GDPR, FERPA).

1. **Functional Requirements:**

Dynamic study schedules based on deadlines, exams, and learning habits.  
 Interactive AI tutoring via chatbot, voice, and video support.  
 Engagement tracking & adaptive recommendations for optimized learning.  
 Gamification elements like badges, leaderboards, and rewards.

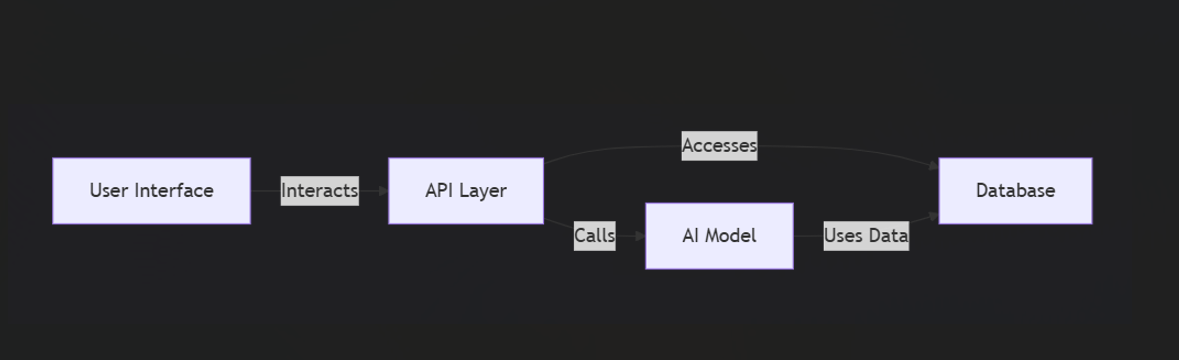
1. **Constraints & Challenges:**

Ensuring real-time AI adaptability for seamless learning experiences.  
 Maintaining accuracy in personalization across diverse learning styles.  
 Managing high user loads while ensuring smooth system performance.  
 Adhering to data privacy laws while collecting and processing user data.

**Phase-3: Project Design**

**Objective:**

Develop the architecture and user flow of the application.



**Key Points:**

1. **System Architecture:**

User Interface: The user enters a study-related query via the UI (e.g., "Create a study plan for NEET in 3 months").

API Layer:The request is sent to the backend, which calls the Google Gemini API.

AI Model:The AI processes the query and generates a structured study plan.

It personalizes schedules based on user preferences, available time, and difficulty levels.

Database:The AI model accesses a database of study materials, past papers, and subject-specific recommendations.

User Experience:The generated study plan is displayed in an intuitive, easy-to-follow format.Users can track progress, set reminders, and get AI-generated study tips

1. **User Flow:**

Step 1: User enters a study-related query (e.g., "Best strategy to cover Physics in 2 weeks").

Step 2: The backend calls the Gemini API to generate a custom study plan.

Step 3: The app displays the study plan, suggesting time slots, breaks, and key resources.

1. **UI/UX Considerations:**

Minimalist & Intuitive UI – Simple navigation for a stress-free experience.

Customizable Filters – Users can adjust based on time, subject difficulty, or exam date.

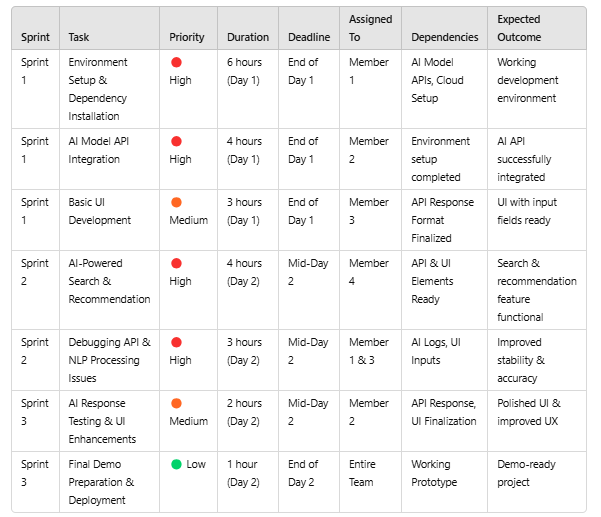
Light & Dark Mode – Ensures comfortable reading for extended study sessions.

Progress Tracking – Interactive checklists and completion indicators.

**Phase-4: Project Planning (Agile Methodologies)**

**Objective:**

Break down development tasks for efficient completion.



**Sprint Planning with Priorities**

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

(🔴 High Priority) Set up the environment & install dependencies.

(🔴 High Priority) Integrate AI Model APIs (e.g., GPT-4, Llama).

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

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

 (🔴 High Priority) Implement AI-powered search & recommendation functionalities.

(🔴 High Priority) Debug API issues & handle errors in NLP processing.

**Sprint 3 – Testing, Enhancements & Submission (Day 2)**

**(**🟠 Medium Priority) Test AI responses, refine UI, & fix UI bugs.

(🟢 Low Priority) Final demo preparation & deployment.

**Phase-5: Project Development**

**Objective:**

Develop and integrate core functionalities of StudBud AI.

**Key Points:**

1. **Technology Stack Used:**

* Frontend: Python, Stream-lit
* Backend: FastAPI, OpenAI API, LangChain
* Database: MySQL, (Oracle)

1. **Development Process:**

* Implement AI-driven personalized study plan generation.
* Develop AI chatbot for instant query resolution.
* Integrate sentiment analysis for engagement tracking.
* Optimize AI model to adapt study plans dynamically.

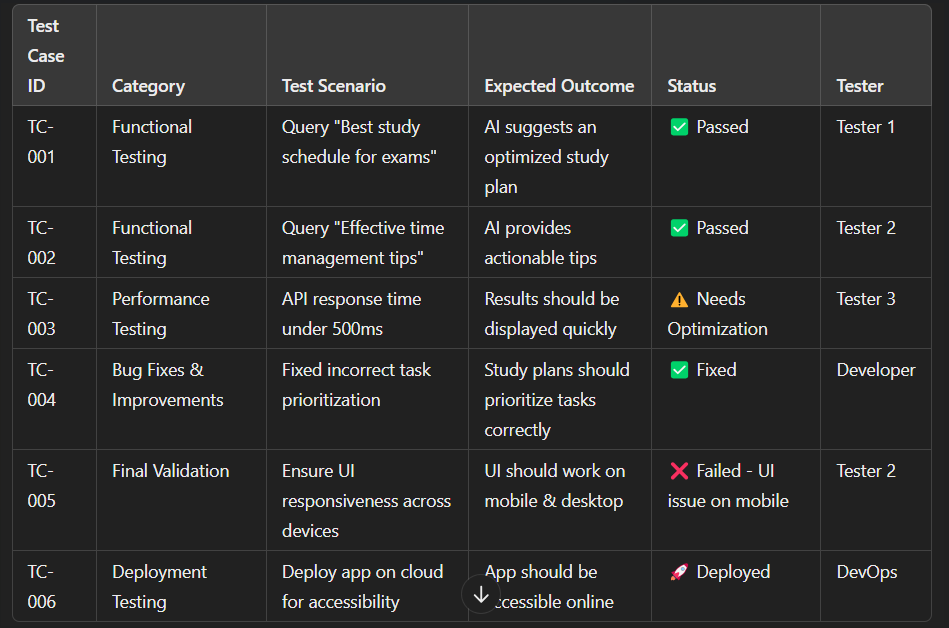
1. **Challenges & Fixes:**

* Challenge: AI response time optimization.
* Fix: Implement model caching and request batching.
* Challenge: Ensuring high personalization accuracy.
* Fix: Fine-tune AI models based on real user feedback.
* Challenge: Scaling AI services efficiently.
* Fix: Use distributed computing with Kubernetes to handle loads.

**Phase-6: Functional & Performance Testing**

**Objective:**

Ensure that the AutoSage App works as expected.



**Final Submission**

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**