Hackathon Project Phases Template for the **AI StudBud App** project.

Hackathon Project Phases Template

Project Title: AI StudBud

AI StudBud App Using BERT(Bidirectional Encoder Representations from Transformers)

Team Name:

Hackstreet Girls

Team Members:

Haleema Khatun

- Kasarla Lakshmi Prasanna
- Akkala Varshitha
- Palle Rani
- Medaboina Bhargavi

Phase-1: Brainstorming & Ideation

Objective: Develop an AI-powered application that helps students **schedule their subjects** effectively based on their strengths and weaknesses, optimizing study time for better academic performance.

Key Points:

1. Problem Statement:

- Many students struggle with **time management** and **subject prioritization** while preparing for exams.
- Generic study plans do not account for **individual strengths and weaknesses**, leading to inefficient learning.
- Lack of **personalized guidance** results in poor academic performance and stress.

2. Proposed Solution:

- An **AI-driven study planner** that dynamically adjusts schedules based on **students' subject proficiency and learning patterns**.
- Provides **personalized study plans** that allocate more time to weaker subjects while reinforcing strengths.

- **Tracks progress** and suggests improvements based on user performance and feedback.

3. Target Users:

- **School and college students** preparing for exams.
- **Competitive exam aspirants** needing an optimized study schedule.
- **Students balancing multiple subjects** and needing better time management.

4. Expected Outcome:

- A functional AI-powered **smart study planner** that helps students **manage their study schedules efficiently**.
- Improved learning outcomes by focusing on **individual needs** and adapting over time.
- **Reduced stress** through structured and strategic study plans.

Phase-2: Requirement Analysis

Objective:

Define the **technical and functional requirements** for the AI-powered **student study planner** application.

---

1. Technical Requirements:

- **Programming Language:** Python
- **Backend:** Google Gemini Flash API
- **Frontend:** Streamlit Web Framework
- **Database:** Not required initially (API-based queries)

---

2. Functional Requirements:

- Ability to **analyze student strengths and weaknesses** based on input data.
- Generate **personalized study schedules** that adapt dynamically.
- Provide **subject-wise progress tracking** and improvement suggestions.
- Display **optimized study plans** in an intuitive UI.
- Offer **AI-driven recommendations** for better time management.

---

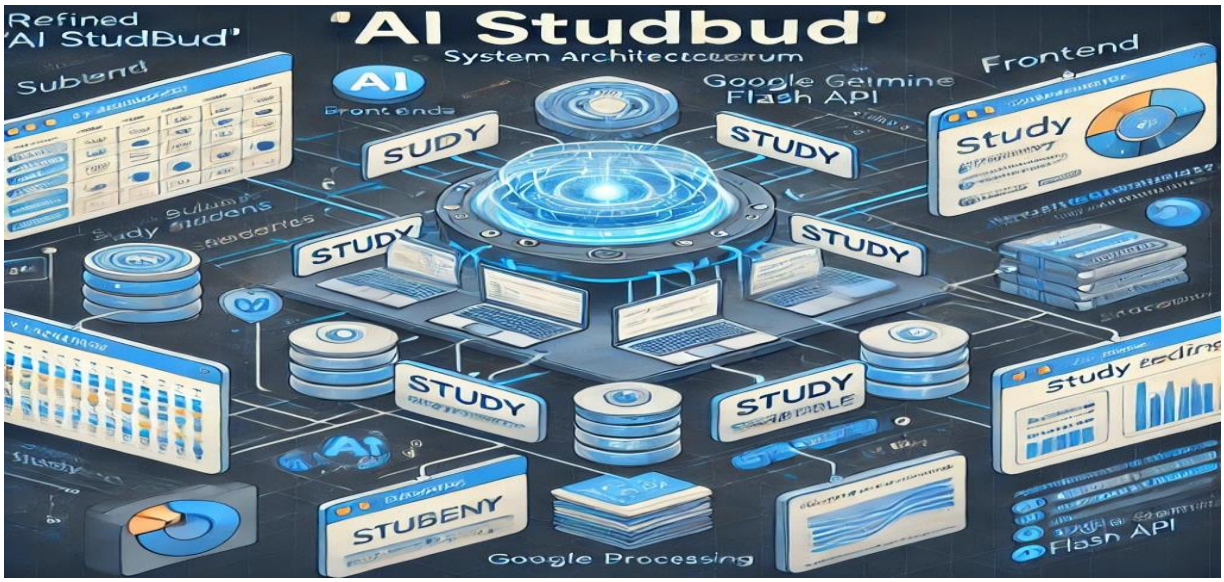
3. Constraints & Challenges:

- Ensuring **real-time AI analysis** of student performance.
- Handling **API rate limits** while optimizing AI responses.
- Maintaining a **smooth and interactive UI** using Streamlit.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

---

1. System Architecture:

- **User enters study-related query** via the UI.
- **Query is processed** using Google Gemini Flash API.
- **AI model analyzes** user strengths, weaknesses, and study patterns.
- **The frontend displays** a personalized study schedule with subject prioritization.

---

2. User Flow:

- **Step 1:** User inputs their subjects, study hours, and confidence levels.
- **Step 2:** The backend calls the **Gemini Flash API** to analyze the input.
- **Step 3:** The app **generates and displays** an optimized study schedule.
- **Step 4:** Users can **track progress** and adjust schedules dynamically.

3. UI/UX Considerations:

- **Minimalist, user-friendly interface** for easy navigation.
- **Drag-and-drop scheduling** for flexibility.
- **Progress visualization** (graphs, study streaks).
- **Dark & light mode** for better accessibility.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks using **Agile methodology** for efficient completion.

Sprint Breakdown

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	Haleema Khatun	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	Medium	2 hours (Day 1)	End of Day 1	Lakshmi Prasanna	API response format finalized	Basic UI with input fields
Sprint 2	Study Plan Generation	High	3 hours (Day 2)	Mid-Day 2	Varshitha	API response, UI elements ready	Personalized study plan functionality
Sprint 2	Error Handling & Debugging	High	1.5 hours (Day 2)	Mid-Day 2	Bhargavi	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	Medium	1.5 hours (Day 2)	Mid-Day 2	Rani	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 **study plan generation** based on user strengths & weaknesses.
✅ **High Priority:** Debug API issues & handle errors in user 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 AI Studbud App to help students schedule subjects based on their strengths and weaknesses.

Key Points:

1. Technology Stack Used:

- **Frontend:** Streamlit
- **Backend:** Google Gemini Flash API
- **Programming Language:** Python

2. Development Process:

- **Implement User Authentication:** Secure login and data storage for students' academic records.
- **Integrate Gemini API for Analysis:** Analyze students' strengths and weaknesses based on input (grades, preferences, and past performance).
- **Develop Smart Scheduling Algorithm:**
  - Prioritize weak subjects while balancing overall workload.
  - Allocate study hours efficiently based on importance and exam dates.
- **Optimize UI for Better Experience:**
  - Simple, interactive dashboard to show study plans.
  - Customization options for students to adjust schedules.

3. Challenges & Fixes:

- **Challenge:** High API processing time for large student datasets.
  - **Fix:** Implement caching for repeated queries to reduce processing time.
- **Challenge:** Generating personalized yet flexible study schedules.
  - **Fix:** Use dynamic weight-based scheduling, allowing user adjustments.
- **Challenge:** Ensuring engagement and usability for students.
  - **Fix:** Add notifications/reminders and allow users to track progress.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the **AI Studbud** application functions correctly, provides accurate subject scheduling, and performs well across different scenarios.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "Generate a study plan for upcoming exams"	A well-structured study schedule should be generated.	✅ Passed	Tester 1
TC-002	Functional Testing	Input academic strengths & weaknesses	Subjects should be prioritized based on weak and strong areas.	✅ Passed	Tester 2
TC-003	Performance Testing	API response time under 500ms	Study plan generation should be fast.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect scheduling logic	The schedule should correctly align with the student's input.	✅ Fixed	Developer
TC-005	UI Testing	Ensure UI is responsive across devices	App should work smoothly on mobile and desktop.	❌ Failed - UI layout issues on mobile	Tester 4
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	❖ Deployed	DevOps