Hackathon Project Phases Template for the AI Study Planer project.

Hackathon Project Phases Template

Project Title:

StudBud : AI Study Planer

Team Name:

AI Visionaries

Team Members:

- **K. Meghana**
- **B. Praneetha**
- **G. Ravali**
- **S. Sirisha**

Phase-1: Brainstorming & Ideation

Objective:

Studbud aims to develop an AI-powered personalized study planner using **BERT** to generate optimized study schedules based on students' **goals, strengths, weaknesses, and preferences**. The system will intelligently adapt and refine study plans to maximize learning efficiency. Our goal is to enhance student productivity and help them achieve their academic targets effectively.

Key Points:

1. **Problem Statement:**
    - Students struggle to create personalized and efficient study plans that align with their goals, strengths, and weaknesses.
    - **Studbud** leverages **AI and BERT** to generate optimized study schedules, enhancing learning efficiency and academic performance.

2. **Proposed Solution:**
    - Develop **Studbud**, an AI-powered study planner using **BERT** to create personalized study schedules based on students' goals, strengths, and weaknesses.
    - Implement an adaptive system that **dynamically adjusts study plans** based on student progress, ensuring optimal learning efficiency.
3. **Target Users:**
    - **Students** of all levels (school, college, competitive exam aspirants) who need personalized and efficient study plans.
    - **Educators & Mentors** looking for AI-driven tools to help guide students with customized study strategies.
4. **Expected Outcome:**
    - Students will receive **personalized, AI-driven study plans** that improve time management, enhance learning efficiency, and boost academic performance.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the **Studbud: AI Personalized Study Planner** to ensure efficient study plan generation and adaptation.

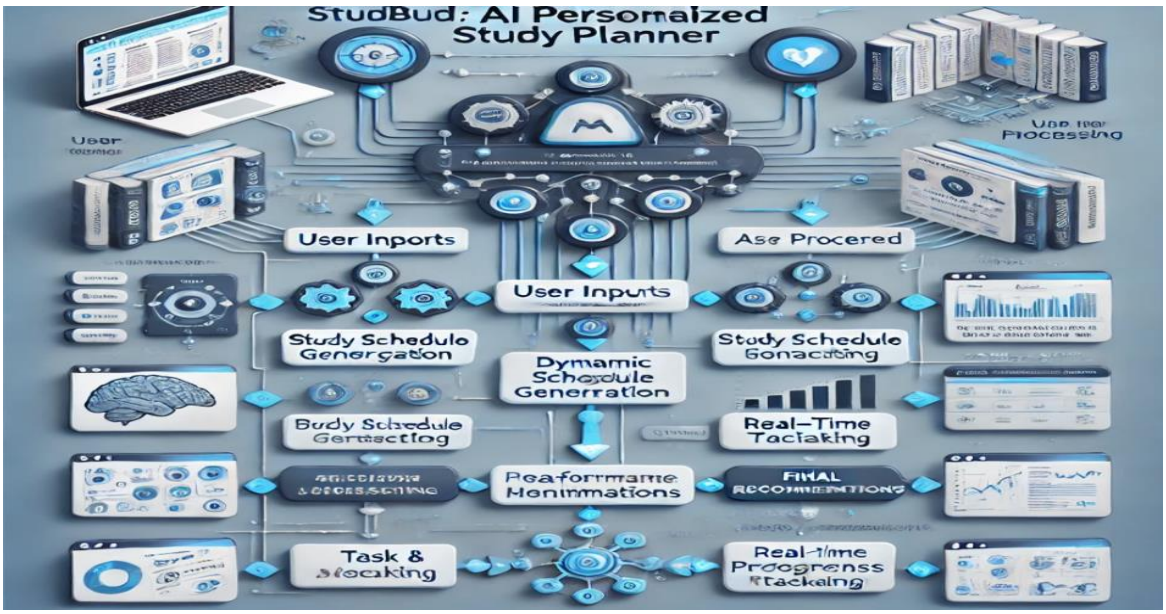
Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Backend: BERT-based NLP Model & AI-driven Recommendation System
    - Frontend: **Streamlit Web Framework, GUI**
    - DatabaseFirebase/Cloud Storage for user data & study plans
2. **Functional Requirements:**
    - Generate **personalized study plans** based on user input (goals, strengths, weaknesses).
    - Adapt study schedules dynamically based on **progress and feedback**.
    - Provide **AI-driven study recommendations** and time management insights.
    - Integrate a **notification system** (WhatsApp/Telegram) for study reminders.
3. **Constraints & Challenges:**
    - Ensuring accurate **NLP-based goal classification** for effective plan generation.
    - Handling **scalability & performance** while processing multiple study plans.
    - Creating a **user-friendly and engaging UI** for an intuitive experience.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - User inputs **study-related queries** (e.g., subjects, goals, weaknesses) via the UI.
    - Query is processed using **BERT-based AI model** for personalized study plan generation.
    - AI model analyzes user input and **creates an optimized study schedule**.
    - The frontend displays the **customized study plan** with recommendations.
2. **User Flow:**
    - **Step 1:** User enters study preferences (e.g., "Prepare for exams in 30 days").
    - **Step 2:** Backend AI processes the input and **generates a structured study plan**.
    - **Step 3:** The system displays the plan, allowing users to **track progress** and adjust schedules.
3. **UI/UX Considerations:**
    - **Simple, intuitive interface** for easy study plan management.
    - **Customizable study preferences** (subjects, time allocation, difficulty levels).
    - **Dark & light mode** for comfortable study sessions at any time.

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 & Model Integration	🔴 High	6 hours (Day 1)	End of Day 1	Member 1	Python, BERT Model, API setup	AI model integrated & working
Sprint 1	Basic UI & Input System	🟡 Medium	2 hours (Day 1)	End of Day 1	Member 2	API response structure finalized	Basic UI with user input fields
Sprint 2	Study Plan Generation Logic	🔴 High	3 hours (Day 2)	Mid-Day 2	Member 1 & 2	AI model response, UI ready	Study plan generation with insights
Sprint 2	Error Handling & Performance Optimization	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Member 3 & 4	API logs, user inputs	Improved system stability & speed
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Member 2 & 3	Model output, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	End of Day 2	Entire Team	Fully functional 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 BERT Model for personalized study plans.
🟡 **Medium Priority** – Build a basic UI with input fields.

Sprint 2 – Core Features & Debugging (Day 2)

🔴 **High Priority** – Implement study plan generation and optimization logic.
🔴 **High Priority** – Debug AI responses & handle errors in user queries.

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

🟡 **Medium Priority** – Test AI predictions, refine UI, & fix UI bugs.
🟢 **Low Priority** – Final demo preparation & deployment. 🚀

Phase-5: Project Development

Objective:

Implement core features of the **Studbud: AI Personalized Study Planner**.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** BERT-based AI Model & Recommendation System
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **user authentication** and study preferences input system.
    - Develop **AI-powered study plan generation** and adaptation logic.
    - Optimize **study recommendations** based on user progress and feedback.
3. **Challenges & Fixes:**
    - **Challenge:** Delayed response in generating study plans.

      Fix: Implement **caching & precomputed suggestions** for faster results.
    - **Challenge:** Handling **large-scale user requests** efficiently.

      Fix: Optimize **backend processing & API calls** for scalability.

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**
   https://github.com/23wh1a12c1/AI-Visionaries
4. **Presentation**