# **Hackathon Project Phases Template**

## **Project Title:**

StudBud: AI Study Planner

## **Team Name:**

Team Rookies

## **Team Members:**

* U.Hasini
* D.Harini
* D.Meenakshi
* A.Tejashwini
* G.Vaishnavi

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

Develop an AI-powered study planner that generates personalized study schedules using free Generative AI models.

### **Key Points:**

**Problem Statement:**

* Students struggle with managing their study time effectively.
* Traditional planners lack adaptability based on subject difficulty and exam schedules.

**Proposed Solution:**

* AI-powered study planner using Gemini Flash to create customized study schedules.
* Recommends optimized learning hours based on user inputs (subjects, available time, difficulty level).
* Provides daily reminders and motivation tips.

**Target Users:**

* Students preparing for exams.
* Professionals looking for structured study plans for certifications.
* Learners needing a personalized study assistant.

**Expected Outcome:**

* A fully functional AI-based study planner that helps users maximize their productivity.

## **Phase-2: Requirement Analysis**

### **Objective:**

Define the technical and functional requirements for StudBud.

### **Key Points:**

#### **Technical Requirements:**

* **Programming Language:** Python
* **Backend:** Cohere API
* **Frontend:** Streamlit Web Framework
* **Database:** Not required initially (API-based query processing)

#### **Functional Requirements:**

* Accept user inputs: subjects, available hours, exam dates, difficulty levels.
* Generate a personalized study plan using AI.
* Display study schedules with recommendations.
* Offer daily learning reminders and motivation tips.

#### **Constraints & Challenges:**

* Ensuring real-time AI-generated study schedules.
* Handling API rate limits.
* Providing a smooth UI experience with Streamlit.

## **Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow of StudBud.

### **Key Points:**

#### **System Architecture:**

1. User enters study-related details via UI.
2. Query is processed using Cohere API.
3. AI model generates an optimized study plan.
4. The frontend displays the personalized study plan.

#### **User Flow:**

1. User inputs subjects, available hours, and difficulty levels.
2. Backend calls Cohere API to retrieve study plans.
3. AI processes and formats study schedules.
4. User gets recommendations with daily study blocks.

#### **UI/UX Considerations:**

* Simple, intuitive design for ease of use.
* Filters for subject according to study preferences.
* Dark & light mode for better usability.

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

### **Sprint Planning with Priorities**

| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected Outcome** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint 1 | Environment Setup & API Integration | 🔴 High | 6 hours | End of Day 1 | Hasini & Meenakshi & | Cohere API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours | End of Day 1 | Tejaswhini | API response format finalized | Basic UI with input fields |
| Sprint 2 | Study Plan Generation Logic | 🔴 High | 3 hours | Mid-Day 2 | Harini & Vaishnavi | API response, UI elements ready | Study schedule with recommendations |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours | Mid-Day 2 | Tejashwini & Harini | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours | Mid-Day 2 | Vaishnavi& Hasini | API response, UI layout completed | Responsive UI, better user experience |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour | End of Day 2 | Entire Team | Working prototype | Demo-ready project |

## **Phase-5: Project Development**

### **Objective:**

Implement core features of StudBud.

### **Technology Stack Used:**

* **Frontend:** Streamlit
* **Backend:** Cohere API
* **Programming Language:** Python

### **Development Process:**

* Implement API key authentication and integrate Cohere API.
* Develop logic for personalized study schedules.
* Optimize AI queries for performance.

### **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 StudBud works as expected.

|  | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Query "Study plan for exams in 2 weeks" | AI-generated plan should be displayed | ✅ Passed | Vaishnavi |
| TC-002 | Functional Testing | Query "Best study method for Physics" | AI should suggest study techniques | ✅ Passed | Meenakshi |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly | ⚠ Needs Optimization | Harini |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses | Data accuracy should be improved | ✅ Fixed | Harini |
| TC-005 | Final Validation | Ensure UI is responsive across devices | UI should work on mobile & desktop | ❌ Failed - UI broken on mobile | Vaishnavi |
| TC-006 | Deployment Testing | Host the app using Streamlit Sharing | App should be accessible online | 🚀 Deployed | Hasini |

## **Final Submission**

* **Project Report** Based on the templates
* **Demo Video** (3-5 Minutes)
* **GitHub/Code Repository Link:**
* **Presentation** for Hackathon Judging