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

# Hackathon Project Phases Template

**Project Title:**

**Studbud: AI Study Planner**

**Team Name:**

CODE WIZARDS

**Team Members:**

* Jyosna Vuttla(22RA1A6632)
* B.Kalyani(22RA1A6634)
* P.Ashritha (22RA1A6613)
* B.Karthik(22RA1A6635)

## Phase-1: Brainstorming & Ideation

**Objective:**

**Improve Academic Performance:** Help students achieve their academicKey Points:

1. **Problem Statement:**

* 1. "Studbud: AI Personalized Study Planner" is an intelligent application designed to create customized student study plans based on their specific goals, strengths, weaknesses, and preferences. Utilizing the BERT (Bidirectional Encoder Representations from Transformers) architecture, this tool helps students optimize their study schedules to achieve their academic targets efficiently

.

1. **Proposed Solution:**

○ AI-Powered Study Planning using **Natural language processing** and **Machine Learning** To improve **academic performance**

* The app offers **Enhanced student engagement** and **Better resource allocations**.

1. **Target Users:**

* 1. **High School Students (ages 14-18):** students manage their coursework, prepare for exams.

○ **College/University Students (ages 18-25):** students in managing their academic workload, achieving academic goals.

○**Graduate Students (ages 25+):** students balance their academic responsibilities with other obligations.

1. **Expected Outcome:**

* Improved academic performance, increased productivity, and developed study habits, leading to sustained academic success and lifelong learning skills.

## Phase-2: Requirement Analysis

**Objective:**

To identify and document the functional, non-functional, and technical requirements of the AI-powered study planning and management tool

.**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language: **Python**

○ Backend: **Google Gemini Flash API**

○ Frontend: **torch,pandas**

1. **Functional Requirements:**

* 1. Ability to **Task management and tracking** using Gemini Flash API.

○ Display **Progress tracking and analytics** in an intuitive UI.

○ Provide **Security and accessibility features**

○ Allow students to prepare **Study planning and scheduling**.

1. **Constraints & Challenges:**

* Time and budget limitations

○ Complexity of AI-powered study planning.

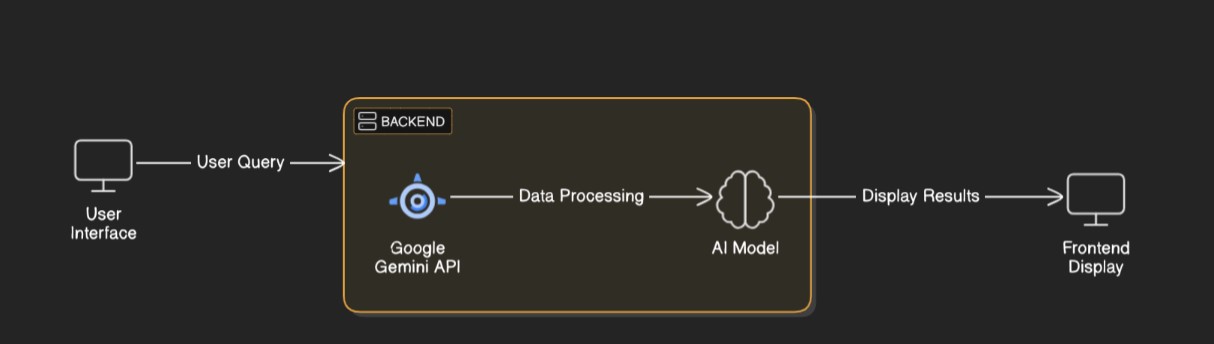
○ Managing conflicting requirements

* Ensuring scalability and flexibility

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the application.



**Key Points:**

1. **System Architecture:**

* Front-end: **React** or **Angular** for web and mobile app development

○ Back-end: **Node.js** or **Django** for server-side logic and API development

○ Database: **MySQL** or **MongoDB** for storing user data and study plans

* AI/ML: **TensorFlow** or **PyTorch** for building AI-powered study planning and recommendation engines.

1. **User Flow:**

* Step 1: User signs up and logs in
* Step 2: User creates a study plan
* Step 3:User views and completes tasks
* Step 4: User tracks progress and achievements
* Step 5: User receives reminders and notifications

1. **UI/UX Considerations:**

* **Clean and Simple Design**: Use a minimalistic design to help users focus on their study plans..
* **User-Centered Design**: Design the app with the user's needs and goals in mind..
* **Mobile-Friendly Navigation**: Use mobile-friendly navigation patterns, such as bottom tabs and gesture-based navigation..

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

**Objective:**

Break down development tasks for efficient completion.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Dependencies** | **Expected**  **Outcome** |
| Sprint 1 | Conduct user research and gather requirements | 🔴 High | 5 hours  (Day 1) | End of Day  1 | None | User research report and project scope statement |
| Sprint 1 | Create wireframes and prototypes | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | User research report | Wireframes and prototypes |
| Sprint 1 | Develop login and registration features | 🔴 High | 4 hours  (Day 1) | Mid-Day 2 | None | Working login and registration features |
| Sprint 2 | Develop study plan creation feature | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Login and registration features | Working study plan creation feature |
| Sprint 2 | Develop task management feature | 🟡  Medium | 2 hours  (Day 2) | Mid-Day 2 | Study plan creation feature | Working task management feature |
| Sprint 2 | Conduct unit testing for login and registration features | 🟢 High | 1 hour (Day 2) | End of Day  2 | None | Test report and debugged code |
| Sprint 3 | Develop AI-powered study planning and recommendation engine | 🟢 High | 1.5 hours  (Day 2) | End of Day 2 | Study plan creation feature | Working AI-powered study planning and recommendation engine |
| Sprint 3 | Integrate AI engine with study plan feature | 🟡  Medium | 1.5 hours  (Day 2) | End of Day 2 | AI-powered study planning and recommendation engine | Integrated AI engine and study plan feature |

**Sprint Planning with Priorities**

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

**(**🔴 **High Priority**) Develop study plan creation feature.

**(**🔴 **High Priority)** Integrate study plan creation feature with database and API

**(**🟡 **Medium Priority)** Develop task management feature

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

**(**🔴 **High Priority)** Study Plan Creation

**(**🔴 **High Priority)** AI-Powered Recommendations

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

**(**🟡 **Medium Priority)** Progress Tracking

**(**🟢 **Low Priority)** Reminders and Notifications

## Phase-5: Project Development

**Objective:**

StudBud's core features include study plan creation, task management, AI-powered recommendations, progress tracking, and reminders.

**Key Points:**

1. **Technology Stack Used:**

* **Frontend:** React Native for building the mobile app

○ **Backend:** Node.js for building the server-side application

○ **Programming Language:** Python

1. **Development Process:**

* Involves requirements gathering, design, frontend and backend development, AI/ML development, testing, debugging, deployment, and maintenance.
* The development methodology used is Agile, with tools such as React Native, Node.js, MongoDB, and TensorFlow.

1. **Challenges & Fixes:**

* + **Challenge:** **Data Accuracy**: Ensuring accurate data collection and analysis for personalized recommendations**.**

**Fix:**. **Data Accuracy**: Implement data validation and verification processes to ensure accurate data collection.

○ **Challenge:** **User Engagement:** Keeping users motivated and engaged with the app..

**Fix:** **User Engagement:** Implement gamification elements, rewards, and reminders to keep users engaged**.**

## Phase-6: Functional & Performance Testing

**Objective:**

To verify that the app's features, user interface, and data storage mechanisms work as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Load Testing | Simulate 100 users logging in concurrently | | Response time < 2 seconds. | ✅ Passed | Tester 1 |
| TC-002 | Stress Testing | Simulate 1000 users accessing the app simultaneously | App remains stable and responsive. | ✅ Passed | Tester 2 |
| TC-003 | scalability  Testing | Increase user load by 50% and measure response time | Response time < 3 seconds. | ✅ Passed | Tester 1 |
| TC-004 | Response Time Testing | Measure response time for a single user. | Response time < 1 second | ✅ Passed | Tester 2 |
| TC-005 | Throughput Testing | Measure throughput for a single user. | Throughput > 10 requests per second . | ✅ Passed | Tester 1 |

## Final Submission

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