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

Hackathon Project Phases Template

# Project Title:

**AI STUDY PLANNER**

# Team Name:

**SMSK**

# Team Members:

* M.VAMSHI
* S.VINESH
* T.SRINATH
* V.AJAY

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

#### ****Objective:****

Develop an AI-powered study planner that helps students optimize their learning schedule, get personalized recommendations, and track progress.

#### ****Key Points:****

1. **Problem Statement:**
   * Many students struggle with organizing their study schedules efficiently.
   * Lack of personalized learning plans that adapt to individual progress and focus areas.
   * Difficulty in balancing multiple subjects while preparing for exams.
2. **Proposed Solution:**
   * An AI-driven study planner that generates customized learning schedules based on syllabus, learning speed, and deadlines.
   * Provides AI-generated flashcards, quizzes, and study tips.
   * Integrates with calendars and sends smart reminders.
3. **Target Users:**
   * Students preparing for exams (high school, college, competitive exams).
   * Self-learners looking for structured study guidance.
   * Teachers and tutors managing student study plans.
4. **Expected Outcome:**
   * A functional AI study planner that creates adaptive study schedules and enhances learning efficiency.

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

#### ****Objective:****

Define the technical and functional requirements of the AI Study Planner.

#### ****Key Points:****

1. **Technical Requirements:**
   * **Programming Language:** Python
   * **Backend:** OpenAI API for AI-generated study plans
   * **Frontend:** React (or Streamlit for MVP)
   * **Database:** Firebase or SQLite for user data storage
2. **Functional Requirements:**
   * Users input subjects, topics, and deadlines.
   * AI generates an optimized study plan with personalized recommendations.
   * Progress tracking and adaptive scheduling.
   * Study reminders via notifications or emails.
   * Option to generate practice quizzes or flashcards.
3. **Constraints & Challenges:**
   * Ensuring accurate and adaptable AI-generated study plans.
   * Managing real-time updates for rescheduling missed sessions.
   * Integration with existing calendar and reminder apps.

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

#### ****Objective:****

Develop the architecture and user flow of the application.

#### ****Key Points:****

1. **System Architecture:**
   * User inputs study preferences, subjects, and deadlines.
   * AI processes data and generates a dynamic study plan.
   * The system tracks progress and adjusts the schedule based on performance.
   * Users receive reminders and study suggestions.
2. **User Flow:**
   * **Step 1:** User enters study goals, subjects, and deadlines.
   * **Step 2:** AI generates a study plan with structured sessions.
   * **Step 3:** The planner tracks progress and adapts based on completion and performance.
3. **UI/UX Considerations:**
   * Simple, clean interface with a calendar view.
   * Drag-and-drop feature for adjusting schedules.
   * Dark and light modes for better accessibility.

### ****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 | Set up environment & API integration | 🔴 High | 6 hours | Day 1 | Member 1 | OpenAI API, Firebase | API connection working |
| Sprint 1 | Basic UI design | 🟡 Medium | 2 hours | Day 1 | Member 2 | API response format | Initial UI ready |
| Sprint 2 | Study Plan Algorithm | 🔴 High | 4 hours | Day 2 | Member 1 & 2 | API and database | Dynamic study plan generated |
| Sprint 2 | Reminders & Notifications | 🔴 High | 3 hours | Day 2 | Member 3 | User data & calendar integration | Alerts system functional |
| Sprint 3 | Testing & Enhancements | 🟡 Medium | 2 hours | Day 2 | Member 3 & 4 | App functionality | Smooth user experience |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour | Day 2 | Entire Team | Working prototype | Ready for demo |

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

#### ****Objective:****

Implement core features of the AI Study Planner.

#### ****Key Points:****

1. **Technology Stack Used:**
   * **Frontend:** React (or Streamlit for MVP)
   * **Backend:** OpenAI API for AI-generated plans
   * **Database:** Firebase or SQLite
2. **Development Process:**
   * Implement AI-generated study plans.
   * Develop user progress tracking and adaptive scheduling.
   * Integrate reminders and notifications.
3. **Challenges & Fixes:**
   * **Challenge:** Managing missed study sessions.
     + **Fix:** Implement AI-driven rescheduling.
   * **Challenge:** Handling multiple subjects and study priorities.
     + **Fix:** Allow users to rank subjects by importance.

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

#### ****Objective:****

Ensure the AI Study Planner works as expected.

| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| TC-001 | Functional Testing | User inputs study schedule | AI generates a dynamic plan | ✅ Passed | Tester 1 |
| TC-002 | Functional Testing | User skips a study session | AI auto-adjusts the plan | ✅ Passed | Tester 2 |
| TC-003 | Performance Testing | API response time < 500ms | Fast results | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Incorrect reminders | Fix reminder logic | ✅ Fixed | Developer |
| TC-005 | UI Testing | Responsive on mobile & desktop | Smooth UI | ❌ Failed - Needs Fixing | Tester 2 |
| TC-006 | Deployment Testing | Host the app | App is accessible online | 🚀 Deployed | DevOps |

### ****Final Submission****

1. **Project Report** – Based on the hackathon template.
2. **Demo Video (3-5 minutes)** – Showcasing the AI Study Planner in action.
3. **GitHub/Code Repository Link** – Providing access to the source code.
4. **Presentation** – Highlighting key features and implementation.