

Hackathon Project Phases Template

Project Title: StudBud

Team Name: MindX

Team Members:

- Meneni Sahithi
 - Avulla Shreya
 - Perela Sindhu
 - Banothu Shravya
-

Phase-1: Brainstorming & Ideation

Objective:

StudBud: AI Study Planner is an intelligent study planning application that personalizes learning schedules based on students' individual goals, strengths, weaknesses, and study preferences. By leveraging BERT (Bidirectional Encoder Representations from Transformers) and Generative AI (GenAI), StudBud dynamically adapts study plans to optimize learning efficiency, ensuring better academic performance.

Key Points:

1. Problem Statement:

Many students struggle with time management, subject prioritization, and study consistency, leading to inefficient learning and suboptimal exam performance. Traditional static study plans fail to accommodate a student's evolving needs, learning pace, and strengths. StudBud solves this problem by using AI-powered adaptive learning, creating dynamic study schedules that adjust in real-time based on user progress and performance.

2. Proposed Solution:

An AI-powered application using BERT and Generative AI (GenAI) to create personalized study plans. The app offers adaptive learning strategies based on user preferences and progress tracking.

3. Target Users:

- Students preparing for exams who need personalized and optimized study plans.
- Learners looking for adaptive learning methods and time management assistance.

4. Expected Outcome:

- A functional AI-powered study planner that provides personalized study schedules, adaptive learning suggestions, and efficient time management strategies.
-

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the AutoSage App.

Key Points:

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 generate personalized study plans using BERT and GenAI.

- Display study schedules, subject-wise plans, and learning recommendations in an intuitive UI.
- Provide adaptive learning suggestions based on user feedback and progress.
- Allow users to input their subjects, difficulty level, and available study time to get a customized plan.
- Track learning progress and modify schedules dynamically to optimize learning.

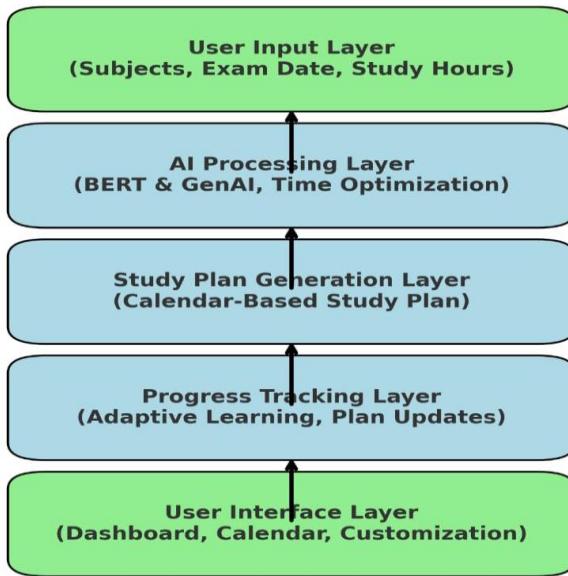
3. Constraints & Challenges:

- Ensuring accurate and adaptive study plan generation using BERT and GenAI.
 - Handling large volumes of user data efficiently while personalizing schedules.
 - Optimizing AI-generated plans to align with different exam formats and study preferences.
 - Developing a smooth UI/UX experience for interactive study plan visualization.
 - Integrating progress tracking to provide real-time feedback to students.
-

Phase-3: Project Design

Objective:

The architecture and user flow of the application.



Key Points:

1. System Architecture:

1. User Input: Student enters subjects, exam date, study hours, and learning preferences.
2. AI Processing: Generates a personalized study schedule, optimizing time distribution.
3. Study Plan Generation: Creates a calendar-based schedule with revision days and practice sessions.
4. Real-Time Progress Tracking: AI monitors study completion and adjusts the plan dynamically.
5. User Feedback & Adaptation: AI suggests additional resources, practice tests, and learning strategies.

2. User Flow:

1. Profile Setup: User enters academic details and study goals.
2. AI Study Plan Creation: Personalized schedule displayed in a calendar view.
3. Daily Study Execution: User follows AI-recommended tasks and reminders.

4. Progress Tracking: AI updates the schedule based on completed topics.

5. Final Exam Mode: Optimized revision sessions and mock tests before exams.

UI/UX Considerations:

- Dashboard: Progress tracking, upcoming tasks, and AI recommendations.
 - Calendar View: Organized study plan visualization.
 - Customization: Users adjust study preferences and reschedule tasks.
 - Dark & Light Mode: Enhances readability.
 - Notifications & Alerts: AI reminders for study sessions.
-

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 & API Integration	● High	6 hours (Day 1)	End of Day 1	Member 1	Python, BERT & GenAI setup	AI model successfully integrated
Sprint 1	Frontend UI Development	● Medium	3 hours (Day 1)	End of Day 1	Member 2	API response format finalized	Basic UI for study planner input
Sprint 2	Study Plan Generation Logic	● High	3 hours (Day 2)	Mid-Day 2	Member 1 & 2	AI model, UI ready	Dynamic study schedule generation
Sprint 2	Progress Tracking & Adaption	● High	3 hours (Day 2)	Mid-Day 2	Member 3	Study Plan logic implemented	AI adapts schedule based on progress
Sprint 3	UI Enhancements & Dashboards	● Medium	2 hours (Day 2)	Mid-Day 2	Member 2 & 3	Core UI built	Responsive, Interactive UI
Sprint 3	Final Presentation & Deployment	● Low	1 hour (Day 2)	End of Day 2	Entire Team	Working prototype	Ready for presentation & submission

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

- (● High Priority) Set up the development environment & install dependencies.
- (● High Priority) Integrate BERT & GenAI models for personalized study plan generation.
- (● Medium Priority) Develop a basic frontend UI to accept study preferences.

Sprint 2 – Core Features & Debugging (Day 2)

- (● High Priority) Implement dynamic study plan generation based on user input.
- (● High Priority) Enable progress tracking and real-time AI-based schedule adjustments.
- (● High Priority) Develop adaptive learning suggestions based on performance

Sprint 3 –UI Enhancements & Testing (Day 2)

- (● Medium Priority) Enhance UI/UX design, making the interface interactive and user-friendly.
- (● Medium Priority) Implement visual progress tracking.
- (● High Priority) Perform testing & debugging to ensure AI-generated schedules are accurate.

Sprint 4 – Deployment & Final Submission (Day 2)

- (● Low Priority) Deploy the application using Flask/Streamlit.
- (● Low Priority) Prepare the final presentation, Github repo, and demo video.

Phase-5: Project Development

Objective:

Implement core features of the StudBud Web.

Key Points:

1. Technology Stack Used:

- **Frontend:** React.js / Streamlit
- **Backend:** Flask
- **AI Model:** BERT & Generative AI

2. Development Process:

- AI Model Integration – Implement BERT & GenAI for personalized study plans.
- Study Plan Generation – AI creates a balanced calendar-based schedule.
- Progress Tracking – Users update progress; AI adjusts schedules dynamically.
- User Interface – Interactive dashboard with calendar view and reminders.
- Testing & Debugging – Validate AI accuracy, optimize response times, fix UI issues.
- Deployment – Host on Flask/Streamlit, ensuring data security and accessibility.

3. Challenges & Fixes:

- Slow AI response → Optimize with query caching.
 - Limited adaptability → Introduce user feedback-driven adjustments.
 - Real-time schedule changes → AI-enabled dynamic rescheduling.
 - Different study priorities → Offer multiple study modes.
-

Phase-6: Functional & Performance Testing

Objective:

Ensure that the StudBud Web Application works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Generate a study plan for 2 weeks with 4 subjects	AI generates a structured study plan.	<input checked="" type="checkbox"/> Passed	Tester 1
TC-002	Functional Testing	Adjust study difficulty level dynamically.	AI modifies study plan based on progress.	<input checked="" type="checkbox"/> Passed	Tester 2
TC-003	Performance Testing	API response time under 500ms	Study plan is generated quickly.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect schedule reallocation.	AI correctly updates study sessions.	<input checked="" type="checkbox"/> Fixed	Developer
TC-005	UI/UX Testing	Ensure UI is responsive on mobile & desktop.	Smooth navigation and usability.	✗ Failed - UI broken on mobile	Tester 2
TC-006	Deployment Testing	Host the app on Flask/Streamlib	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
4. Presentation