

Hackathon Project Phases Template for the AutoSage App project.

Hackathon Project Phases Template

Project Title:

AI Study Planner

Team Name:

Team AI

Team Members:

- M Samson
 - P Charan
 - M Vaishnavi
 - M Supraja
-

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-Study Planner expert tool using Gemini Flash to help users compare and analyze of study planner that are useful to students to plan the time table.

Key Points:

1. Problem Statement:

"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.

2. Proposed Solution:

AI Personalized Study Planner addresses this problem by leveraging the BERT architecture to analyze student-provided information about their goals, strengths, weaknesses, and preferences. This intelligent application generates customized

study plans tailored to each student's unique needs, optimizing their study schedules for efficiency and effectiveness, ultimately improving their chances of achieving their academic targets.

3. Target Users:

- High School Students
- College/University Students
- Graduate Students
- Students with Learning Differences
- Distance Learners
- Individuals seeking to improve study habits

4. Expected Outcome:

- **Improved academic performance:** Better grades, test scores, and overall learning outcomes.
- **Enhanced study efficiency:** Optimized use of study time, leading to more effective learning. —

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the AutoSage App.

Key Points:

1. Technical Requirements:

- Programming Language: **Python**
- Backend: **BERT**
- Frontend: **Streamlit Web Framework**
- Database: **Not required initially (BERT-based queries)**

Functional Requirements:

- **Personalized Study Plan Generation:** AI-powered creation of study plans based on individual user data.
- **Schedule Management:** Visualization, editing, and tracking of the generated study schedule.
- **Progress Tracking:** Monitoring user progress towards goals and providing feedback.
- **User Account Management:** Securely storing and managing user data and preferences.
- **Platform Accessibility:** Compatibility across various devices and operating systems.

1 Constraints & Challenges:

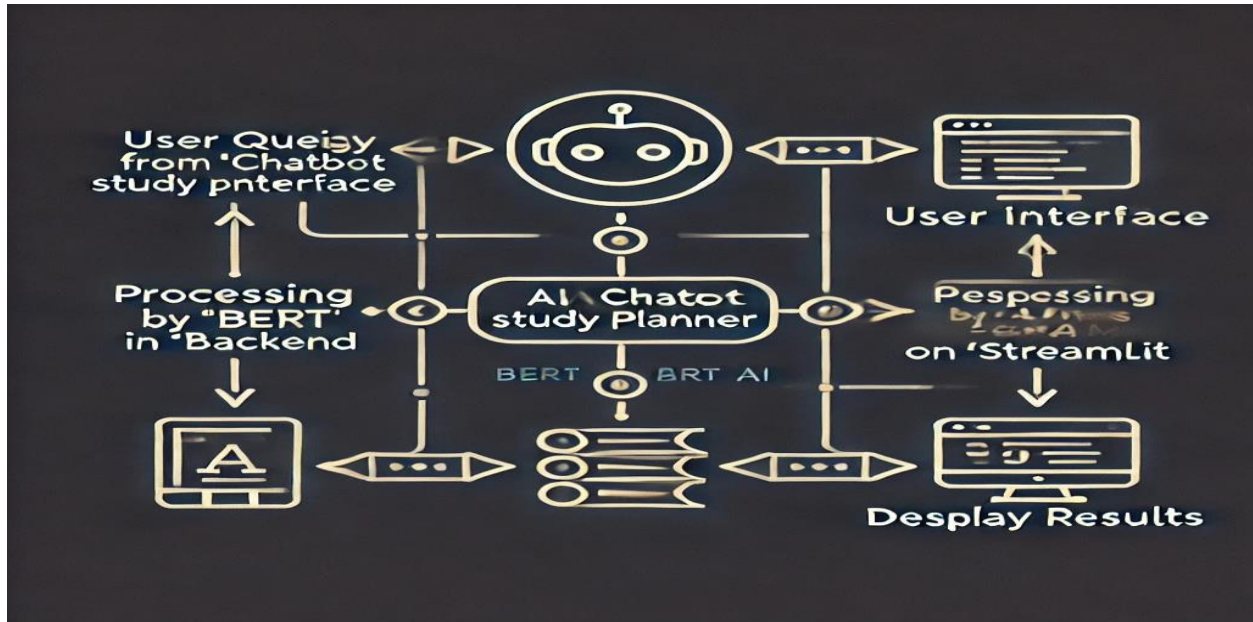
- Ensuring real-time updates from **BERT**.
- Providing a **smooth UI experience** with Streamlit.

—

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. System Architecture:

- User enters study-related query via UI.
- Query is processed using **BERT**
- AI model fetches and processes the data.
- The frontend displays **vehicle details, reviews, and comparisons**.

2. User Flow:

- Step 1: User enters a subject
- Step 2: The backend **calls the BERT** to retrieve subject data.
- Step 3: The app processes the requirement of user and give the related study planner to user

3. UI/UX Considerations:

- ❑ **Smooth BERT connection:** Frontend and backend communicate flawlessly.
- ❑ **Smart data handling:** Data is mapped and managed efficiently.
- ❑ **Easy user interaction:** Forms and displays are intuitive.
- ❑ **Solid error handling:** Loading and errors are managed gracefully.
- ❑ **Structured frontend:** Components support backend interaction.–

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	Vaishnavi	BERT, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	● Medium	2 hours (Day 1)	End of Day 1	Samson	BERT response format finalized	Basic UI with input fields
Sprint 2	subject Search & Comparison	● High	3 hours (Day 2)	Mid-Day 2	Charan	BERT response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	● High	1.5 hours (Day 2)	Mid-Day 2	Samson, Charan, Vaishnavi	BERT logs, UI inputs	Improved BERT stability
Sprint 3	Testing & UI Enhancements	● Medium	1.5 hours (Day 2)	Mid-Day 2	Supraja	BERT response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	● Low	1 hour (Day 2)	End of Day 2	Samson, Charan, Vaishnavi, Supraja	Working 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**.
- (● Medium Priority) Build a **basic UI with input fields**.

Sprint 2 – Core Features & Debugging (Day 2)

- (● High Priority) Implement **search & comparison functionalities**.
- (● High Priority) Debug BERT issues & handle **errors in queries**.

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

- (● Medium Priority) Test BERT responses, refine UI, & fix UI bugs.
- (● Low Priority) Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Implement core features of the AI Study Planner.

Key Points:

1. Technology Stack Used:

- **Frontend:** Streamlit
- **Backend:** BERT
- **Programming Language:** Python

2. Development Process:

- Implement **BERT key authentication** and **BERT integration**.
- Develop **Study comparison** and **study planning tips logic**.
- Optimize **search queries** for performance and relevance.

3. Challenges & Fixes:

- **Challenge:** Delayed BERT response times.
Fix: Implement **caching** to store frequently queried results.
- **Challenge:** Limited BERT calls per minute.
Fix: Optimize queries to fetch **only necessary data**.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AI Study Planner works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Verify that a new study plan can be created with valid inputs.	The system should successfully create and save a new study plan.	✅ Passed	Vaishnavi
TC-002	Functional Testing	Test the validation process when creating a study plan with missing/invalid data.	The system should prompt the user to fill or correct missing/invalid fields.	✅ Passed	Charan
TC-003	Performance	Check the response time under a simulated load of	The system should	⚠ Needs	Samson

	Testing	100 concurrent users.	respond within 2 seconds on average.	Optimization	
TC-004	Bug Fixes & Improvements	Verify the fix for the incorrect scheduling bug (plans were duplicating dates).	The schedule should be generated correctly without duplications.	✓ Fixed	Supraja

Final Submission

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