

Hackathon Project Phases Template that ensures students can complete it efficiently while covering all six phases. The template is structured to capture essential information without being time-consuming.

Hackathon Project Phases Template

Project Title:

Studbud: AI Study Planner

Team Name:

4BitDone

Team Members:

- Dhruv Shah
- Jashwanth Reddy
- Karan Sai
- Hrushikesh Kumar
- Guna Praneeth

Phase-1: Brainstorming & Ideation

Objective:

- The **AI Study Planner** tackles the issue of poor time management and ineffective study schedules faced by students and professionals. It provides personalized, adaptive study plans with AI-driven recommendations and gamifying features to improve productivity, motivation, and learning outcomes. The project aims to make personalized learning tools accessible to everyone, including underprivileged students, promoting more equitable education.

Key Points:

- **Problem Statement:** Students and professionals struggle with effective time management and consistent study schedules, leading to reduced productivity and poor learning outcomes.

- **Proposed Solution:** The **AI Study Planner** is an adaptive tool that creates personalized, dynamic study plans, tracks progress, and offers AI-driven recommendations tailored to individual learning styles and needs. Key features include:
 - Gamifying elements (rewards, badges, streaks)
 - AI-powered task plan creation
- **Target Users:**
 - **Students:** Those looking to improve their time management and academic performance.
 - **Professionals:** Individuals preparing for certifications, courses, or skill development.
- **Expected Outcome:** The **AI Study Planner** is expected to deliver significant benefits across various groups. For students, it will improve time management, reduce procrastination, and lead to better grades. Professionals will benefit from more efficient scheduling, supporting their career growth and skill enhancement. Additionally, the project will have a social impact by democratizing access to personalized learning strategies, particularly for underprivileged students, thereby promoting more equitable educational opportunities for all.

Phase-2: Requirement Analysis

Objective: The **AI Study Planner** will use **React.js** for the frontend, **Node.js** for the backend, **MongoDB** for storage, and **LangChain and mistral APIs** for personalized recommendations. It will include user authentication with **JWT**, AI-driven study suggestions, progress tracking, and gamifying features.

Key Points:

1. Technical Requirements:

- **Frontend: React.js (for building the user interface)**
- **Backend:** Node.js with Express.js (for handling server-side logic)
- **Database:** MongoDB (for storing user data, study plans, and progress)
- **AI/ML Models:** LangChain + MistralAPIs (for personalized study material recommendations and AI-driven insights)
- **Authentication:** JWT (for user registration, login, and session management)

2. Functional Requirements:

- **Study Plan Creation & Management:** Users should be able to create, modify, and manage their personalized study schedules.
- **AI-Powered Recommendations:** Personalized study material suggestions based on user progress, learning style, and weaknesses, using LangChain and MistralAI APIs.
- **Gamifying Features:** Streaks, rewards, badges, and motivational nudges to keep users engaged and motivated.

- **Progress Tracking:** Track study session completion, tasks, streaks, and overall progress.
- **Responsive Design:** Ensure the platform works seamlessly across different devices and screen sizes.

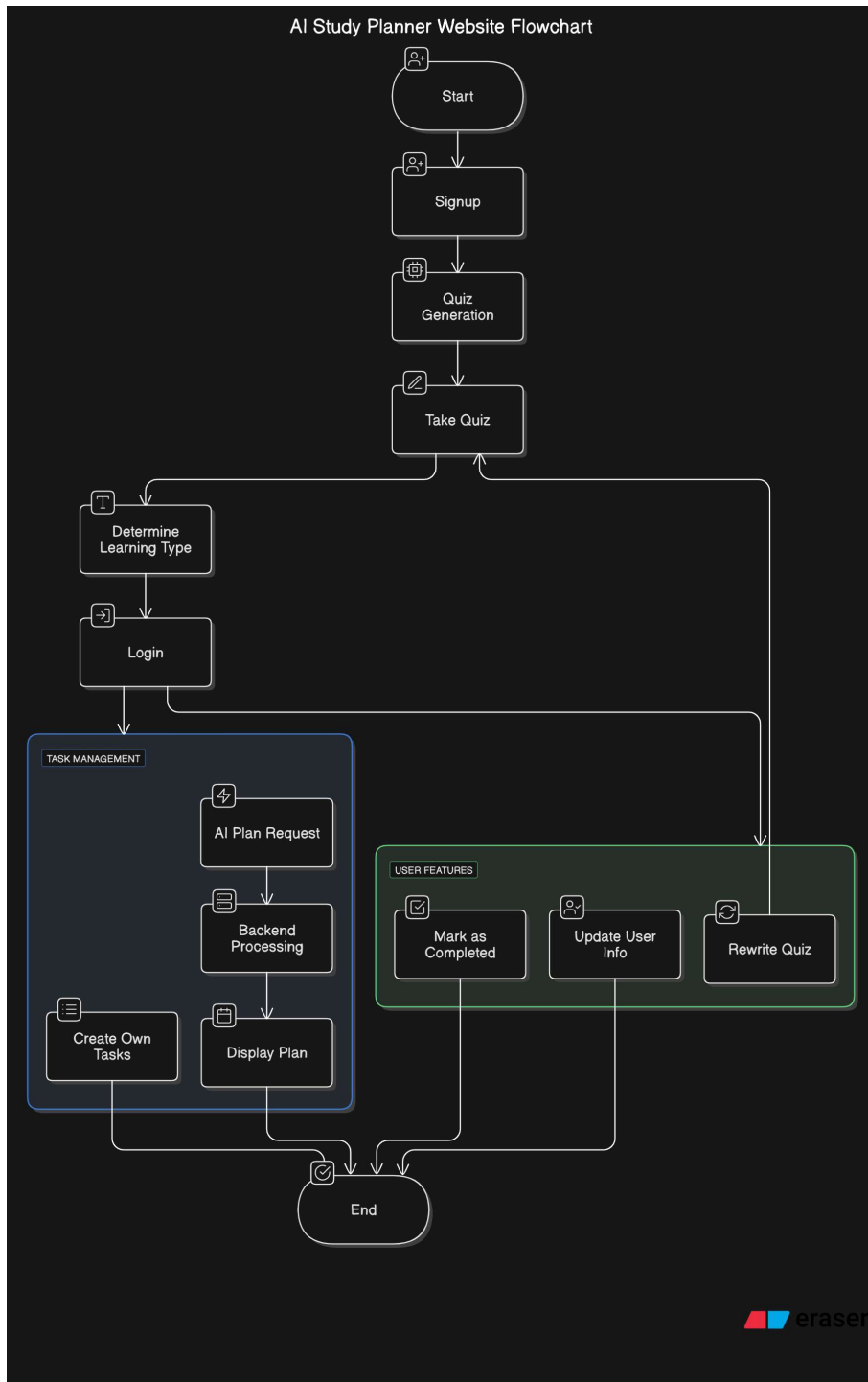
3. Constraints & Challenges:

- **Accuracy of AI Recommendations:** Fine-tuning the AI to provide relevant and personalized study material recommendations based on individual progress and preferences may require continuous improvement.
- **Cross-Platform Compatibility:** Ensuring that the platform functions smoothly across different browsers and devices, including mobile, can be a potential challenge.
- **Data Privacy & Security:** Safeguarding user data, especially sensitive study information, requires strict privacy controls and compliance with data protection regulations.

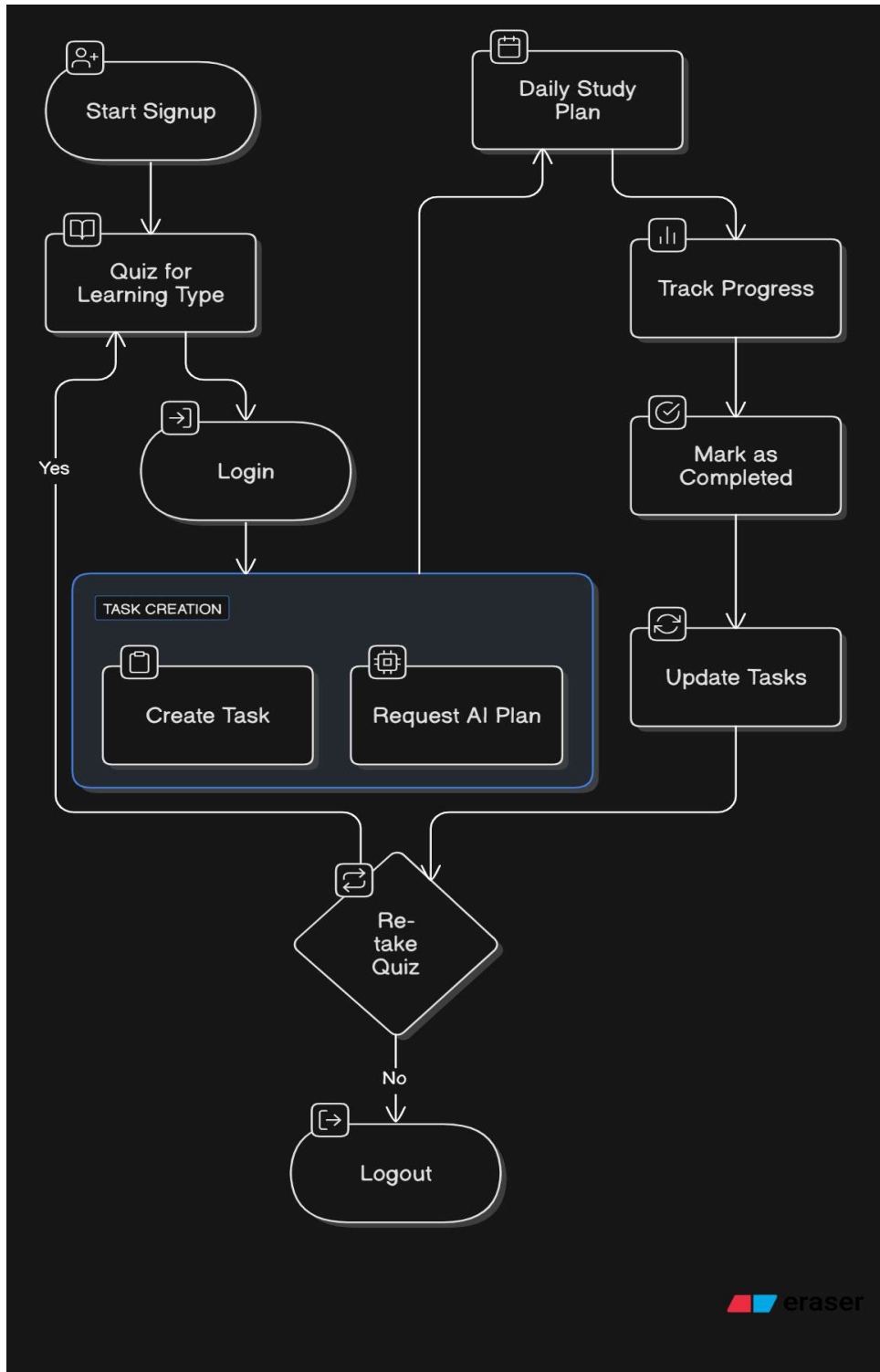
Phase-3: Project Design

Key Points:

1. System Architecture Diagram:



2. User Flow:



Phase-4: Project Planning (Agile Methodologies).

Key Points:

1.Sprint Planning:

The project will be divided into a single 24-hour sprint with clearly defined tasks that need to be accomplished. The team will work continuously on tasks and collaborate effectively to meet the 24-hour deadline.

- **Sprint 1 (0-6 hours):**
 - **Objective:** Set up the project environment (React.js, Node.js, MongoDB), integrate user authentication (JWT).
 - **Deliverables:** Initial setup and authentication functionality.
- **Sprint 2 (6-12 hours):**
 - **Objective:** Develop the study planner interface and basic task management features.
 - **Deliverables:** A working study planner where users can add, edit, and delete tasks.
- **Sprint 3 (12-18 hours):**
 - **Objective:** Integrate AI-powered study plan recommendations (LangChain, Mistral APIs) and implement gamifying features (streaks, rewards).
 - **Deliverables:** AI-driven personalized study plans and gamifying functionalities.
- **Sprint 4 (18-24 hours):**
 - **Objective:** Finalize progress tracking, test all features, fix bugs, and populate the application with basic data for initial use.
 - **Deliverables:** Fully functional study planner with all features integrated. Bug-free application deployed.

1.

2.Task Allocation:

- **Frontend Development (React.js, TailwindCSS):**
 - **Team Member 1:** Focus on the **UI design**, **user authentication**, and **dashboard layout**.
 - **Team Member 2:** Work on **study planner UI**, and **responsive design**.
- **Backend Development (Node.js, Express.js, MongoDB):**
 - **Team Member 3:** Set up **backend server**, **API routes** for task creation and management.
 - **Team Member 4:** Integrate **database** (MongoDB), manage user data, and handle backend logic for study plan updates and progress tracking.

- **AI Integration (LangChain, Mistral APIs):**
 - **Team Member 5:** Focus on integrating **AI-powered recommendations**, **personalized study plans**, and **gamifying** features like progress tracking and user engagement.

Phase-5: Project Development

Key Points:

1. Technology Stack Used:

- **Frontend:**
 - **React.js:** For building the interactive user interface.
 - **TailwindCSS:** For fast and responsive styling of the application.
- **Backend:**
 - **Node.js:** For building the backend server.
 - **Express.js:** For handling HTTP requests and creating the API.
 - **MongoDB:** For storing user data, progress, and tasks.
- **AI & NLP:**
 - **LangChain:** For AI-powered task recommendation and personalized study plans.
 - **Mistral AI:** For generating AI-driven study materials and insights.
- **Authentication:**
 - **JWT:** For handling user authentication and session management.

2. Development Process:

- **Initial Setup:**
 - Set up the development environment with React.js for the frontend and Node.js with Express for the backend.
 - Configured JWT for user login and authentication.
- **Frontend Development:**
 - Built the study planner UI using React.js and TailwindCSS.
 - Developed a dashboard layout where users can view, edit, and delete tasks.
- **Backend Development:**
 - Set up the Node.js server with Express.js to handle API requests from the frontend.
 - Configured MongoDB to store user data, study plans, and task progress.

- Developed APIs to manage user accounts, study tasks, and progress tracking.

- **AI Integration:**

- Integrated LangChain and MistralAI API to provide personalized study recommendations based on the user's preferences and learning style.
- Developed AI-driven features like spaced repetition and active recall for better learning retention.

- **Testing & Debugging:**

- Conducted integration tests to ensure the functionality of the study planner and AI recommendations.
- Fixed bugs related to task creation, UI responsiveness, and AI performance.

3. Challenges & Fixes:

- **Challenge 1: AI Integration Complexity:**

- Integrating AI-driven study plans using LangChain and MistralAI was complex due to dynamic user input and varying preferences.
- **Fix:** Worked closely with the AI team to refine the algorithm and implemented a feedback loop to improve AI-generated study plans based on user feedback.

- **Challenge 2: Synchronizing Frontend and Backend:**

- Ensuring seamless data flow between the frontend and backend was challenging, particularly with user data and real-time updates.
- **Fix:** Utilized RESTful APIs and ensured proper data formatting between the frontend and backend. Used state management tools like Redux for better data flow.

- **Challenge 3: Mobile Responsiveness Issues:**

- Initial layouts did not scale well on different devices.
- **Fix:** Tweaked TailwindCSS utility classes and performed thorough testing on mobile devices to ensure the app was fully responsive.

Phase-6: Functional & Performance Testing

Key Points:

1. Test Cases Executed:

- **User Authentication:**

- Test the signup, login, and logout processes using JWT.
- Verify user data is correctly stored in the database and sessions are managed properly.

- **AI-Generated Quiz:**

- Test the functionality of the AI-generated quiz for different **ages** and **subjects** to determine the user's learning type (visual, auditory etc.).
 - Verify that the quiz is auto-generated based on the selected subject and age group.
 - Ensure quiz results correctly map to personalized learning paths.
- **Task Management:**
 - Test adding, editing, and deleting study tasks.
 - Check if tasks are saved correctly and appear in the user's study planner.
- **AI Recommendations:**
 - Test the AI-generated study plan based on user preferences, learning type (derived from the quiz), and progress.
 - Verify the AI provides accurate and personalized study materials.
 - Test if AI recommendations adjust when tasks are marked as completed or missed.
- **Progress Tracking:**
 - Test progress tracking and task completion features.
 - Verify that streaks, badges, and rewards are displayed correctly based on user progress.
- **Mobile Responsiveness:**
 - Test the app on multiple devices (mobile, tablet, desktop) to ensure responsiveness.
 - Verify the user interface scales properly on various screen sizes.

2. Bug Fixes & Improvements:

- **Bug 1: Task Synchronization Issue:**
 - **Issue:** Users faced issues when trying to edit tasks, and the changes didn't reflect immediately.
 - **Fix:** Implemented better state management for task updates using Zustand and ensured data was properly synchronized with the backend.
- **Bug 2: Mobile Layout Problems:**
 - **Issue:** Some buttons and elements were misaligned on mobile screens.
 - **Fix:** Adjusted TailwindCSS classes for responsive design to fix layout issues on smaller screens.
- **Bug 3: Authentication Errors:**
 - **Issue:** Some users encountered session timeouts unexpectedly.
 - **Fix:** Improved session handling with JWT and implemented automatic session renewal.
- **Bug 4: AI Plan Generation Delay:**
 - **Issue:** AI recommendations took longer than expected due to large input data.
 - **Fix:** Optimized the AI model and improved query handling to speed up the

recommendation process.

- **Improvement** Enhanced user experience by adding tooltips, progress bars, and smoother transitions to improve overall interactivity and engagement.

3. Final Validation:

- The project meets all the **initial requirements** defined during the brainstorming phase, including:
 - User authentication and secure login.
 - AI-generated quiz functionality that determines the learner's type based on age and subject, enabling personalized learning paths.
 - Fully functional study planner with task management features.
 - Personalized AI recommendations and learning paths based on the quiz results.
 - Gamifying features (progress tracking, badges, streaks).
 - Mobile responsiveness and cross-platform functionality.
- **Testing has confirmed** that all major functionalities are working as expected, with no critical bugs remaining.

Final Submission

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