Hackathon Project Phases Template

Project Title:

STUDBUD – AI PERSONALIZED STUDY PLANNER

Team Name:

VORTEX

Team Members:

● K . LASYA

● B. HASINI REDDY

● G . SIVA DURGA

● L . SAI KRUSHINI

Phase-1: Brainstorming & Ideation

Objective:

**StudyPro is an AI-powered study planner that enables students to create personalized study plans, track their progress, manage study resources, and set study goals. By analyzing user preferences and behaviors, StudyPro aids in identifying areas of weakness and recommending tailored study schedules to enhance learning efficiency and effectiveness.**

**1. Problem Statement:**

* **Many users find it challenging to access reliable and current information about two-wheelers and four-wheelers when deciding on a purchase.**
* **Users require guidance on how to maintain their vehicles and make eco-friendly choices regarding their transportation options.**

**2. Proposed Solution:**

* **An AI-powered application that utilizes Gemini Flash to deliver real-time vehicle specifications, user reviews, and direct comparisons between different vehicles.**
* **The app includes features for providing maintenance tips and insights into eco-friendly vehicle options tailored to user preferences.**

**3. Target Users:**

* **Potential vehicle buyers who need detailed specifications and comparisons before making a purchase.**
* **Existing vehicle owners seeking practical maintenance tips for different seasons.**
* **Eco-conscious consumers interested in exploring hybrid and electric vehicle alternatives.**

**4. Expected Outcome:**

* **A fully operational AI-powered vehicle information application that offers valuable insights based on up-to-date data and user inquiries, helping users make informed vehicle-related decisions.**

**Phase-2: Requirement Analysis**

**StudyPro App: Requirement Analysis**

**Objective:**

**Define what the StudyPro App needs to work effectively.**

**1. Technical Requirements:**

**- Programming Language: Use Python to build the app.**

**- Backend:Create custom functions to generate study tasks and track progress.**

**- Frontend: Use Tkinter to make it user-friendly.**

**- Database:No database is needed at first; data will be kept in memory during use.**

**2. Functional Requirements:**

**- Let users create accounts to access the app (sign up, log in, or use as a guest).**

**- Collect users' study preferences like goals, available study time, and preferred methods.**

**- Generate personalized study tasks based on what the user wants.**

**- Include a timer for study sessions that reminds users when to take breaks.**

**- Show motivational messages and reminders to stay hydrated and take breaks.**

**- Allow users to keep track of books or resources for their studies.**

**- Help users see their progress by showing completed tasks and study hours.**

**3. Constraints & Challenges:**

**- Ensure the app is responsive and easy to use even with many users.**

**- Create study tasks in real-time without making the app confusing.**

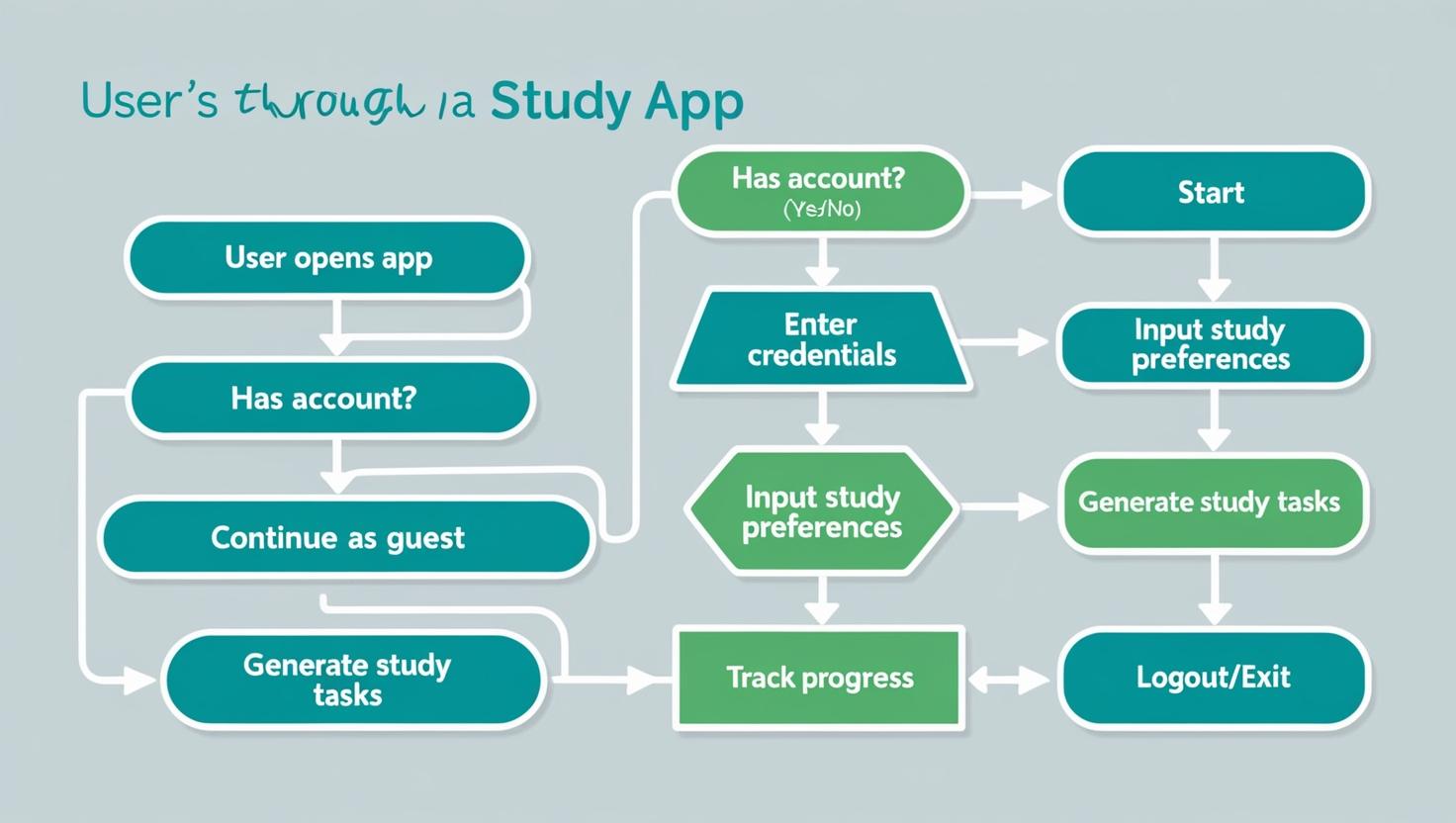
**- Provide a smooth experience when switching between different features.**

**- Keep user data secure, especially since there’s no database in the beginning.**

**Phase-3: Project Design**

**objective:**

**Develop the architecture and user flow of the StudyPro application.**

****

**Keypoints:**

**1. System Architecture:**

**- User logs in or continues as a guest via the UI.**

**- User inputs their study preferences and goals.**

**- The app processes this information to generate personalized study tasks.**

**- The frontend displays the generated tasks, study timer, and motivational messages.**

**2. User Flow:**

**- Step 1: User logs in or starts as a guest.**

**- Step 2:User enters their study goals, available hours, and preferred study techniques.**

**- Step 3: The app creates personalized study tasks based on user input.**

**- Step 4: Users can view tasks, set the study timer, and receive reminders and motivational messages.**

**3. UI/UX Considerations:**

**- Simple and intuitive interface for easy navigation.**

**- Options to filter tasks by category or priority.**

**- Dark and light mode for comfortable viewing in different lighting conditions.**

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

|  |  |
| --- | --- |
| **Component** | **Details** |
| **Vision** | **AI-powered study planner for personalized learning experiences.** |
| **Goals** | **Personalize schedules, implement AI, enhance engagement, ensure usability.** |
| **User Stories** | **Input preferences, track progress, receive reminders, get motivational quotes.** |
| **Backlog** | **Must Have: Authentication, Schedule, Tracking; Should Have: Calendar, Feedback; Could Have: Social sharing, Gamification.** |
| **Sprint Planning** | **Sprint 1: User Onboarding; Sprint 2: Schedule Generation; Sprint 3: Progress Tracking; Sprint 4: Reminders.** |
| **Roles** | **Product Owner: Vision & backlog; Scrum Master: Facilitation; Dev Team: Code; QA/Testers: Testing; Designer: Design.** |
| **Communication** | **Daily stand-ups, weekly meetings, tools like Trello/Jira/Asana.** |
| **Risk Management** | **Risks: Misalignment (surveys) and delays (workload reviews).** |
| **Testing** | **Unit tests, integration tests, user acceptance testing (UAT).** |
| **Deployment** | **CI/CD practices, staging environments, rollback strategies.** |
| **Feedback & Iteration** | **Regular surveys for feedback and planned iterations based on user input.** |

**Phase-5: Project Development**

**Objective:**

**To implement core features of the AI Study Planner App to provide users with a personalized and efficient study experience through tailored study schedules and progress tracking.**

**Key Points:**

**1. Technology Stack Used:**

**- Frontend:Streamlit**

**- Backend:Custom AI algorithms (potentially leveraging ML frameworks)**

**- Programming Language: Python**

**2. Development Process:**

**- Implement user authentication and profile management.**

**- Develop personalized schedule generation logic based on user inputs.**

**- Optimize progress tracking and reminder notification features.**

**3. Challenges & Fixes:**

**- Challenge: Delayed response from AI algorithms when processing preferences.**

**- Fix:Implement caching of user preferences and frequently accessed data to improve retrieval times.**

**- Challenge: High complexity in generating personalized schedules for varied learning styles.**

**- Fix: Streamline the algorithm using simplified heuristics and user feedback loops to refine output.**

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

**Objective:**

**Here are the key objectives for Functional and Performance Testing specifically tailored for a study planner web application:**

**1. User Registration/Login:**

**- Ensure users can register, log in, and log out successfully.**

**2. Profile Management:**

**- Verify users can create, edit, and delete their profiles.**

**3. Schedule Creation:**

**- Confirm users can create, modify, and delete study schedules and to-do lists.**

**4. Resource Management:**

**- Ensure users can add, edit, and remove study resources and materials.**

**5. Notification System:**

**- Check that users receive notifications and reminders for study sessions.**

**6. Collaboration Features:**

**- Test features allowing users to share schedules or resources with friends.**

**7. Search Functionality:**

**- Validate the search feature works for finding schedules, resources, and notes.**

**8. Reports and Analytics:**

**- Ensure users can generate and view performance reports and analytics.**

**Performance Testing Objectives:**

**1. Response Time:**

**- Measure the time taken for the application to respond to user actions (e.g., loading schedules).**

**2. Load Testing:**

**- Test how the application performs under a high number of concurrent users.**

**3. Stress Testing:**

**- Determine the application's breaking point by pushing it beyond normal load conditions.**

**4. Data Retrieval Speed:**

**- Evaluate the speed of retrieving and displaying large datasets (schedules, notes).**

**5. Page Load Times:**

**- Measure the loading speed of different pages in the application.**

**6. API Performance:**

**- Assess the response times of APIs used for data retrieval and user actions.**

**7. Mobile Performance:**

**- Ensure the application performs well on various mobile devices and screen sizes.**

**8. Resource Utilization:**

**- Monitor CPU and memory usage under different loads to assess efficiency.**

**Final Submission**

**1. Project Report Based on the templates**

**2. Demo Video (3-5 Minutes)**

**3. GitHub/Code Repository Link**

**4. Presentation**