**Hackathon Project Phases Template**  for the **STUDYMATE**  project.

**Hackathon Project Phases Template**

**Project Title:**

**StudyMate: An AI-Powered PDF-Based Q&A System for Students**

**Team Name:**

Logicrew

**Team Members:**

* K. Tejaswari
* K. Rishitha sri
* L. Jahnavi

**Phase-1: Brainstorming & Ideation**

**Objective:**

**StudyMate** is an AI-powered academic assistant designed to enhance how students interact with their study materials, especially textbooks and PDFs. The core objective is to make learning more efficient, interactive, and personalized by enabling users to ask natural language questions about their documents and receive intelligent, accurate, and explainable answers.

1. **Problem Statement:**
   * Despite the growing availability of digital textbooks and academic PDFs, students often struggle to **efficiently extract and understand information** from them.
   * Traditional study methods such as highlighting, searching, and note-taking are **time-consuming**, **passive**, and often **fail to support deep learning**, especially when students are under time pressure or preparing for competitive exams..
2. **Proposed Solution:**
   * To address the challenges students face when studying from static, text-heavy PDFs, **StudyMate** proposes an AI-powered, interactive study assistant that transforms how learners interact with academic content.
   * For transparency and trust, the system highlights or cites the **source paragraph** or sentence from the document used to generate the answer.
3. **Target Users:**
   * Primary target group. StudyMate is designed to help students learn more efficiently by interacting with their own study materials.
   * Especially helpful for adult learners revisiting academic topics or upskilling independently.
4. **Expected Outcome:**

The successful implementation of StudyMate is expected to result in a powerful and user-friendly AI platform that transforms how students and learners engage with their study materials.

**Phase-2: Requirement Analysis**

**Objective:**

The primary objective of the requirement analysis phase is to gather, define, and document all functional and non-functional requirements necessary to develop a robust, user-centric AI-powered study assistant.

**Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: PyMuPDF
   * Frontend: HTTP API,streamlit
   * Database: File Storage
2. **Functional Requirements:**
   * User ManagementDisplay **specifications, reviews, and comparisons** in an intuitive UI.
   * Document Upload & Process Question Answering
3. **Constraints & Challenges:**
   * Ensuring real-time updates from **HTTP API**.
   * Handling **API rate limits** and optimizing API calls.
   * 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 Interface (UI)**

** Input Module**

** Text Extraction**

** Question Parsing**

** NLP & Summarization Engine.**

1. **User Flow:**

**STEP 1: Upload PDF**  
 Student uploads study material (notes, textbooks, etc.).

**STEP 2:PDF Processing**  
 System extracts and preprocesses the text from the uploaded file.

**STEP 3:Ask a Question**  
 Student types a question related to the uploaded PDF.

**STEP 4:Question Understanding**  
 System uses NLP to analyze and understand the question.

**STEP 5:Context Retrieval**  
 Relevant parts of the PDF are identified using semantic search (vector DB).

**STEP 6:Answer Generation**  
 AI model (e.g., GPT) generates a precise answer based on retrieved content.

**STEP 7:Answer Display**  
 Student receives a clear, formatted answer (with highlights or explanations).

**Continue Learning**  
Student can ask more questions or upload new PDFs.

1. **UI/UX Considerations:**1. **Simple & Clean Interface**

**2.Easy PDF Upload**

3. **Question Input Box**

4. **Responsive Design**

**Phase-4:**

**Project Planning (AI BASED STUDY MATE)**

**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 | k.tejaswari | Google API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Member 2 | API response format finalized | Basic UI with input fields |
| Sprint 2 | Vehicle Search & Comparison | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | jahanvi | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1 hour (Day 2) | Mid-Day 2 | Member 1&4 | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 2 hours (Day 2) | Mid-Day 2 | Rishitha sri | API response, UI layout completed | Responsive UI, better user experience |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 2 hours (Day 2) | End of Day 2 | Entire Team | 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 **Google Gemini API**.  
 **(🟡 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 API issues & handle **errors in queries**.

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

**(🟡 Medium Priority)** Test API responses, refine UI, & fix UI bugs.  
 **(🟢 Low Priority)** Final **demo preparation & deployment**.

**Phase-5: Project Development**

**Objective:**

Implement core features of the AutoSage App.

**Key Points:**

1. **Technology Stack Used:**

**FORTEND:** HTTP API, streamlit

**BACKEND:** PyMuPDF

**PROGRAMMING LANGUAGE :** Python

**FILE HANDLING**: Pymupdf

**SEMATIC SEARCH** : Faiss

**Development Process:** Identify the problem: Students find it hard to search or understand content in long PDFs.

Goal: Create an AI-powered app to extract content and answer questions from PDFs.

1. **Challenges & Fixes:**

**StudyMate – Challenges & Fixes**

| **Challenge** | **Fix / Solution Implemented** |
| --- | --- |
| **Delayed API response times** | Implemented **caching** to store frequently asked questions and their answers. |
| **Limited API requests per minute (Rate limiting)** | Added **request throttling** and **retry mechanism** to manage Gemini API usage efficiently. |
| **Incorrect or irrelevant AI answers** | Improved **prompt formatting** and passed better context using **text chunking** logic. |
| **Extracted PDF text had formatting issues** | Used **fitz)** for better layout-preserving text extraction. |
| **UI broken on mobile or smaller devices** | Applied **responsive design** tweaks using **PyMuPDF (**Streamlit layout and tested across screen sizes. |
| **Streamlit app failed to load large PDFs** | Added **file size validation** and optimized text parsing for performance. |
| **Security risks in file upload** | Restricted upload to .pdf files only and added file type/size checks. |

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

**Final Submission**

**Project Report**  
– Submit a detailed project report using the provided template (includes architecture, tech stack, features, challenges, testing, etc.).

**Demo Video (3–5 minutes)**  
– Record a walkthrough showing how StudyMate works:

* Uploading PDF
* Asking questions
* Getting AI-generated answers
* Bonus: Show caching, mobile view, etc.

**GitHub/Code Repository Link**  
– Share the public GitHub repo with:

* Clean, commented code
* README.md explaining setup and usage
* Requirements file (requirements.txt)

**Final Presentation (PPT)**  
– Create a project presentation covering:

* Problem Statement
* Solution (StudyMate)
* Features
* Tech Stack
* System Architecture