HACKATHON

Project Title: StudBud Ai Study Planner

Team Name:

Code Crafters

Team Members:

- A.Akash Kumar
- A.Madhukar Reddy
- GK.Aashish
- D.Vinay Reddy
- K.Vignesh

Phase-1: Brainstorming & Ideation

Objective:

The objective of the StudBud AI Study Planner is to create personalized study schedules based on individual learning preferences, goals, and deadlines. It optimizes study time, reduces stress, and tracks progress to ensure academic success. The AI adapts to improve focus, motivation, and performance over time.

Key Points:

1. **Problem Statement:**

   Design and implement an AI-powered **Study Planner** using Python that helps students create efficient, personalized study schedules based on their available time, academic goals, and deadlines. The system should prioritize tasks, track progress, and adapt the study plan based on performance. It should suggest effective study techniques, resources, and provide feedback to enhance productivity and academic success.

   2.Proposed Solution:

   The AI-powered Study Planner will use Machine Learning to generate personalized study schedules based on student inputs like goals,

deadlines, and study preferences. Reinforcement Learning will adapt the plan in real-time, optimizing task prioritization and study techniques. Predictive Analytics will track progress and offer personalized feedback, while recommendation algorithms (Content-based and Collaborative Filtering) will suggest relevant study resources. The system will continuously improve by learning from the student's performance, ensuring an efficient, tailored learning experience.

3.Target Users:

- **Students: High school, college, and graduate students needing personalized study schedules and task management.**
- **Working Professionals: Individuals studying for certifications or learning new skills alongside their jobs.**
- **Tutors and Educators: Teachers and tutors seeking tools to help students structure their study sessions.**
- **Parents: Parents who want to help their children stay organized and focused on their studies.**

4.Expected Outcome:

- **Improved Time Management: Efficiently balancing study time and deadlines.**
- **Increased Academic Performance: Better exam scores through personalized plans.**
- **Reduced Stress: Lower anxiety with organized, adaptive study schedules.**
- **Higher Engagement: Increased motivation with continuous feedback and tracking.**
- **Optimized Learning: Effective study techniques and resource recommendations.**

Phase-2: Requirement Analysis

Objective:

The objective of the Requirement Analysis for the AI Study Planner is to clearly define the functional, non-functional, system, and user requirements

that the system must fulfill. This analysis ensures the AI Study Planner is tailored to meet the needs of its target users, delivering personalized study schedules, task prioritization, progress tracking, and resource recommendations. It also outlines the technical infrastructure, performance expectations, and security considerations necessary to create an intuitive, adaptive, and scalable study planning tool.

Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Backend: **Google Gemini API.**
    - Frontend: **Streamlit Web Framework**
    - Database: **Mongo DB**
2. **Functional Requirements:**
    - **User Management**: Allow account creation, login, and profile customization (study goals, time, deadlines).
    - **Study Schedule**: Generate personalized study timetables based on user inputs and optimize with techniques like Pomodoro.
    - **Task Prioritization**: Prioritize tasks based on urgency, deadlines, and difficulty, with dynamic adjustments.
        - **Progress Tracking**: Track completed tasks and provide progress feedback and adjustments as needed.
        - **Resource Recommendations**: Suggest relevant study resources based on user progress and subject focus.
        - **User Interface: Provide an intuitive interface for easy schedule viewing and task management.**
          Security: Ensure user data is securely stored and protected with strong encryption.

Constraints & Challenges:

- **Data Privacy and Security**: Ensuring user data is securely stored and complies with regulations (e.g., GDPR).

- **AI Accuracy**: Providing accurate study plans and recommendations based on user input.

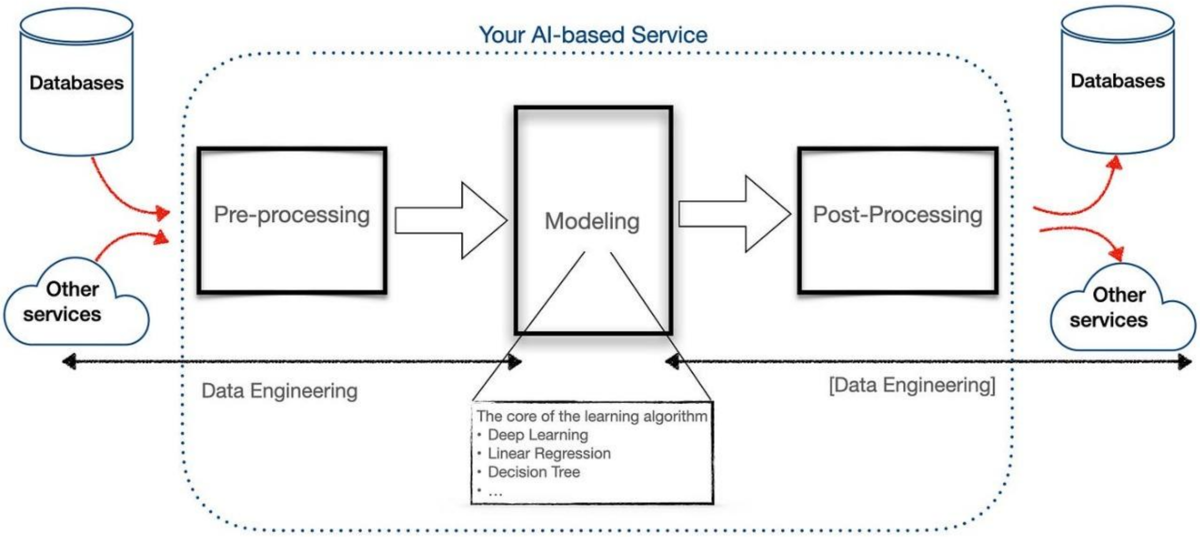
- **Adaptability**: Continuously adjusting study plans in real-time based on user performance.

- **Usability**: Designing a user-friendly interface for students with varying tech skills.
- **Handling Diverse Learning Styles**: Accommodating different study preferences and learning methods.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1.System Architecture:

- **Client-Side (Frontend)** – Built using **React (Next.js) for Web** and **Flutter for Mobile**, featuring a **user dashboard, smart calendar, and AI-powered study planner**.

- **Backend (Microservices Architecture)** – Powered by **Node.js (NestJS) or Python (FastAPI)** with dedicated services for **user management, study planning, progress tracking, notifications, and AI-based recommendations**.

- **Database & Storage** – Uses **PostgreSQL for structured data**, **Redis for caching**, and **Firebase/AWS S3 for storing study materials** like notes and PDFs.
- **AI & Recommendation Engine** – Implements **Machine Learning (TensorFlow, PyTorch)** to personalize study plans, track user progress, and optimize learning strategies dynamically.
- **Cloud & DevOps** – **Deployed on AWS/GCP/Azure** with **Docker + Kubernetes** for scalability, **CI/CD pipelines (GitHub Actions/Jenkins)** for continuous deployment, and **monitoring tools (Prometheus, Grafana)** for performance tracking.

2.User Flow:

- **AI-Powered Study Plan Generation**

  AI analyzes **user input, deadlines, and difficulty levels**
  Creates a **personalized study schedule** with optimal time slots
- **Study Session & Progress Tracking**

  Users follow the **AI-recommended study plan**
  Log completed tasks, take notes, and use timers
  AI **analyzes efficiency and suggests improvements**
- **Adaptive Learning & Smart Recommendations**

  AI adjusts the **study plan dynamically** based on progress
  Provides **smart tips, difficulty-based scheduling, and reminders**
- **Performance Insights & Gamification**

  Dashboard shows **progress analytics, study streaks, and achievements**
  Users get **motivation boosts through rewards & challenges**

3.UI/UX Considerations:

- **Minimalist & User-Friendly Design – Clean, distraction-free UI with dark/light mode and a consistent design system.**
- **Intuitive Navigation – Simple onboarding, clear dashboard, and easy task management (drag & drop, quick edits).**
- **AI-Powered Personalization – Smart study plan suggestions, adaptive learning, and gamification (streaks, badges, rewards).**

- **Cross-Platform & Mobile-First – Responsive design, offline mode, and cloud sync for a seamless experience across devices.**
- **Engagement & Motivation Features – Pomodoro study timers, social learning, and AI chatbot assistance for an interactive experience.**

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	Akash Kumar	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	Akash kumar, Madhukar Vignesh	API response format finalized	Basic UI with input fields
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	end-Day 2	Akash	API logs, UI inputs	Improved API stability
Sprint 3	Backend Development	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Akash Aashish	API response, UI	Responsive UI, better user

					vinay	layout completed	experience
Sprint 3	Final Presentation & Deployment	🟢 medium	1 hour (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)Backend Development.

(🟡 Medium Priority) final Presentation

Phase-5: Project Development

Objective:

Implement core features of the Study Planner.

Key Points:

1. **Technology Stack Used:**

   - **Frontend:** Streamlit
   - **Backend:** Google Gemini Flash Api
   - **Programming Language:** Python

2. **Development Process:**

   - Implement **API key authentication** and **Gemini API integration**.
   - Develop **Study Plansand maintenance tips logic**.
   - Optimize **search queries for performance and relevance**.

3. **Challenges & Fixes:**

- **Challenge: AI Study Plan Accuracy** – Users have different learning speeds and study patterns.
  Fix: Use **adaptive AI algorithms** that analyze user progress and adjust schedules dynamically.
- **Challenge: User Engagement & Retention** – Users may lose motivation over time.
  Fix: Implement **gamification (streaks, rewards, social study groups)** and **personalized reminders** to keep users engaged.
- **Challenge: Scalability & Performance** – Handling a large number of users efficiently.
  Fix: Use **microservices architecture, caching (Redis), and auto-scaling cloud servers** for seamless performance.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the StudBud Ai Study Plannerworks as expected.

Test Case ID	**Category**	**Test Scenario**	**Expected Outcome**	**Status**	**Tester**

TC-001	Functional Testing	Query "Learning Platform”	Verified the learning course	✅ Passed	Akash
TC-002	Functional Testing	Query "AI Chatbox for learing”	Seasonal tips should be provided.	✅ Passed	Madhukar
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Aashish
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Vignesh
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	Vinay
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	🚀 Deployed	DevOps

---

Final Submission

1. https://github.com/Akashkumar208/fictional-invention
2. https://1drv.ms/p/c/ff62ddd3c3a13230/ESkxZingbDRNi1N1Kz3834QBTcstJ6D9-vK7laQA-gkMug?e=6ZBx2a