
Project Title:

StudBud : AI Study planner

Team Name:

Team Innovators

Team Members:

- S.Dileep Sagar
 - B.Siri Vennela
-

Phase-1: Brainstorming & Ideation

Objective:

The Study Timetable Generator is a web-based tool that helps students create optimized study plans by efficiently allocating study hours based on subjects, chapters, and available time. It allows dynamic subject management and features a modern UI.

Key Points:

1. Problem Statement:

- Students struggle with time management and effective study planning, leading to inefficient exam preparation.
- Existing study planners lack personalization based on subjects, chapter distribution, and available study hours.

2. Proposed Solution:

- Develop StudBud AI Study Planner, an intelligent tool that generates optimized study schedules based on user inputs.
- Utilize AI-powered recommendations to allocate study time efficiently, ensuring a well-structured preparation plan.

3. Target Users:

- Students preparing for exams across various academic levels.
- Educators & Institutions seeking personalized study strategies for their students..

4. Expected Outcome:

- Improved **time management** and **study efficiency** with a structured AI-generated timetable.
 - Higher **academic performance** due to optimized study schedules tailored to individual needs.
-

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the StudBud : AI Study Planner.

Key Points:

1. Technical Requirements:

- Programming Language: **Python**
- Backend: **OpenAI API (GPT -4)**
- Frontend: **Streamlit Web Framework**
- Database: **Not required initially (API-based queries)**

2. Functional Requirements:

- Generate **personalized study plans** based on subjects, chapters, available days, and study hours.
- Use AI to **optimize time allocation** based on subject difficulty and study hours.
- Provide an **intuitive and interactive UI** for entering study details and viewing schedules.
- Allow **dynamic addition/removal of subjects** for flexibility.

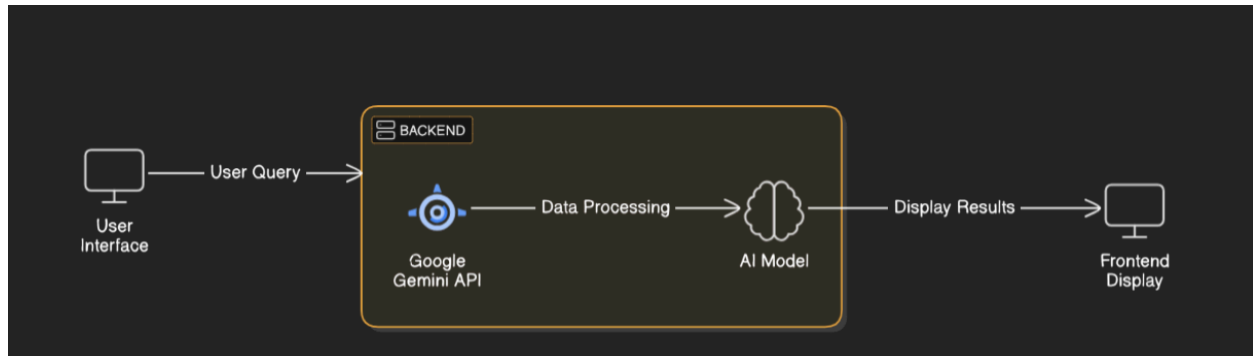
3. Constraints & Challenges:

- **Ensuring accurate study plan generation** based on varied user inputs.
 - **Managing API rate limits** for OpenAI's GPT-4.
 - **Optimizing UI performance** in Streamlit for a seamless experience.
-

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. System Architecture:

- **User inputs study details** (subjects, number of chapters, available days, and daily study hours) via the UI.
- **Query is processed using OpenAI's GPT-4 API** to generate an optimized study plan.
- **AI model structures the study schedule** based on subject difficulty and time constraints.
- **The frontend displays the study timetable** in a clear and structured format.

2. User Flow:

- Step 1: User enters details like subjects, chapters, available days, and daily study hours.
- Step 2: The backend calls **OpenAI's API** to generate an optimized study plan.
- Step 3: The app **processes and displays the study timetable** in a well-organized table format.

3. UI/UX Considerations:

- **Minimalist, visually appealing UI** with a blurred book-themed background.
 - **Dynamic subject management** allowing users to add/remove subjects easily.
 - **Intuitive layout** for entering details and viewing structured schedules.
 - **Interactive elements** like buttons for generating or modifying timetables.
-

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	Siri Vennela	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	● Medium	2 hours (Day 1)	End of Day 1	Dileep Sagar	API response format finalized	Basic UI with input fields
Sprint 2	Study Plan Generation Logic	● High	3 hours (Day 2)	Mid-Day 2	Siri Vennela	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	● High	1.5 hours (Day 2)	Mid-Day 2	Dileep Sagar	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	● Medium	1.5 hours (Day 2)	Mid-Day 2	Siri Vennela	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	● Low	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 OpenAI API (GPT-4) for study plan generation.
- (● Medium Priority) Build a basic UI with input fields using Streamlit.

Sprint 2 – Core Features & Debugging (Day 2)

- (● High Priority) Implement dynamic subject management (add/remove subjects).
- (● High Priority) Optimize AI-generated study plans based on time constraints.
- (● High Priority) Debug API issues, handle errors, and improve response accuracy.

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

- (🟡 **Medium Priority**) Test API responses, refine UI, and fix layout inconsistencies..
 - (🟢 **Low Priority**) Final demo preparation, deployment, and project documentation.
-

Phase-5: Project Development

Objective:

Implement core features of the AutoSage App.

Key Points:

1. Technology Stack Used:

- **Frontend:** Streamlit
- **Backend:** OpenAI GPT-4 API
- **Programming Language:** Python

2. Development Process:

- Implement **API key authentication** and integrate OpenAI API for study plan generation.
- Develop **dynamic subject management** (add/remove subjects) and time allocation logic..
- Optimize study schedules based on **user inputs** (subjects, difficulty level, study hours, exam date).

3. Challenges & Fixes:

- **Challenge:** Generating a balanced study schedule across multiple subjects.
Fix: Implement weighted time allocation based on subject difficulty and chapter count.
 - **Challenge:** Ensuring quick API responses for real-time study plan updates.
Fix: Optimize API calls by caching recent responses and limiting redundant queries.
 - **Challenge:** UI responsiveness and readability
Fix: Improve UI design with a blurred book-themed background, clear text, and structured tables for study plans.
-

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	User enters subjects, chapters, and available days/hours.	AI should generate a structures and optimized timetable.	✅ Passed	Dileep Sagar
TC-002	Functional Testing	User tries to add more subjects dynamically.	Subjects should be added without errors.	✅ Passed	Siri Vennela
TC-003	Performance Testing	API response time under 500ms.	Study plan should be created quickly.	⚠ Needs Optimization	Dileep Sagar
TC-004	Bug Fixes & Improvements	Fixed incorrect time allocation for subjects.	Study Plan accuracy should be improved	✅ Fixed	Siri Vennela
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on desktop.	❌ Failed - UI broken on mobile	Dileep Sagar
TC-006	Deployment Testing	Host the website using Streamlit Sharing	Website should accessible online..	🚀 Deployed	Siri Vennela

Final Submission

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