**Project Title:**

**FitSync AI: Real-Time Fitness Adjustments**

**Team Name:**

The\_Avengers

**Team Members:**

* Vangolu Venugopal
* Aasritha Mulkalapally
* Karthikeya Rankireddy
* M Sai Sampath
* S Adithya Reddy

## Phase-1: Brainstorming & Ideation

**Objective:**

Develop an AI-powered fitness tool that provides real-time workout adjustments and personalized coaching using the Gemini Flash model.

**Key Points:**

1. **Problem Statement:**

* Many fitness enthusiasts struggle to maintain motivation or break through plateaus because their workout routines become stagnant.
* Users need dynamic, data-driven guidance that adapts to their evolving performance and recovery needs.

1. **Proposed Solution:**

* An AI-powered application that leverages Gemini Flash to analyze workout inputs, wearable metrics, and historical performance to generate personalized workout adjustments, new exercise suggestions, and recovery tips..

1. **Target Users:**

* Fitness enthusiasts and athletes seeking to optimize their training.
* Individuals looking for real-time coaching to prevent injuries and maximize workout efficiency.

1. **Expected Outcome:**

* A functional application that delivers real-time, personalized fitness advice and adjustments based on user data and queries.

## Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for the FitSync AI.

**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language: **Python**

○ Backend: **Google Gemini Flash API**

○ Frontend: **Streamlit Web Framework**

○ Database: **MongoDB**

**Functional Requirements:**

○ Fetch and process user workout data (manual entry or integration with wearables like Apple HealthKit, Google Fit).

○ Generate personalized exercise adjustments, intensity modifications, and recovery suggestions using Gemini Flash.

○ Display real-time workout insights and progress dashboards in an intuitive UI.

○ Provide seasonal or situational maintenance tips (e.g., recovery strategies during high-intensity periods).

1. **Constraints & Challenges:**

* 1. Ensuring real-time response and accurate recommendations from the Gemini Flash API.

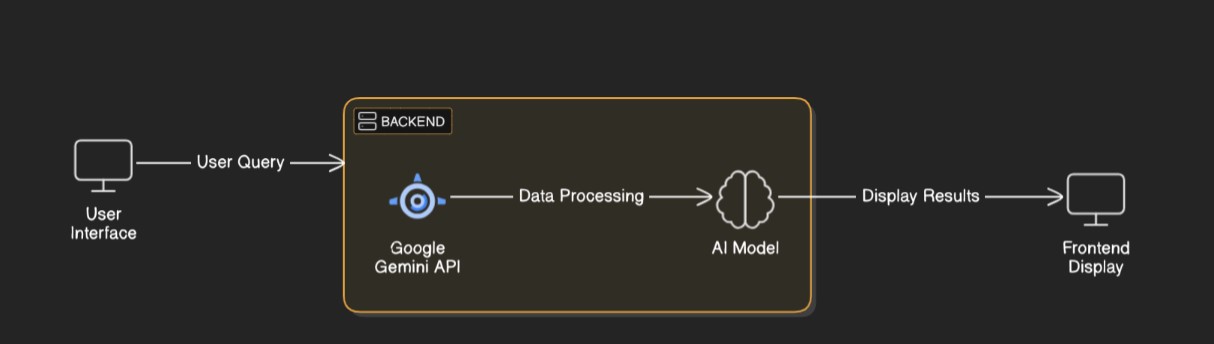
○ Managing API rate limits while delivering consistent performance.

○ Creating a seamless and engaging user interface with minimal latency.

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the application.



**Key Points:**

1. **System Architecture:**

* 1. User enters his and his workout details

○ Query is processed using **Google Gemini API**.

○ AI model fetches and processes the data.

○ The frontend displays **Gemini Response Of Strengths, weakness of current plan , new workout plan, Nutrition Diet plan, Intensity details, Extra Advices, tracking plans and comparisons**.

1. **User Flow:**

* 1. Step 1: User enters a query ("age": 20,

    "gender": "Male",

    "fitness\_level": "Intermediate",

    "weight": 73,  # kg

    "height": 180,  # cm

    "goal":"six pack(abs)",

    "workout\_history": [

        {"exercise": "Squat", "duration": 45, "intensity": "High", "sets": 3, "reps": 10, "weight\_used": 60},

        {"exercise": "Bench Press", "duration": 40, "intensity": "Medium", "sets": 3, "reps": 8, "weight\_used": 50}

    ],

    "performance\_metrics": {

        "heart\_rate": 140,

        "calories\_burned": 450,

        "progress\_over\_time": "Improving"

    },

    "recovery\_data": {

        "sleep\_quality": "Good",

        "rest\_days": 2,

        "soreness\_levels": "Mild"

    }

)

○ Step 2: The backend **calls the Gemini Flash API** to retrieve response.

○ Step 3: The app processes the data and **displays results** in an easy-to-read format.

1. **UI/UX Considerations:**

**○ Minimalist, user-friendly interface** for seamless navigation.

○ **Filters for price, mileage, and features**.

○ **Dark & light mode** for better user 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 | Member 1 | 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 | Real-Time Workout Plan Generation | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Member 1& 2 | API response, UI elements ready | AI-generated fitness recommendations & plan generation |
| Sprint 2 | Error Handling &  Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | Member 1&4 | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Member 2& 3 | 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 Overview:**

* Sprint 1 – Setup & Integration (Day 1):
  + Tasks:
    - Set up the development environment (Python, Streamlit, or chosen frontend framework).
    - Integrate the Google Gemini Flash API with proper API key authentication.
    - Build a basic UI to accept workout data inputs.
  + Priority: High
  + Expected Outcome: A stable API connection and a basic input interface for workout data.
* Sprint 2 – Core Features & Debugging (Day 2, Mid-Day):
  + Tasks:
    - Implement core features for processing user data and generating real-time fitness recommendations.
    - Integrate filters and UI components for viewing exercise adjustments and recovery tips.
    - Debug API issues and optimize query handling (e.g., rate limit management, caching).
  + Priority: High
  + Expected Outcome: Functional real-time search and personalized adjustment features.
* Sprint 3 – Testing, Enhancements & Finalization (Day 2, End of Day):
  + Tasks:
    - Conduct functional and performance testing to ensure reliability and responsiveness.
    - Refine UI elements based on feedback to improve usability.
    - Prepare the final demo and deploy the application.
  + Priority: Medium to Low
  + Expected Outcome: A demo-ready, fully functional prototype of FitSync AI.

## Phase-5: Project Development

**Objective:**

Implement core features of the AutoSage App.

**Key Points:**

1. **Technology Stack Used:**

* 1. **Frontend:** Streamlit

○ **Backend:** Google Gemini Flash API

○ **Programming Language:** Python

1. **Development Process:**

* 1. Develop algorithms that combine real-time data with historical performance to offer actionable fitness advice.

○ Ensure the AI model can generate dynamic suggestions (e.g., adjusting workout intensity, suggesting recovery tips)..

○ Implement caching and query optimization to handle API rate limits and ensure quick responses.

1. **Challenges & Fixes:**

* 1. **Challenge:** Delayed API response times.

**Fix:** Implement **caching** to store frequently queried results.

○ **Challenge:** Limited API calls per minute.

**Fix:** Optimize queries to fetch **only necessary data**.

## 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 | Query: "Generate a workout plan for six-pack abs" | AI generates a structured weekly workout plan with personalized recommendations. | ✅ Passed | Tester 1 |
| TC-002 | Functional  Testing | User modifies intensity and duration in UI | AI dynamically adjusts workout recommendations | ✅ Passed | Tester 2 |
| TC-003 | Performance  Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | AI suggestions contain incorrect workout metrics | Fix applied; AI now generates accurate intensity levels | ✅ Fixed | Develop er |
| TC-005 | Final Validation | Ensure UI is responsive on mobile and web | UI should work smoothly on all devices | ❌ Failed - UI broken on mobile | Tester 2 |
| TC-006 | Deployment  Testing | Host the app using  Streamlit Sharing | App should be accessible online. | 🚀 Deployed | DevOps |

## Final Submission

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