Hackathon Project

FINANCIA

# Project Idea:

**Personal Finances Chat-bot Using IBM Granite**

# Team Name:

Team ArmaRouge

# Team Members:

KENGAM MOHAN KRISHNA

KROVI MOHAN KOUSHIK

KRISHNA VARDHAN

# Phase-1: Brainstorming & Ideation

## Objective:

Develop an AI-powered personal finance chatbot using IBM Granite & Gemini Flash to help users manage budgets, track expenses, and plan investments.

## Key Points:

1. **Problem Statement:**

* Many users struggle with budgeting, expense tracking, and investment planning due to lack of financial literacy.
* Users need personalized financial advice tailored to their income, goals, and risk tolerance.

1. **Proposed Solution:**

* An AI AI-powered chatbot using IBM Granite & Gemini Flash to provide real-time financial insights, budget summaries, and investment recommendations.
* The app offers personalized financial planning based on user profiles and goals.

1. **Target Users:**

* Students learning to manage finances.
* Young Professionals starting their financial journey.
* Mid-Career Professionals planning for long-term goals.
* Retirees managing savings and investments.

1. **Expected Outcome:**

* A functional AI-powered finance chatbot that provides actionable insights and personalized advice.

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the Personal Finance Chatbot.

## Key Points:

1. **Technical Requirements:**
   * **Programming Language:**Python
   * **Backend:**IBM Granite 3.2.2b Hugging Face API, Gemini Flash API
   * **Frontend:**Streamlit Web Framework, Gradio
   * **Database:**Session state for temporary data storage
2. **Functional Requirements:**
   * Ability to fetch financial insights using Gemini Flash API.
   * Display budget summaries, expense breakdowns, and investment recommendations in an intuitive UI.
   * Provide personalized financial advice based on user profiles.
   * Allow users to input and track financial data.
   * Providing Critical Thinking Analysis of Solution using IBM Granite.
3. **Constraints & Challenges:**
   * Ensuring real-time responses from Gemini API.
   * Handling API rate limits and optimizing queries.
   * Providing a user-friendly interface with Streamlit and Gradio

# Phase-3: Project Design

## Objective:

Develop the architecture and user flow of the application.

┌───────────────────────┐

│ USER INTERFACE │

│ • Data Input Form │

│ • Chat Query Box │

└───────────┬───────────┘

│

┌───────────▼───────────┐

|GEMINI FLASH API & IBM GRANITE │

│ DEPLOYED LINK |

| • Process Data │

│ • Generate Insights │

└───────────┬───────────┘

│

┌───────────▼───────────┐

│ OUTPUT DASHBOARD │

│ • Budget Charts │

│ • Text Advice │

└───────────────────────┘

## Key Points:

## System Architecture:

## User inputs financial data and queries via UI.

## Query is processed using Google Gemini API.

## AI model generates personalized financial advice.

## The frontend displays budget summaries, insights, and recommendations.

## User Flow:

## Step 1: User sets up profile and inputs financial data.

## Step 2: The backend calls the Gemini Flash API to generate insights.

## Step 3: The app displays personalized financial advice in an easy-to-read format.

## UI/UX Considerations:

## Minimalist, user-friendly interface for seamless navigation.

## Interactive charts for visualizing financial data.

## 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 | KENGAM MOHAN KRISHNA | Google API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day 1 | Members 2&3 | API response format finalized | Basic UI with input fields |
| Sprint 2 | Financial Data Input & Analysis | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | KENGAM MOHAN KRISHNA | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | KROVI MOHAN KOUSHIK | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Members 1&2 | 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 **Google Gemini API** & **IBM GRANITE**

**(**🟡 **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 Personal Finance Chatbot.

**Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** IBM Granite & Google Gemini Flash API
   * **Programming Language:** Python
2. **Development Process:**
   * Implement **API key authentication** and **Gemini API integration**.
   * Develop **budget analysis and investment recommendation logic**.
   * Optimize **queries for performance and relevance**.
3. **Challenges & Fixes:**
   * **Challenge:** Delayed API response times.

**Fix:**  Implement **caching** for 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 Financia App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Query "Generate budget summary" | Detailed Budget Summary Displayed. | Passed | KROVI MOHAN  KOUSHIK |
| TC-002 | Functional Testing | Query "Investment advice for my profile" | Personalized Investment  Recommendations. | Passed | KROVI MOHAN KOUSHIK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | Needs Optimization | KENGAM MOHAN KRISHNA |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | Fixed | KENGAM  MOHAN  KRISHNA |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | Failed - UI broken on mobile | KROVI MOHAN KOUSHIK |
| TC-006 | Deployment Testing | Host the app using Streamlit Sharing | App should be accessible online. | Deployed | KENGAM MOHAN KRISHNA |

# Final Submission

1. **Project Report**
2. **GitHub/Code Repository Link**
3. **Demo Video:** https://drive.google.com/file/d/1IPPkDRddaAIYoXaO323gQa2AbAk8EqEM/view?usp=drive\_link
4. **Presentation**