

# Hackathon Project Phases Template

## Project Title:

Gemini Pro Financial Decoder Using Gemini Pro AI

## Team Name:

(Financial decoders)

## Team Members:

- Srisumana Pottabathini
  - Laya Uppu
  - Vennapusa Rajeswari
  - Harshitha Reddy
- 

## Phase-1: Brainstorming & Ideation

### Objective:

Develop an AI-powered financial data expert tool using Gemini Pro AI to help users compare and analyze financial data specifications, reviews, and eco-friendly options.

### Key Points:

#### 1. Problem

#### Statement:

- Many users struggle to find reliable, up-to-date information about various financial instruments before making a purchase decision.
- Users also need guidance on financial data maintenance and eco-friendly financial data choices.

#### 2. Proposed

#### Solution:

- An AI-powered application using Gemini Pro AI to provide real-time financial data specifications, reviews, and comparisons.
- The app offers maintenance tips and eco-friendly financial data insights based on user preferences.

### 3. Target

**Users:**

- Investors looking for stock analysis and financial insights.
- Financial analysts requiring automated risk assessment.
- Eco-conscious consumers searching for hybrid and electric financial data options.

### 4. Expected

**Outcome:**

- A functional AI-powered financial data information app that provides insights based on real-time data and user queries.

## Phase-2: Requirement Analysis

### Objective:

Define the technical and functional requirements for the Gemini Pro Financial Decoder.

### Key Points:

#### 1. Technical

**Requirements:**

- Programming Language: Python, TensorFlow, Pandas
- Backend: Google Gemini Pro AI API
- Frontend: Web-based Dashboard & Mobile App
- Database: Cloud-based financial data warehouse for real-time insights

#### 2. Functional

**Requirements:**

- Ability to fetch financial data details using Gemini Pro AI API.
- Display **specifications, reviews, and comparisons** in an intuitive UI.
- Provide real-time financial data maintenance tips based on seasons.
- Allow users to search eco-friendly financial datas based on emissions and incentives.

#### 3. Constraints

**&**

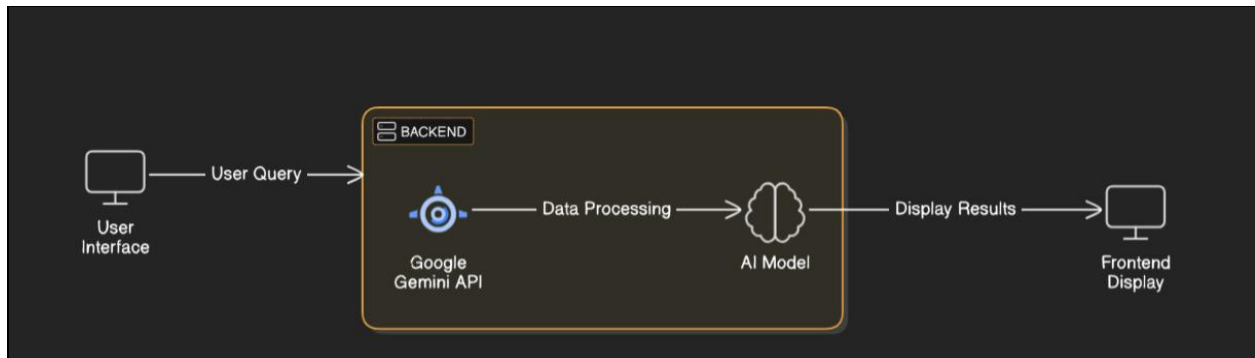
**Challenges:**

- Ensuring real-time updates from **Gemini 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 enters financial data-related query via UI.
- Query is processed using AI-powered financial models for trend analysis.
- AI model fetches and processes the data.
- The frontend displays financial data details, reviews, and comparisons.

### 2. User

### Flow:

- Step 1: User enters a query (e.g., "Best motorcycles under ₹1 lakh").
- Step 2: The backend calls the Gemini Pro AI API to retrieve financial data data.
- Step 3: The app processes the data and **displays results** in an easy-to-read format.

### 3. 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	Shanawaz	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	anwar	API response, UI elements ready	Search functionality with filters
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 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 Gemini Pro Financial Decoder.

Key Points:

1. Technology

Stack

Used:

  - Frontend: Streamlit
  - Backend: Google Gemini Pro AI API
  - Programming Language: Python, TensorFlow, Pandas
2. Development

Process:

  - Develop secure API connections for financial data retrieval and AI-powered analysis.
  - Develop financial data comparison and maintenance tips logic.
  - Optimize AI-driven financial forecasting models for real-time analysis.
3. Challenges

&

Fixes:

  - Challenge: Ensuring real-time financial data processing.  
Fix: Implement AI model optimization for faster financial predictions.
  - Challenge: Handling large-scale financial datasets with minimal latency.  
Fix: Implement data pre-processing and AI-driven summarization.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Gemini Pro Financial Decoder works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "Best budget cars under ₹10 lakh"	Relevant budget cars should be displayed.	✅ Passed	Tester 1
TC-002	Functional Testing	Query "Motorcycle maintenance tips for winter"	Seasonal tips should be provided.	✅ 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	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Developer

TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	✗ 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**