**Hackathon Project Phases Template** for the **Gemini Pro Financial Decoder** project

Hackathon Project Phases Template

# Project Title:

# Gemini Pro Financial Decoder

# Team Name:

# Algos

# Team Members:

# Harshitha

# Akshaya

# Tejaswi

# Phase-1: Brainstorming & Ideation

## Objective:

Develop an AI-powered financial analysis tool using **Gemini Pro** to help users decode financial data, analyze trends, and provide intelligent insights for informed decision-making.

## Key Points:

1. **Problem Statement:**

* Many individuals and businesses struggle to understand complex financial data and market trends.
* Traditional financial analysis tools require manual effort and expertise to interpret financial statements, stock trends, and market reports.

1. **Proposed Solution:**

* An AI-powered application utilizing **Gemini Pro** to process financial data and provide real-time market insights.
* The system will assist users by summarizing reports, predicting trends, and offering actionable investment suggestions based on data analysis.

1. **Target Users:**

* Individual investors looking for stock market insights.
* Financial analysts who need quick summaries of reports.
* Businesses seeking AI-driven financial forecasting.

1. **Expected Outcome:**

* A functional AI-driven **financial decoder** capable of analyzing market trends, summarizing financial reports, and providing investment insights.

# 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 & Node.js
* **Backend:** Gemini Pro API
* **Frontend:** React.js (or Streamlit for a simple interface)
* **Database:** MongoDB for storing user queries and analysis history

1. **Functional Requirements:**

* Fetch and analyze financial reports using **Gemini Pro API**.
* Provide trend predictions and market analysis.
* Summarize financial statements and offer investment insights.
* Enable users to query specific stocks, industries, or economic trends.

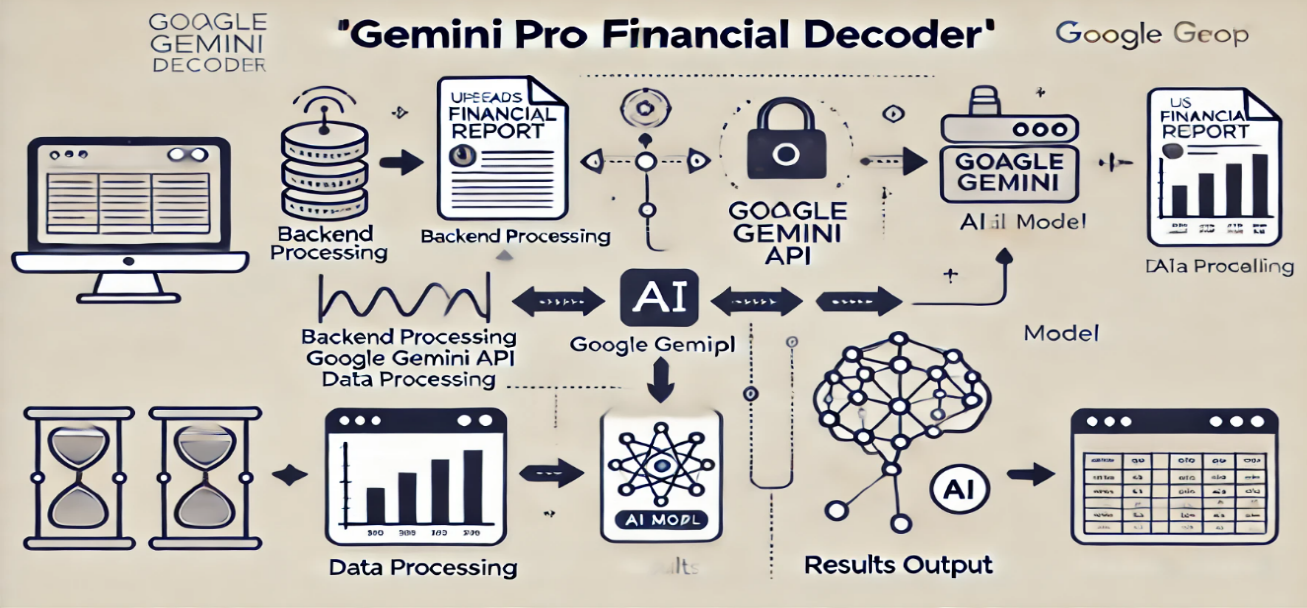
1. **Constraints & Challenges:**

* Ensuring real-time financial data updates.
* Handling API rate limits efficiently.
* Providing an intuitive and user-friendly interface.

# Phase-3: Project Design

## Objective:

Develop the architecture and user flow of the application.

`

## Key Points:

1. **System Architecture:**

* User inputs financial-related queries via UI.
* Query is processed using **Gemini Pro API**.
* AI model fetches and analyzes the financial data.
* The frontend displays insights, summaries, and trend predictions.

1. **User Flow:**

* **Step 1:** User enters a financial query (e.g., "Analyze Tesla's Q4 earnings").
* **Step 2:** The backend calls the **Gemini Pro API** to retrieve financial data.
* **Step 3:** AI processes the data and generates insights in a structured format.

1. **UI/UX Considerations:**

* Minimalist, user-friendly interface for easy navigation.
* Graphical representation of trends and financial analysis.
* Light & dark 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 | Harshitha | API Key, Backend Setup | API connection established |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day 1 | Akshaya | API response format | Basic UI with  Input fields |
| Sprint 2 | Financial Data Analysis | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Tejaswi | API response, UI elements | Financial Data  Processing |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | Harshitha & Akshaya | API logs, UI inputs | Improved  API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Tejaswi & Akshaya | API response, UI layout | Responsive UI,  Better 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 development environment & install dependencies. **🔴 High Priority -** Integrate Google Gemini API for financial data analysis. **🟡 Medium Priority -** Build a basic UI for input fields and output display.

**Sprint 2 – Core Features & Debugging (Day 2)**

**🔴 High Priority -** Implement financial data processing & decoding functionalities.  
**🔴 High Priority -** Debug API responses & handle errors effectively.

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

**🟡 Medium Priority -** Test API responses, refine UI, and fix UI bugs. **🟢 Low Priority -** Prepare final demo & deploy the application.

# Phase-5: Project Development

## Objective:

Implement core features of the **Gemini Pro Financial Decoder**.

## Key Points:

1. **Technology Stack Used:**

* **Frontend:** React.js / Streamlit
* **Backend:** Node.js & Python
* **Database:** MongoDB
* **API:** Gemini Pro API

1. **Development Process:**

* Implement **API authentication** and Gemini Pro API integration.
* Develop **financial analysis algorithms** for trend detection.
* Optimize **queries for accurate and fast financial insights**.

1. **Challenges & Fixes:**

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

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

* **Challenge:** Handling inconsistent or missing financial data

**Fix:** Implement fallback mechanisms and data validation

# 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 | Upload a financial report (PDF/CSV) | File should be successfully uploaded. | ✅ Passed | Tejaswi |
| TC-002 | Functional Testing | Generate AI summary for the uploaded file | AI should provide an  accurate summary. | ✅ Passed | Akshaya |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC-003 | Performance Testing | AI summary generation  time under 3 seconds | Summary should be generated quickly. | ⚠ Needs Optimization | Harshitha |
| TC-004 | Bug Fixes & Improvements | Fix incorrect extraction  of key financial metrics | AI should extract correct financial insights. | ✅ Fixed | Harshitha |
| TC-005 | Final Validation | Ensure AI summary output  is formatted correctly | Summary should be clear and structured. | ❌ Failed - Formatting issues | Tejaswi |
| TC-006 | Deployment Testing | Integrate data visualization for summary insights | Visualization should be displayed correctly | ✅Passed | Akshaya |

# Final Submission

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**