Hackathon Project Phases Template

# Project Title:

**DataQueryAI: Intelligent Data Analysis with Google TAPAS**

# Team Name:

QueryCrafters

# Team Members:

* Pendli Akshaya
* Mandula Likhita Sree
* Kokkula Sadhvika

# Phase-1: Brainstorming & Ideation

## Objective:

Develop an AI-powered data analysis tool using Google TAPAS to help users analyze and extract insights from CSV data through natural language queries.

## Key Points:

1. **Problem Statement:**
   * Users struggle to analyze large datasets manually, requiring technical expertise in tools like Excel, SQL, or Python.
   * Businesses need faster, more accurate data analysis to make data-driven decisions.
   * Non-technical users find it challenging to derive insights from raw data without coding knowledge.
2. **Proposed Solution:**
   * An AI-powered application using Google TAPAS to process natural language queries over tabular data (CSV files).
   * The app offers **maintenance tips** and **eco-friendly vehicle insights** based on user preferences.
3. **Target Users:**
   * **Business Analysts** who need quick insights from data.
   * **Non-technical users** who want to analyze data without coding.
   * **Educators and Students** learning data analysis concepts.
4. **Expected Outcome:**
   * A functional AI-powered data analysis tool that provides accurate, real-time insights from uploaded CSV files.

# Phase-2: Requirement Analysis

## Objective:

Define the technical and functional requirements for the DataQueryAI. Define the technical and functional requirements for DataQueryAI.

## Key Points:

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **Google TAPAS Model**
   * Frontend: **Streamlit Web Framework**
2. **Functional Requirements:**
   * Ability to upload CSV files and process natural language queries.
   * Display query results in tables, charts, or text format.
   * Provide accurate, context-aware answers using Google TAPAS.
3. **Constraints & Challenges:**
   * Handling large CSV files efficiently.
   * Ensuring real time query processing with minimum latency.
   * Optimizing Google TAPAS for diverse datasets and query types.

# Phase-3: Project Design

## Objective:

Develop the architecture and user flow of DataQueryAI.

## Key Points:

1. **System Architecture:**
   * User uploads a CSV file and enters query via the streamline UI.
   * The backend processes the query using Google TAPAS.
   * The model analyzes the data and returns the result to the frontend.
2. **User Flow:**
   * Step 1: User uploads a CSV file.
   * Step 2: User enters a query (e.g., "What is the total revenue for Q2?")
   * Step 3: The app processes the query and displays the results.
3. **UI/UX Considerations:**
   * **Simple, intuitive interface** for uploading files and entering queries.
   * **Interactive display** of results (tables, charts, or text).
   * **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 0** | Environment Setup & Planning | High | 2 hours | 2 hours | Member 1 | Python, Streamlit, Google Colab | Ready-to-code environment. |
| **Sprint 1** | Integrate Google TAPAS & Set Up Streamlit | High | 6 hours | 8 hours | Member 1 & 2 | TAPAS API, Streamlit | TAPAS integrated, basic app running. |
| **Sprint 2** | Build File Uploader & Query Input | High | 4 hours | 12 hours | Member 2 & 3 | Streamlit UI setup | File uploader and query input functional. |
| **Sprint 3** | Connect Frontend to Backend & Display Results | High | 6 hours | 18 hours | Member 1 & 2 | TAPAS integration, UI ready | Query processing and results displayed. |
| **Sprint 4** | Test, Debug & Optimize | High | 6 hours | 24 hours | Member 1 & 3 | Test logs, API responses | Stable app with minimal bugs. |
| **Sprint 5** | Add Visualizations & Deploy App | Medium | 5 hours | 29 hours | Member 2 & 4 | Query results ready | Visualizations added, app deployed. |
| **Sprint 6** | Prepare Demo & Final Submission | High | 1 hour | 30 hours | Entire Team | All tasks completed | Demo video, report, and GitHub link submitted. |

## ****Sprint Planning with Priorities****

### **Sprint 0 – Environment Setup & Planning (2 Hours)**

* **(🔴 High Priority)** Set up the environment (Python, Streamlit, Google Colab).
* **(🔴 High Priority)** Finalize the project plan and assign roles.

### **Sprint 1 – Integration & Basic Setup (6 Hours)**

* **(🔴 High Priority)** Integrate Google TAPAS for query processing.
* **(🔴 High Priority)** Set up a basic Streamlit app skeleton.

### **Sprint 2 – Core UI Development (4 Hours)**

* **(🔴 High Priority)** Build a file uploader for CSV files.
* **(🔴 High Priority)** Add a query input field for natural language queries.

### **Sprint 3 – Query Processing & Results Display (6 Hours)**

* **(🔴 High Priority)** Connect the frontend to the backend (Google TAPAS).
* **(🔴 High Priority)** Display query results in tables or text format.

### **Sprint 4 – Testing & Debugging (6 Hours)**

* **(🔴 High Priority)** Test the app with diverse datasets and queries.
* **(🔴 High Priority)** Debug and optimize performance (e.g., API response times).

### **Sprint 5 – Enhancements & Deployment (5 Hours)**

* **(🟡 Medium Priority)** Add visualizations (e.g., charts, graphs) for query results.
* **(🔴 High Priority)** Deploy the app on Streamlit Sharing.

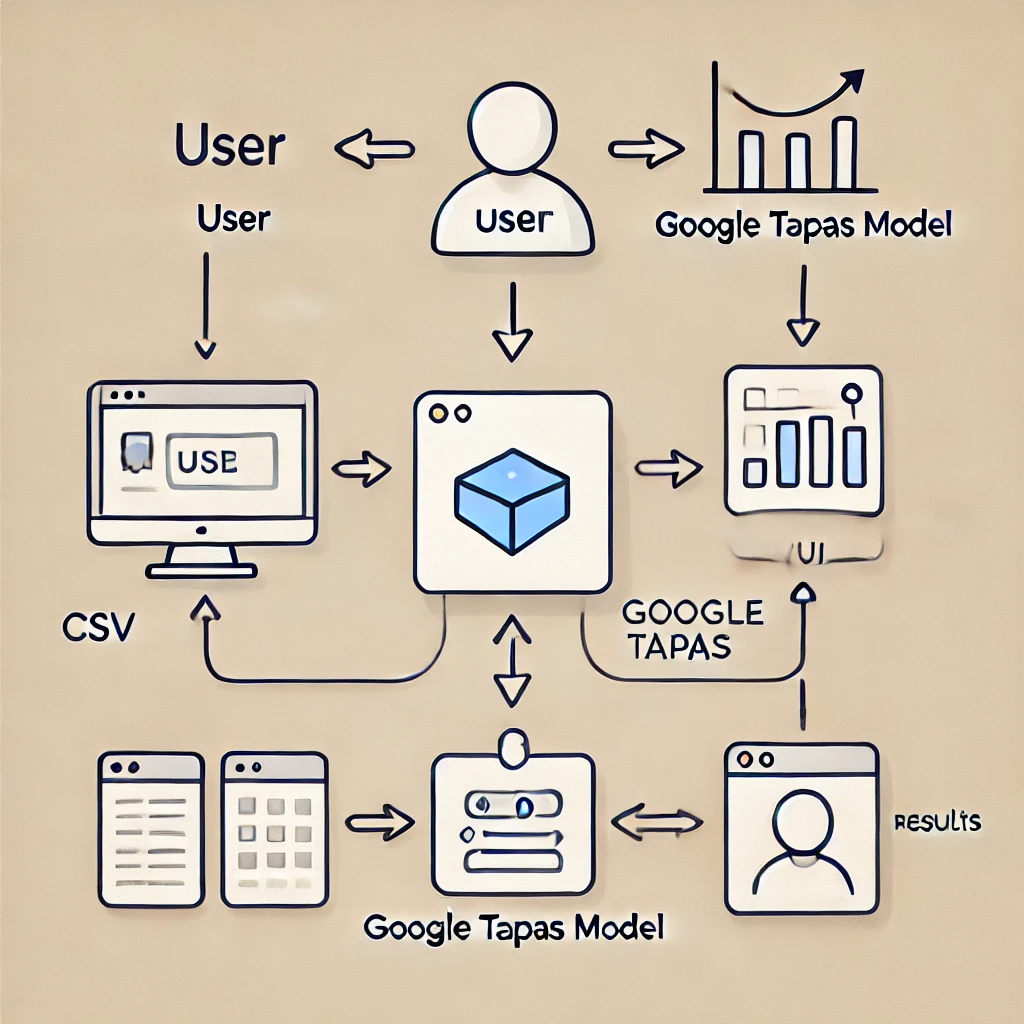
### **Sprint 6 – Final Submission (1 Hour)**

* **(🔴 High Priority)** Prepare a 3-5 minute demo video.
* **(🔴 High Priority)** Finalize documentation and submit the project (report, GitHub link).

# Phase-5: Project Development

## Objective:

Implement core features of the DataQueryAI App.



## Key Points:

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend:** Google TAPAS
   * **Programming Language:** Python
2. **Development Process:**
   * Integrate Google TAPAS for query processing.
   * Develop the Streamlit UI for file uploads and query input.
   * Optimize query handling for performance and accuracy..
3. **Challenges & Fixes:**
   * **Challenge:** Delayed response times for large datasets.
   * **Fix:** Implement data preprocessing and caching.
   * **Challenge:** Difficulty in comparing the columns.
   * **Fix:** Proceeded by validating the columns before comparing.

# Phase-6: Functional & Performance Testing

## Objective:

Ensure that the DataQueryAI works as expected.

| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| **TC-001** | Functional Testing | Query "What is the highest value in units sold?" | Highest value should be displayed | ✓ Passed | Tester 1 |
| **TC-002** | Functional Testing | Query "Compare units sold by region?" | Visualization of comparison should be displayed | ✓ Passed | Tester 2 |
| **TC-003** | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | Fixed | Tester 3 |
| **TC-004** | Final Validation | Ensure UI is responsive across devices. | UI should work on desktop. | Passed | Tester 2 |
| **TC-005** | 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**