Project Title:

DataQueryAI - AI-Powered Data Analysis

Team Name:

(Provide your team's name)

Team Members:

- Vishnu Vardhan Gaddam
- Ashik Gottam
- Koushik Reddy
- Adidala Nidvitha
- Dontham Varshitha

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered data querying tool that simplifies data analysis by providing insightful and contextually accurate responses based on CSV file uploads.DataQueryAI can enhance data querying, analysis, and decision-making by leveraging AI-driven automation, natural language processing (NLP), and advanced analytics. The goal is to identify key features, potential integrations, and user experience improvements that will make data querying more intuitive, efficient, and accessible for a wide range of users, from data analysts to non-technical stakeholders.

Key Points:

1. **Problem Statement:**
    - Traditional data querying requires SQL or technical expertise, making it inaccessible to non-technical users.
    - Analysts spend excessive time writing, debugging, and optimizing queries, slowing down decision-making.

2. **Proposed Solution:**
    - **AI-powered natural language querying** to enable users to retrieve insights without SQL knowledge.
    - **Integration with various databases and BI tools** for seamless workflow incorporation.
    - **Automated query optimization** for faster and more efficient data processing.
3. **Target Users:**
    - **Business analysts & decision-makers** who need quick insights without technical expertise.
    - **Executives & managers** who require real-time analytics for strategic decisions.
    - **Marketing, sales, and finance teams** needing instant data retrieval for operations
4. **Expected Outcome:**
    - **Reduced dependency on technical teams**, enabling self-service data access.
    - **Enhanced business intelligence** with smarter, AI-optimized query suggestions

---

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the DataQueryAI.

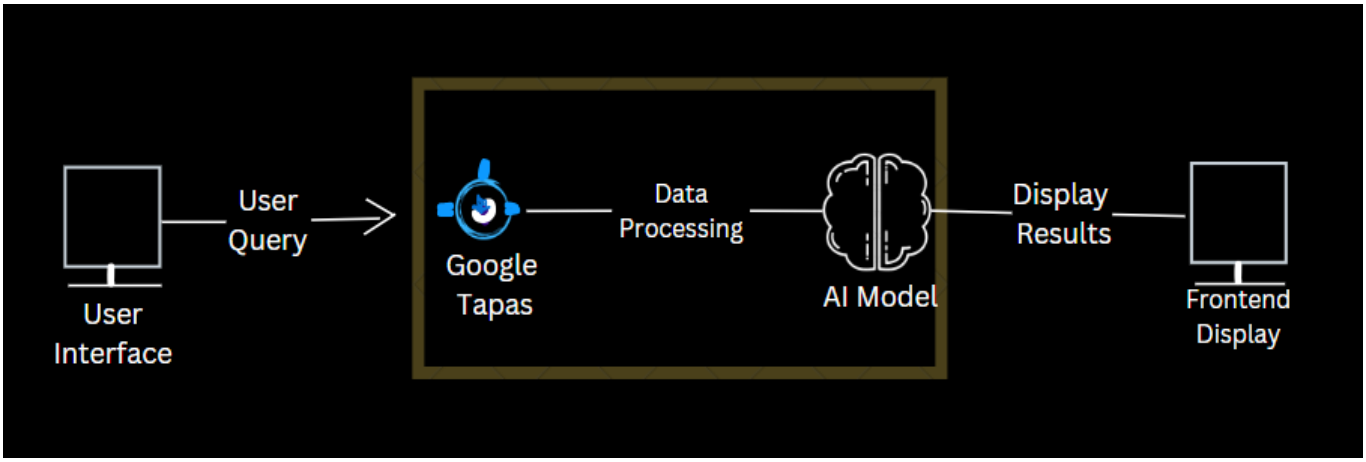
Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - OpenAI API / Generative AI models
    - Frontend: **Streamlit Web Framework**
    - Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
    - Ability to fetch data insights using AI-powered analysis.
    - Display processed results in an intuitive UI.
    - Provide real-time statistical insights and data
    - Allow users to upload CSV files for automated analysis.
3. **Constraints & Challenges:**
    - Ensuring real-time AI-generated responses.
    - 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 uploads a CSV file via the UI.
    - Query is processed using the Generative AI model.
    - AI model fetches, processes, and interprets the data.
    - The frontend displays structured insights, trends, and visualizations.
2. **User Flow:**
    - **Step 1:** User uploads a CSV file.
    - **Step 2:** Backend processes the file using AI-driven models.
    - **Step 3:** The app presents insights and responses 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	Ashik	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	? Medium	2 hours (Day 1)	End of Day 1	Vishnu	API response format finalized	Basic UI with input fields
Sprint 2	Query Processing & AI Integration	? High	3 hours (Day 2)	Mid-Day 2	Nidvitha	API response, UI elements ready	AI-powered natural Language Queries implemented
Sprint 2	Error Handling & Debugging	? High	1.5 hours (Day 2)	Mid-Day 2	Ashik	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	? Medium	1.5 hours (Day 2)	Mid-Day 2	Koushik	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 OpenAI API for query processing.
(? Medium Priority) Build a basic UI with input fields using Streamlit.

Sprint 2 – Core Features & Debugging (Day 2)

(? High Priority) Implement natural language query processing. (? High Priority) Debug API issues & handle errors in queries.

(? Medium Priority) Implement query optimization & AI-assisted suggestions.

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 DataQueryAI.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend: Generative AI API**
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and AI model integration.
    - Develop CSV file processing and query interpretation logic.
    - Optimize query processing for performance and relevance.
3. **Challenges & Fixes:**
    - **Challenge:Slow AI response times. Fix: Implement caching for frequently queried results.**
    - **Challenge: Handling large datasets. Fix: Optimize AI processing to focus on key insights.**

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	Upload CSV with Sales data	Accurate trend analysis	✅ Passed	Vishnu
TC-002	Functional Testing	Query dataset for summary stats	Display correct summary.	✅ Passed	Ashik

TC-003	Performance Testing	response time under 500ms	Quick AI processing	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect data insights.	Improved accuracy.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive	UI adapts to all screens	❌ Failed - UI broken on mobile	Tester 2
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App 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**