Project Proposal: AI-Powered Data Analyst (PoC to MVP)

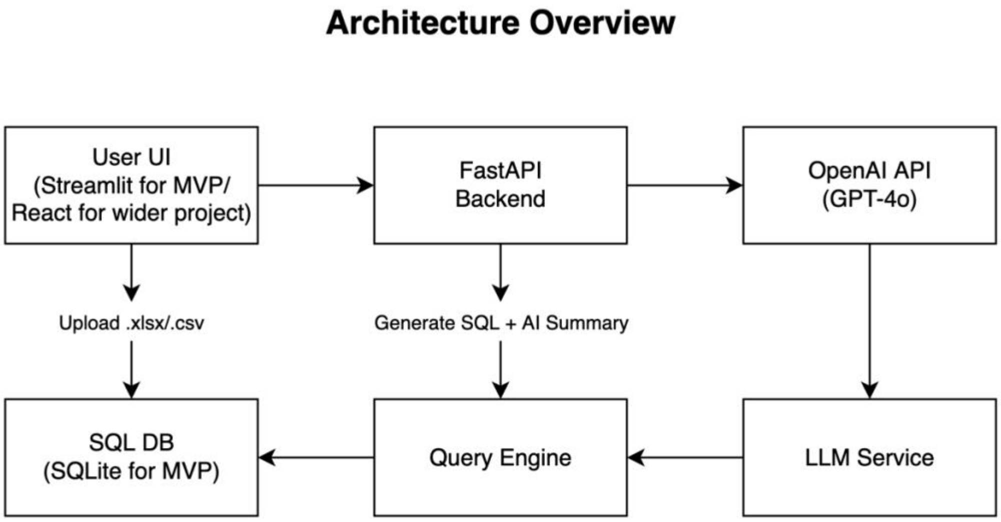
▪ Overview

This proposal outlines the technical and project planning aspects of an AI-powered solution for **Data Quality Management** through natural language interaction.

The goal is to allow non-technical users to ask questions about their datasets and receive both **SQL-generated results** and **insightful analysis** from AI.

▪ Technical Design

1. Architecture Overview



Component Roles:

- **UI(Streamlit):** Uploads Excel, takes user questions, displays tables & AI insights
- **FastAPI:** Optional layer to expose LLM & SQL services (optional if sticking with Streamlit)
- **OpenAI API:** Handles natural language → SQL and result interpretation
- **QueryEngine:** Executes SQL queries against a lightweight SQLite DB
- **LLM Service:** Generates SQL + summarizes DataFrame results with AI

2. Technologies Used

Component	Technology	Purpose
UI	Streamlit	Simple frontend for user input/output
AI Model	OpenAI GPT-4o / mini	SQL generation + analysis summarization
Backend (optional)	FastAPI	Serve LLM logic via REST (optional)
Database	SQLite	Store & query ingested Excel data
File Upload	pandas + openpyxl	Load Excel/CSV into DB
Containerization	Docker	Portable & reproducible runtime

3. Deployment Plan

- MVP/Poc
  - Local containerized deployment via Docker
  - .env based configuration for secrets and paths
- Production (Future)
  - Deploy to cloud (e.g., **Azure, AWS, etc**)
  - Add cloud-managed DB (e.g., **PostgreSQL**)
  - Add authentication and multi-user support

- Project Planning

1. Development Plan for MVP

Week	Tasks
Week 1	a) Finalize UI layout + file ingestion b) Implement SQL generation via Open AI c) Basic SQL execution
Week 2	a) AI-based interpretation of results b) Add retry/error handling for bad SQL c) Dockerize app
Week 3	a) User-friendly UX (loading states, error messages) b) Add optional chart visualization c) Internal user testing
Week 4	a) Package demo b) Prepare documentation c) Live walkthrough with client

Functionable MVP is feasible in 3 – 4 weeks

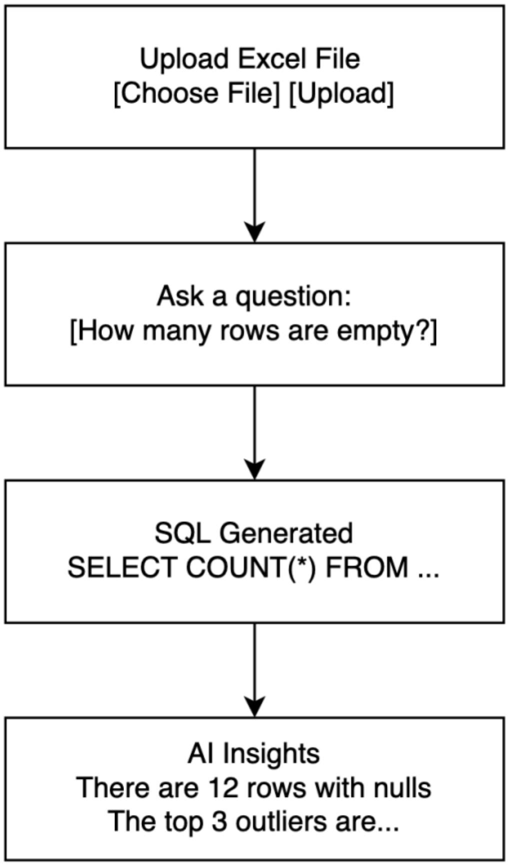
2. Resources Needed

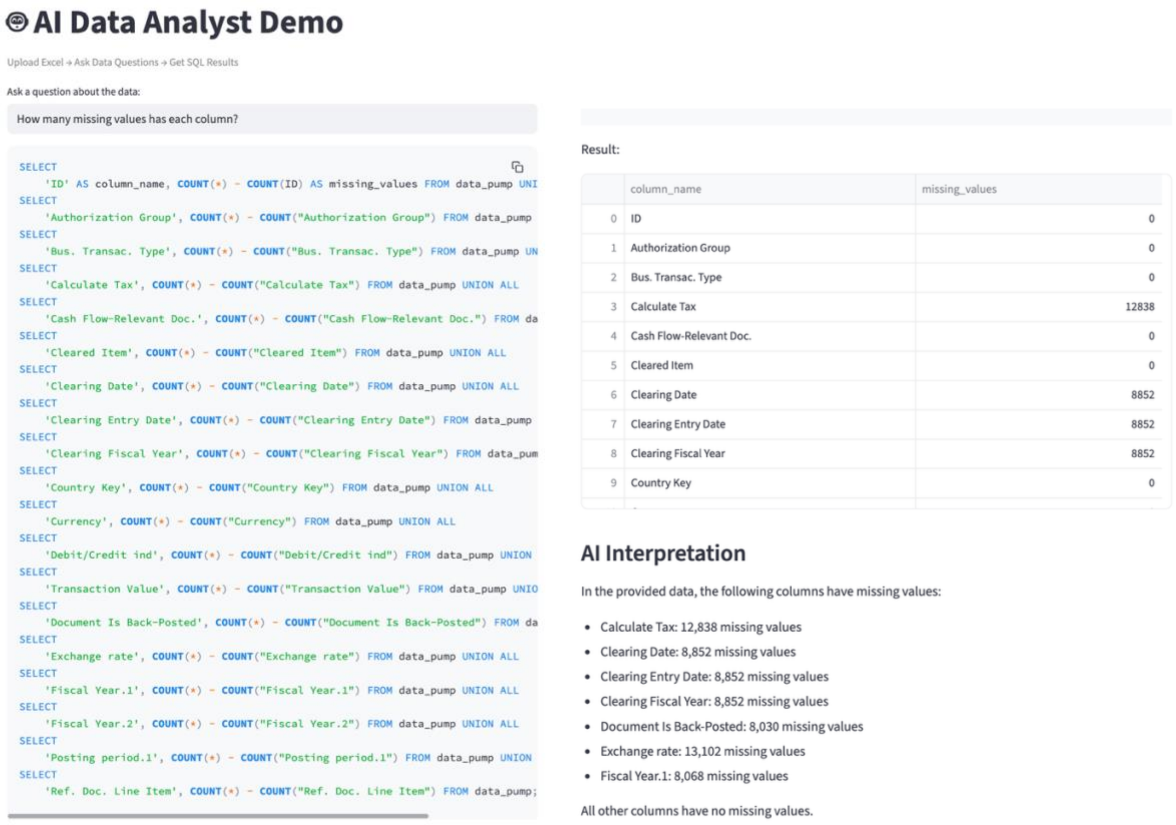
Role	Profile
1xAI Engineer	Python, OpenAI API, prompt engineering
1xFullstack Engineer	Python, Streamlit, FastAPI, SQL
(Optional) UI Designer	UX wireframes / enhancements

For MVP, **1–2 people** can deliver the product efficiently.

SCRUM method is used as project-management method.

- UI Mockup





AI Data Analyst Demo

Upload Excel → Ask Data Questions → Get SQL Results

Ask a question about the data:

How many missing values has each column?

```
SELECT
    'ID' AS column_name, COUNT(*) - COUNT(ID) AS missing_values FROM data_pump UNI
SELECT
    'Authorization Group', COUNT(*) - COUNT("Authorization Group") FROM data_pump
SELECT
    'Bus. Transac. Type', COUNT(*) - COUNT("Bus. Transac. Type") FROM data_pump UN
SELECT
    'Calculate Tax', COUNT(*) - COUNT("Calculate Tax") FROM data_pump UNION ALL
SELECT
    'Cash Flow-Relevant Doc.', COUNT(*) - COUNT("Cash Flow-Relevant Doc.") FROM da
SELECT
    'Cleared Item', COUNT(*) - COUNT("Cleared Item") FROM data_pump UNION ALL
SELECT
    'Clearing Date', COUNT(*) - COUNT("Clearing Date") FROM data_pump UNION ALL
SELECT
    'Clearing Entry Date', COUNT(*) - COUNT("Clearing Entry Date") FROM data_pump
SELECT
    'Clearing Fiscal Year', COUNT(*) - COUNT("Clearing Fiscal Year") FROM data_pum
SELECT
    'Country Key', COUNT(*) - COUNT("Country Key") FROM data_pump UNION ALL
SELECT
    'Currency', COUNT(*) - COUNT("Currency") FROM data_pump UNION ALL
SELECT
    'Debit/Credit ind', COUNT(*) - COUNT("Debit/Credit ind") FROM data_pump UNION
SELECT
    'Transaction Value', COUNT(*) - COUNT("Transaction Value") FROM data_pump UNIO
SELECT
    'Document Is Back-Posted', COUNT(*) - COUNT("Document Is Back-Posted") FROM da
SELECT
    'Exchange rate', COUNT(*) - COUNT("Exchange rate") FROM data_pump UNION ALL
SELECT
    'Fiscal Year.1', COUNT(*) - COUNT("Fiscal Year.1") FROM data_pump UNION ALL
SELECT
    'Fiscal Year.2', COUNT(*) - COUNT("Fiscal Year.2") FROM data_pump UNION ALL
SELECT
    'Posting period.1', COUNT(*) - COUNT("Posting period.1") FROM data_pump UNION
SELECT
    'Ref. Doc. Line Item', COUNT(*) - COUNT("Ref. Doc. Line Item") FROM data_pump;
```

Result:

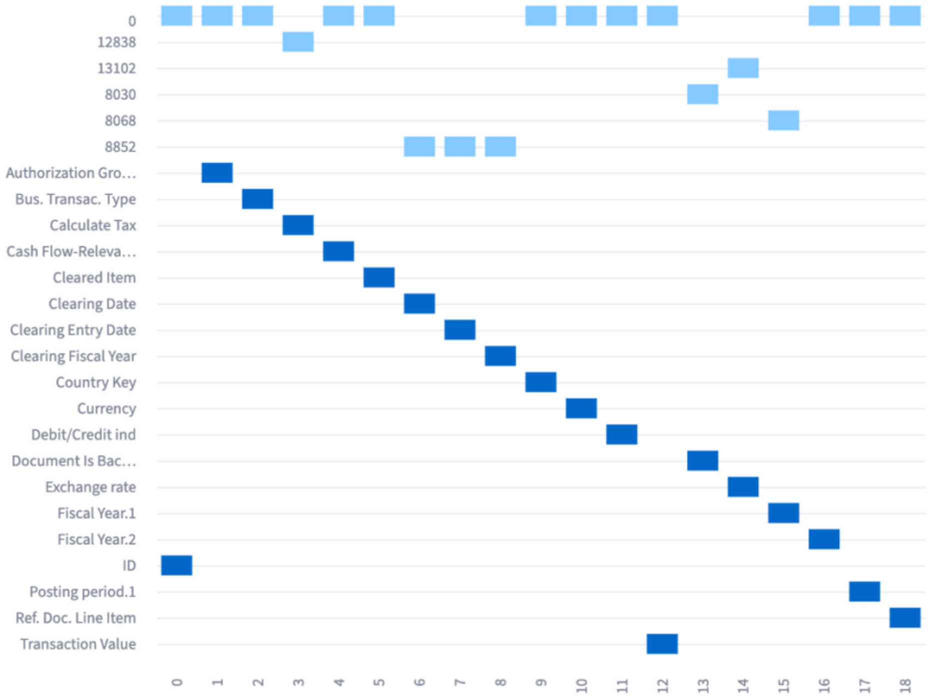
	column_name	missing_values
0	ID	0
1	Authorization Group	0
2	Bus. Transac. Type	0
3	Calculate Tax	12838
4	Cash Flow-Relevant Doc.	0
5	Cleared Item	0
6	Clearing Date	8852
7	Clearing Entry Date	8852
8	Clearing Fiscal Year	8852
9	Country Key	0

AI Interpretation

In the provided data, the following columns have missing values:

- Calculate Tax: 12,838 missing values
- Clearing Date: 8,852 missing values
- Clearing Entry Date: 8,852 missing values
- Clearing Fiscal Year: 8,852 missing values
- Document Is Back-Posted: 8,030 missing values
- Exchange rate: 13,102 missing values
- Fiscal Year.1: 8,068 missing values

All other columns have no missing values.



- Summary

This PoC demonstrates a scalable path to an AI data quality assistant that empowers non-technical users to interact with their data naturally.

With minimal infrastructure and a clean user experience, the system proves both the **technical feasibility** and **business value** of the solution.