**Fidelity Dashboard Project**

**Overview**

This Power BI project presents an interactive dashboard for Fidelity clients, designed to visualize portfolio performance, transaction trends, and customer engagement metrics. It includes:

* **Key Metrics**: Total Portfolio Value, Total Transactions, Average ROI, Net Cash Flow, Active Customer Ratio, Weighted ROI.
* **Visual Insights**: Portfolio distribution by asset type, transaction volume over time.
* **Interactivity**: Slicers for Location, Risk Level, and Date range.
* **Branding**: Fidelity color palette and logo for a professional look.

**Files in Repository**

* Fidelity\_Dashboard.pbix: Power BI Desktop file containing the full report.
* Dashboard\_Documentation.pdf: PDF with project approach, embedding note, and dashboard screenshots.
* README.md: This document.

**Data Model**

* **Tables**:
  + Customer\_Data: Customer demographics and risk tolerance.
  + Portfolio\_Data: Portfolio holdings with asset type, current value, ROI, and risk level.
  + Transactions: Buy/Sell transactions with timestamps and amounts.
* **Relationships**: Linked on Customer\_ID (one-to-many) from Customer\_Data to each fact table.

**Key DAX Measures**

* Total Portfolio Value = SUM(Portfolio\_Data[Current\_Value])
* Total Transactions = SUM(Transactions[Amount])
* Average ROI = AVERAGE(Portfolio\_Data[ROI])
* Net Cash Flow = [Total Buys] - [Total Sells]
* Active Customer Ratio = DIVIDE(DISTINCTCOUNT(Transactions[Customer\_ID]), DISTINCTCOUNT(Customer\_Data[Customer\_ID]))
* Weighted ROI = DIVIDE(SUMX(Portfolio\_Data, Portfolio\_Data[Current\_Value] \* Portfolio\_Data[ROI]), [Total Portfolio Value])

**Embedding & Security Note**

This report can be embedded using Power BI Embedded (App-owns-data) by registering an Azure AD app and using a service principal to generate embed tokens. We enforce Row-Level Security using dynamic roles based on USERPRINCIPALNAME() to ensure users see only their data. For large datasets, we recommend Import mode with incremental refresh partitions and aggregated tables to optimize both performance and resource usage.