**Financial Data Analytics Project Documentation**

## Project Overview

The project aims to develop a comprehensive financial data analysis and visualization platform using IEX Cloud API, Azure SQL, Tableau, Python, and Excel. The system will enable users to extract, clean, analyze, and visualize historical stock prices, financial indicators, and economic data for JSE Top 40 companies. The project will be staged into 4 parts: Data Extract using APIs, Database creation, Data Visualization, Forecasting

## Preparation

1. We need to prepare the local repository for us to organize our code. To achieve this we use the command prompt: mkdir PartXXX\_of\_the\_project
   * Directory Creation: Four directories are created for different project parts: data extraction, database creation, data visualization, and forecasting.
   * Directory Structure: Each directory is created separately using the mkdir command. They are organized within the specified project path.
   * Purpose of Organization: Segments the project into distinct parts for focused development and management. This will allow us to later refactor this code in a modulus structure
   * Project Preparation: Initial setup phase to prepare the local environment for subsequent development tasks.
2. Creating our virtual environment for working on this project. This is where we will also install our required libraries from requirements.txt file using:
   * Bash: python -m venv jse40
   * Bash: pip install -r requirements.txt

## Objectives

**Part 1: Data Extraction with Python**

* Set Up IEX Cloud API: Obtain API keys and authenticate your requests.
* Retrieve Historical Stock Prices: Use Python libraries like requests or specialized financial libraries to fetch historical stock prices for JSE Top 40.
* Fetch Financial Indicators and Economic Data: Utilize IEX Cloud API endpoints or other relevant sources to gather financial indicators and economic data.
* Store Data as CSV: Use Python's csv module or pandas to store the collected data in CSV format.

**Part 2: Data Cleaning and Database Management with SQL and Python**

* Connect to Azure SQL Database: Use Python libraries like pyodbc or sqlalchemy to connect to your Azure SQL database.
* Data Cleaning and Feature Engineering: Write SQL queries or use Python's data manipulation libraries like pandas to clean the data and perform feature engineering.
* Create Views and Manage Database: Write SQL scripts to create views and manage your database schema as required.
* Upload Data to Azure SQL: Execute SQL commands through Python to upload cleaned and processed data to your Azure SQL database.

**Part 3: Visualization with Tableau and Excel**

* Tableau Dashboard Creation: Connect Tableau to your Azure SQL database and create interactive dashboards. Ensure to include relevant visualizations and insights.
* Publish Tableau Dashboard to Tableau Public: Publish your Tableau dashboard to Tableau Public for easy sharing and accessibility.
* Excel Dashboard Creation: Use Excel's data connection features to connect to your Azure SQL database and create a dashboard with charts and pivot tables.

**Part 4: Forecasting and Portfolio Creation**

* Time Series Forecasting: Use Python libraries like statsmodels or prophet to perform time series forecasting on historical stock prices.
* Portfolio Creation: Implement portfolio optimization techniques using Python libraries like cvxpy or custom algorithms to construct an optimized portfolio based on forecasted data and risk tolerance.