**1. Overview**

This guide explains how the **Streamlit Risk Management App** was built, breaking down the code **line by line** with explanations on why each part is necessary. The app allows users to **simulate trading scenarios**, analyze risks, and visualize trading performance.

|  |  |
| --- | --- |
| **2. Required Libraries** | A screenshot of a computer code  Description automatically generated |

**Why These Libraries?**

* **numpy**: Used for numerical calculations and simulating random trade results.
* **pandas**: Helps store trade results in a structured DataFrame.
* **streamlit**: Provides an interactive UI for user input and visualizations.
* **plotly.graph\_objects & plotly.express**: Used to create dynamic and visually appealing charts.
* **scipy.stats**: Helps generate the normal distribution curve for trade return analysis.

**3. Streamlit Page Configuration & UI Setup**

A close-up of text

Description automatically generated

**Why This is Important?**

* Expanding the layout allows for **side-by-side visualizations**.
* The **banner** makes the app visually appealing.
* The **title** gives context to the user

**4. Scenario Input Sliders**

A close up of a text

Description automatically generated

* **Why Columns?** Organizing the inputs into **three columns** makes them easy to compare.

**Function to Collect User Inputs for a Scenario**

A black background with text

Description automatically generated with medium confidence

* **Why Define a Function?** This ensures **reusability** for multiple scenarios.
* **Markdown Headers**: Helps differentiate each scenario visually.
* **Number Inputs**: Users can **fine-tune** risk parameters.

**Storing Scenario Inputs**

A group of black and red text

Description automatically generated

* Assigning **colors** to make each scenario visually distinct.

**5. Simulating Risk Management Scenarios**

|  |  |
| --- | --- |
| A screen shot of a computer code  Description automatically generated | * **Why Set a Random Seed?** Ensures **reproducibility** of random outcomes. * **Capital Tracking**: Initializes the trading balance. * **Lists to Store Trade Data**: Used for plotting results later. |

**Simulating Trades**

A computer code on a black background

Description automatically generated

* **Win/Loss Simulation**: Uses **random choice** based on win rate.
* **Randomized Trade Returns**: Gains/losses fluctuate within **±20% of average**.
* **Capital Adjustment**: Updates capital after each trade.
* **Drawdown Calculation**: Tracks largest drop in capital.

**Store Simulation Results in DataFrame**



**Run Simulations for All Scenarios**

A close-up of words

Description automatically generated

**6. KPI Metrics**

* **KPI Cards** display the final balance and percent gain/loss for each scenario.

A black background with colorful lights

Description automatically generated