Name Suggestion

- **BI Streamline**
- **InsightBuilder**
- **Report GenX.io**
- **InsightAutomator.io**
- **BIReportPro.io**
- **BIReportPro.com**

Before we begin, ensure you have the following:

1. **Python**: Install Python on your system. You can download it from the official website (https://www.python.org/downloads/).

2. **Power BI Service**: Access to Power BI Service, where your reports will be hosted (https://powerbi.microsoft.com/en-us/).

3. **Data Sources**: Identify the data sources you want to use for report generation. This could include databases, spreadsheets, APIs, or any other data repositories.

Steps to Build the Tool

1. Data Extraction and Transformation

- Write Python scripts to extract data from your chosen data sources. Libraries like Pandas, SQLAlchemy, or specific database connectors will be useful here.

- Perform data transformation as necessary. Clean, aggregate, and preprocess the data to match your reporting requirements.

2. Data Analysis and Visualization

- Use Python's data analysis libraries, such as Pandas and NumPy, to analyze the transformed data. Calculate key performance indicators (KPIs) relevant to your reports.

- Utilize data visualization libraries like Matplotlib or Seaborn to create charts and graphs representing your data.

3. Power BI REST API Integration

- Research and familiarize yourself with the Power BI REST API documentation (https://docs.microsoft.com/en-us/rest/api/power-bi/).

- Write Python scripts to programmatically interact with the Power BI REST API. This includes creating datasets, pushing data, defining data models, and creating reports.

4. Automate the Process

- Automate the entire process using Python's scheduling libraries like `schedule` or `cron`. Schedule your script to run at specific intervals or based on data triggers.

5. User-Friendly Interface (Optional)

- Consider building a user-friendly interface using Python frameworks like Flask or Django to allow users to configure report generation parameters and monitor the process.

6. Testing and Validation

- Test your automated process thoroughly to ensure it generates accurate and up-to-date reports.

7. Deployment

- Deploy your tool to a server or cloud platform, if necessary, to run scheduled tasks automatically.

8. Documentation and User Guides

- Create documentation and user guides to help others understand how to use your tool.