Project Plan: LiveDataViz - Real-time Data Dashboard

LiveDataViz is a real-time data visualization dashboard project that fetches live data from APIs and visualizes it using interactive charts. The project combines Python for backend API development and JavaScript (React, D3.js, or Plotly) for frontend visualization. This PDF contains the step-by-step guide and features to implement as you start working on the project.

Steps to Get Started:

1. **Set Up Your Development Environment:**

- Install Python (and `pip` for package management) on Ubuntu.

- Install Node.js and npm for JavaScript (React and frontend dependencies).

- Install a code editor like VSCode or Sublime Text.

2. **Backend Setup with Python:**

- Install Flask/Django for the backend framework:

** - Flask: `pip install flask`**

** - Django: `pip install django`**

- Create a REST API that returns mock or real-time data (e.g., weather, stock, or cryptocurrency).

- Set up WebSocket (optional) for real-time data streaming.

- Deploy Backend to a platform like PythonAnywhere or Heroku.

3. **Frontend Setup with JavaScript:**

- Set up React using `create-react-app`:

** - `npx create-react-app my-app`**

- Install charting libraries (D3.js, Plotly, or Chart.js):

** - `npm install d3` or `npm install plotly.js`**

- Connect backend (Flask/Django) to the frontend using Axios or Fetch API to pull data.

4. **Create Interactive Visualizations:**

- Use D3.js or Plotly to create dynamic charts (line, bar, or scatter charts).

- Handle real-time data updates using WebSockets or periodic polling for updates.

5. **Implement Interactivity:**

- Add filtering options (e.g., select date range, type of data, etc.).

- Allow dynamic updates to the charts based on user selections.

6. **Export Data or Charts (optional):**

- Implement functionality for exporting charts or data to CSV or image files.

7. **Deploy the Application:**

- Deploy the backend (Flask/Django) and frontend (React) to platforms like Heroku, Netlify, or AWS.

8. **Document the Project:**

- Write documentation on how to set up and use the project.

- Include API endpoints and installation instructions.

9. **Iterate and Add Features:**

- Continue improving and adding features based on feedback or your own ideas.