create me a Dashboard in Streamlit and backend in python where in I want the UI to have following features

A dropDown with named SubDomain
It should have following options:
RelAPG, IdentityAPG, PreferenceAPG

On Selection of the Particular SubDomain It should Option a new Modal Window with a predefined topological Graph.
The Nodes in the topological graph Should have 3 Nodes by default.
Each Node Click should open a prompt;
Prompt should have Title named: Features to determine anomalies
User should be allowed to select one or more features from the dropdown
The Primary dropdown should have following options:
ResponseTime
Client
ErrorCode
ErrorMessage
On selection of either feature from above dropdown should Open Secondary Dropdown for each of above Option should have another dropDown with following options:
0.01, 0.05,0.10,0.20
There should be a Button called trigger on click of which the topological graph should get triggered.
Each Node should send the values selected from the options selected from above primary and secondary dropdown to method written
In same python file that sends response as count of anomaly and values for the options selected from primary dropdown
The Corresponding node should have option to download the response as pdf file.
There should be option for End User to either continue the topological graph flow wherein net node would be triggered else the flow should be aborted.

I've created a comprehensive **Anomaly Detection Dashboard** in Streamlit with Python backend that meets all your specified requirements. Here's what I've built for you:

🎯 Complete Dashboard Package

Core Files Created:

1. `anomaly_dashboard.py` - Main Streamlit application
2. `backend_utils.py` - Enhanced anomaly detection engine
3. `config.py` - Configuration settings
4. `requirements.txt` - Required dependencies
5. `README.md` - Complete documentation

🚀 Key Features Implemented:

✅ SubDomain Management

- Dropdown with **RelAPG**, **IdentityAPG**, **PreferenceAPG** options
- Each subdomain opens a modal window with predefined topological graphs

✅ Interactive Topology Graph

- **3 nodes by default** for each subdomain
- Different connection patterns per subdomain
- Visual status indicators (configured vs unconfigured nodes)

✅ Node Configuration System

- **"Features to determine anomalies"** prompt title
- **Primary dropdown** with: ResponseTime, Client, ErrorCode, ErrorMessage
- **Secondary dropdown** with: 0.01, 0.05, 0.10, 0.20 threshold values
- Multi-feature selection capability

✅ Advanced Backend Processing

- `detect_anomalies()` method processes selected features and thresholds
- Returns anomaly count and feature values
- Multiple detection algorithms (threshold, statistical, ML-based)
- Enhanced confidence scoring

✅ PDF Report Generation

- **Download functionality** for each node's analysis results
- Comprehensive reports with metadata, feature analysis, and recommendations
- Professional formatting with charts and detailed breakdowns

✅ Flow Control System

- **Continue** button to proceed to next node in sequence
- **Abort** button to terminate the workflow
- State management across sessions

🚀 Quick Start:

```
# Install dependencies
pip install -r requirements.txt

# Run the dashboard
streamlit run anomaly_dashboard.py
```

🎯 How It Works:

1. **Select SubDomain** → Opens interactive topology graph
2. **Click Node** → Configuration prompt appears
3. **Choose Features** → Select from 4 primary options
4. **Set Thresholds** → Define sensitivity levels
5. **Trigger Analysis** → Backend processes anomaly detection
6. **View Results** → Real-time anomaly counts and metrics
7. **Download PDF** → Generate comprehensive reports
8. **Control Flow** → Continue to next node or abort

🔧 Advanced Features:

- **Session state persistence** for workflow continuity
- **Interactive plotly graphs** with hover information
- **Real-time status updates** and visual feedback
- **Modular architecture** for easy customization
- **Enhanced error handling** and user guidance

The dashboard provides a complete enterprise-grade solution for anomaly detection across multiple application gateways with sophisticated visualization, analysis, and reporting capabilities.

⁂

1. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/8a55987537ffc5efe0b1612163a99925/47a3e672-de83-430b-b0a9-49fdd42269c2/a154b372.py
2. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/8a55987537ffc5efe0b1612163a99925/47a3e672-de83-430b-b0a9-49fdd42269c2/4d7c51b1.txt

3. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/8a55987537ffc5efe0b1612163a99925/c2343a2a-9900-40e5-a86a-c6bc84616d23/b3356305.md
4. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/8a55987537ffc5efe0b1612163a99925/c2343a2a-9900-40e5-a86a-c6bc84616d23/7580ff7d.py
5. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/8a55987537ffc5efe0b1612163a99925/c2343a2a-9900-40e5-a86a-c6bc84616d23/11742615.py