🚁Router Path Visualizer

A network path visualizer built with **Flask** and **Dijkstra's algorithm**, which finds and displays the shortest routing path between network IP nodes using real-world coordinate data.

> 🔍Visualize the path your data takes from a given IP to a central gateway IP, powered by an interactive frontend and intelligent backend logic.

Features

- 🧮**Dijkstra's Algorithm** for efficient shortest path routing.
- 🗒️Visualizes paths based on geographical coordinates (lat/lng).
- 🗺️Dynamic IP-to-IP graph with city metadata.
- ⚠️ Error handling for invalid IPs and missing coordinate data.
- 🗜️Flask backend API + interactive HTML frontend.

🏗️Tech Stack

- **Backend**: Python, Flask, CSV parsing, Dijkstra's algorithm
- **Frontend**: HTML, CSS (Bootstrap), JavaScript (for dynamic UI)
- **Dataset**: IP-to-IP mappings with cost and geolocation data (`Updated_DSADataset.csv`)

Project Structure

```
Router-Path-Visualizer/
├── server.py                # Flask app with graph logic and API routes
├── index.html               # Frontend form/UI
├── Updated_DSADataset.csv   # CSV containing IP graph and coordinates (not
uploaded)
├── static/                  # Folder for static assets like CSS
│   └── style.css            # Custom styling (if used)
├── templates/               # Flask template directory
│   └── Deployment.html      # Main HTML page rendered at "/"
```

🚀 Getting Started

Requirements

- Python 3.7+
- Flask

🕋 Installation & Run

```
# Clone the repo
git clone https://github.com/Niteshjai/Router-Path-Visualizer.git
cd Router-Path-Visualizer

# Install dependencies
pip install flask

# Run the app
python server.py
```

Open your browser and go to: http://127.0.0.1:5000

How It Works

Step-by-Step Flow:

1. The app reads a graph of IPs and coordinates from `Updated_DSADataset.csv`.
2. When a user inputs a **starting IP**, the backend uses **Dijkstra's algorithm** to calculate the **shortest path** to the default gateway `1.108.102.183`.
3. The API returns:
4. The full IP-to-IP network graph (`/map-data`)
5. The specific path found from the source IP (`/path`)
6. The frontend renders this data (can be extended to display on a map using Leaflet or Google Maps).

Endpoints

Route	Method	Description
`/`	GET	Renders the HTML UI (`Deployment.html`)
`/map-data`	GET	Returns full graph (nodes + edges) as JSON
`/path`	POST	Takes JSON input `{startIP: "x.x.x.x"}` and returns shortest path

Important Notes

- The default **gateway IP** is hardcoded as `1.108.102.183` in `server.py`. You can change it if needed.
- Ensure the file `Updated_DSADataset.csv` is present in the root directory. Format:

```
SourceIP, DestinationIP, SrcLng, SrcLat, Cost, City
```

🏢Demo Preview (Coming Soon)

Include screenshots or a screen recording of path visualization once ready.

🏡Author

Nitesh Jaiswal\ GitHub: @Niteshjai

License