

Shortest Path Finder for Pakistani Cities

Course: Design and Analysis of Algorithms (DAA)

Semester: Fall 2025

1. Project Goal: Build a Practical Route Planner

This project is your chance to take a core computer science concept—graph algorithms—and apply it to a practical problem: finding the fastest routes between cities in Pakistan.

The goal is to build a [simple web application](#) that lets a user pick a starting city and an ending city, and then immediately shows the best route using a classic algorithm.

This Matters on Your C.V.: This project demonstrates a complete cycle of software development: handling real-world data, implementing a fundamental algorithm, and building a user-friendly application. Recruiters love seeing projects that connect theory (Dijkstra) to practice (a working app).

2. The Three Steps of the Project

You are building this system in three straightforward phases: Data Cleanup, Algorithm Implementation, and App Interface.

Phase 1: Data Preparation (The Cleanup)

Start with the provided dataset

- **Filter:** Your first task is to clean up the data. You must only keep the cities located in Pakistan.
- **Extract:** Once filtered, keep only the necessary columns: City Name, Latitude, and Longitude.
- **Tools:** You can use the [Pandas](#) and [GeoPandas](#) libraries in Python. It makes data filtering very simple!

Phase 2: Algorithm Implementation

This is where you implement the core logic for the shortest path.

- **Graph Creation:** Every city you kept in Phase 1 is a [node](#) in your graph. The connection (or [edge](#)) between any two cities is the distance between them.
 - For this you are provided the code.
- **Edge Weights:** You need to calculate the [actual](#) distance between all pairs of cities using their Latitude and Longitude coordinates. For this you are provided the Python function you can use it.
- **The Core Algorithm:** Implement [Dijkstra's Algorithm](#) from scratch. This algorithm will take your calculated distances and find the sequence of cities that results in the absolute shortest route.
 - **Input:** You give it a starting city (u) and an ending city (v).
 - **Output:** It returns the list of cities in the shortest path and the total distance traveled (in kilometers).

Phase 3: Creating the App Interface

You will use [Streamlit](#) to build the simple frontend interface for your tool.

- **Selection:** The app must have two dropdown menus: one for the Source City and one for the Destination City.
- **Results:** Once the user clicks "Calculate," the app must display the final calculated shortest path (e.g., "Lahore → Faisalabad → Islamabad") and the total distance clearly.

3. Collaboration with AI Tools

Using generative AI (like ChatGPT) to help you write or debug code is [allowed](#), provided you follow a simple transparency rule:

1. **Submit Your Prompts:** Keep a document of all the prompts you use with the AI. This shows us [how](#) you approached the problem and what you asked for.
2. **Understand Your Code:** You must be able to explain the code you submit.

4. What to Submit

1. **The Code:** A zipped folder or a GitHub link containing your Python files.
2. **The Prompt Log:** A PDF listing the conversation/prompts with your AI helper.
3. **A recorded video Demo:** A recorded demonstration explaining that how you have done it.

5. Quick Hints

- **Pandas Filtering:** Use the `==` operator on your dataframe to quickly filter for cities where the 'Country' column equals 'Pakistan'.

Start simple, take it one phase at a time, and you'll have a great project for your C.V.!