

# Quickest *Path* User Manual

Project 3  
Team 5

Last updated on  
07/05/2025

## ALGOSUP 2025



# Table of Contents

- **Introduction**
  - **Project Overview**
  - **Audience**
  - **Scope**
- **Prerequisites**
  - **Basic Requirements**
  - **Minimum System Requirements**
  - **Data File (optional)**
- **Installation Guide**
  - **Download The Software**
  - **Setting Up The Environment**
  - **Building the Project**
  - **Compile the Project**
  - **Running the Software**
  - **Running Tests**
- **Usage Instructions**
  - **Finding the Quickest Path**
- **Troubleshooting**
- **FAQs**
- **Legal Mention**
  - **Copyright**
  - **License & Usage**
  - **Liability Disclaimer**
  - **Modifications & Updates**
  - **Contact Information**
- **Appendix**
  - **Glossary**

# 1. Introduction

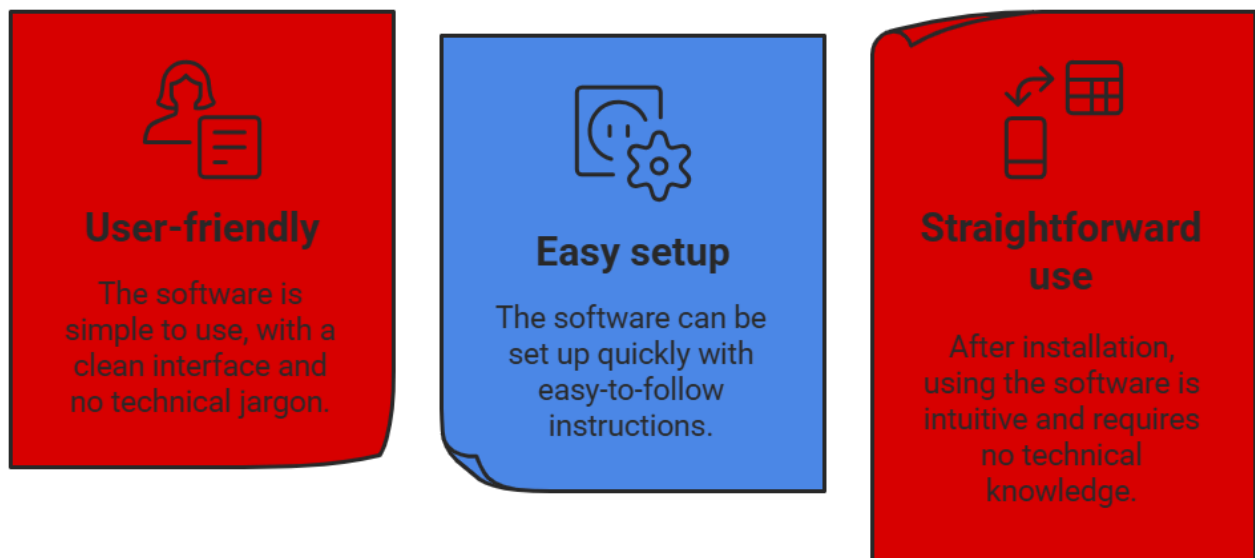
## I. Project Overview

This user manual provides guidance on how to operate and interact with the **Quickest Path Calculator** software. It calculates the fastest road between two landmarks in the United States using a large dataset of connected nodes. Its primary applications include trip planning, route optimization, and travel time reporting.

## II. Audience

This user manual is designed for everyone, no technical requirements or computer-related background are expected. The software is built to be easy to use, with a simple interface that doesn't require technical knowledge. Once installed, the tool is ready for use and allows you to perform your tasks without needing to understand complex systems.

### Key Points for the audience:



## III. Scope

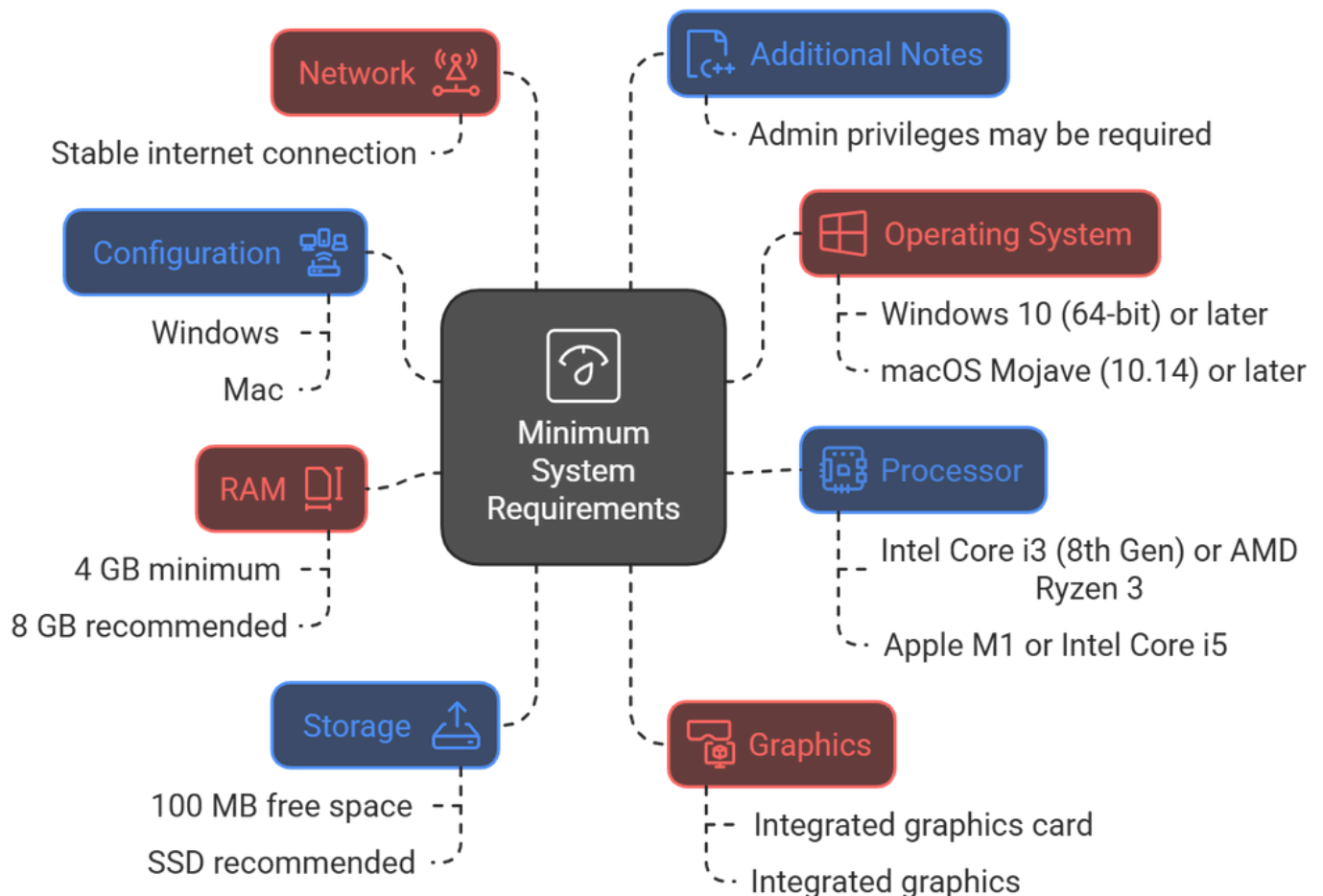
The software provides a **REST API** that allows users to find travel times and calculate optimal routes between landmarks. By using a CSV dataset of road connections, the API processes user inputs and returns travel data in both XML and JSON formats. This enables easy integration with other systems and allows users to retrieve and interpret travel information in a way that suits their needs.

# 2. Prerequisites

## I. Basic Requirements

- Input: Keyboard and mouse/trackpad.
- Device: PC or laptop.
- Power: Electricity or battery.
- Network: Internet for initial setup.

## II. Minimum System Requirements



## How to check system specifications?

### Windows Method

Press Windows + R.  
Type dxdiag and press Enter.  
In the DirectX Diagnostic Tool, you can find your Processor, Memory (RAM), and Graphics listed.



### Mac Method

Click the Apple logo in the top-left corner.  
Select About This Mac.  
The window that appears will show details about your Processor, Memory (RAM), Storage, and Graphics.

## III. Data File (optional)

- **USA-roads.csv:**

This is a file that contains information about roads and travel times between landmarks. It's very important for the software to work correctly.

- You can download it from

(<https://www.dropbox.com/transfer/AAAAAOehAVYlkjxxZRCMqTskDm1obt66RxD2g8zqp0-QqnCZofEMkjw>). Once you have it, keep it in the same folder as the software.

## 3. Installation Guide

### I. Download the Software

To start, you need to get the software on your computer. There are two options for this:

#### Case 1: Using Git (recommended if you have Git installed):

1. Open the **Command Prompt** (Windows) or **Terminal** (Mac).
2. Type the following command to download the software:

```
git clone https://github.com/algosup/2024-2025-project-3-  
quickest-path-team-5
```

Tip: If you don't have Git installed, please visit this (<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>) to install Git.

#### Case 2: Manual Download:

1. Go to the [GitHub repository](#) and click Download ZIP. Extract the ZIP file to a location on your computer. Note: Make sure the folder you download is **easy** to access.

## II. Setting Up the Environment

To run the software:

### CMake (Required)

- **CMake** is a tool that simplifies the compilation of complex projects.
  - Visit the official website: [CMake \(https://cmake.org/download\)](https://cmake.org/download) .
  - Download and install the version suitable for your system (Windows, Mac, Linux).

### Clang (C++ Compiler):

- **On Mac:** Run brew install llvm to install Clang via the LLVM package.
- **On Windows:** Use Chocolatey to install Clang by running choco install llvm.

### Crow Web Framework:

- You'll need to install the Crow web framework. Follow the Crow installation guide for CMake-based projects.

### MinGW (on Windows for Makefiles):

- If you're working on Windows and need mingw32-make, install MinGW. Instructions can be found in the [MinGW installation guide](#).

## III. Building the Project

### Generate Build Files Using CMake:

Create a build directory and navigate into it:

```
mkdir build
```

```
cd build
```

For Windows (MinGW):

- Run : `cmake -S .. -B . -G "MinGw Makefiles"`

For Mac/Linux:

- Run : `cmake -S .. -B .`

## VI. Compile the Project:

For Windows(with MinGW):

- Run : `mingw32-make`

For Mac/Linux:

- Run : `make`

## V. Running the Software

Running cap:

After building, navigate to the bin-directory and run the software:

- Run : `cd ../bin`  
`./cap`

Running DAG (Directed Acyclic Graph) Validation:

If you want to run the DAG validation:

- Navigate to the `dag_graph_validator` directory:
- Run : `cd dag_graph_validator`
- Run the **CMake** commands to build:
  - Take in consideration if your are on windows or mac, on this example both are printed.
- Run :

```
cmake -S . -B build -G "MinGw Makefiles"  
cd build  
mingw32-make || make  
cd ..
```

- Then, run the validation tool:
- Run : `cd bin`  
`./dag`

# VI. Running Tests

## Test Suite:

- To verify everything is working correctly, run the test suite:
  - Navigate to the test directory:
  - Run : `cd /Yourpath/to/tests`
  - Run the tests: `/tests`


## 4. Usage Instructions

### I. Finding the Quickest Path

- **How It Works**
  - The software calculates the shortest route between two landmarks using the provided road network data.
- **Understanding The Output**
  - Travel Time: Displays the estimated time to traverse the path.
  - Path: Lists the sequence of landmarks to follow for the route.
- **The interface is easy to use, and takes the form of a single page containing all the information.**
  - Enter your two landmarks, your source and your destination



**CAP Interface**  
Visualize optimal paths with real-time calculation based on cap API




Start Node:  
345

End Node:  
675748

Choose a response format:  
JSON

Find Optimal Path

view target map



**Path Details**

Total Travel Time: 10698757

Step 1: 345 → 198

Step 2: 198 → 109

- Choose your download format

Choose a response format:

JSON

Any

JSON

XML

**Path Details**

Total Travel Time: 10698757

- Submit your request

Choose a response format:

JSON

Find Optimal Path

**Path Details**

Total Travel Time: 10698757

- Waiting to access the travel times

### Path Details

Total Travel Time: 10698757	
Step 1	345 → 198
Step 2	198 → 199
Step 3	199 → 147
Step 4	147 → 203

## 5. Troubleshooting

Issue	Problem	Solution
<b>Missing or Incorrect USA-roads.csv File</b>	The file is missing or doesn't match the required format.	<ol style="list-style-type: none"> <li>1. Ensure the file is in the same folder as the software.</li> <li>2. The file should look like: <code>Landmark_A_ID, Landmark_B_ID, Time.</code></li> </ol>
<b>API Not Responding</b>	The server is not running.	<ol style="list-style-type: none"> <li>1. Check if the server is running by typing: <code>curl http://localhost:&lt;PORT&gt;/api/path?source_id=1&amp;destination_id=5</code></li> <li>2. Ensure the server is on port 8080.</li> <li>3. Ensure no other program is using the same port.</li> </ol>
<b>Unexpected Results</b>	The API is giving wrong results.	<ol style="list-style-type: none"> <li>1. Double-check the IDs you're using.</li> <li>2. Ensure the dataset is correct.</li> </ol>
<b>Bad Request</b>	Missing or incorrect data in your request.	<ol style="list-style-type: none"> <li>1. Ensure you include both <code>source_id</code> and <code>destination_id</code>.</li> <li>2. Example: <code>http://localhost:&lt;PORT&gt;/api/path?source_id=1&amp;destination_id=5</code></li> </ol>
<b>Not Found</b>	The requested resource doesn't exist.	<ol style="list-style-type: none"> <li>1. Verify the URL and endpoint are correct.</li> </ol>
<b>Internal Server Error</b>	Something went wrong on the server.	<ol style="list-style-type: none"> <li>1. Check the server logs for specific errors.</li> <li>2. Ensure the <code>USA-roads.csv</code> file is correct.</li> </ol>
<b>Server Fails to Start</b>	The server won't run.	<ol style="list-style-type: none"> <li>1. Check that you have installed all required dependencies.</li> <li>2. Look at error messages for missing files or libraries.</li> </ol>
<b>Slow API Response</b>	The response	<ol style="list-style-type: none"> <li>1. Ensure your dataset doesn't have</li> </ol>

	takes too long.	unnecessary data. 2. Try limiting how many requests are being made at once.
<b>Permission Denied Errors</b>	You can't access files or run commands.	1. Run the software as an administrator (use <b>sudo</b> on Linux). 2. Ensure the <b>USA-roads.csv</b> file can be read.

# 6. FAQs

## Common Questions



### Invalid Inputs

The software returns an error message for invalid inputs.



### Additional Data

Additional data can be added to the CSV file with a specific format.



### Input Format

CSV files must have exactly: NodeA, NodeB, Distance.



### Missing Files

The software notifies users if the data file is missing or corrupted.



### Compatibility

The program is compatible with multiple all the operating systems, including Windows and macOS.



### Performance Metrics

The program calculates travel time, shortest path, and compliance metrics.



### Visualization

Currently, the program does not support visualization but allows exporting results.



### Test Cases

Users can add more test cases by modifying the test suite files.



### Weighted Graphs

The software supports weighted graphs with various cost metrics.



### Project Reset

To reset or rebuild the project, delete the build directory and rerun commands.



### Bug Reporting

Users can report bugs or request features via GitHub and contact the team.

# 7. Legal Mentions

---

## I. Copyright

All content within this document, including text, images, and code snippets, is the property of Team 5 and is protected under applicable copyright laws. Unauthorized reproduction, distribution, or modification of this document is strictly prohibited.

© 2025 Team 5. All rights reserved.

## II. License & Usage

This document is provided for informational purposes only. It may not be copied, shared, or used for commercial purposes without prior written consent from Team 5.

For more information on copyright and usage, you may refer to the following legal articles:

- **Berne Convention for the Protection of Literary and Artistic Works**
- **EU Directive 2001/29/EC**

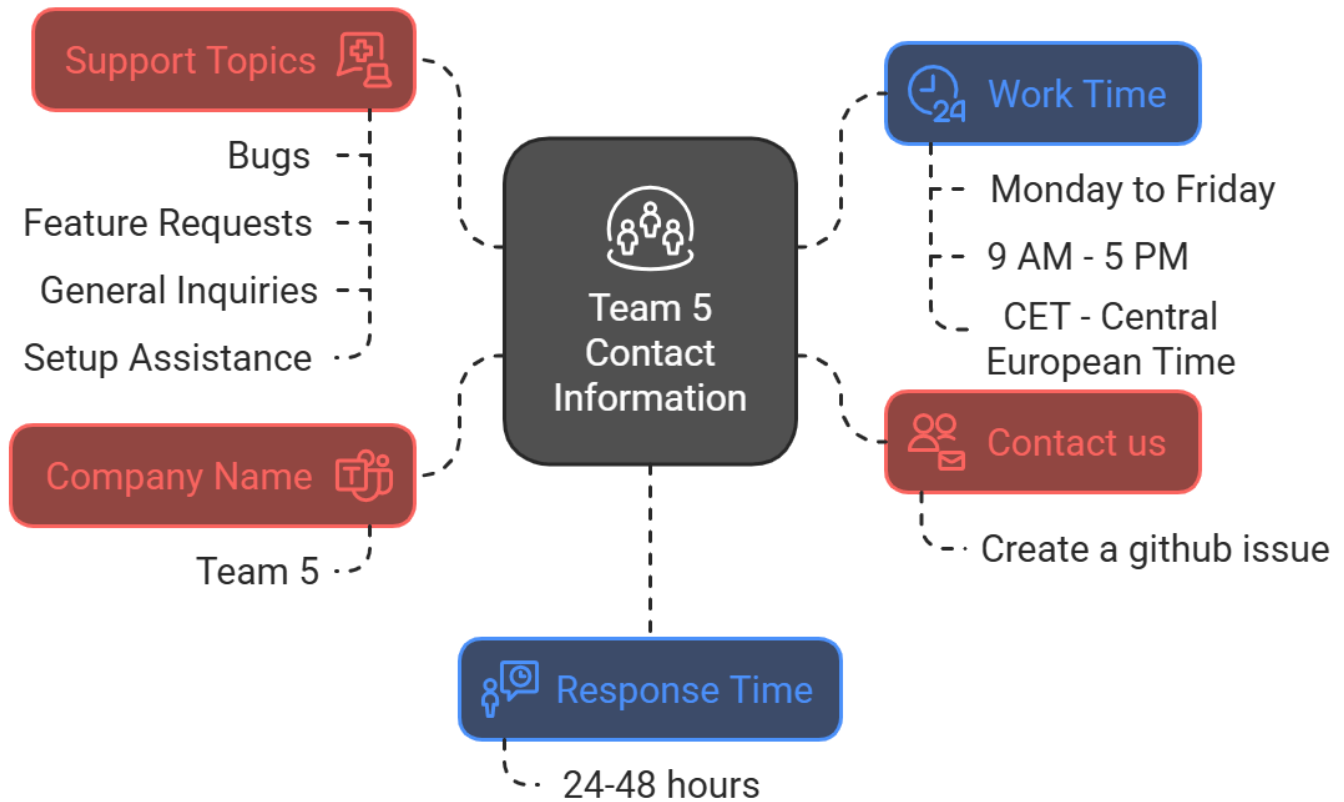
## III. Liability Disclaimer

Team 5 provides this software free of charge, without collecting personal data. The software is offered "as is," with no express or implied warranties. While we strive for accuracy and reliability, we do not guarantee uninterrupted or error-free operation. Users are responsible for their own use of the software. Team 5 shall not be liable for any indirect or incidental issues arising from its use, except in cases of gross negligence or willful misconduct.

## IV. Modifications & Updates

Team 5 reserves the right to modify, update or delete any part of its services, whether documents or software.

## V. Contact Information



# 8. Appendix

---

## I. Glossary:

- **Landmark:** A point of interest in the graph (e.g., city, intersection). It's like a location or stop along your journey.
- **CSV (Comma-Separated Values):** A simple text file format where data is stored in rows and columns. Each value is separated by a comma. It's similar to a spreadsheet, but without the rows and columns you see in Excel.
- **API (Application Programming Interface):** A set of rules that lets different software applications talk to each other. It's like a menu at a restaurant where the software makes a request and gets a response back.
- **Build Directory:** A folder where temporary files are stored during the process of making a program. Think of it as a workspace where all the pieces of your program come together.
- **Makefiles:** A set of instructions that tell the computer how to compile (build) the software. They're like a recipe with step-by-step instructions on how to make the program.
- **HTTP Server:** A program that allows other software, like a web browser, to interact with your system. It's like a waiter that brings requests from the internet and delivers responses back.
- **Git:** A tool that helps manage and track changes in software. It's like a time machine for your code, allowing you to go back to earlier versions if something goes wrong.
- **Dataset:** A collection of information or data that the software uses to calculate routes or times. It's like a map with all the road information you need for your trip.
- **Compile:** The process of turning the code you've written into a program that can be run on your computer. It's like putting ingredients together to make a meal.
- **Terminal:** A program where you type commands to interact with your computer. It's like talking directly to your computer using text.
- **Executable File:** A type of file that contains a program that can be run on your computer. It's like a machine that performs a task when you click it.
- **Environment Variables:** Special settings that help your system know where to find certain tools or files. They're like a map that tells your computer where to look for things it needs.

# ALGOSUP

---

International Software Development School