Echo Global Logistics
EV Routing App Instruction Manual
Professional Training and Operations Guide
May 2025

Table of Contents

1. What This Manual Covers

This manual explains everything needed to set up, deploy, and maintain the Echo Global Logistics EV Routing Application, even if you have no previous coding or technical experience.

It covers:

- What each tool (Python, Github, Render) is and why it matters
- How to install all required software
- How to upload the app to the internet
- How to configure the app to calculate EV vs Diesel routes
- How to maintain and update the app over time

This manual is designed so that **even someone who has never programmed before** can follow it successfully.

---

2. Prerequisites

Before starting, make sure you have:

- A Windows or Mac computer
    - You need basic access to a computer where you can install programs.
- An internet connection
    - Needed to download software, APIs, and upload your app.
- A Google Account
    - Required to access Google APIs for location services.
- An OpenRouteService Account
    - Required for calculating truck routes.

- About 1 GB free space on your computer
    - To store all project files.

No special technical knowledge is required.

---

3. What is Python?

- Python is a free, open-source programming language.
- It allows people to write instructions that a computer can understand and execute.
- In this project, Python powers the "brain" of the app:
    - Fetching location data (with API calls)
    - Calculating truck routes
    - Creating maps
    - Managing cost and CO2 calculations

Official Python Website:
https://www.python.org/

Why it matters:
Without Python, none of the app's functionality could work.

---

4. Installing Python

1. Go to https://www.python.org/downloads/
2. Click the yellow "Download Python" button.
3. Open the installer file after downloading.
4. On the first installer screen:
    - **Very Important:** Check the box that says **"Add Python to PATH"**.

        - This ensures your computer knows where Python is located.

5. Click **Install Now** and allow the installation to complete.

Why checking “Add Python to PATH” matters:
If you forget, you won't be able to run Python commands later in your terminal (See Part 5: Testing Python Installation).

---

5. Testing Python Installation

After installing Python:

1. Open **Command Prompt** (Windows) or **Terminal** (Mac).
2. Type:
   ```
   python --version
   ```
3. You should see something like:
   ```
   Python 3.11.5
   ```

If you see an error:

- It means Python wasn’t added to PATH.
- You should reinstall Python and ensure you check "Add Python to PATH".

Why this matters:
Testing installation confirms your system is ready for everything else later (like running the app).

---

6. What is Visual Studio Code?

- Visual Studio Code (VS Code) is a free application designed for editing code.
- It is lightweight, easy to install, and beginner-friendly.
- We use VS Code to:

- Open the EV Routing project files
- Edit any settings
- Launch the app locally for testing

Official VS Code Website:
https://code.visualstudio.com/

Why it matters:
Without a good editor, managing multiple files and making changes would be much harder.

---

7. Installing Visual Studio Code

1. Go to https://code.visualstudio.com/
2. Click "Download for Windows" (or Mac if on a Mac computer).
3. Open the downloaded installer.
4. Proceed through the installation with default options.
5. Open VS Code once installation is complete.

No special options need to be changed during installation.

---

8. Installing Necessary Extensions in VS Code

Inside Visual Studio Code:

1. Click the Extensions icon (looks like four squares).
2. Search for and install:
   - **Python** (published by Microsoft)
   - **Pylance** (optional but highly recommended for code suggestions)

Why these extensions matter:

- The Python extension tells VS Code how to recognize .py files and interact properly with Python.
- Pylance provides code hints and error checking while you work.

Without these extensions, you would not get useful feedback while editing files.

---

9. What is Github?

- Github is an online service where people can upload and store files.
- It was designed mainly for code, but you can upload any type of project.
- Github helps you:
  - Store your app files safely in the cloud
  - Track changes you make over time
  - Connect to Render (hosting service) easily

Official Github Website:
https://github.com/

Why it matters:
Render pulls your project directly from Github, meaning you never have to manually upload files again once it is set up.

---

10. Creating a Github Account

Step-by-step:

1. Visit https://github.com/
2. Click "Sign Up."
3. Enter your email address.
4. Create a username and password.

5. Follow the email verification steps to activate your account.

Once finished, you have a free Github account that will store your EV Routing Project files.

---

11. Setting up a Github Repository

A "repository" is like a folder inside Github where your project lives.

1. Log into Github.
2. Click the "+" icon in the top-right corner and select "New repository."
3. Fill out the form:
   - Repository name: ev-routing-project
   - Description: (Optional) EV Routing Application Project
   - Public or Private: Choose Public or Private (either works)
4. Leave "Initialize this repository with a README" **unchecked**.
5. Click "Create Repository."

Your repository is now created, but still empty.

Why it matters:
You cannot upload files to Github without first creating a repository.

---

12. Uploading Your Project to Github

1. Open the Echo Mapping Project folder on your computer.
2. Select **all** files and folders inside it (not just one file).
3. Drag and drop them into the Github page where it says "Drag files here to upload."
4. Scroll down and click "Commit changes."

Now your EV Routing App files are uploaded to Github.

Why it matters:
Render (your hosting service) needs your app files stored somewhere permanent online. Github provides that.

---

13. What is Render?

- Render is an online hosting service that can take your app from Github and make it available live on the internet.
- It connects directly to your Github repository so you don't need to manually upload anything once connected.
- Render automatically restarts your app if it goes offline or if you make updates.

Official Render Website:
https://render.com/

Why it matters:
Render handles the heavy work of keeping your app available online 24/7 without you needing to run it manually.

---

14. Creating a Render Account

Step-by-step:

1. Visit https://render.com/
2. Click "Sign Up."
3. Choose "Sign Up with Github."
    - This allows Render to access your repositories automatically.
4. Authorize Render to connect to your Github account when prompted.

You now have a Render account that can deploy apps.

---

15. Connecting Github to Render

1. In Render, click "New +" → "Web Service."
2. Select your "ev-routing-project" repository from the Github list.
3. Set basic settings:
    - Name: ev-routing-app
    - Environment: Python
    - Build Command: pip install -r requirements.txt
    - Start Command: python app.py
4. Click "Create Web Service."

Render will now begin deploying your app automatically.

---

16. Deploying the Web App

Render will:

- Install required Python libraries listed in requirements.txt.
- Connect your Github files.
- Build your application.
- Start a web server to host it.
- Provide a public link like:
  https://ev-routing-app.onrender.com/

Deployment time:

- First deployment may take 5–10 minutes.

Why it matters:
Once deployed, your EV Routing App is live and ready for anyone to access.

17. Setting Up API Keys (Google and OpenRouteService)

You must obtain two API keys to allow the app to calculate routes.

1. **Google API Key (for Geocoding addresses):**
    - Go to https://console.cloud.google.com/
    - Create a new project.
    - Enable the "Geocoding API."
    - Generate an API Key.
2. **OpenRouteService API Key (for Truck Routing):**
    - Go to https://openrouteservice.org/
    - Create an account.
    - Create a new API Key.
    - Allow permissions for "driving-hgv" (Heavy Goods Vehicles).

18. Placing API Keys into app.py

Once you have the keys:

1. Open app.py in Visual Studio Code.
2. Find the lines:

```
google_api_key = "your-google-api-key"
openrouteservice_api_key = "your-openrouteservice-api-key"
```

3. Paste your real API keys between the quotation marks.

Why this matters:
Without valid API keys, your app cannot fetch addresses or calculate driving routes.

19. Understanding the Project Folder Structure

Your project files are organized like this:

- app.py
  - The main Python program that:
    - Receives user input
    - Calls APIs
    - Calculates EV vs Diesel routes
    - Creates maps
- requirements.txt
  - Lists all the Python libraries the app needs.
  - Render uses this list to know what to install.
- templates/ (Folder)
  - HTML files that create the web pages the user sees.
    - index.html: The homepage where users input data.
    - result.html: Page displaying the result for a single route.
    - batch_result.html: Page displaying results from uploaded Excel files.
- static/ (Folder)
  - Files that stay the same (logos, Excel charger lists, optional styling).

Why this matters:
The folder structure must stay exactly like this or the app won't work properly.

---

20. Understanding How the App Works (Line-by-Line app.py Walkthrough)

app.py is divided into clear sections:

- **Imports**
    - Loads necessary libraries like Flask, pandas, folium, requests.
- **API Key Variables**
    - Stores your Google and OpenRouteService API keys.
- **Flask App Setup**
    - Starts the web server.
- **Route Handlers**
    - index(): Displays the homepage form.
    - result(): Processes a single route.
    - batch_result(): Processes multiple routes from Excel upload.
- **Functions**
    - Calls Google Geocoding API to get latitude and longitude.
    - Calls OpenRouteService API to get truck-safe routes.
    - Creates maps using folium.
    - Calculates total cost and emissions for Diesel and EV.
- **Return Pages**
    - Shows the generated maps and output values on web pages.

Why this matters:
Understanding app.py helps you make changes later if needed (like adjusting truck battery range or charger rules).

---

21. Running the App Locally (On Your Computer)

Before deploying online, you can test it on your machine.

1. Open Visual Studio Code.

2. Open the Echo Mapping Project folder.

3. Open the Terminal inside VS Code.

4. Type and run:

   ```
   pip install -r requirements.txt
   ```

   (Installs necessary libraries.)

5. Run the app by typing:

   ```
   python app.py
   ```

6. Open your browser and visit:
   http://127.0.0.1:5000/

Why this matters:
Testing locally ensures everything works before hosting online.

---

22. Using the App: Single Route Mode

Single Route Mode lets you manually input a start and destination city.

How it works:

- You type in:
  - Start City
  - Start State
  - Destination City
  - Destination State
- The app:
  - Gets coordinates for both cities.
  - Calculates a heavy-truck Diesel route and an EV route.

  - Inserts charger stops every 225 miles on the EV route.
  - Calculates costs and carbon emissions.
- Results are shown with two maps and a summary table.

---

23. Using the App: Batch Upload Mode

Batch Upload Mode lets you upload a full Excel file of routes.

How it works:

- Excel file must have these columns:
  - Start City
  - Start State
  - Destination City
  - Destination State
  - MPG (Optional — defaults to 9 if blank)
  - Annual Trips (Minimum 1 trip)
- After uploading, the app:
  - Processes all routes automatically.
  - Calculates Diesel vs EV costs and emissions for each route.
  - Creates a downloadable Excel output with all results.

Why this matters:
Batch mode saves time when you have many routes to analyze.

---

24. Understanding the Output Files

The output Excel files show:

- Diesel Route Distance
- EV Route Distance
- Diesel Annual Mileage
- EV Annual Mileage
- Diesel Total Annual Cost
- EV Total Annual Cost
- Diesel Total Annual CO2 Emissions
- EV Total Annual CO2 Emissions
- EV Possible? (Yes/No depending if chargers fit)

Why this matters:
The output helps you quickly compare costs and carbon savings between Diesel and EV options.

---

25. Troubleshooting Common Errors

Problem	Cause	Solution
"Flask not found" error	Flask library not installed	Run pip install flask
"KeyError: API key"	API keys missing or incorrect	Recheck app.py and paste keys correctly
"Excel upload fails"	Wrong format	Ensure the uploaded file has correct column names
"Map not showing"	folium library not installed	Run pip install folium

Problem	Cause	Solution
Render app not deploying	Wrong Build or Start Command	Double-check Build: pip install -r requirements.txt, Start: python app.py

Tip:

Always read the Terminal or Render logs carefully — they usually explain what went wrong.

---

26. Maintenance and Updating the Website

Updating the website:

1. Make your edits locally (in Visual Studio Code).
2. Save the changed files.
3. Commit changes to Github:
    - Open Github website.
    - Upload updated files.
4. Render will automatically rebuild and redeploy the app.

Other Maintenance Tasks:

- Refresh your API keys once per year.
- Update EV Charger Excel files quarterly.
- Test new deployments locally before uploading major changes.

---

27. Glossary (Beginner Tech Dictionary)

Term	Meaning
Python	Programming language used to create the app

Term	Meaning
Visual Studio Code	Program used to open and edit files
Github	Website that stores your project files
Repository	A storage folder on Github
Render	Online service that hosts your website
API	Tool that lets one app communicate with another app
Geocoding	Turning an address into latitude and longitude
Static files	Files that don't change (like images or Excel lists)
Terminal	Where you type commands on your computer
Deployment	Publishing your app live on the internet

---

28. Constants and Settings Breakdown

Important settings inside app.py:

- diesel_mpg = 9
    - Default miles per gallon for Diesel trucks.
- diesel_emission_factor = 1.617
    - Kilograms of CO2 emitted per mile for Diesel trucks.
- ev_battery_range = 20.39
    - Miles one EV truck can drive per full charge.
- ev_charge_cost = 2.208
    - Cost in dollars to recharge one full EV battery.

Why this matters:

You can adjust these numbers later if truck technology changes.

---

29. Full Folder Map and Explanations

Folder layout:

```
Echo Mapping Project/ ├── app.py
├── requirements.txt
├── templates/
│ ├── index.html (homepage)
│ ├── result.html (single route output)
│ └── batch_result.html (batch upload output)
├── static/
│ ├── uiuc_logo.png (logo file)
│ ├── echo_logo.png (logo file)
│ ├── 1.xlsx to 7.xlsx (charger station lists)
│ └── style.css (optional website styling)
```

Important:

- Do not move or rename folders.
- Flask expects "templates/" and "static/" folders to be exactly named.
- The app.py must stay in the main project folder.