PLANNING A TRAVEL JOURNEY

This Prolog program serves as a transportation management system. It helps determine available travel routes between different locations, including both direct and indirect connections. It defines a database of locations, travel modes, and routes, and includes utilities for querying travel possibilities and verifying operations.

- This project is a program designed to assist in planning travel journeys from one location to another.
- It identifies the necessary modes of transportation to move between locations and determines the intermediate stops required along the route.
- The program’s knowledge base includes a set of locations with varying dimensions, three available transportation methods for traveling between these locations, and the various paths that connect them.

Locations and transports

The system's database includes a group of 10 distinct places, 3 Transports

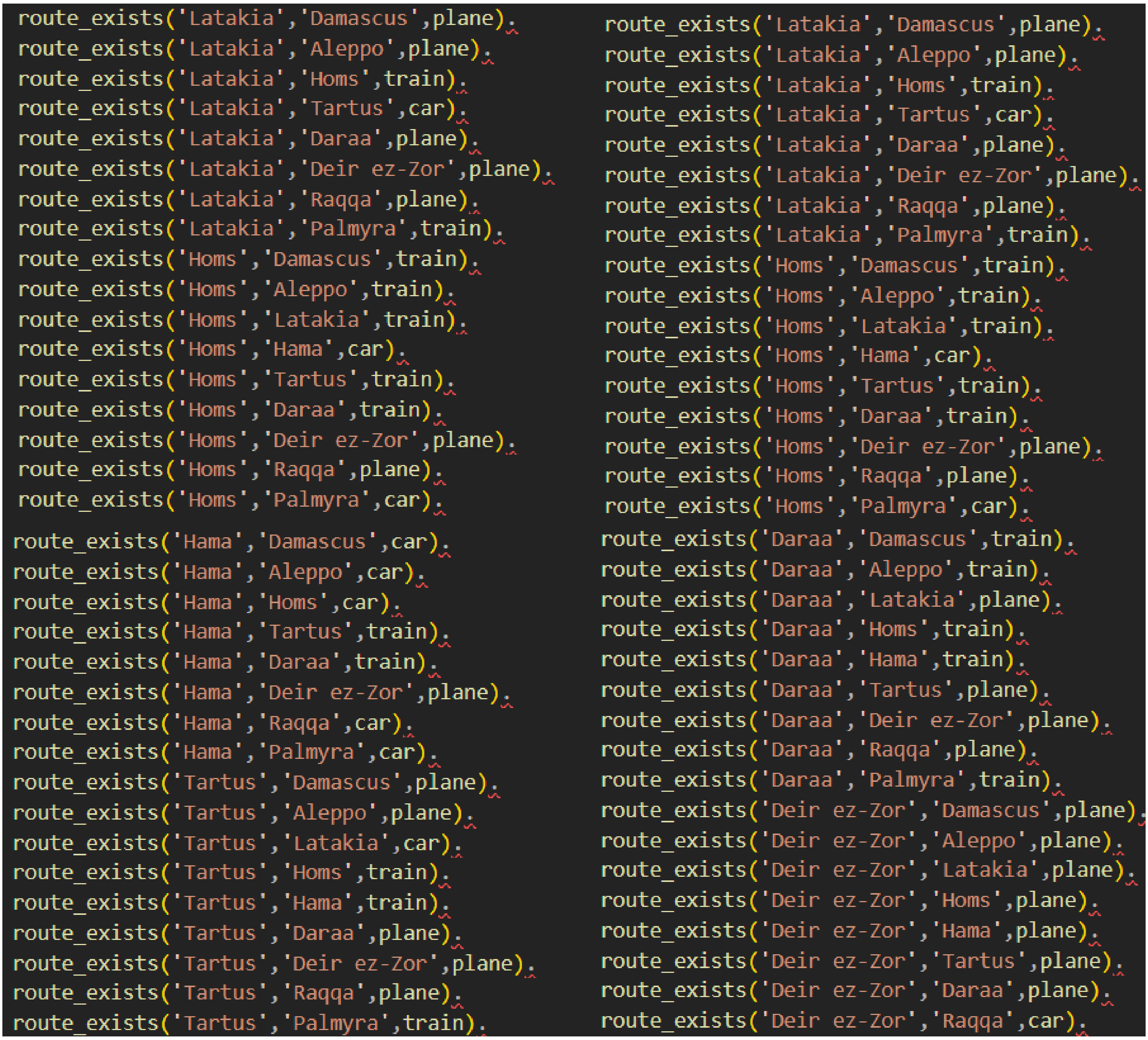
such as :

- Damascus
- Aleppo
- Latakia
- Homs
- Hama
- Tartus
- Daraa
- Deir ez-Zor
- Raqqa
- Palmyra

such as :

- car
- train
- plane

All routes between different places are represented as facts in the program, specifying the starting location, destination, and the mode of transportation. These facts form the basis for the rules utilized in the program. **it's in the code CALLED route_exists()**



```
route_exists('Latakia','Damascus',plane).
route_exists('Latakia','Aleppo',plane).
route_exists('Latakia','Homs',train).
route_exists('Latakia','Tartus',car).
route_exists('Latakia','Daraa',plane).
route_exists('Latakia','Deir ez-Zor',plane).
route_exists('Latakia','Raqqa',plane).
route_exists('Latakia','Palmyra',train).
route_exists('Homs','Damascus',train).
route_exists('Homs','Aleppo',train).
route_exists('Homs','Latakia',train).
route_exists('Homs','Hama',car).
route_exists('Homs','Tartus',train).
route_exists('Homs','Daraa',train).
route_exists('Homs','Deir ez-Zor',plane).
route_exists('Homs','Raqqa',plane).
route_exists('Homs','Palmyra',car).
route_exists('Hama','Damascus',car).
route_exists('Hama','Aleppo',car).
route_exists('Hama','Homs',car).
route_exists('Hama','Tartus',train).
route_exists('Hama','Daraa',train).
route_exists('Hama','Deir ez-Zor',plane).
route_exists('Hama','Raqqa',car).
route_exists('Hama','Palmyra',car).
route_exists('Tartus','Damascus',plane).
route_exists('Tartus','Aleppo',plane).
route_exists('Tartus','Latakia',car).
route_exists('Tartus','Homs',train).
route_exists('Tartus','Hama',train).
route_exists('Tartus','Daraa',plane).
route_exists('Tartus','Deir ez-Zor',plane).
route_exists('Tartus','Raqqa',plane).
route_exists('Tartus','Palmyra',train).
```

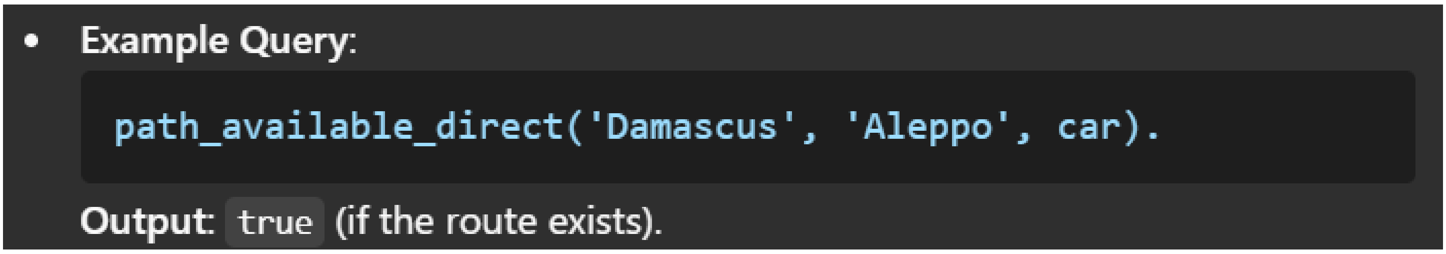
```
route_exists('Latakia','Damascus',plane).
route_exists('Latakia','Aleppo',plane).
route_exists('Latakia','Homs',train).
route_exists('Latakia','Tartus',car).
route_exists('Latakia','Daraa',plane).
route_exists('Latakia','Deir ez-Zor',plane).
route_exists('Latakia','Raqqa',plane).
route_exists('Latakia','Palmyra',train).
route_exists('Homs','Damascus',train).
route_exists('Homs','Aleppo',train).
route_exists('Homs','Latakia',train).
route_exists('Homs','Hama',car).
route_exists('Homs','Tartus',train).
route_exists('Homs','Daraa',train).
route_exists('Homs','Deir ez-Zor',plane).
route_exists('Homs','Raqqa',plane).
route_exists('Homs','Palmyra',car).
```

```
route_exists('Daraa','Damascus',train).
route_exists('Daraa','Aleppo',train).
route_exists('Daraa','Latakia',plane).
route_exists('Daraa','Homs',train).
route_exists('Daraa','Hama',train).
route_exists('Daraa','Tartus',plane).
route_exists('Daraa','Deir ez-Zor',plane).
route_exists('Daraa','Raqqa',plane).
route_exists('Daraa','Palmyra',train).
route_exists('Deir ez-Zor','Damascus',plane).
route_exists('Deir ez-Zor','Aleppo',plane).
route_exists('Deir ez-Zor','Latakia',plane).
route_exists('Deir ez-Zor','Homs',plane).
route_exists('Deir ez-Zor','Hama',plane).
route_exists('Deir ez-Zor','Tartus',plane).
route_exists('Deir ez-Zor','Daraa',plane).
route_exists('Deir ez-Zor','Raqqa',car).
```

FUNCTIONS

DIRECT PATH:

path_available_direct(Start, End, Mode):



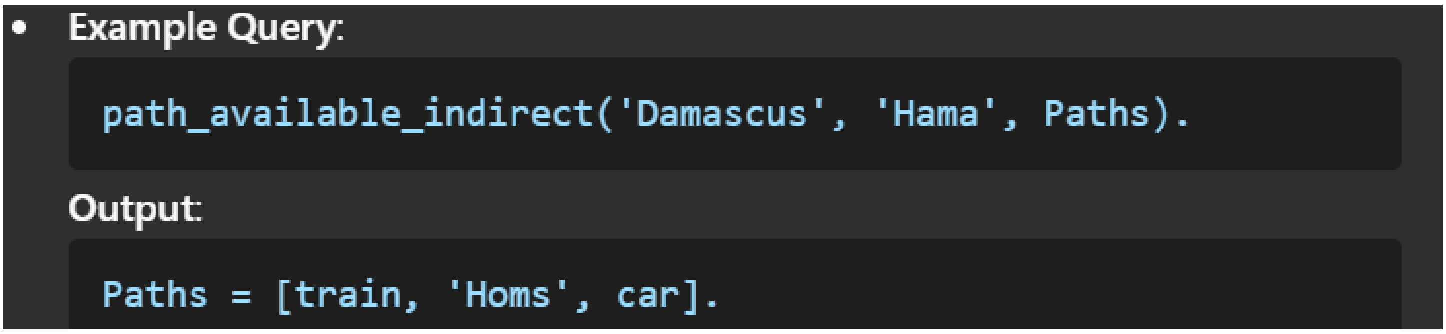
- Example Query:

```
path_available_direct('Damascus', 'Aleppo', car).
```

Output: `true` (if the route exists).

INDIRECT PATH:

path_available_indirect(Start, End, Mode):



- Example Query:

```
path_available_indirect('Damascus', 'Hama', Paths).
```

Output:

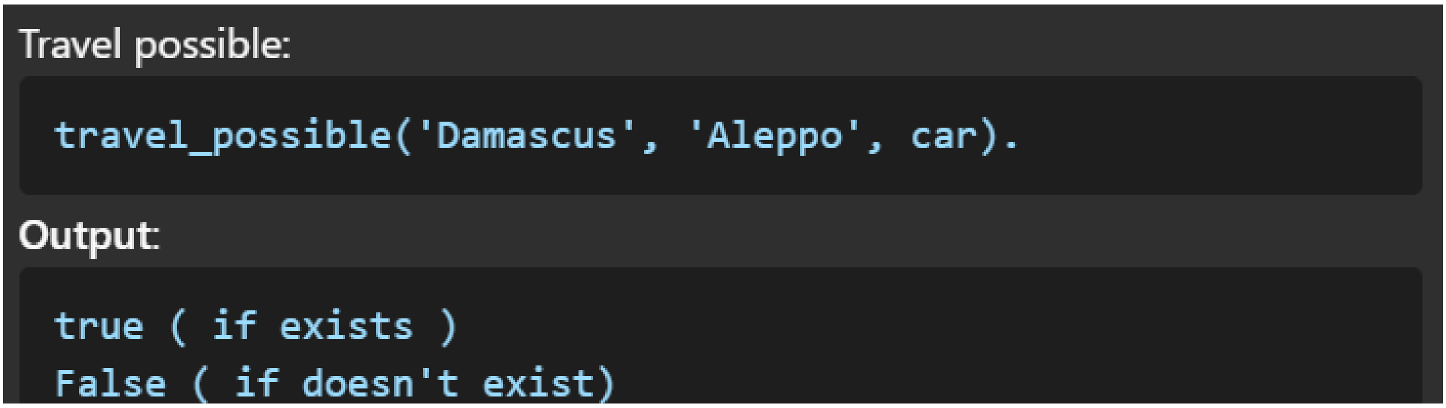
```
Paths = [train, 'Homs', car].
```

Finds an indirect travel route from Start to End using an intermediate location.

Returns the travel path as a list, including the intermediate location and the transport modes.

TRAVEL POSSIBLE

travel_possible(Start, End, Paths):



Travel possible:

```
travel_possible('Damascus', 'Aleppo', car).
```

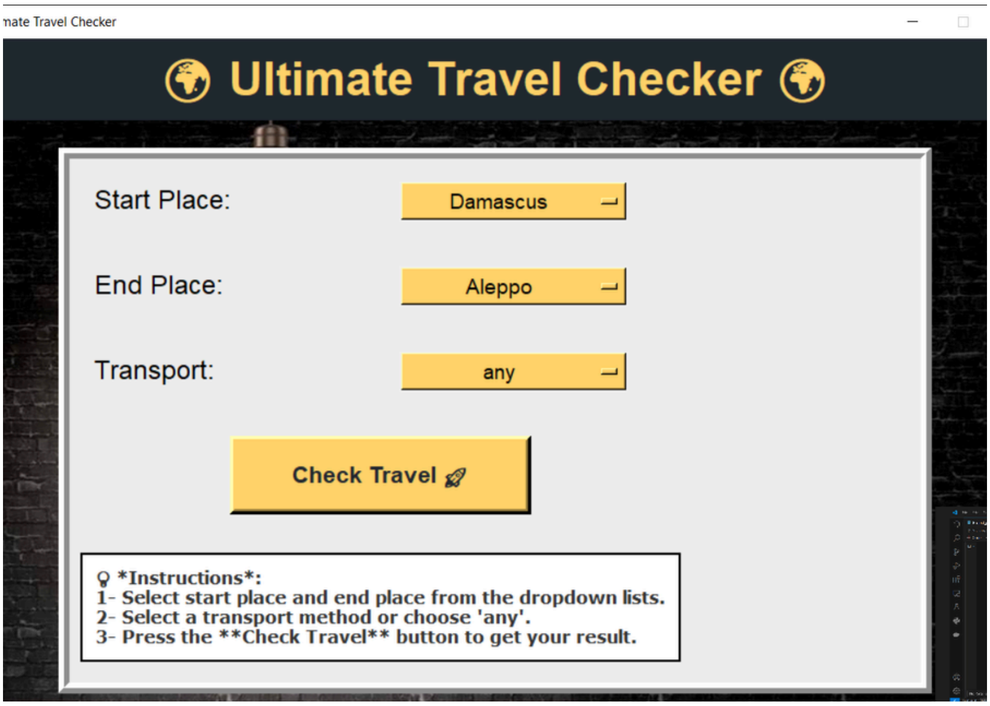
Output:

```
true ( if exists )
False ( if doesn't exist)
```

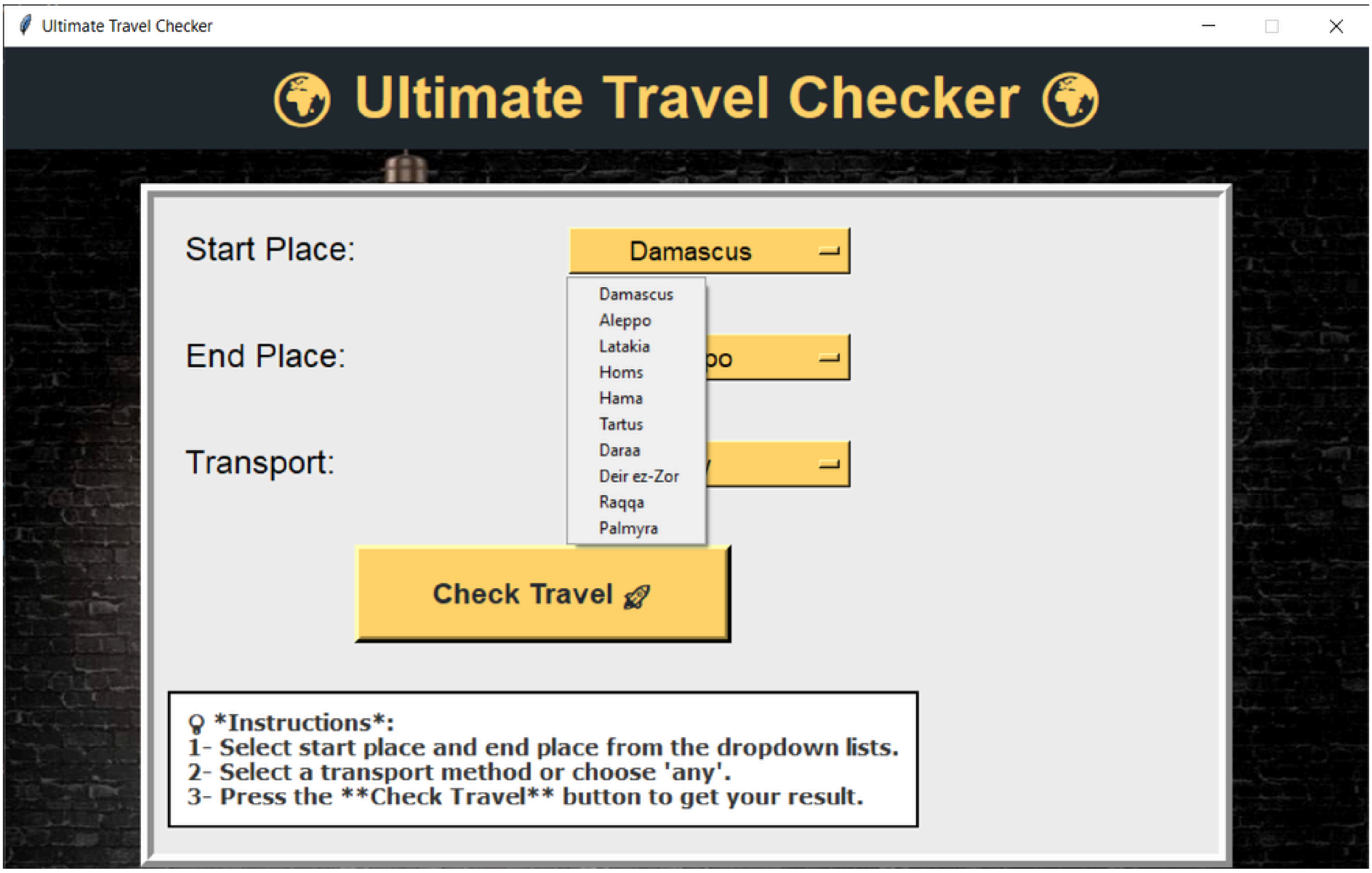
1. First, checks for a direct route using path_available_direct
2. If no direct route exists, it searches for an indirect path using path_available_indirect.

THE GUI :

The interface :

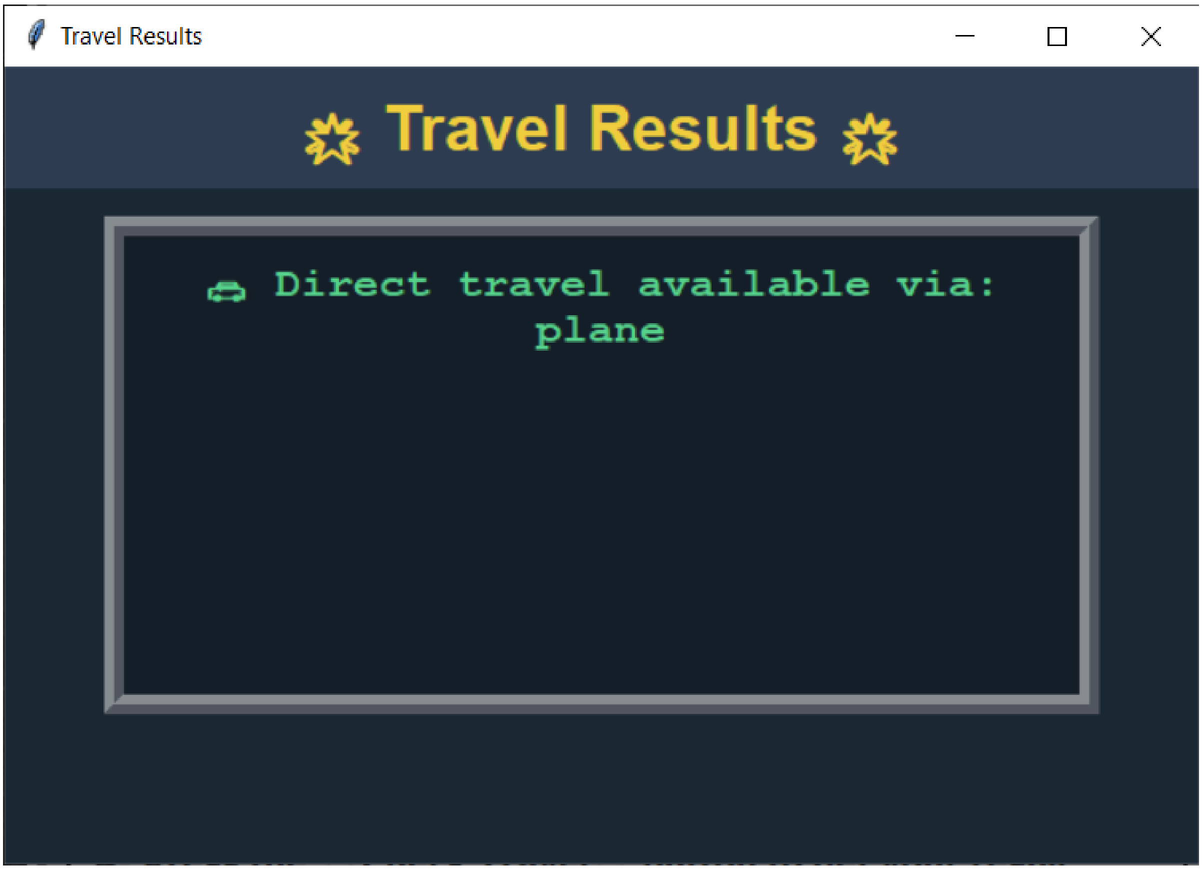


Instructions:
1- Select start place and end place from the dropdown lists.
2- Select a transport method or choose 'any'.
3- Press the **Check Travel** button to get your result.



Instructions:
1- Select start place and end place from the dropdown lists.
2- Select a transport method or choose 'any'.
3- Press the **Check Travel** button to get your result.

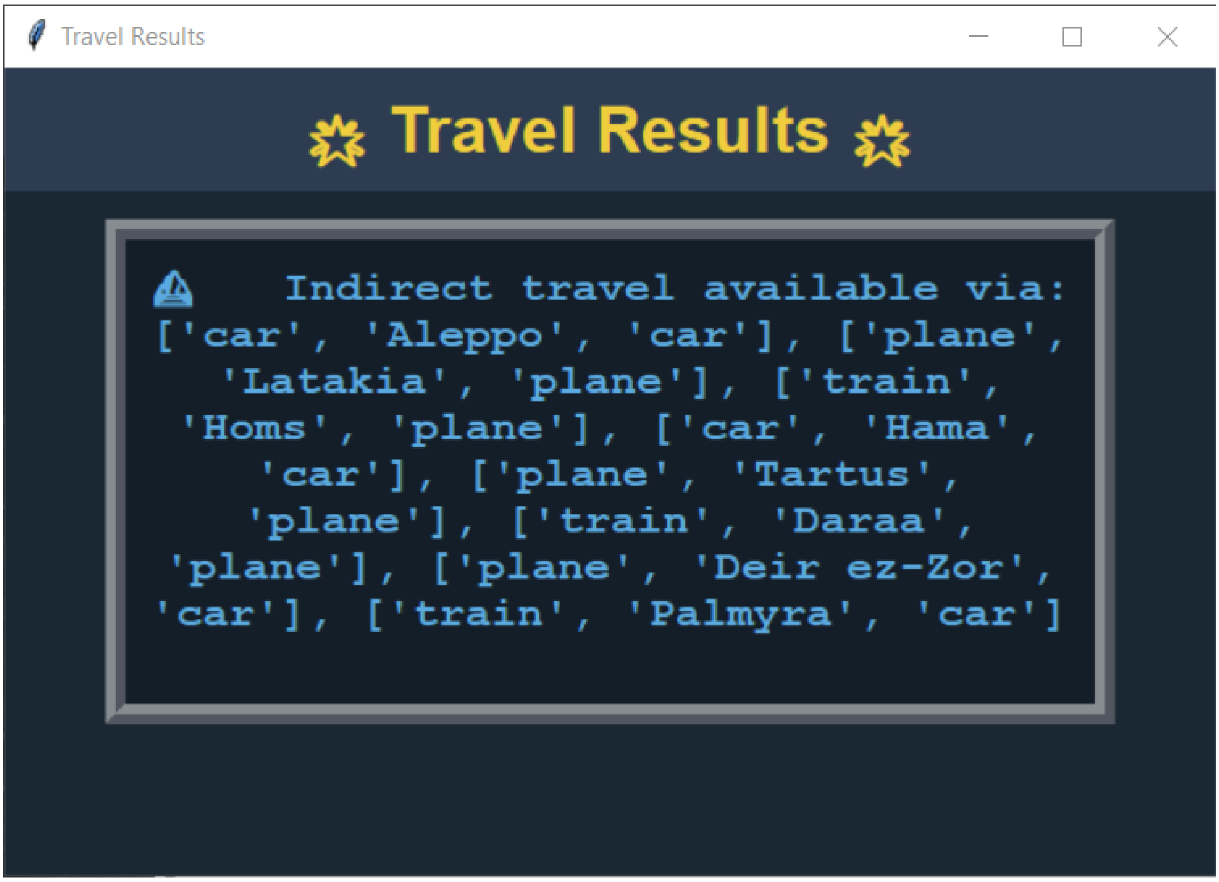
If There’s a direct travel , this interface will show



Travel Results

```
🚗 Direct travel available via:
plane
```

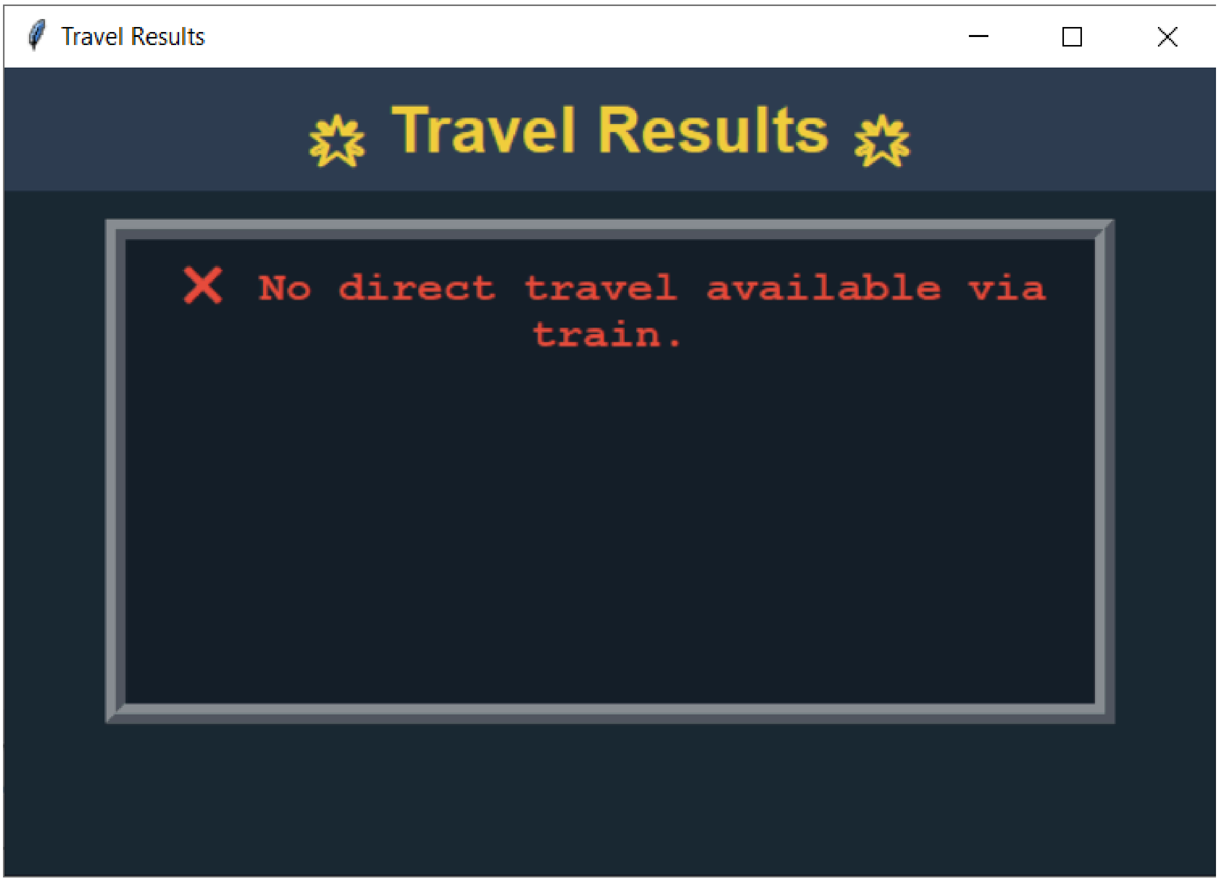
If There’s a indirect travel , this interface will show



Travel Results

```
⚠   Indirect travel available via:
['car', 'Aleppo', 'car'], ['plane',
'Latakia', 'plane'], ['train',
'Homs', 'plane'], ['car', 'Hama',
'car'], ['plane', 'Tartus',
'plane'], ['train', 'Daraa',
'plane'], ['plane', 'Deir ez-Zor',
'car'], ['train', 'Palmyra', 'car']
```

If There’s a no direct travel , this interface will show



✨ Travel Results ✨

❌ No direct travel available via train.