Interactive Map using Leaflet-web mapping report

Introduction

Web maps can display more information than paper maps or other types of maps. Especially when interactive features or animations are added, they become more attractive to users and can display any information in a more flexible and free way. This project attempts to link the city's transportation system with tourist landmarks, offering basic functions to provide visitors with basic information about the city's tourism and transportation. Although there are currently many interactive maps on this topic, some of which are very expressive, because this project is based on offline LEAFLET services, all data is local. This means that data must be downloaded locally, processed, and then called locally rather than using existing service APIs. As a result, the richness of data and functionality may not be as extensive as other projects that widely use online service APIs. However, it still has some basic interactive features for users to use and explore.

The target group for this project is mainly tourists. Visitors who wish to visit Salzburg can use this map to learn basic information about the city's transportation system and the 10 most popular landmarks. Some basic interactive features can help visitors get a preliminary understanding of the city. For example, visitors can click on each landmark icon to display some information and pictures, and there is also a link to the landmark's official website. Visitors can also see the nearest station and the surrounding bus routes. If visitors want to know more detailed transport information, there are also links in the information column on the left that take them directly to the city's transportation website. A more detailed description of the map function will be provided in the following section.

Data sources

This project mainly utilised Salzburg city traffic data, Salzburg boundaries, and data for the top ten most popular landmarks in Salzburg. All traffic data for Salzburg was sourced from OpenStreetMap, including:

- Salzburg transport lines (line features)
- Bicycle lane (line features)
- Bus station (point features)
- Footpath (line features)

The city boundary data for Salzburg also comes from OpenStreetMap, including:

- Salzburg city boundary (line features)

The data on the top ten most popular landmarks in Salzburg, including their names and related attribute information, is sourced from the official Salzburg travel guide website (https://www.salzburg.info/en). The exact coordinates of these landmarks are obtained from Google Maps. The top ten landmarks in Salzburg include:

- Landmarks (point features).

Since these data have a lot of attribute information, only the attribute fields used in the map are shown here. For city traffic data, only the name fields of the station and bus route data are used for display. Due to a large number of bicycle lanes and walking paths and some missing name fields, no attribute information is displayed for the bicycle lanes and footpaths. Only line features are displayed on the map. Also, since there's no need to show any extra information, the city boundaries of Salzburg are only shown as line features on the map.

For point features of landmarks, in order to display some related information, the attribute fields that need to be used are :

- Name
- Brief introduction
- Image URL
- Link URL for more information.

Other data used includes three different basemaps: the standard OpenStreetMap, the World Satellite Image Map, and the Open Topographic Map. All three maps are provided by OpenStreetMap services.

Methodology

1. Data preprocessing

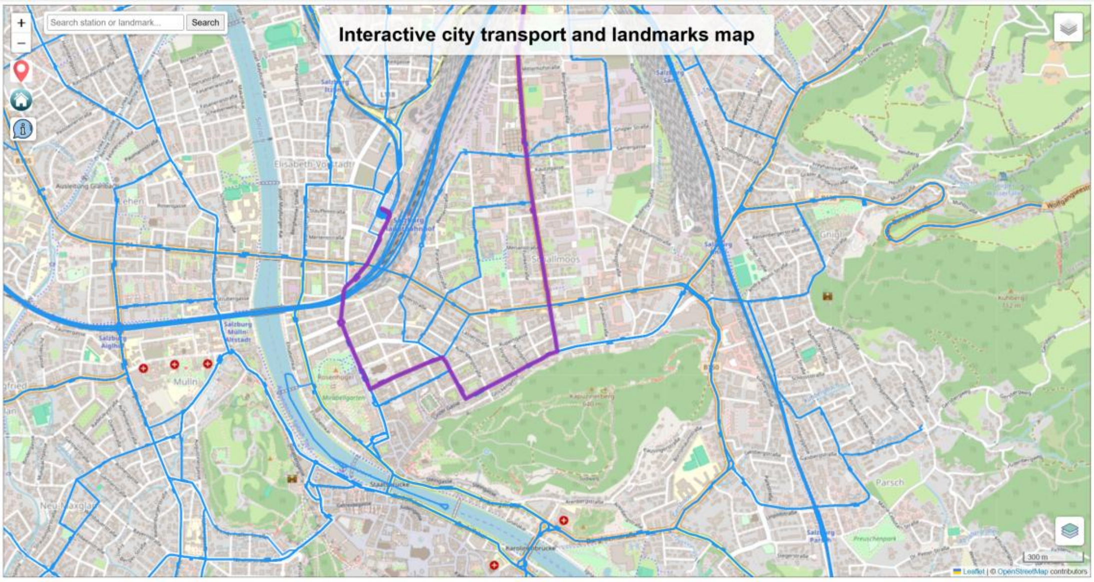
The first step is to download the city transportation data. In this case, the data was downloaded from OpenStreetMap in Shapefile format. This format can be directly opened in geographic information software such as ArcGIS Pro or QGIS. The data must first be filtered, as the source data contains a wide variety of data categories. Here, the specific data categories relevant to this case must be filtered out first. Next, the filtered data is clipped using vector data of the Salzburg boundary, retaining only the data within the city boundary. During this process, it is important to ensure that the attribute table remains intact. Finally, the pre-processed data is exported into GeoJSON format according to the different layer categories displayed on the map, and these data are stored in the project's data folder. Since the data for the ten landmarks was compiled into a table after obtaining all the information and directly imported into ArcGIS Pro to generate point features, no pre-processing is required here. Simply convert the shapefile format to GeoJSON and save it in the correct folder.

2. Map functions

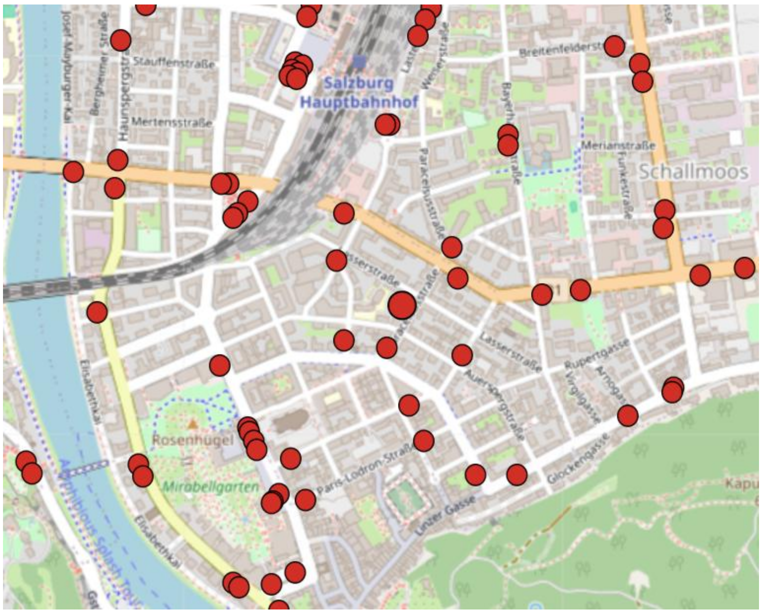
In order to better show Salzburg's landmarks and transportation system, the following interactive events have been added to the map.

- On the right side of the map, two layer control buttons are used at the top and bottom to organize different base maps and data layers. The two buttons are distinguished by different icons: the upper button is used to switch between base maps, and the lower button is used to switch between data layers. After clicking the button, there will be a title to distinguish the functions of the different control buttons. Clicking the control button will also display a list of data layers and base maps.
- For bus lines, when the user moves the mouse over a particular bus line, it can be highlighted in a different colour. Bus stops also have this feature. When hovering the mouse over a bus stop, the dot icon will enlarge.

Here, two different bus line icons are defined, and a mouse hover event is set up. When the mouse hovers over a bus line, the original icon is replaced with a highlighted icon. The highlighted display at the station also uses this method.



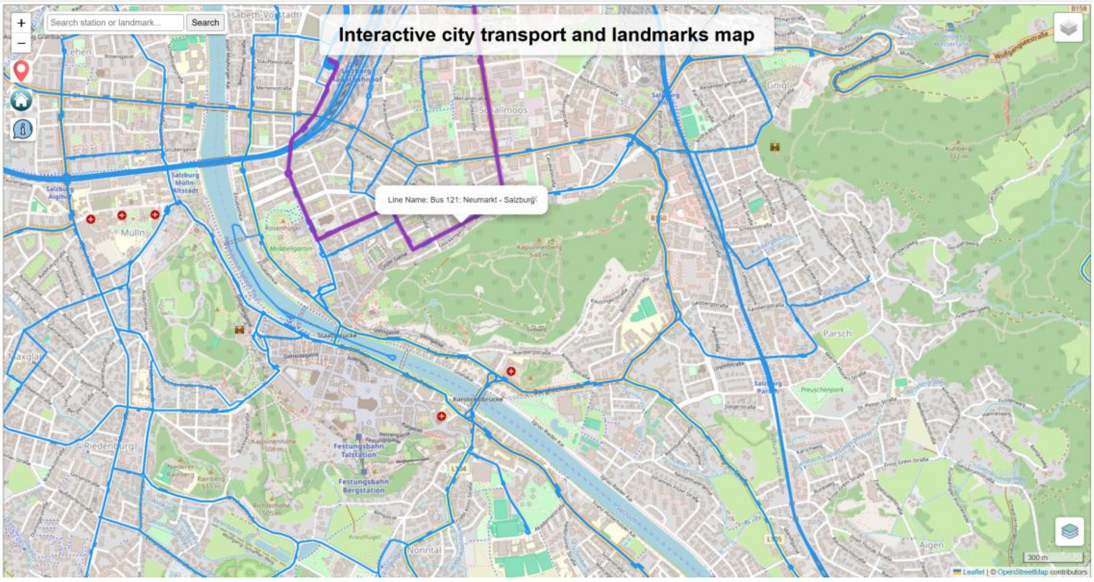
(Figure 1. When the user hovers the mouse over a bus line, it will be highlighted.)



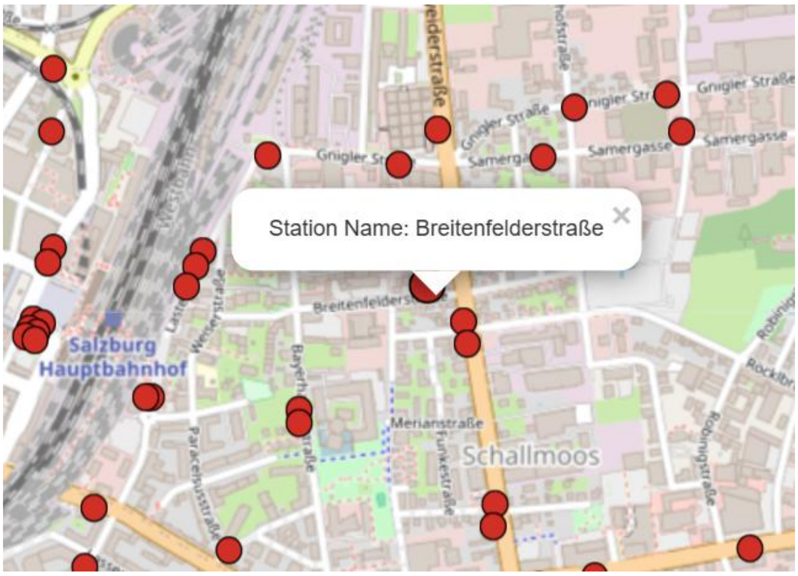
(Figure 2. When the mouse hovers over the station dot icon, it will enlarge.)

- Bus line and station icons also have a pop-up function. When users click on a route or icon, the station name and the bus route number, as well as the starting and ending stations, will be displayed.

Here, simply needs to bind a pop-up window function and set the content to the fields that want to display in the data attributes.



(Figure 3. A pop-up window displaying bus line attributes.)

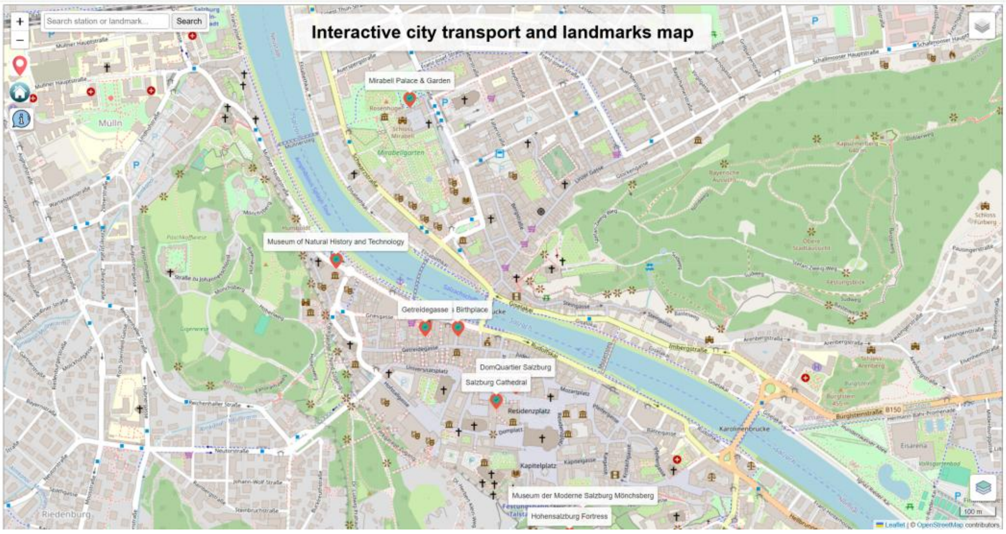


(Figure 4. A pop-up window displaying the station name.)

The other three features (Bicycle path, Salzburg city boundary, and footpath) are distinguished and displayed using different colours only, without pop-up windows or highlighting functions.

For the display of landmarks, a number of additional features have been added here. Similarly, when the mouse hovers over a landmark icon, the icon will also be enlarged. This is the same method as before.

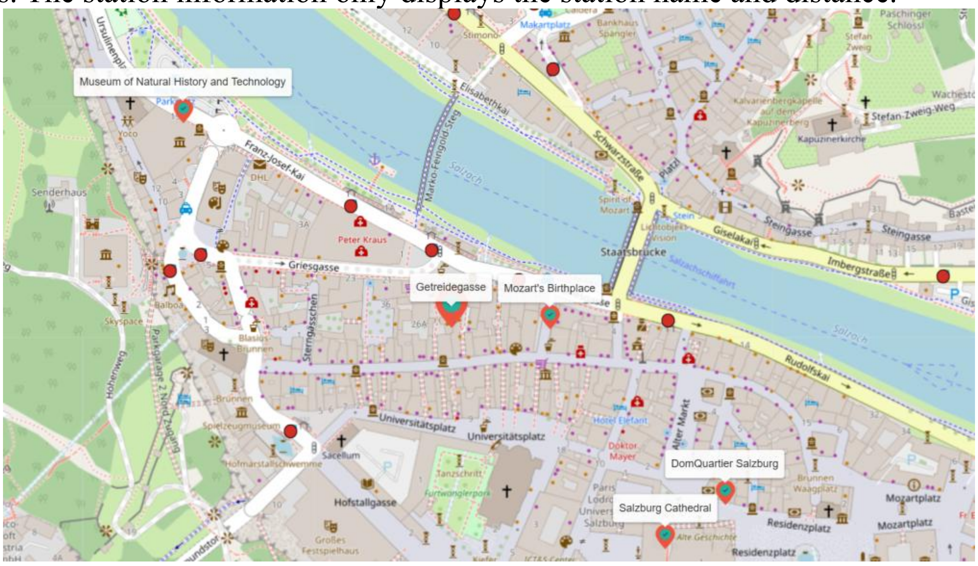
- Firstly, the labels were added to the landmark data to display the names of landmarks, which will be displayed when the landmark layer is opened.



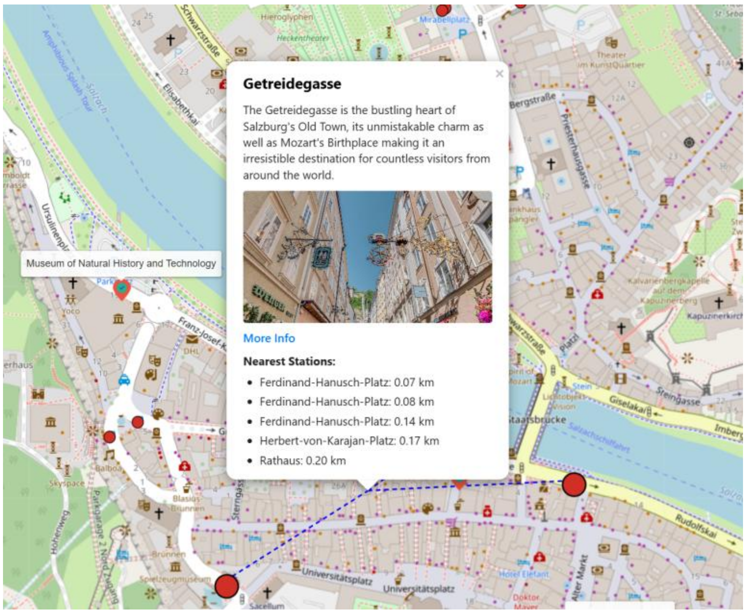
(Figure 5. Add a label above the landmark icon to display the name.)

- When users click on the landmark icon, a pop-up window will appear displaying the name of the landmark, a brief introduction, a picture, and a link button for more information.
- In order to connect landmarks with urban transportation, the pop-up window also displays the five stations closest to the landmark, along with their names and distances.
- At the same time, lines will connect the nearest stations and landmarks, and the selected station icon will also be enlarged when the landmark is clicked. This makes it easier for users to recognise.

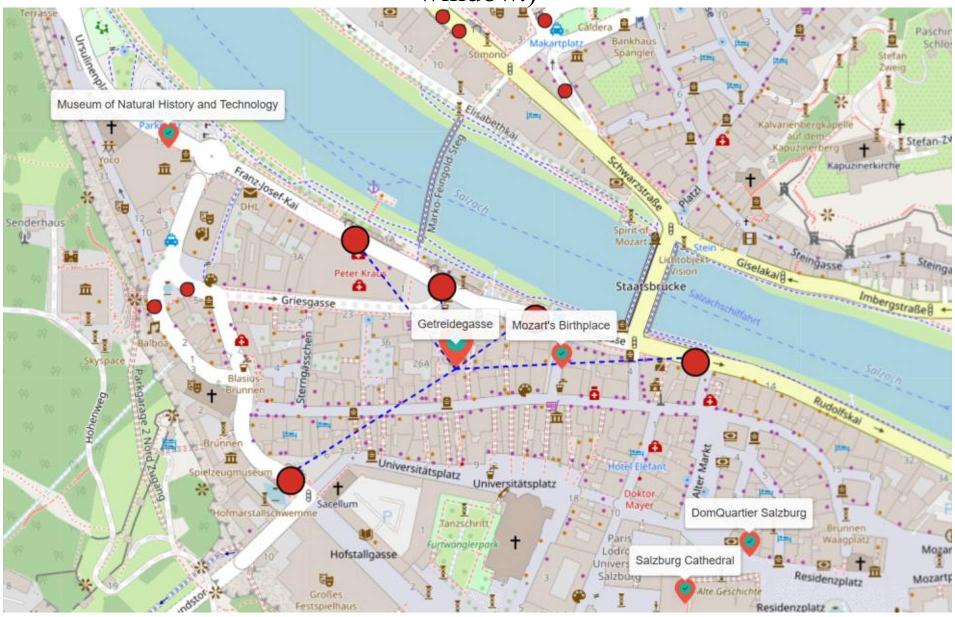
The method here is to first define a function to calculate the straight-line distance, use this function to calculate the distance between each station and the landmark, and then select the five stations with the shortest distances to be added to the array. Since the landmark data already has fields for name, brief description, image URL, and link URL, so only need to define the style of the pop-up window here. Then display the above attributes of the landmark data, and link them to the information of the five selected stations. The station information only displays the station name and distance.



(Figure 6. When the mouse is placed over a landmark icon, the icon will be enlarged.)



(Figure 7. When clicking the icon, the information will be displayed in a pop-up window.)

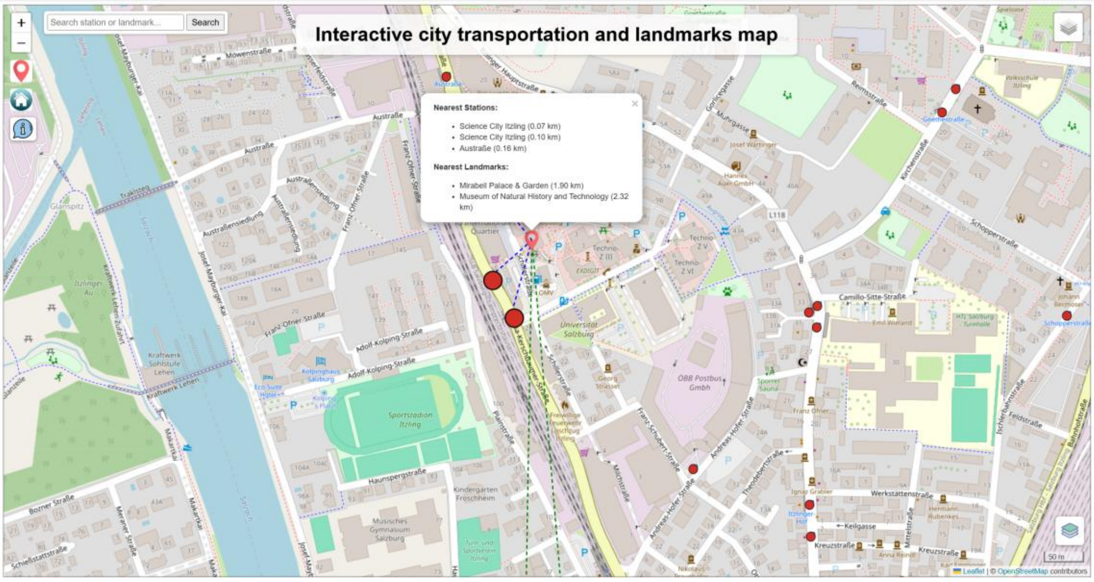


(Figure 8. The five stations closest to the landmark will be displayed in an enlarged icon.)

In addition, a function to display the user's location has been added to make it easier for users to find their location.

- Click the red button in the upper right corner of the map, and a pop-up window will appear asking if the user wants to allow the website to access their location. Once their location is obtained, the map will move to their current location.
- While moving to the current location, a pop-up window will display the three nearest stations and two nearest landmarks. The icons for the nearest stations will also be highlighted to help users find them easily.

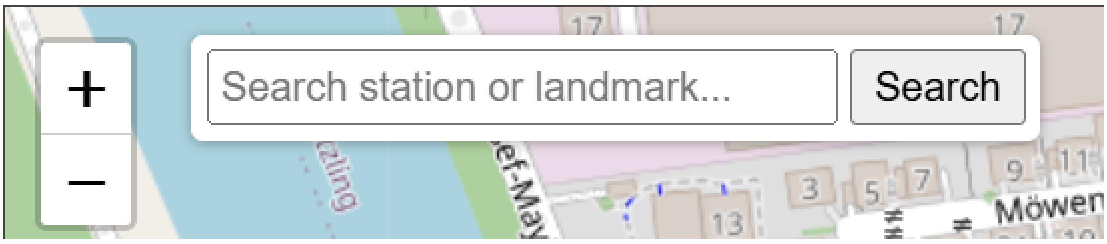
Here, user coordinates are obtained using the HTML5 Geolocation API, and then the view is reset to the user's coordinates. The method for calculating the shortest distance and displaying it in the pop-up window is the same as before.



(Figure 9. Display a pop-up window while locating the user's position and highlight the nearest station.)

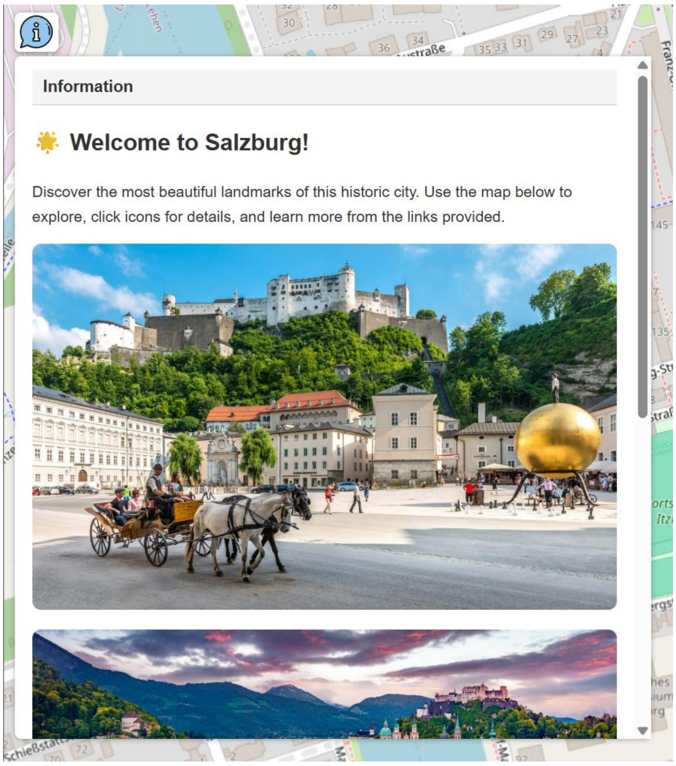
Other functions include:

- Double-click anywhere on the map to zoom in to the nearest bus stop.
- Some basic map control functions: hold down the mouse button and drag the map, and scroll the wheel to zoom in and out on the map.
- Control buttons for zooming in and out on the map.
- The search box is used to search for and locate target locations by station and landmark names.
- Go back to the original map view position.
- The information bar contains a brief introduction to the map, as well as data sources, links to more information about city transportation and landmarks, and also the map author.



(Figure 10. Zoom in/out button and search bar.)

(Figure 11. Return to original map view button.)



(Figure 12. Information bar containing more information.)

Limitations and potential improvements

Since this project is based on an offline Leaflet service, the richness and diversity of the data are crucial to the display and interactivity of the map. Although OpenStreetMap can provide a very rich dataset, the ideal dataset for urban transportation data should provide detailed urban road data (even for very small footpaths), detailed public transport route information, railway information, and also ensure that there is no missing or incorrect attribute information for each data record. However, the current data has some issues, as the data provided by OpenStreetMap contains some errors and omissions. For example, some bus line data on the map is interrupted or duplicated. The station data attributes are missing information about which routes pass through these stations and which bus routes use these stations. The data for bicycle lanes and footpaths is very messy, and most of it is missing basic attribute information such as names.

Obtaining data from government departments, in particular, those institutions responsible for managing urban public transport systems, represents a highly feasible and promising approach. In the initial efforts, an attempt was made to acquire data from the Salzburg urban transport system, and successfully downloaded the General Transit Feed Specification (GTFS) dataset. While this dataset provided a broad range of detailed information, it was primarily structured in tabular form. Geographic Information System (GIS) software such as ArcGIS Pro or QGIS can indeed convert coordinate data into shapefiles (SHP) while preserving the associated tabular data within the attribute table. However, a significant limitation remains: the dataset lacks essential attribute information regarding bus routes—specifically, which stations are served by each route and which buses stop at each station. This type of information is critical for the effective construction of a comprehensive city transportation map.

Moving forward, it may be worthwhile to explore alternative data sources or consider establishing direct communication with relevant authorities to obtain more detailed and structured datasets.

Another existing limitation is that local data sources are insufficient, and the styles and functionalities are relatively limited. Therefore, extensively utilising external online API services is a good method to enhance the richness of maps. However, it is important to consider that some APIs require payment for use, or for free users, there may be limited call quotas or restrictions on simultaneous calls. This leads to significant limitations for online maps, as calculating distances often requires iterating through each station, which may necessitate multiple or simultaneous API calls, potentially exceeding free users' limits. Additionally, API call failures can prevent the map from loading or cause errors, unless a command is used to skip the failed call and proceed to the next step, in which case the map may fail to display properly.

At the same time, while calling more online API services can effectively enhance map elements, it is important to note that in this situation, the online map cannot be used in environments without internet connectivity or with poor network conditions. Therefore, a potential improvement for this issue is to carefully select data, optimise its structure, and present more information within a reasonable data size. Additionally, through coding, it is possible to achieve some excellent styles and functionalities. This requires map designers to extensively read and absorb inspiration from existing online map cases, learn from good patterns, and stimulate creativity to design more attractive online maps.

Finally, regarding the calculation of the distance to the nearest station from a landmark, this project uses a straight-line distance rather than an actual distance, which is another limitation. In some cases, the straight-line distance may appear short, but the actual distance is longer than expected, which is related to the city's road network. However, incorporating accurate distance calculations into the project requires downloading detailed city road data and storing it in a local database. Although OpenStreetMap provides some detailed road data, it still requires filtering and organisation. On the other hand, actual distances often have multiple options, as can be seen in online mapping software like Google Maps. After selecting a destination, there are multiple ways to get there, and even when choosing to walk, there may still be more than one route planning option. The shortest distance here also needs to be further clarified: is it based on the shortest distance or the shortest arrival time? Using static road data often leads to discrepancies between road planning and actual conditions, as it is impossible to update the data in real time, and users are also unable to report actual conditions to the service centre.

A potentially good approach to this problem is to call some relevant online service APIs. These online services can update city road data in real time and have some artificial intelligence algorithms that are more flexible and can calculate and formulate the shortest route based on user preferences and needs. Of course, using detailed road data and storing it in a local database is still another good solution, but because it cannot be updated in real time, it will always result in reduced practicality. This point is worth further exploration in future research and practice.

Conclusions and Discussion

This project has initially achieved a static association between city landmarks and public transportation, focusing on displaying basic information about city landmarks, city transportation, and the distance to the nearest station. It also provides users with links to access more detailed information. The next step is to further improve the project based on its existing shortcomings and deficiencies. For urban transportation, since static data always has shortcomings, it is possible to consider using some existing APIs to compensate for these shortcomings. The interactive features of the map can be further enhanced by incorporating video and animation elements to increase user interest in landmarks. Additionally, surrounding facilities such as restaurants, cafes, parking lots, and commercial areas can be included in the map's display scope. The final goal is to create a real-time, attractive, and comprehensive interactive online map that offers a rich and complete user experience. At the same time, it is also possible to explore the integration of user-generated content and feedback mechanisms, enabling continuous updates based on user input and enhancing the connection between users and maps.

Note: Please review the detailed comments in the code and the readme file for more explanations about the code and functions.