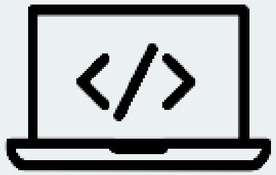
# THE CITY LIFE RANK

JavaScript Advanced Project



edited by Matteo Battilani

# INDEX

# PROJECT DESCRIPTION



The project 'The City Life Rank' was created as the final test of the Start2Impact course on 'Javascript Advanced'. The requirement was to create a page consisting of a textbox to allow the user to enter the name of a specific city.

Once the user clicks on the appropriate button, the application must contact the API of the external service Teleport to obtain the information to be displayed on the page.

Once the application has received the data from the external service, it must display information about the categories, description and city score of the city on the page.

## MINIMUM REQUIREMENTS

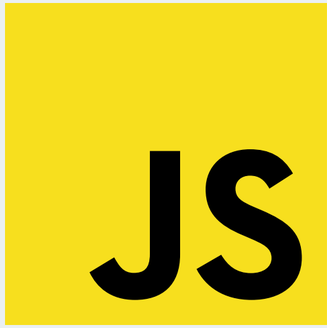
- an HTML page with a search bar and button
- a CSS page with styling
- a JS page with calls to the Teleport API
- show categories, description and city score for each searched city

## THE REQUIREMENTS I ADDED

- ranking of the 10 best and worst cities according to city score (managed with 2 specific functions)
- display of the national flag for the cities in the ranking (REST Countries API)
- listener allowing access to the data of a city in the ranking by clicking on its name
- handling of possible homonyms during the search by creating a list of search results

# TECHNOLOGIES ADOPTED





# THE SEARCH BAR

- The search bar presents a placeholder inviting the user to type the name of a city.
- When the bar is in focus (clicked on by the user), its edges are highlighted in a darker colour.
- By clicking on the 'Search' button, the user is shown a list of search results to choose from.
- A title and a description of the service accompany the search bar.

# CHARTS

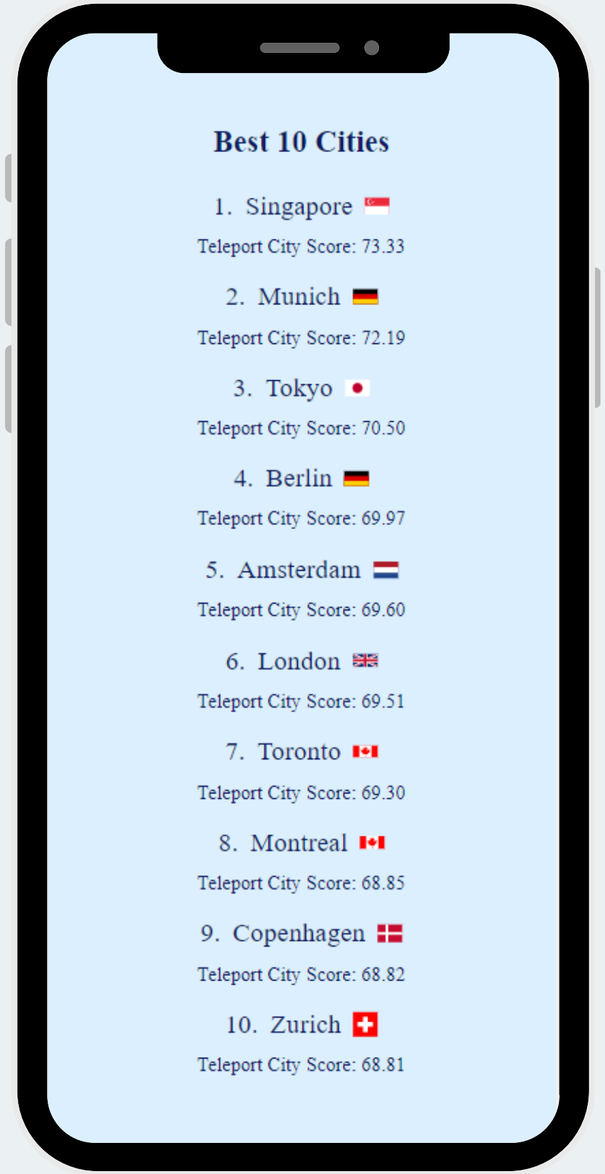
## Desktop version

- The two rankings show the best and worst cities respectively according to the score assigned by Teleport.
- Each city is also accompanied by its country flag (this is possible by calling the REST Countires API).
- Clicking on a city name displays the description, categories and score of that city.

# CHARTS

## Mobile Version

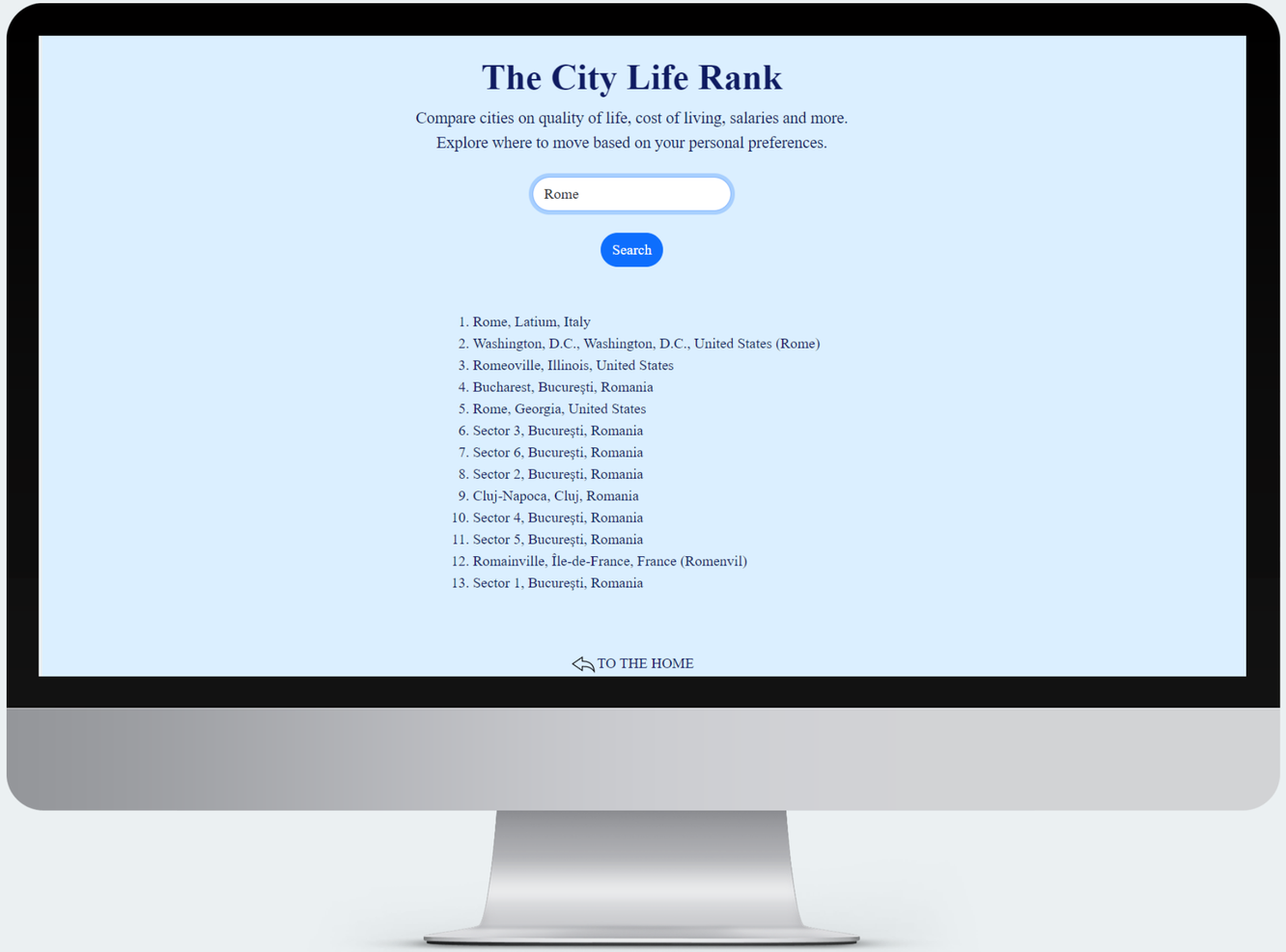
- The mobile version has a different layout, more suitable for vertical viewing.
- The rankings are no longer displayed side by side, but one below the other.



# SEARCH RESULTS

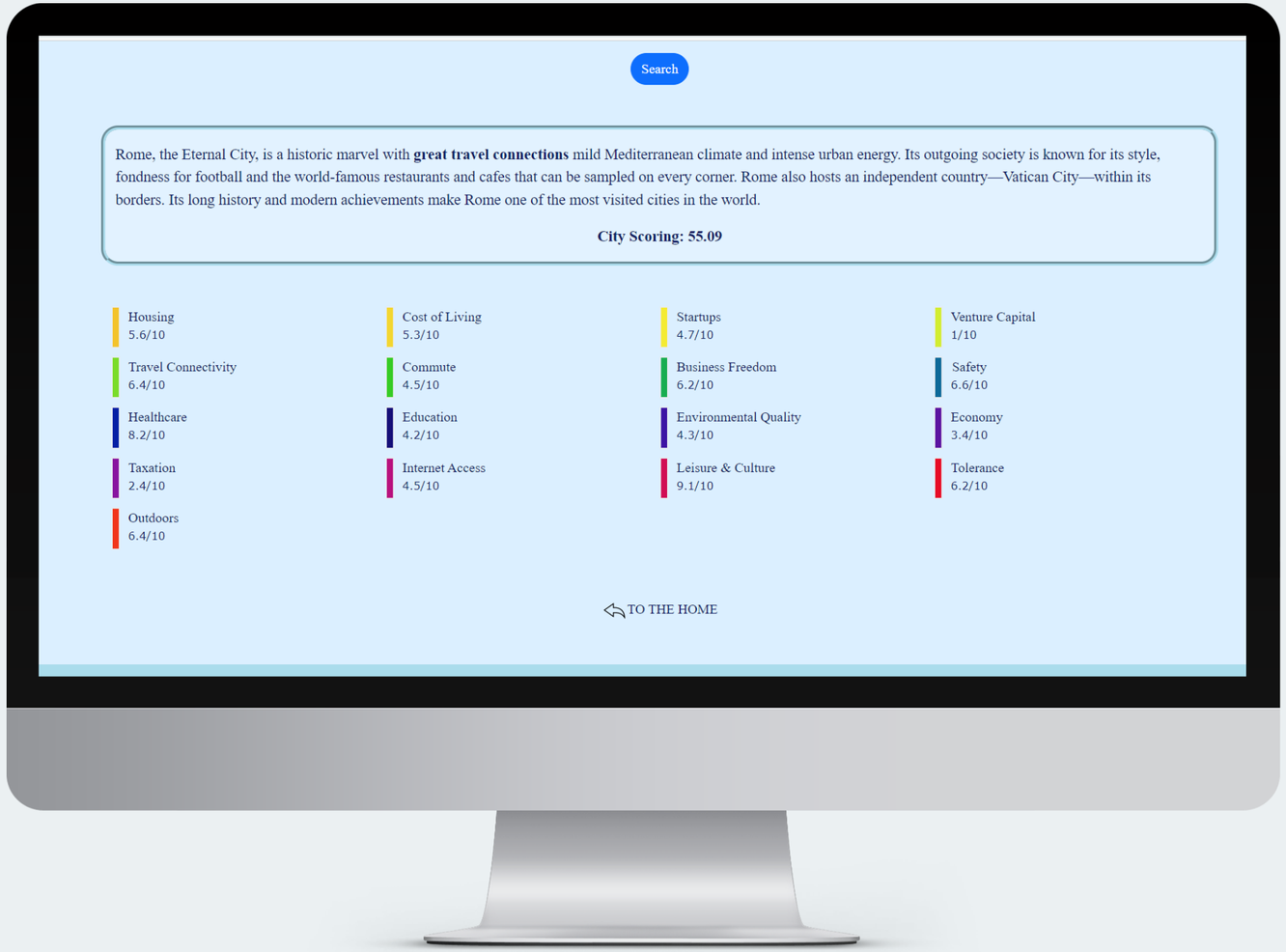
- Once the search button is pressed, the user can choose the city whose details they want to obtain from the list of results. This handles any typing errors and homonyms.

- At the foot of the page, the 'TO THE HOME' button allows the user to display the search bar again with the rankings of the best and worst cities.



# DATA VISUALISATION

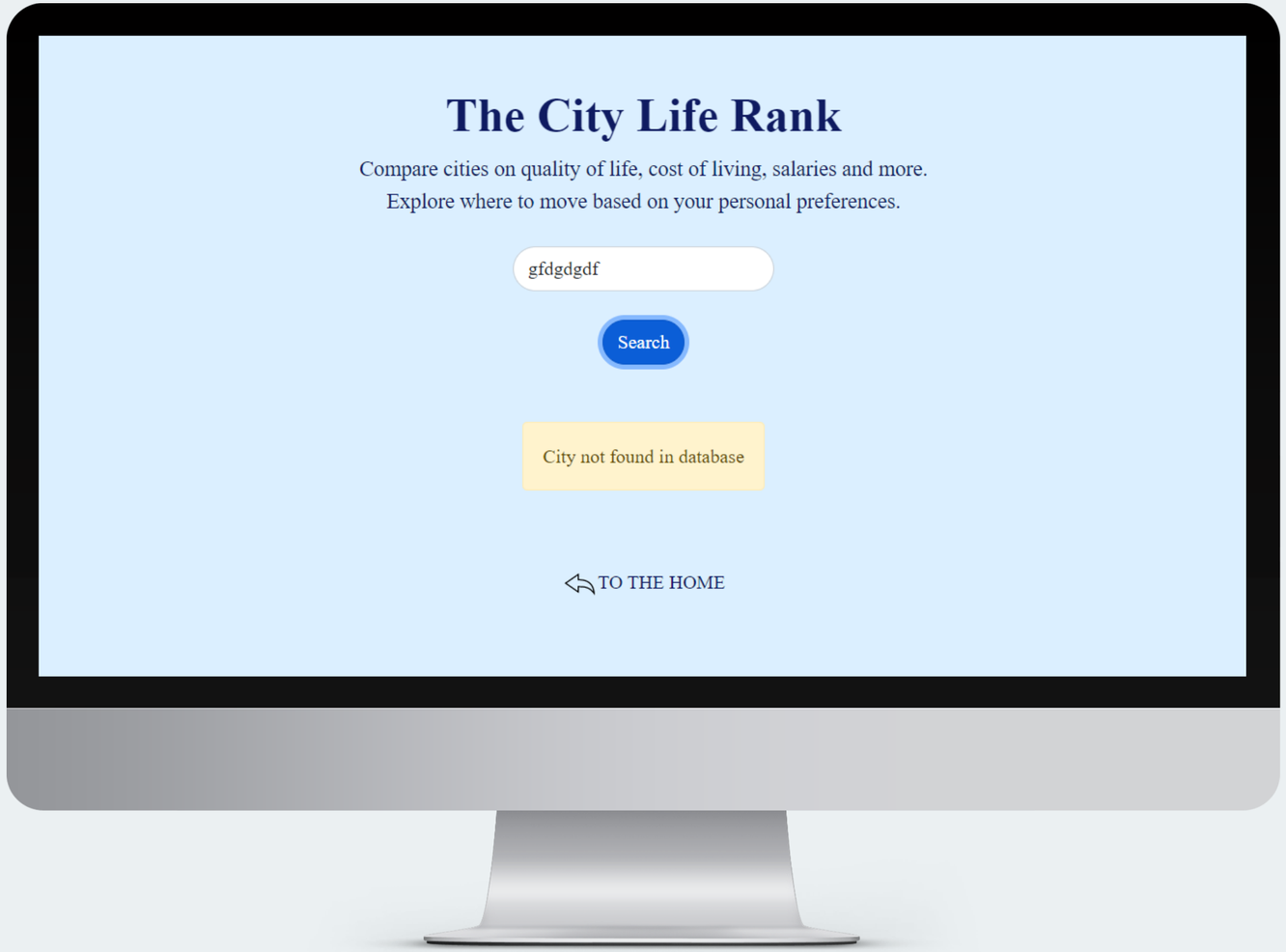
- A coloured box delimits the description of the city.
- The categories, accompanied by their scores, are arranged in a grid. The colour of the categories is also obtained via the Teleport API.
- At the foot of the page, the 'TO THE HOME' button allows the user to view the search bar again with the rankings of the best and worst cities.



# ERROR MANAGEMENT

- If the user enters an invalid city name or one that does not exist in the Teleport database, a message is displayed to warn the user.

- Even if the city is present in Teleport's databases, but has no description, category or score, it is excluded from the search results and the same message is shown to the user.



# CONCLUSIONS

## what I learnt

- The project, which was the final test of the Javascript Advanced course, was instrumental in familiarising myself with this technology and brushing up on HTML/CSS.
- I learnt how to dynamically edit web pages through customised functions.
- I learnt how to handle async/await and how to use the Axios library to make calls to external APIs.

**I am very satisfied with the result, but this is only the beginning of the journey to learn the front-end side.**



# CONCLUSIONS

## what I learnt



### TECHNICAL SKILLS ACQUIRED

- use and consolidation of HTML
- use and consolidation of CSS
- use and consolidation of Bootstrap
- use and consolidation of Visual Studio Code
- use of and consolidation of Javascript
- use of axios library for API calls
- deployment of a web page on GitHub

**Source code on GitHub**

**Live Demo Counter**

**Start2Impact Profile**

**LinkedIn profile**