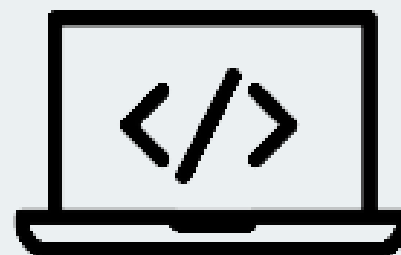




THE CITY LIFE RANK

Progetto JavaScript Advanced



a cura di Matteo Battilani

INDICE

02

1. DESCRIZIONE DEL PROGETTO	3
2. TECNOLOGIE UTILIZZATE	4
3. LA SEARCH BAR	5
4. CLASSIFICHE	6
4.1 Versione Desktop	6
4.2 Versione Mobile	7
5. RISULTATI DI RICERCA	8
6. VISUALIZZAZIONE DATI	9
7. GESTIONE ERRORI	10
8. CONCLUSIONI	11

DESCRIZIONE DEL PROGETTO

03

Il progetto "The City Life Rank" nasce come prova finale del corso Start2Impact relativo a "Javascript Advanced". La richiesta è stata quella di creare una pagina composta da una textbox per permettere all'utente di inserire il nome di una specifica città.

Una volta che l'utente clicca su un apposito button, l'applicazione deve contattare le API del servizio esterno Teleport per ottenere la informazioni da visualizzare in pagina.

Una volta ricevuti i dati dal servizio esterno l'applicazione dovrà visualizzare in pagina le informazioni riguardanti le categorie, la descrizione e il city score della città.

I REQUISITI MINIMI RICHIESTI

- una pagina HTML con search bar e pulsante
- una pagina CSS con lo stile
- una pagina JS con le chiamate alle API di Teleport
- mostrare categorie, descrizione e city score per ogni città ricercata

I REQUISITI CHE HO AGGIUNTO

- classifica delle 10 migliori e peggiori città in base al city score (gestite con 2 funzioni specifiche)
- visualizzazione della bandiera nazionale per le città della classifica (API di REST Countries)
- listener che permette di accedere ai dati di una città della classifica cliccando sul suo nome
- gestione delle possibili omonimie durante la ricerca creando un elenco di risultati di ricerca

TECNOLOGIE UTILIZZATE

HTML



CSS

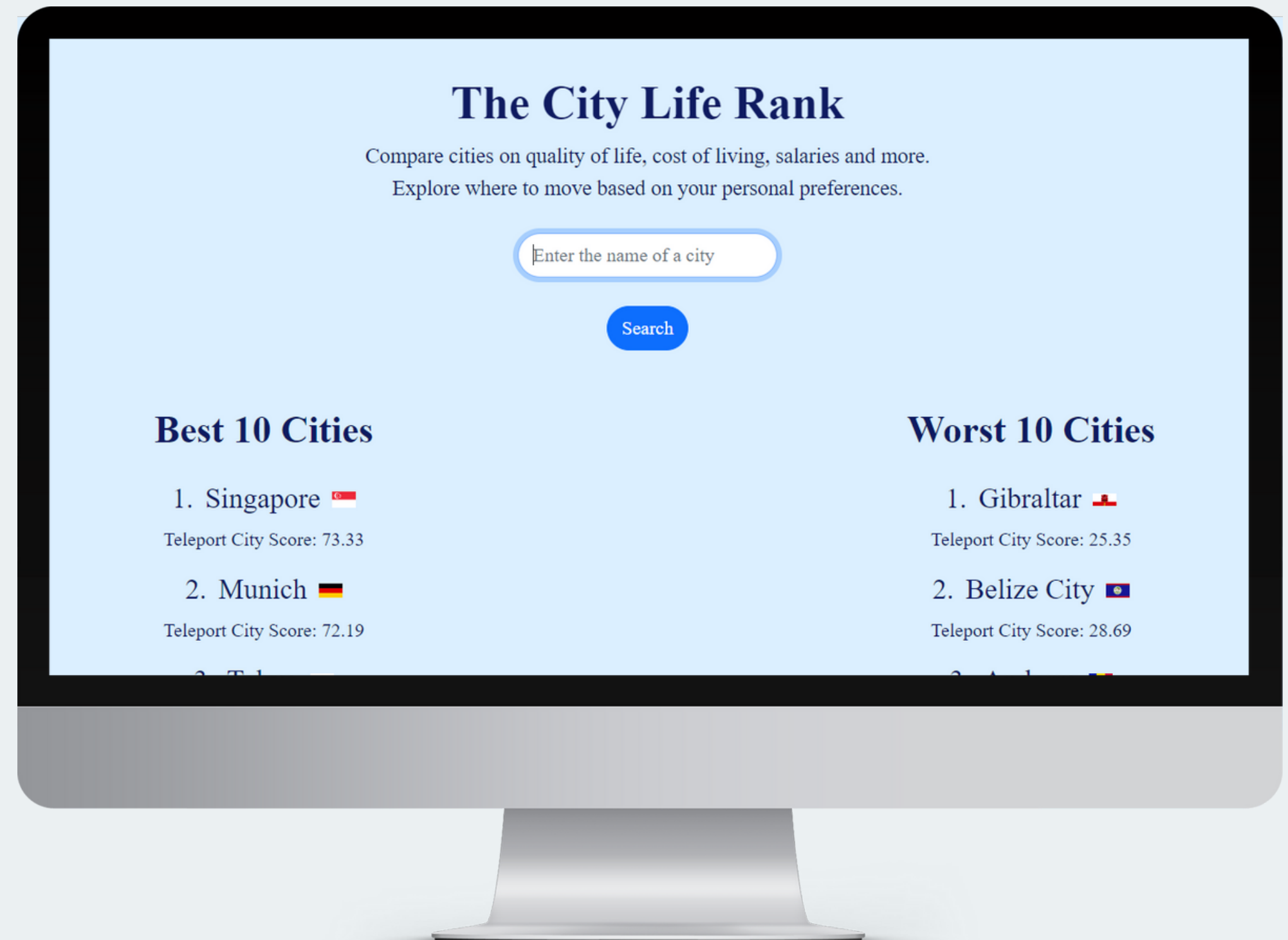


JS



LA SEARCH BAR

- La search bar presenta un placeholder che invita l'utente a digitare il nome di una città.
- Quando la barra è in focus (cliccata dall'utente), i suoi bordi si evidenziano con un colore più scuro.
- Cliccando sul pulsante "Search", all'utente viene mostrata una lista dei risultati di ricerca tra cui scegliere.
- Un titolo ed una descrizione del servizio accompagnano la search bar.



CLASSIFICHE

Versione Desktop

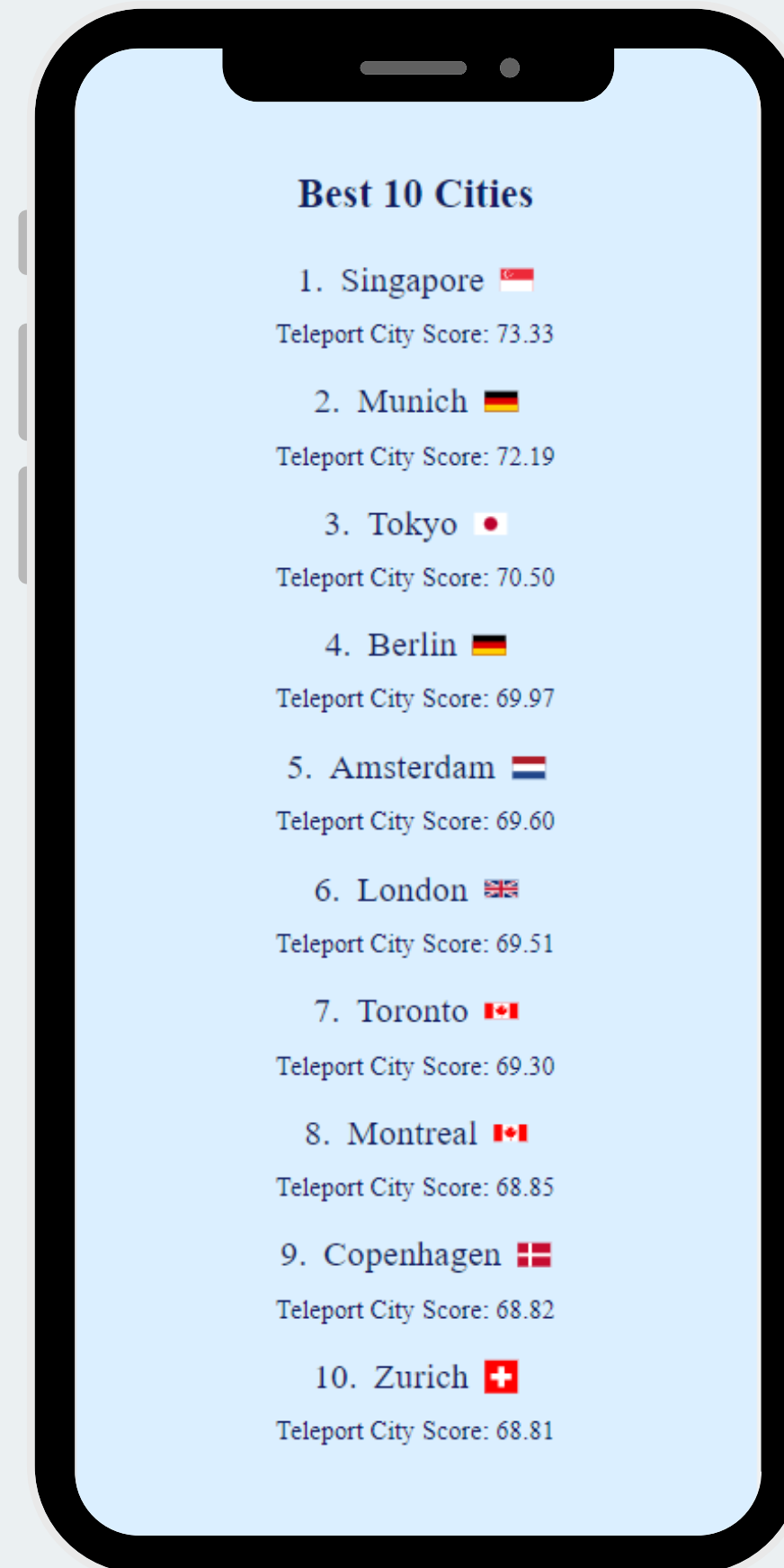
- Le due classifica mostrano rispettivamente le migliori e peggiori città in base al punteggio assegnato da Teleport.
- Ogni città è accompagnata anche dalla bandiera della nazione di appartenenza (ciò è possibile chiamando le API di REST Countries).
- Cliccando sul nome di una città si visualizzano la descrizione, le categorie ed il punteggio di tale città.



CLASSIFICHE

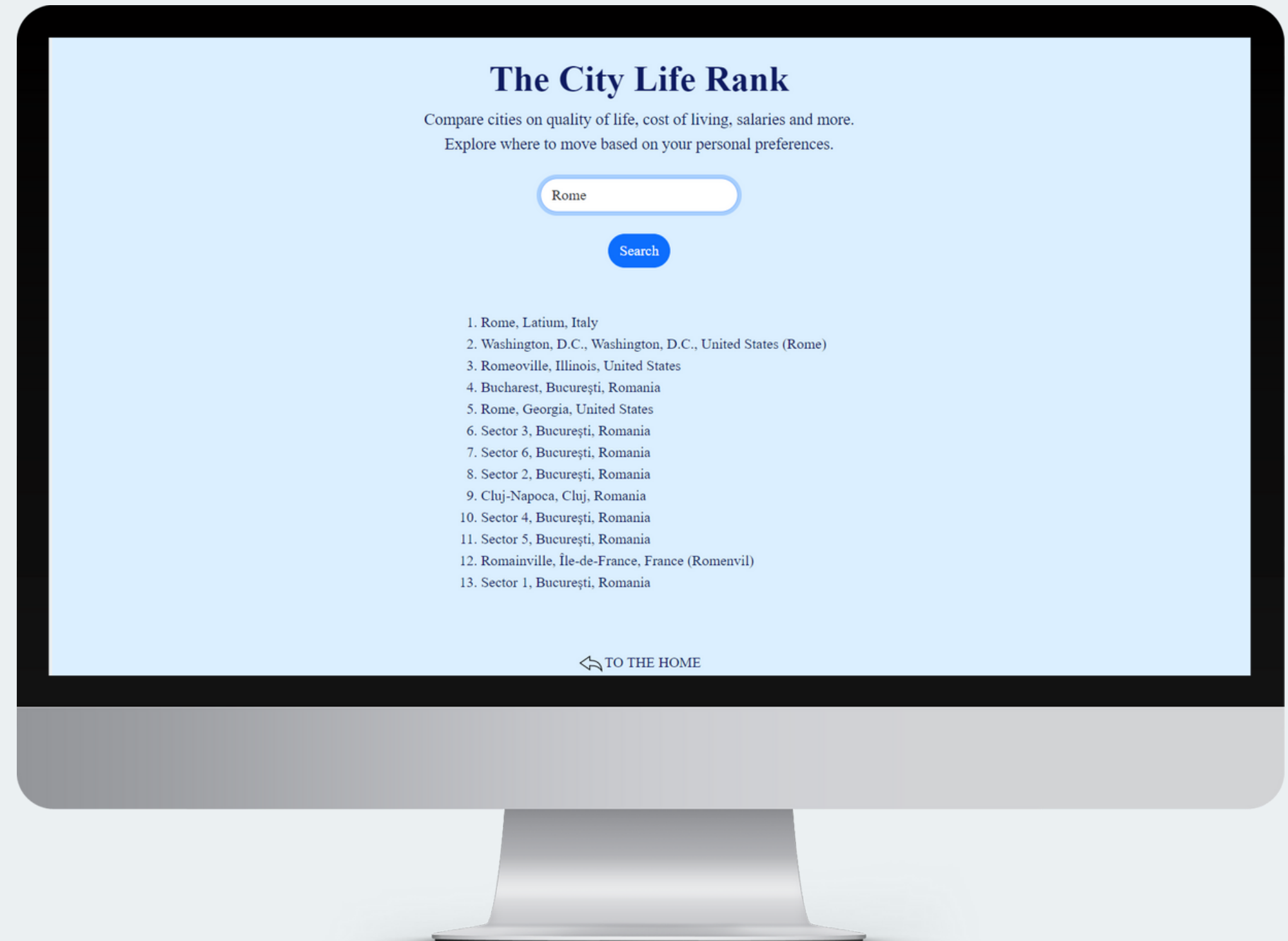
Versione Mobile

- La versione mobile, presenta un layout diverso, più adatto alla sua visualizzazione verticale.
- Le classifiche non vengono più visualizzate affiancate, bensì una soto l'altra.



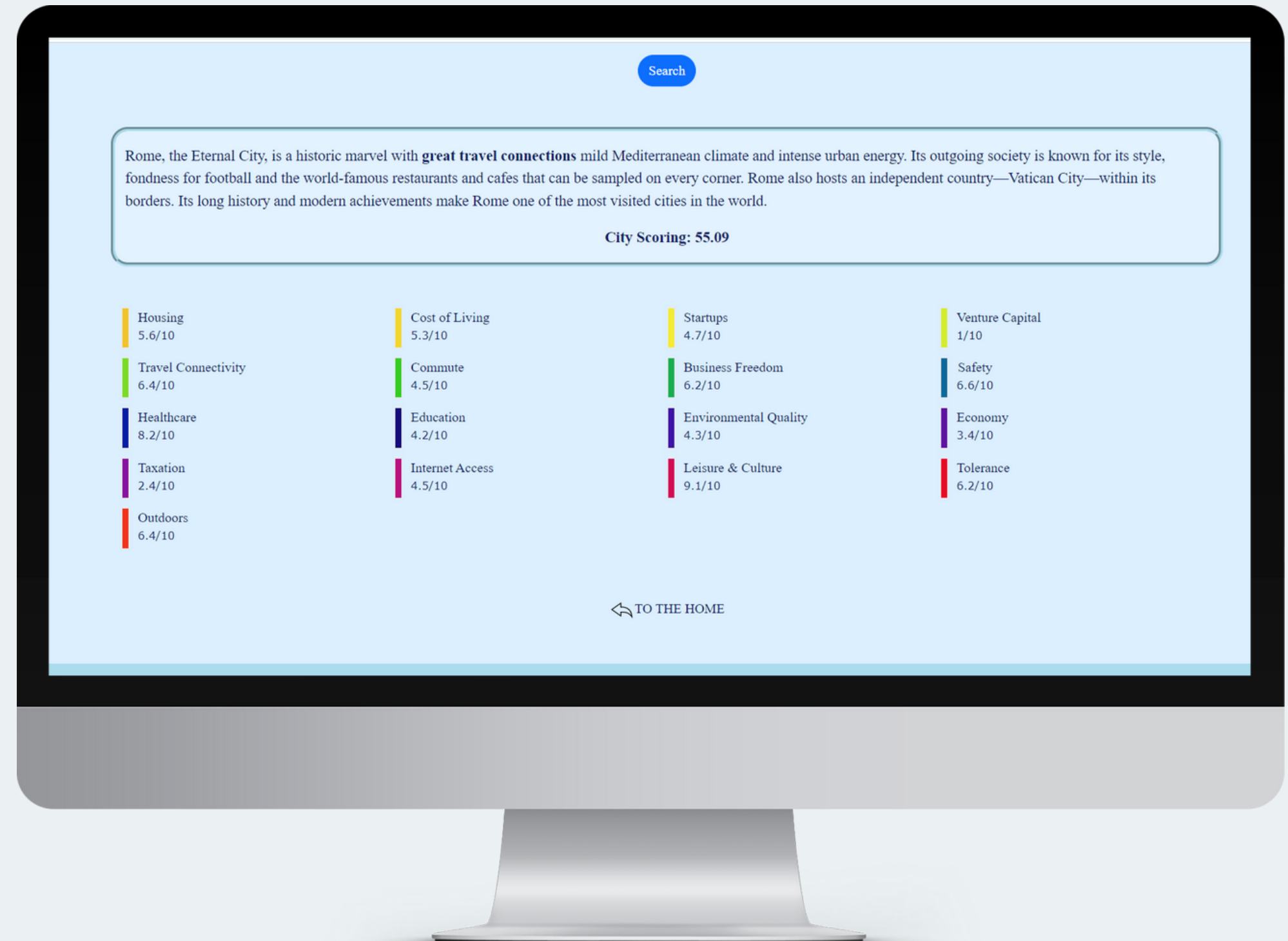
RISULTATI RICERCA

- Una volta premuto il pulsante di ricerca, l'utente può scegliere la città di cui vuole ottenere i dettagli dall'elenco dei risultati. Ciò gestisce eventuali errori di digitazione ed omonimie.
- Ai piedi della pagina, il pulsante "TO THE HOME" permette all'utente di visualizzare nuovamente la barra di ricerca con le classifiche delle migliori e peggiori città.



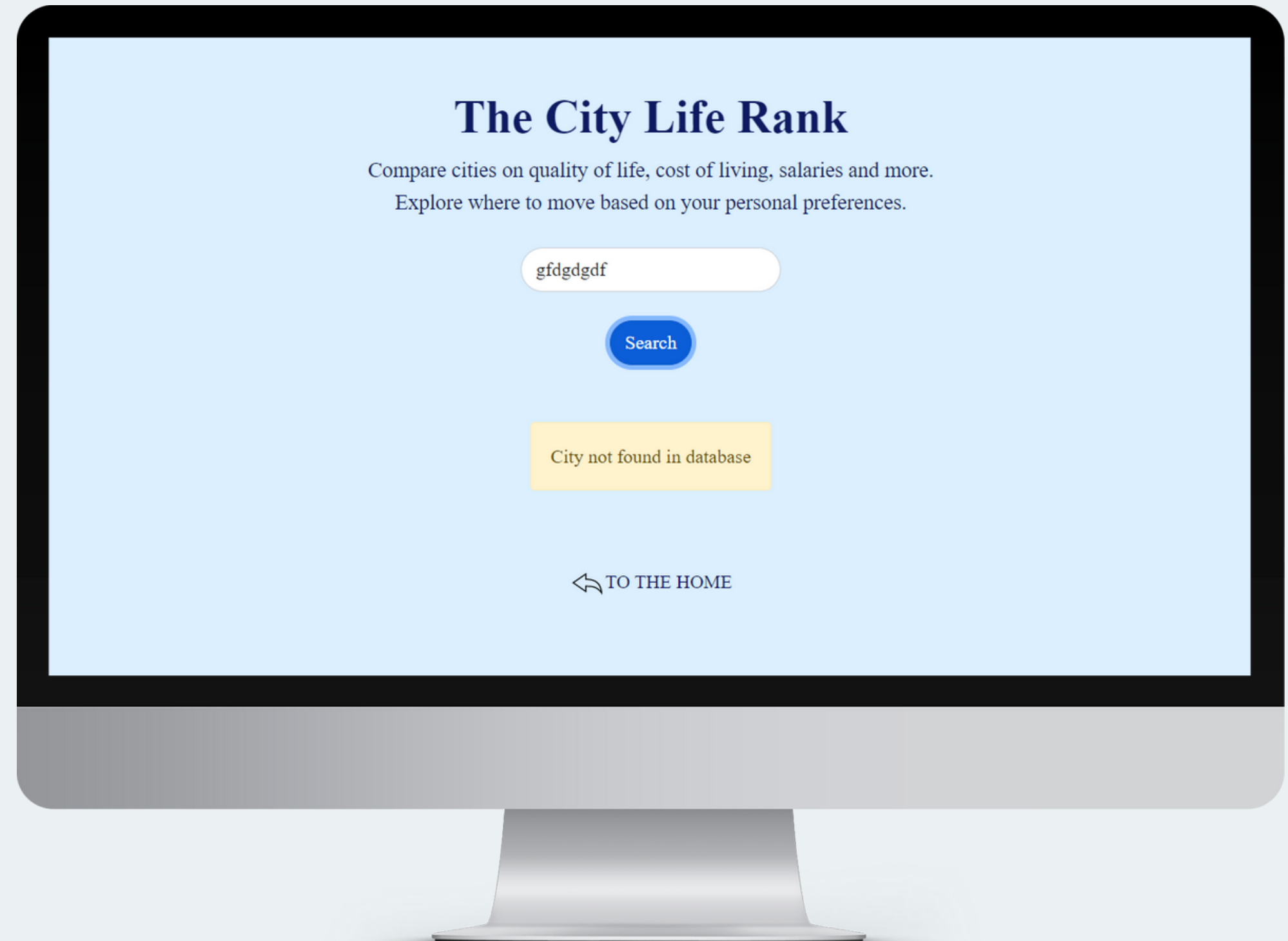
VISUALIZZAZIONE DATI

- Un riquadro colorato delimita la descrizione della città.
- Le categorie, accompagnate dai relativi punteggi, sono disposte a griglia. Anche il colore delle categorie è ottenuto tramite le API di Teleport.
- Ai piedi della pagina, il pulsante "TO THE HOME" permette all'utente di visualizzare nuovamente la barra di ricerca con le classifiche delle migliori e peggiori città.



GESTIONE ERRORI

- Qualora l'utente inserisca un nome di città non valido o non presente nei database di Teleport, viene visualizzato un messaggio che avvisa l'utente.
- Anche nel caso in cui la città sia presente nei database di Teleport, ma non abbia descrizione, categorie e punteggio, questa viene esclusa dai risultati di ricerca e lo stesso messaggio viene mostrato all'utente.



CONCLUSIONI

cosa ho imparato

- Il progetto, che è stata la prova finale del corso Javascript Advanced, è stato fondamentale per prendere confidenza con questa tecnologia e ripassare HTML/CSS.
- Ho imparato a modificare dinamicamente le pagine web attraverso funzioni personalizzate.
- Ho imparato a gestire async/await e ad utilizzare la libreria Axios per effettuare chiamate ad API esterne.

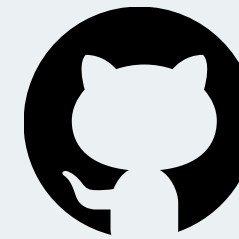
Sono molto soddisfatto del risultato ottenuto, ma questo è solo l'inizio del cammino per imparare il lato front-end.

CONCLUSIONI

cosa ho imparato

SKILLS TECNICHE ACQUISITE

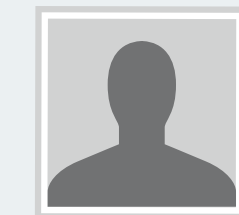
- utilizzo e consolidamento di HTML
- utilizzo e consolidamento di CSS
- utilizzo e consolidamento di Bootstrap
- utilizzo e consolidamento di Visual Studio Code
- utilizzo e consolidamento di Javascript
- utilizzo libreria axios per chiamate API
- deploy di una pagina web su GitHub



Codice Sorgente su GitHub



Live Demo Counter



Profilo Start2Impact



Profilo LinkedIn