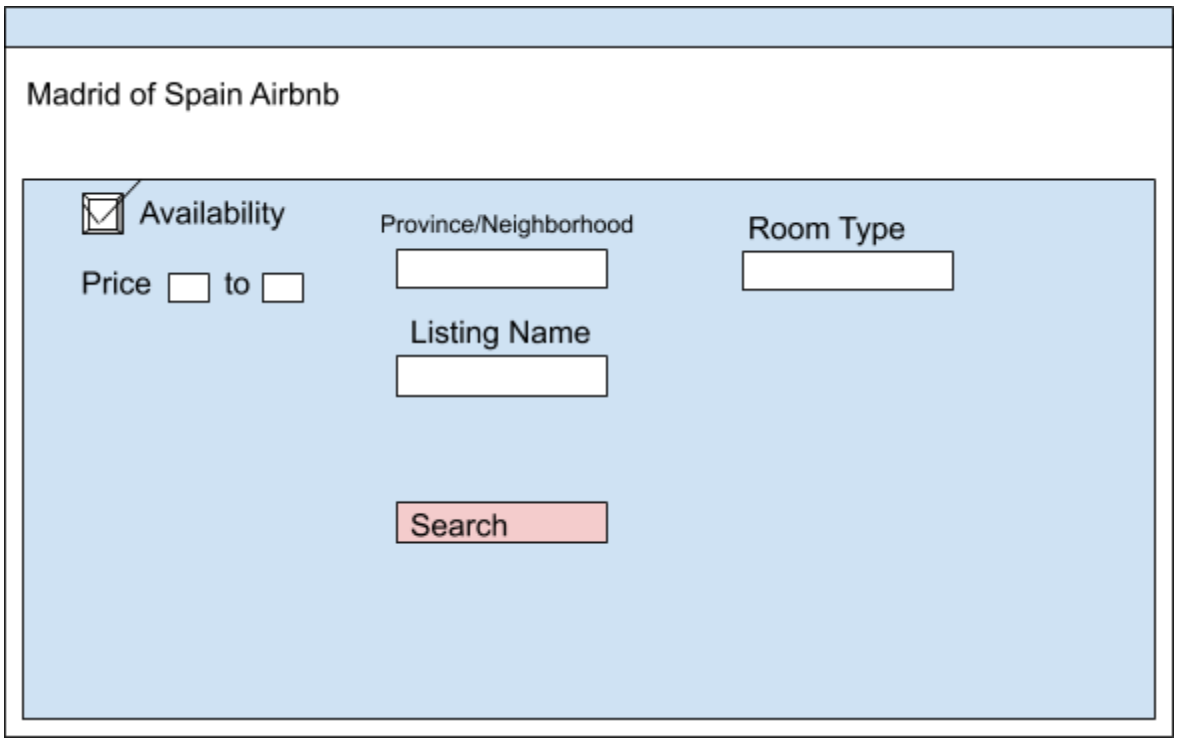
Airbnb - Lab 3 artifact

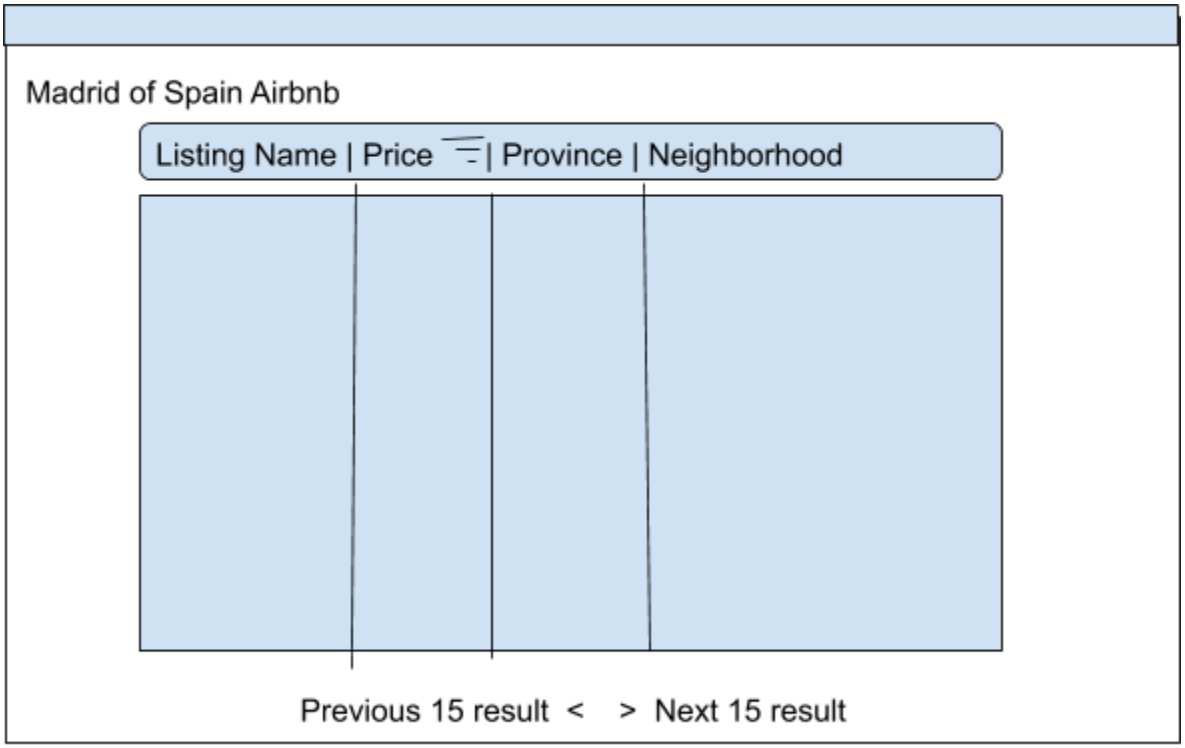
Features (user stories) to Implement in Next Sprint:

- **Pricing**
  (listings.csv)
    - **Feature 1**: as a user, I want to view listings, sorted by price.
    - **Feature 2**: as a user, I want to search for Airbnb listings in Madrid within a price range.
    - **Feature 3**: as a user, I want to search for an Airbnb listing by listing name.
    - **Feature 4**: as a user, I want to search for an Airbnb listing by room type.
- **Neighborhoods**
  (neighborhoods.csv)
    - **Feature 5**: as a user, I want to search for Airbnb listings only in a specific province in Madrid.
    - **Feature 6**: as a user, I want to search for Airbnb listings only in a specific neighborhood in Madrid.
- **Availability**
  (calender.csv)
    - **Feature 7**: as a user, I want to search for available Airbnb listings in Madrid.

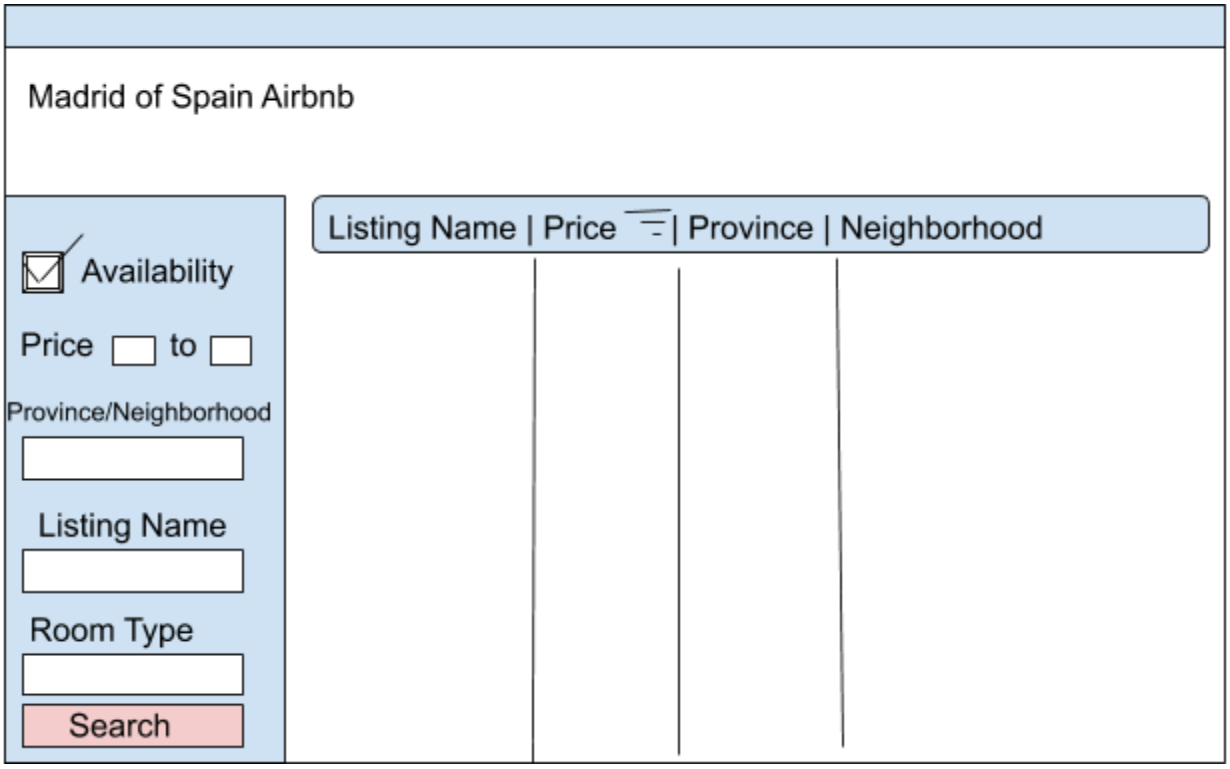
GUI

- Option 1: search page that redirects to another page that lists up to the first 10 listings that fit the criteria





- Option 2: Search bar and result page included on same page



Sort by price (high to low), price range, listing name, room type, province, neighborhood, available

Test Cases

- **Feature 1 test cases**: as a user, I want to view listings, sorted by price.
    - Test case 1: as a user, I select “price: high to low” to view listings’ price from high to low.
        - Correct output: website displays the first 15 listings, sorted by price in descending order.
    - Test case 2: as a user, I select “price: low to high” to view listings’ price from high to low.
        - Correct output: website displays the first 15 listings, sorted by price in ascending order.
- **Feature 2 test cases**: as a user, I want to search for Airbnb listings in Madrid within a price range.
    - Test case 1: as a user, I search for listings by inputting range $100 to $150 to view listings within that range.
        - Correct output: website displays the first 15 listings with price greater than 100 and less than 150.
    - Test case 2: as a user, I search for listings by inputting range $50 to $100 to view listings within that range.
        - Correct output: website displays the first 15 listings with price greater than 50 and less than 100.
    - Test case 3: as a user, I search for listings by inputting range $150 to $100 to view listings within that range.
        - Correct output: website displays an error that reads “The range provided is not correct.”
    - Test case 4: as a user, I search for listings by inputting range $150 into the first box to view listings above $150.
        - Correct output: website displays the first 15 listings with price greater than 150.
    - Test case 4: as a user, I search for listings by inputting range $150 into the second box to view listings less than $150.
        - Correct output: website displays the first 15 listings with price less than $150.

- **Feature 3**: as a user, I want to search for an Airbnb listing by listing name.
    - Test case 1: as a user, I enter a name that exists in the database into the search bar.
        - Correct output: website displays the matched airbnb information.
    - Test case 2: as a user, I enter a name that does not exist in the database into the search bar.
        - Correct output: website displays an error that reads “We do not find anything for that listing.”
- **Feature 4**: as a user, I want to search for an Airbnb listing by room type.
    - Test case 1: as a user, I search for listings that are for a private room.
        - Correct output: website displays the first 15 listings that are a private room.
    - Test case 1: as a user, I search for listings that are for an entire home or apartment.
        - Correct output: website displays the first 15 listings for an entire home or apartment.
- **Feature 5**: as a user, I want to search for Airbnb listings only in a specific province in Madrid.
    - Test case 1: as a user, I want to view the listings in Arganzuela.
        - Correct output: website displays the first 15 listings inside the province Arganzuela, which includes listings in neighborhoods: Acacias, Atocha, Chopera, Delicias, Imperial, Legazpi, Palos de Moguer.
    - Test case 2: as a user, I want to view the listings outside of Madrid, Spain.
        - Correct output: website displays “Location not found”.
- **Feature 6**: as a user, I want to search for Airbnb listings only in a specific neighborhood in Madrid
    - Test case 1: as a user, I want to view the listings in Acacias.
        - Correct output: website displays the first 15 listings inside the neighborhood Acacias, but not the other neighborhoods in the province of Arganzuela which includes neighborhoods: Atocha, Chopera, Delicias, Imperial, Legazpi, Palos de Moguer.

        - Test case 2: as a user, I want to view the listings outside of Madrid, Spain.
            - Correct output: website displays “Location not found”.

- **Feature 7**: as a user, I want to search for available Airbnb listings in Madrid.
    - Test case 1: as a user, I click the “availability” button to view all Airbnb Madrid listings available.
        - Correct output: website displays the first 15 available listings and not unavailable listings.
    - Test case 2: as a user, I want to view available listings in Acacias
        - Correct output: website displays only available listings inside the neighborhood Acacias, but not the other neighborhoods in the province of Arganzuela which includes neighborhoods: Atocha, Chopera, Delicias, Imperial, Legazpi, Palos de Moguer.
    - Test case 3: as a user, I want to view available listings in Arganzuela
        - Correct output: website displays the first 15 listings inside the province Arganzuela, which includes listings in neighborhoods: Acacias, Atocha, Chopera, Delicias, Imperial, Legazpi, Palos de Moguer.

TODO LIST

Done list of last sprint:

- Not Applicable

ToDo task list for the next sprint:

- Create a frontend form component for searching by price range.
    - Acceptance criteria: the form receives data inputted by the user and sends information to the backend when the user submits the form.
- Create a frontend form component for searching a listing by room type.
    - Acceptance criteria: the form receives data inputted by the user and sends information to the backend when the user submits the form.
- Create a frontend form component for searching a listing by province/neighborhood.
    - Acceptance criteria: the form receives data inputted by the user and sends information to the backend when the user submits the form.
- Create a frontend form component for searching a listing by availability.
    - Acceptance criteria: the form receives data inputted by the user and sends information to the backend when the user submits the form.
- Create a frontend form component for searching a listing by listing name.
    - Acceptance criteria: the form receives data inputted by the user and sends information to the backend when the user submits the form.

- Backend program to search listing.csv for sorted listings (display 15 results).
    - Acceptance criteria: the function should return 15 listings that are either the 15 lowest prices of a specific search criteria or the 15 highest prices of a specific search criteria.
- Backend program to search listing.csv for listings within a price range (display 15 results).
    - Acceptance criteria: the function should return 15 listings within price range.
- Backend program to search listing.csv for listings with a specific listing name.

    - Acceptance criteria: the function should return the specific name or an error.
- Backend program to search listing.csv for listings within a specific room type (display 15 results).
    - Acceptance criteria: the function should return 15 listings that are of that specific room type (received from client).
- Backend program to search neighborhoods.csv for listings within a province (display 15 results).
    - Acceptance criteria: the function should return 15 listings within the province (received from client).
- Backend program to search neighborhoods.csv for listings within a neighborhood (display 15 results).
    - Acceptance criteria: the function should return 15 listings within the neighborhood (received from client).
- Backend program to search calender.csv for available listings (display 15 results).
    - Acceptance criteria: the function should return 15 available listings.