Managing Data on the World-Wide Web

Assignment 3 – Frontend Programming

The assignment is 15% Takef.

Submission date: 20.12.2021 23:55 PM

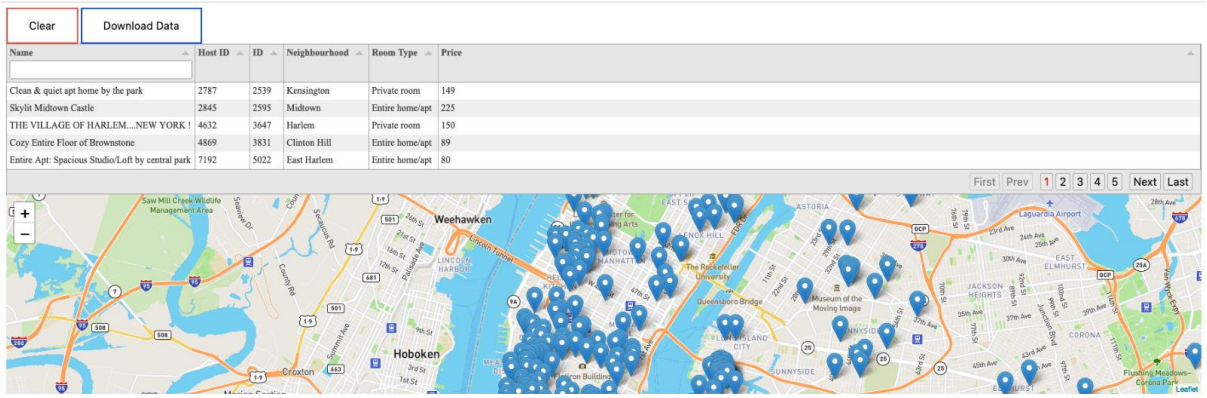
Introduction

This assignment focuses on frontend programming. You will implement a page that visualizes data using frontend tools: HTML APIs, JavaScript libraries, etc.

Specifically, in this assignment, you will implement a page that visualizes an uploaded dataset of *Airbnb locations in New York*. The Airbnb locations are presented as entries in a table and markers on a map.

Selecting an entry in the table will:

1. Show additional information about the selected entry.
2. Zoom into that specific location on the map.



Dataset

A dataset of Airbnb's places in New York is provided as part of this assignment. The dataset's format is a **CSV** file.

The provided dataset is a subset of the *New York City Airbnb Open Data* dataset in Kaggle. The structure of the data is described on Kaggle.

Displaying the Data

Uploading/Downloading the Data Set

The first step is to upload the CSV file that contains the Airbnb locations. The initial state of the page should present an uploading UI of your choice (without showing any maps or tables).

The page changes after uploading the data, and the uploading option is hidden. In that state, there should be an option to:

1. Download the uploaded dataset as CSV.
2. Clear out the uploaded dataset and return the page to its initial state.

Data as Interactive Table

Upon uploading the dataset, the entries are displayed as an interactive table, allowing exploration and a better view of the data. The table should have the following capabilities:

- Sorting – can sort the data by **any** column.
- Filtering – filter Airbnbs by name.
- Pagination – display ten rows per page, and enable pagination.

You are required to use the **Tabulator** JS library, allowing easy access to all the above.

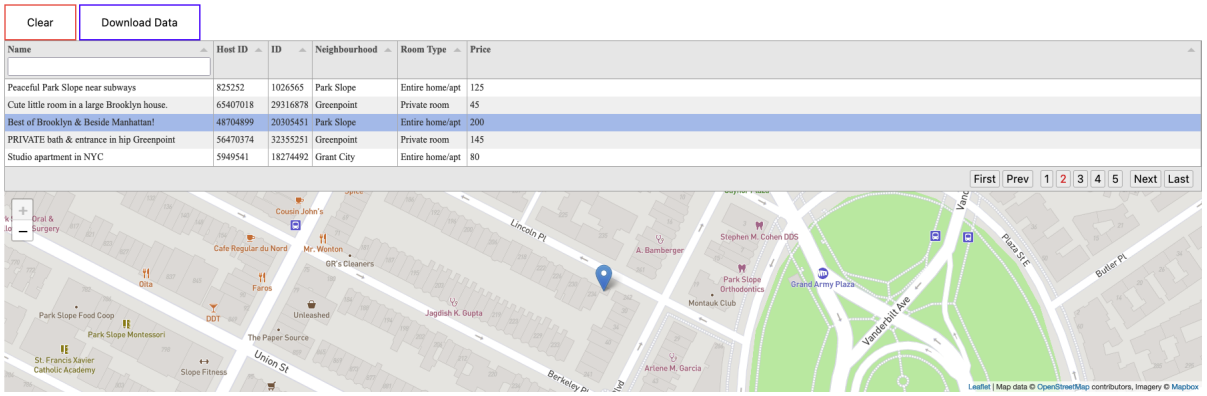
The table should present the Airbnb items from the provided dataset and should contain only the following subset of the attributes: *name*, *host ID*, *ID*, *neighborhood*, *room type*, and *price*.

Maps

Now, we want to use visual geographic information to view the Airbnbs. For this part, you will use **Leaflet**, a JS library for mapping and geo-information.

You are required to implement a map with markers for **all** the Airbnbs locations in the dataset.

Selecting an entry in the table will zoom into that specific location on the map as shown in the image below.
Make sure to automatically deselect the selected table entry when the user moves the map.



To display the map you will need to use *mapbox* which has a free maps API. You need to create a *mapbox* account to get an access token which you will use in your Javascript code.

Additional Information "Card"

When selecting an entry in the table, additional information will appear showing all of the attributes, including those that are not in the table. For example, the additional information can be shown as a "card" below the map or next to the entry. Make sure to discard this additional information when deselecting the table entry.

Finishing Touches

Finally, you are required to display your results on a **single page**, like a reactive application. The flow should be simple and you should avoid unneeded complexity.

You are welcome to be creative. Outstanding creative work will get a small bonus.

Getting Started

Installing Libraries

You can install all the required libraries using the **npm package manager**. You can download **npm** here.

After you've installed npm you can install the libraries by running the following commands from the terminal:

- Tabulator – *"npm install tabulator*"
- Leaflet – *"npm install leaflet"*

Using the Libraries

These libraries are simple to use and have a lot of online examples.

Tabulator – You can find various examples and documentation here.

Leaflet – You can find here everything you need. Again, there are many features you don't need to know.

Tips and Guidelines

- You can work on each part separately and connect them all at the end.
- Use the given links! Almost everything you need to implement becomes easy if you understand how to use the libraries.
- For parsing CSV files you can use the Papa Parse library.
- The single page can be straightforward. Do not waste your time implementing very complicated control-flow logic.
- Make sure that your application works in Chrome and Firefox.

Submission

You should submit a zip file named hw3.zip containing the following files:

• hw3.html – the HTML file of your application.

• hw3_design.css – the CSS file of your application.

• Any other files needed for your application.

Good Luck!