Project 3

Zillow National Home Prices

Project Scope

An analysis of pre-pandemic national housing price and days on market trends

Source: Zillow House Price Data from Kaggle*

1. **Days on Market by State**
   - 1 CSV file, 53 records
   - Collects how long listings remain on Zillow by state
   - Gathers monthly from Jan 2010 - Feb 2020

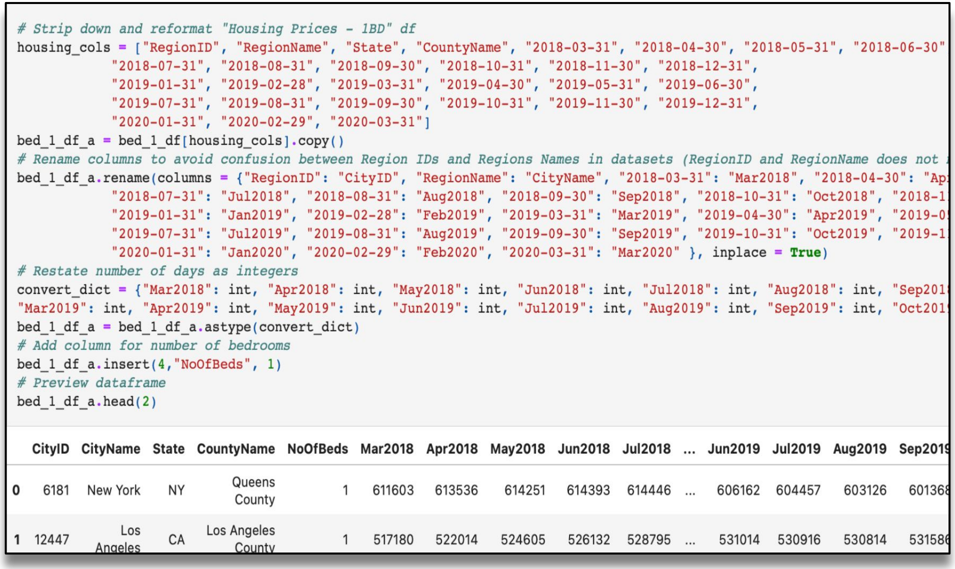
2. **Housing Price by Number of Bedrooms**
   - 3 CSV files: 1 bedroom, 14927 records; 2 bedroom 23396 records, 3 bedroom 25735 records
   - Collects Zillow Home Value Index, which measures typical home value by city
   - Gathers monthly from Jan 1996 - Mar 2020

*https://www.kaggle.com/datasets/paultimothymooney/zillow-house-price-data

Cleaning CSV Files

Sample code

1. Start with 4 CSV files
   - Days on market
   - 1, 2, and 3 bedroom housing price
2. Strip down and reformat columns to only gather data from the most recent 2 years
3. Add number of bedrooms column to each housing price dataframe
4. Append housing price datasets together
5. End with 2 CSV files
   - Days on market
   - Housing prices



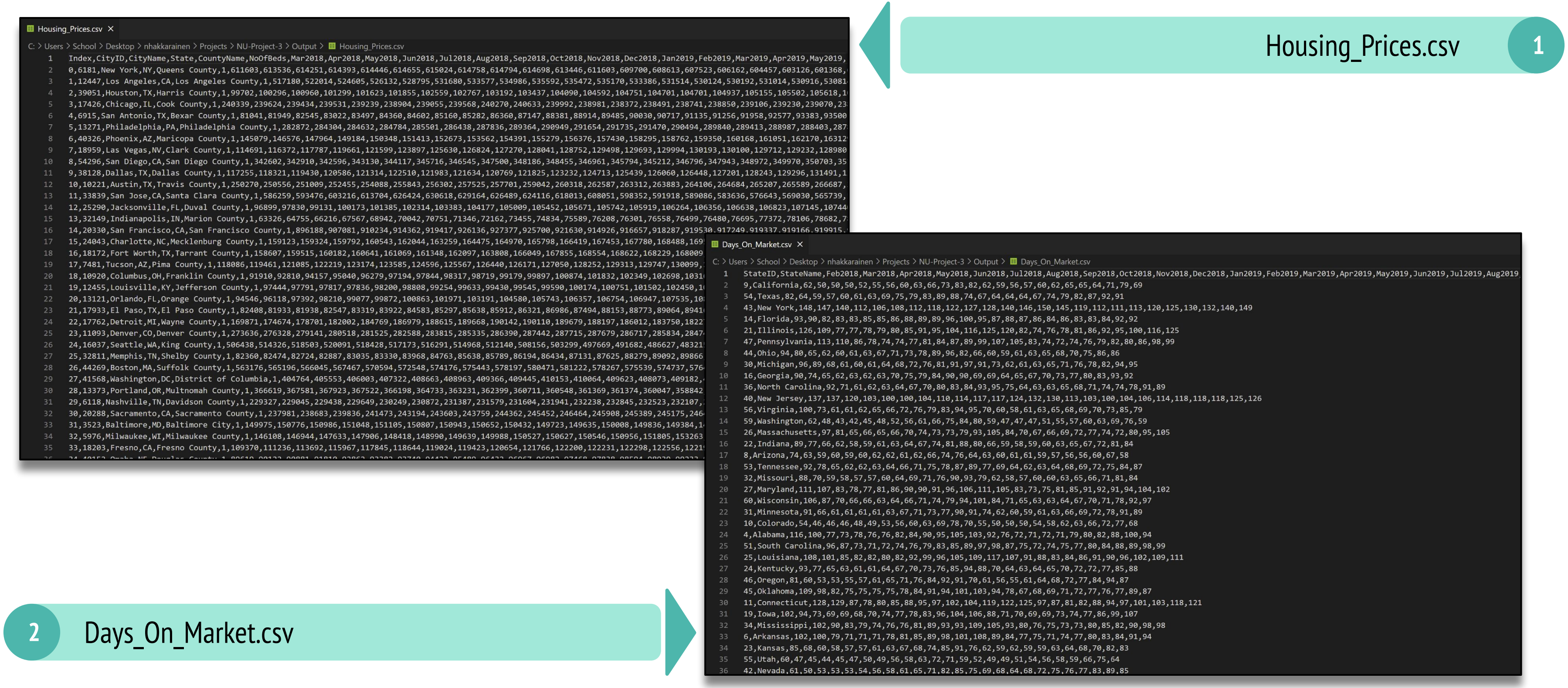
```
# Strip down and reformat "Housing Prices - 1BD" df
housing_cols = ["RegionID", "RegionName", "State", "CountyName", "2018-03-31", "2018-04-30", "2018-05-31", "2018-06-30"
            "2018-07-31", "2018-08-31", "2018-09-30", "2018-10-31", "2018-11-30", "2018-12-31",
            "2019-01-31", "2019-02-28", "2019-03-31", "2019-04-30", "2019-05-31", "2019-06-30",
            "2019-07-31", "2019-08-31", "2019-09-30", "2019-10-31", "2019-11-30", "2019-12-31",
            "2020-01-31", "2020-02-29", "2020-03-31"]
bed_1_df_a = bed_1_df[housing_cols].copy()
# Rename columns to avoid confusion between Region IDs and Regions Names in datasets (RegionID and RegionName does not
bed_1_df_a.rename(columns = {"RegionID": "CityID", "RegionName": "CityName", "2018-03-31": "Mar2018", "2018-04-30": "Ap
            "2018-07-31": "Jul2018", "2018-08-31": "Aug2018", "2018-09-30": "Sep2018", "2018-10-31": "Oct2018", "2018-1
            "2019-01-31": "Jan2019", "2019-02-28": "Feb2019", "2019-03-31": "Mar2019", "2019-04-30": "Apr2019", "2019-0
            "2019-07-31": "Jul2019", "2019-08-31": "Aug2019", "2019-09-30": "Sep2019", "2019-10-31": "Oct2019", "2019-1
            "2020-01-31": "Jan2020", "2020-02-29": "Feb2020", "2020-03-31": "Mar2020" }, inplace = True)
# Restate number of days as integers
convert_dict = {"Mar2018": int, "Apr2018": int, "May2018": int, "Jun2018": int, "Jul2018": int, "Aug2018": int, "Sep201
"Mar2019": int, "Apr2019": int, "May2019": int, "Jun2019": int, "Jul2019": int, "Aug2019": int, "Sep2019": int, "Oct201
bed_1_df_a = bed_1_df_a.astype(convert_dict)
# Add column for number of bedrooms
bed_1_df_a.insert(4,"NoOfBeds", 1)
# Preview dataframe
bed_1_df_a.head(2)
```

	CityID	CityName	State	CountyName	NoOfBeds	Mar2018	Apr2018	May2018	Jun2018	Jul2018	...	Jun2019	Jul2019	Aug2019	Sep2019
0	6181	New York	NY	Queens County	1	611603	613536	614251	614393	614446	...	606162	604457	603126	601368
1	12447	Los Angeles	CA	Los Angeles County	1	517180	522014	524605	526132	528795	...	531014	530916	530814	531586

Cleaning CSV Files

Screenshot examples

1 Housing_Prices.csv



Housing_Prices.csv

C: > Users > School > Desktop > nhakkarainen > Projects > NU-Project-3 > Output > Housing_Prices.csv

```
Index,CityID,CityName,State,CountyName,NoOfBeds,Mar2018,Apr2018,May2018,Jun2018,Jul2018,Aug2018,Sep2018,Oct2018,Nov2018,Dec2018,Jan2019,Feb2019,Mar2019,Apr2019,May2019,
0,6181,New York,NY,Queens County,1,611603,613536,614251,614393,614446,614655,615024,614758,614794,614698,613446,611603,609700,608613,607523,606162,604457,603126,601368,
1,12447,Los Angeles,CA,Los Angeles County,1,517180,522014,524605,526132,528795,531680,533577,534986,535592,535472,535170,533386,531514,530124,530192,531014,530916,53081
2,39051,Houston,TX,Harris County,1,99702,100296,100960,101299,101623,101855,102559,102767,103192,103437,104090,104592,104751,104701,104701,104937,105155,105502,105618,1
3,17426,Chicago,IL,Cook County,1,240339,239624,239434,239531,239239,238904,239055,239568,240270,240633,239992,238981,238372,238491,238741,238850,239106,239230,239070,23
4,6915,San Antonio,TX,Bexar County,1,81041,81949,82545,83022,83497,84360,84602,85160,85282,86360,87147,88381,88914,89485,90030,90717,91135,91256,91958,92577,93383,93500
5,13271,Philadelphia,PA,Philadelphia County,1,282872,284304,284632,284784,285501,286438,287836,289364,290949,291654,291735,291470,290494,289840,289413,288987,288403,287
6,40326,Phoenix,AZ,Maricopa County,1,145079,146576,147964,149184,150348,151413,152673,153562,154391,155279,156376,157430,158295,158762,159350,160168,161051,162170,16312
7,18959,Las Vegas,NV,Clark County,1,114691,116372,117787,119661,121599,123897,125630,126824,127270,128041,128752,129498,129693,129994,130193,130100,129712,129232,128980
8,54296,San Diego,CA,San Diego County,1,342602,342910,342596,343130,344117,345716,346545,347500,348186,348455,346961,345794,345212,346796,347943,348972,349970,350703,35
9,38128,Dallas,TX,Dallas County,1,117255,118321,119430,120586,121314,122510,121983,121634,120769,121825,123232,124713,125439,126060,126448,127201,128243,129296,131491,1
10,10221,Austin,TX,Travis County,1,250270,250556,251009,252455,254088,255843,256302,257525,257701,259042,260318,262587,263312,263883,264106,264684,265207,265589,266687,
11,33839,San Jose,CA,Santa Clara County,1,586259,593476,603216,613704,626424,630618,629164,626489,624116,618013,608051,598352,591918,589086,583636,576643,569030,565739,
12,25290,Jacksonville,FL,Duval County,1,96899,97830,99131,100173,101385,102314,103383,104177,105009,105452,105671,105742,105919,106264,106356,106638,106823,107145,10744
13,32149,Indianapolis,IN,Marion County,1,63326,64755,66216,67567,68942,70042,70751,71346,72162,73455,74834,75589,76208,76301,76558,76499,76480,76695,77372,78106,78682,7
14,20330,San Francisco,CA,San Francisco County,1,896188,907081,910234,914362,919417,926136,927377,925700,921630,914926,916657,918287,919530,917249,919337,919166,919915,
15,24043,Charlotte,NC,Mecklenburg County,1,159123,159324,159792,160543,162044,163259,164475,164970,165798,166419,167453,167780,168488,169
16,18172,Fort Worth,TX,Tarrant County,1,158607,159515,160182,160641,161069,161348,162097,163808,166049,167855,168554,168622,168229,168009
17,7481,Tucson,AZ,Pima County,1,118086,119461,121085,122219,123174,123585,124596,125567,126440,126171,127050,128252,129313,129747,130099,
18,10920,Columbus,OH,Franklin County,1,91910,92810,94157,95040,96279,97194,97844,98317,98719,99179,99897,100874,101832,102349,102698,1031
19,12455,Louisville,KY,Jefferson County,1,97444,97791,97817,97836,98200,98808,99254,99633,99430,99545,99590,100174,100751,101502,102450,1
20,13121,Orlando,FL,Orange County,1,94546,96118,97392,98210,99077,99872,100863,101971,103191,104580,105743,106357,106754,106947,107535,10
21,17933,El Paso,TX,El Paso County,1,82408,81933,81938,82547,83319,83922,84583,85297,85638,85912,86321,86986,87494,88153,88773,89064,8941
22,17762,Detroit,MI,Wayne County,1,169871,174674,178701,182002,184769,186979,188615,189668,190142,190110,189679,188197,186012,183750,1822
23,11093,Denver,CO,Denver County,1,273636,276328,279141,280518,281525,282588,283815,285335,286390,287442,287715,287679,286717,285834,2847
24,16037,Seattle,WA,King County,1,506438,514326,518503,520091,518428,517173,516291,514968,512140,508156,503299,497669,491682,486627,48321
25,32811,Memphis,TN,Shelby County,1,82360,82474,82724,82887,83035,83330,83968,84763,85638,85789,86194,86434,87131,87625,88279,89092,89866
26,44269,Boston,MA,Suffolk County,1,563176,565196,566045,567467,570594,572548,574176,575443,578197,580471,581222,578267,575539,574737,576
27,41568,Washington,DC,District of Columbia,1,404764,405553,406003,407322,408663,408963,409366,409445,410153,410064,409623,408073,409182,
28,13373,Portland,OR,Multnomah County,1,366619,367581,367923,367522,366198,364733,363231,362399,360711,360548,361369,361374,360047,358842
29,6118,Nashville,TN,Davidson County,1,229327,229045,229438,229649,230249,230872,231387,231579,231604,231941,232238,232845,232523,232107,
30,20288,Sacramento,CA,Sacramento County,1,237981,238683,239836,241473,243194,243603,243759,244362,245452,246464,245908,245389,245175,246
31,3523,Baltimore,MD,Baltimore City,1,149975,150776,150986,151048,151105,150807,150943,150652,150432,149723,149635,150008,149836,149384,1
32,5976,Milwaukee,WI,Milwaukee County,1,146108,146944,147633,147906,148418,148990,149639,149988,150527,150627,150546,150956,151805,153263
33,18203,Fresno,CA,Fresno County,1,109370,111236,113692,115967,117845,118644,119024,119423,120654,121766,122200,122231,122298,122556,1221
34,40152,Omaha,NE,Douglas County,1,89610,90122,90881,91810,92862,93382,93740,94422,95480,96432,96967,96982,97468,97828,98504,98030,99222
```

Days_On_Market.csv

C: > Users > School > Desktop > nhakkarainen > Projects > NU-Project-3 > Output > Days_On_Market.csv

```
StateID,StateName,Feb2018,Mar2018,Apr2018,May2018,Jun2018,Jul2018,Aug2018,Sep2018,Oct2018,Nov2018,Dec2018,Jan2019,Feb2019,Mar2019,Apr2019,May2019,Jun2019,Jul2019,Aug2019
9,California,62,50,50,50,52,55,56,60,63,66,73,83,82,62,59,56,57,60,62,65,65,64,71,79,69
54,Texas,82,64,59,57,60,61,63,69,75,79,83,89,88,74,67,64,64,64,67,74,79,82,87,92,91
43,New York,148,147,140,112,106,108,112,118,122,127,128,140,146,150,145,119,112,111,113,120,125,130,132,140,149
14,Florida,93,90,82,83,83,85,85,86,88,89,89,96,100,95,87,88,87,86,84,86,83,83,84,92,92
21,Illinois,126,109,77,77,78,79,80,85,91,95,104,116,125,120,82,74,76,78,81,86,92,95,100,116,125
47,Pennsylvania,113,110,86,78,74,74,77,81,84,87,89,99,107,105,83,74,72,74,76,79,82,80,86,98,99
44,Ohio,94,80,65,62,60,61,63,67,71,73,78,89,96,82,66,60,59,61,63,65,68,70,75,86,86
30,Michigan,96,89,68,61,60,61,64,68,72,76,81,91,97,91,73,62,61,63,65,71,76,78,82,94,95
16,Georgia,90,74,65,62,63,62,63,70,75,79,84,90,90,69,69,64,65,67,70,73,77,80,83,93,92
36,North Carolina,92,71,61,62,63,64,67,70,80,83,84,93,95,75,64,63,63,65,68,71,74,74,78,91,89
40,New Jersey,137,137,120,103,100,100,104,110,114,117,117,124,132,130,113,103,100,104,106,114,118,118,118,125,126
56,Virginia,100,73,61,61,62,65,66,72,76,79,83,94,95,70,60,58,61,63,65,68,69,70,73,85,79
59,Washington,62,48,43,42,45,48,52,56,61,66,75,84,80,59,47,47,47,51,55,57,60,63,69,76,59
26,Massachusetts,97,81,65,66,65,66,70,74,73,73,79,93,105,84,70,67,66,69,72,77,74,72,80,95,105
22,Indiana,89,77,66,62,58,59,61,63,64,67,74,81,88,80,66,59,58,59,60,63,65,67,72,81,84
8,Arizona,74,63,59,60,59,60,62,62,61,62,66,74,76,64,63,60,61,61,59,57,56,56,60,67,58
53,Tennessee,92,78,65,62,62,63,64,66,71,75,78,87,89,77,69,64,62,63,64,68,69,72,75,84,87
32,Missouri,88,70,59,58,57,57,60,64,69,71,76,90,93,79,62,58,57,60,60,63,65,66,71,81,84
27,Maryland,111,107,83,78,77,81,86,90,90,91,96,106,111,105,83,73,75,81,85,91,92,91,94,104,102
60,Wisconsin,106,87,70,66,66,63,64,66,71,74,79,94,101,84,71,65,63,63,64,67,70,71,78,92,97
31,Minnesota,91,66,61,61,61,61,63,67,71,73,77,90,91,74,62,60,59,61,63,66,69,72,78,91,89
10,Colorado,54,46,46,46,48,49,53,56,60,63,69,78,70,55,50,50,50,54,58,62,63,66,72,77,68
4,Alabama,116,100,77,73,78,76,76,82,84,90,95,105,103,92,76,72,71,72,71,79,80,82,88,100,94
51,South Carolina,96,87,73,71,72,74,76,79,83,85,89,97,98,87,75,72,74,75,77,80,84,88,89,98,99
25,Louisiana,108,101,85,82,82,80,82,92,99,96,105,109,117,107,91,88,83,84,86,91,90,96,102,109,111
24,Kentucky,93,77,65,63,61,61,64,67,70,73,76,85,94,88,70,64,63,64,65,70,72,72,77,85,88
46,Oregon,81,60,53,53,55,57,61,65,71,76,84,92,91,70,61,56,55,61,64,68,72,77,84,94,87
45,Oklahoma,109,98,82,75,75,75,75,78,84,91,94,101,103,94,78,67,68,69,71,72,77,76,77,89,87
11,Connecticut,128,129,87,78,80,85,88,95,97,102,104,119,122,125,97,87,81,82,88,94,97,101,103,118,121
19,Iowa,102,94,73,69,69,68,70,74,77,78,83,96,104,106,88,71,70,69,69,73,74,77,86,99,107
34,Mississippi,102,90,83,79,74,76,76,81,89,93,93,109,105,93,80,76,75,73,73,80,85,82,90,98,98
6,Arkansas,102,100,79,71,71,71,78,81,85,89,98,101,108,89,84,77,75,71,74,77,80,83,84,91,94
23,Kansas,85,68,60,58,57,57,61,63,67,68,74,85,91,76,62,59,62,59,59,63,64,68,70,82,83
55,Utah,60,47,45,44,45,47,50,49,56,58,63,72,71,59,52,49,49,51,54,56,58,59,66,75,64
42,Nevada,61,50,53,53,53,54,56,58,61,65,71,82,85,75,69,68,64,68,72,75,76,77,83,89,85
```

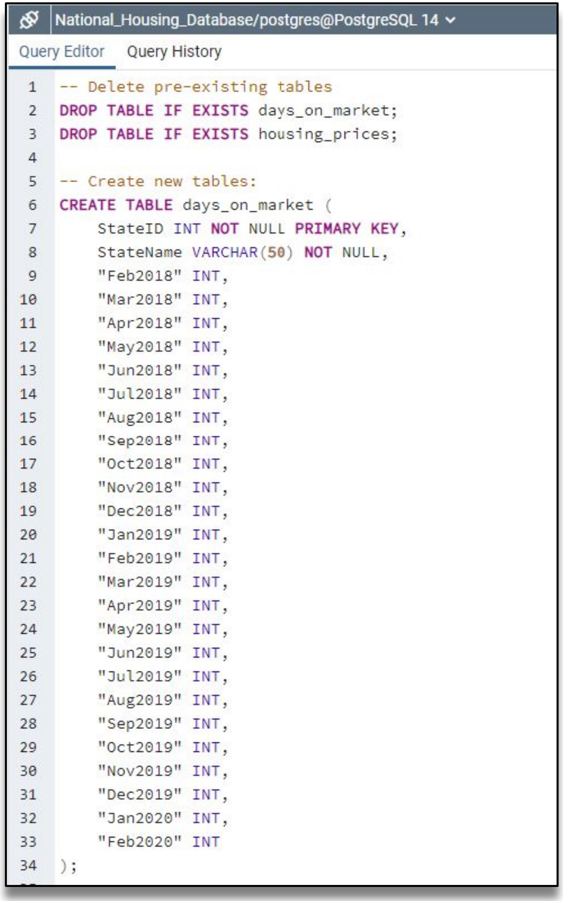
2 Days_On_Market.csv

PostgreSQL Database

Sample code

1. Initiate database and drop existing tables with desired table names
2. Create two tables for days_on_market CSV file and housing_prices CSV file
3. Import relevant datasets with headers
4. Perform SELECT ALL query to ensure datasets imported correctly

Troubleshooting: *needed to rename all date columns, removing hyphen from column names, in order for Flask query to work.*



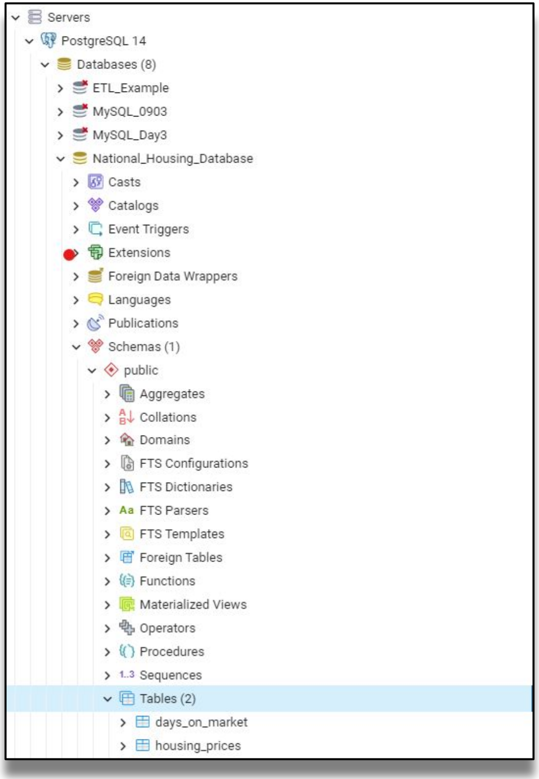
```sql
-- Delete pre-existing tables
DROP TABLE IF EXISTS days_on_market;
DROP TABLE IF EXISTS housing_prices;

-- Create new tables:
CREATE TABLE days_on_market (
    StateID INT NOT NULL PRIMARY KEY,
    StateName VARCHAR(50) NOT NULL,
    "Feb2018" INT,
    "Mar2018" INT,
    "Apr2018" INT,
    "May2018" INT,
    "Jun2018" INT,
    "Jul2018" INT,
    "Aug2018" INT,
    "Sep2018" INT,
    "Oct2018" INT,
    "Nov2018" INT,
    "Dec2018" INT,
    "Jan2019" INT,
    "Feb2019" INT,
    "Mar2019" INT,
    "Apr2019" INT,
    "May2019" INT,
    "Jun2019" INT,
    "Jul2019" INT,
    "Aug2019" INT,
    "Sep2019" INT,
    "Oct2019" INT,
    "Nov2019" INT,
    "Dec2019" INT,
    "Jan2020" INT,
    "Feb2020" INT
);
```

PostgreSQL Database

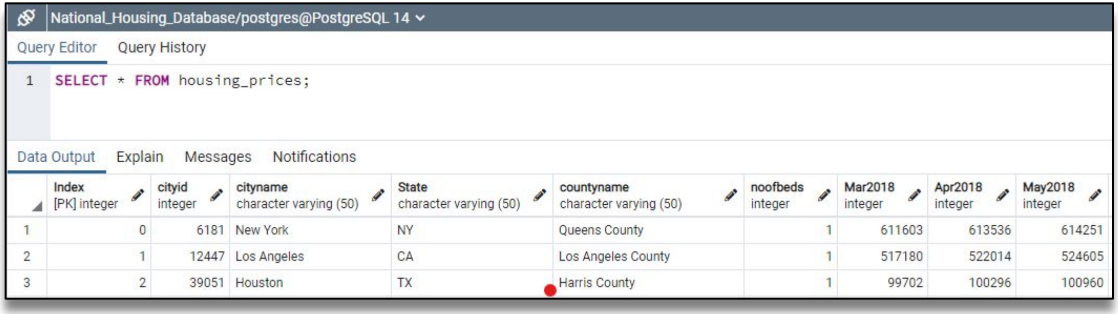
Screenshot example

1 Create “National Housing Database”



- Servers
  - PostgreSQL 14
    - Databases (8)
      - ETL_Example
      - MySQL_0903
      - MySQL_Day3
      - National_Housing_Database
        - Casts
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas (1)
          - public
            - Aggregates
            - Collations
            - Domains
            - FTS Configurations
            - FTS Dictionaries
            - FTS Parsers
            - FTS Templates
            - Foreign Tables
            - Functions
            - Materialized Views
            - Operators
            - Procedures
            - Sequences
            - Tables (2)
              - days_on_market
              - housing_prices

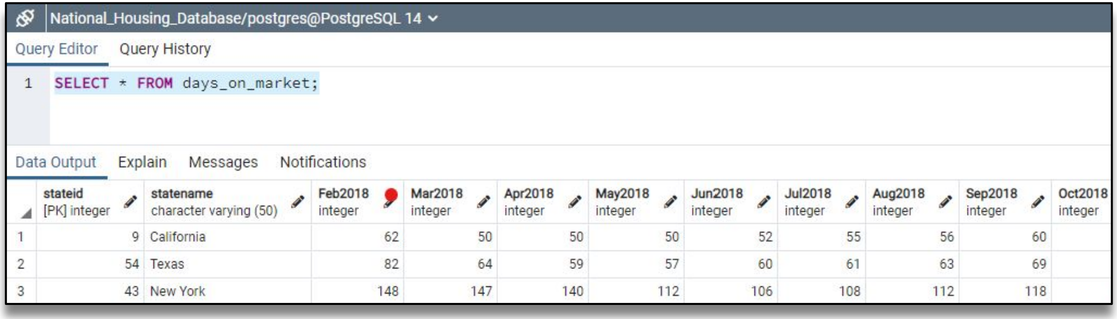
2 Create “Housing Prices” table and import CSV file with headers



```sql
SELECT * FROM housing_prices;
```

	Index [PK] integer	cityid integer	cityname character varying (50)	State character varying (50)	countyname character varying (50)	noofbeds integer	Mar2018 integer	Apr2018 integer	May2018 integer
1	0	6181	New York	NY	Queens County	1	611603	613536	614251
2	1	12447	Los Angeles	CA	Los Angeles County	1	517180	522014	524605
3	2	39051	Houston	TX	Harris County	1	99702	100296	100960

3 Create “Days on Market” table and import CSV file with headers



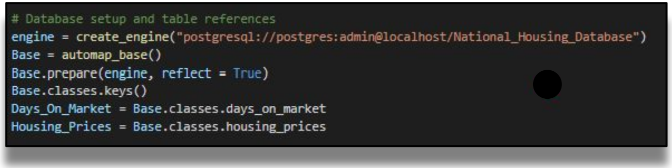
```sql
SELECT * FROM days_on_market;
```

	stateid [PK] integer	statename character varying (50)	Feb2018 integer	Mar2018 integer	Apr2018 integer	May2018 integer	Jun2018 integer	Jul2018 integer	Aug2018 integer	Sep2018 integer	Oct2018 integer
1	9	California	62	50	50	50	52	55	56	60	
2	54	Texas	82	64	59	57	60	61	63	69	
3	43	New York	148	147	140	112	106	108	112	118	

Flask-Powered API

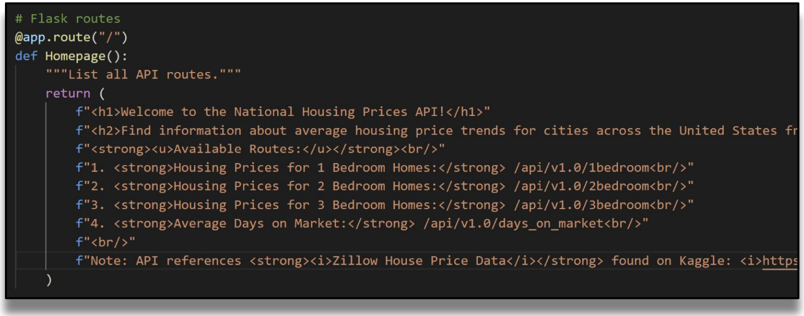
Sample code

1 Establish connection with PostgreSQL database and assign variables for each table



```
# Database setup and table references
engine = create_engine("postgresql://postgres:admin@localhost/National_Housing_Database")
Base = automap_base()
Base.prepare(engine, reflect = True)
Base.classes.keys()
Days_On_Market = Base.classes.days_on_market
Housing_Prices = Base.classes.housing_prices
```

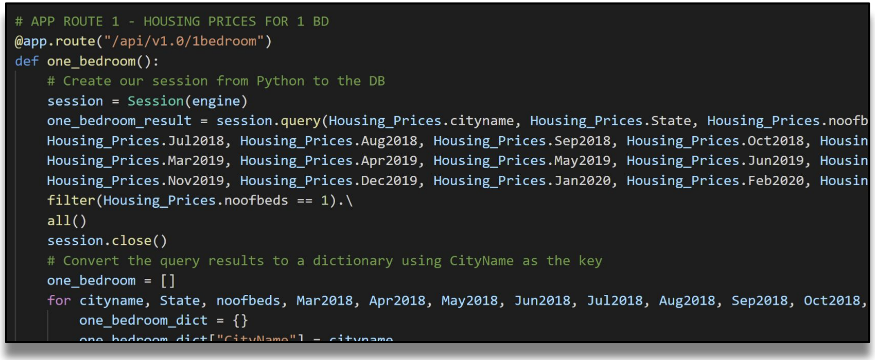
2 Setup Flask and list all available API routes on homepage



```
# Flask routes
@app.route("/")
def Homepage():
    """List all API routes."""
    return (
        f"<h1>Welcome to the National Housing Prices API!</h1>"
        f"<h2>Find information about average housing price trends for cities across the United States fr
        f"<strong><u>Available Routes:</u></strong><br/>"
        f"1. <strong>Housing Prices for 1 Bedroom Homes:</strong> /api/v1.0/1bedroom<br/>"
        f"2. <strong>Housing Prices for 2 Bedroom Homes:</strong> /api/v1.0/2bedroom<br/>"
        f"3. <strong>Housing Prices for 3 Bedroom Homes:</strong> /api/v1.0/3bedroom<br/>"
        f"4. <strong>Average Days on Market:</strong> /api/v1.0/days_on_market<br/>"
        f"<br/>"
        f"Note: API references <strong><i>Zillow House Price Data</i></strong> found on Kaggle: <i>https
    )
```

3 Create API routes:

- Create session from Python to database
- Query table for desired subset of data
- Convert query results into a dictionary
- Return JSON representation of data

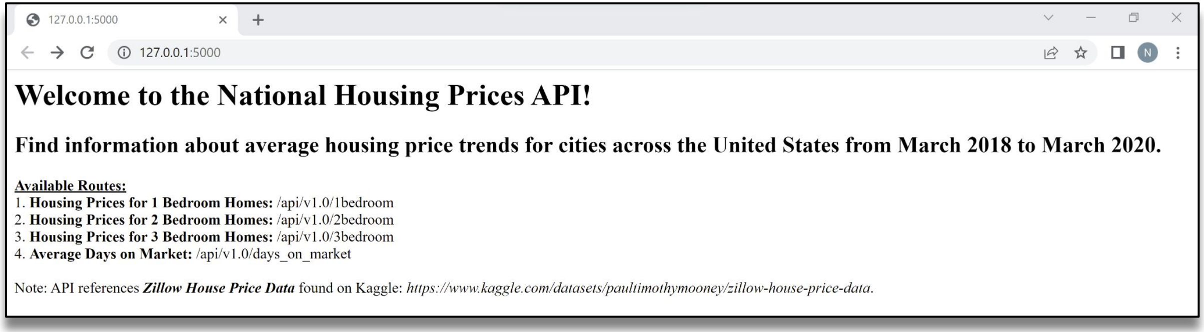


```
# APP ROUTE 1 - HOUSING PRICES FOR 1 BD
@app.route("/api/v1.0/1bedroom")
def one_bedroom():
    # Create our session from Python to the DB
    session = Session(engine)
    one_bedroom_result = session.query(Housing_Prices.cityname, Housing_Prices.State, Housing_Prices.noofb
    Housing_Prices.Jul2018, Housing_Prices.Aug2018, Housing_Prices.Sep2018, Housing_Prices.Oct2018, Housin
    Housing_Prices.Mar2019, Housing_Prices.Apr2019, Housing_Prices.May2019, Housing_Prices.Jun2019, Housin
    Housing_Prices.Nov2019, Housing_Prices.Dec2019, Housing_Prices.Jan2020, Housing_Prices.Feb2020, Housin
    filter(Housing_Prices.noofbeds == 1).\
    all()
    session.close()
    # Convert the query results to a dictionary using CityName as the key
    one_bedroom = []
    for cityname, State, noofbeds, Mar2018, Apr2018, May2018, Jun2018, Jul2018, Aug2018, Sep2018, Oct2018,
        one_bedroom_dict = {}
        one_bedroom_dict["CityName"] = cityname
```

Flask-Powered API

Screenshot example

1 Homepage listing four API Routes…



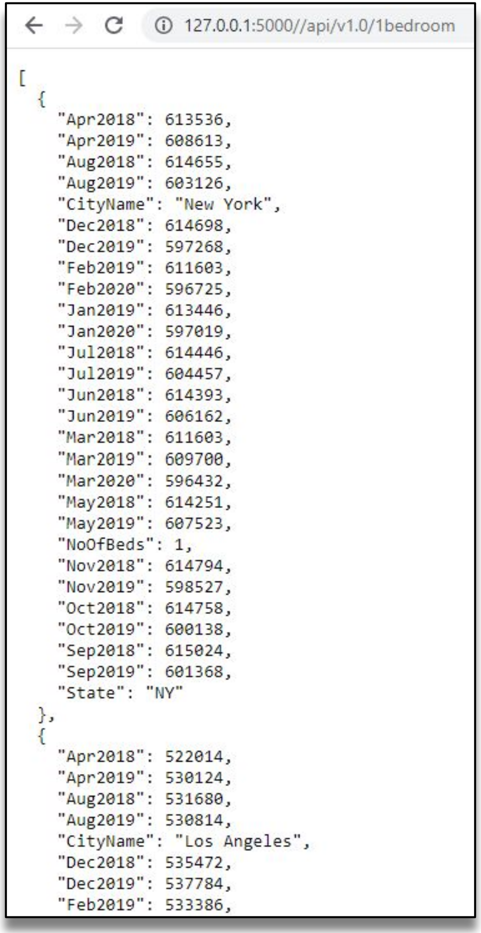
Welcome to the National Housing Prices API!

Find information about average housing price trends for cities across the United States from March 2018 to March 2020.

Available Routes:
1. **Housing Prices for 1 Bedroom Homes:** /api/v1.0/1bedroom
2. **Housing Prices for 2 Bedroom Homes:** /api/v1.0/2bedroom
3. **Housing Prices for 3 Bedroom Homes:** /api/v1.0/3bedroom
4. **Average Days on Market:** /api/v1.0/days_on_market

Note: API references ***Zillow House Price Data*** found on Kaggle: *https://www.kaggle.com/datasets/paultimothymooney/zillow-house-price-data.*

2 …JSONified dataset



```
[
  {
    "Apr2018": 613536,
    "Apr2019": 608613,
    "Aug2018": 614655,
    "Aug2019": 603126,
    "CityName": "New York",
    "Dec2018": 614698,
    "Dec2019": 597268,
    "Feb2019": 611603,
    "Feb2020": 596725,
    "Jan2019": 613446,
    "Jan2020": 597019,
    "Jul2018": 614446,
    "Jul2019": 604457,
    "Jun2018": 614393,
    "Jun2019": 606162,
    "Mar2018": 611603,
    "Mar2019": 609700,
    "Mar2020": 596432,
    "May2018": 614251,
    "May2019": 607523,
    "NoOfBeds": 1,
    "Nov2018": 614794,
    "Nov2019": 598527,
    "Oct2018": 614758,
    "Oct2019": 600138,
    "Sep2018": 615024,
    "Sep2019": 601368,
    "State": "NY"
  },
  {
    "Apr2018": 522014,
    "Apr2019": 530124,
    "Aug2018": 531680,
    "Aug2019": 530814,
    "CityName": "Los Angeles",
    "Dec2018": 535472,
    "Dec2019": 537784,
    "Feb2019": 533386,
```

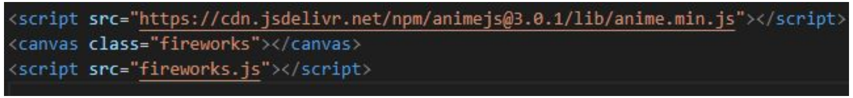
Visualizations

Sample code

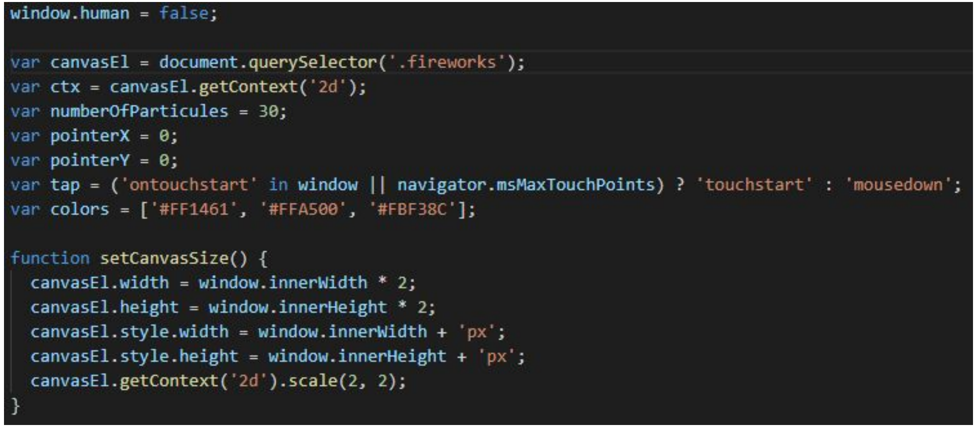
1. Create html framework and download the anime.js library

2. Set Canvas size and choose colors for the fireworks

3. Create circles for the firework effect and add an clicking option to the page



```html
<script src="https://cdn.jsdelivr.net/npm/animejs@3.0.1/lib/anime.min.js"></script>
<canvas class="fireworks"></canvas>
<script src="fireworks.js"></script>
```



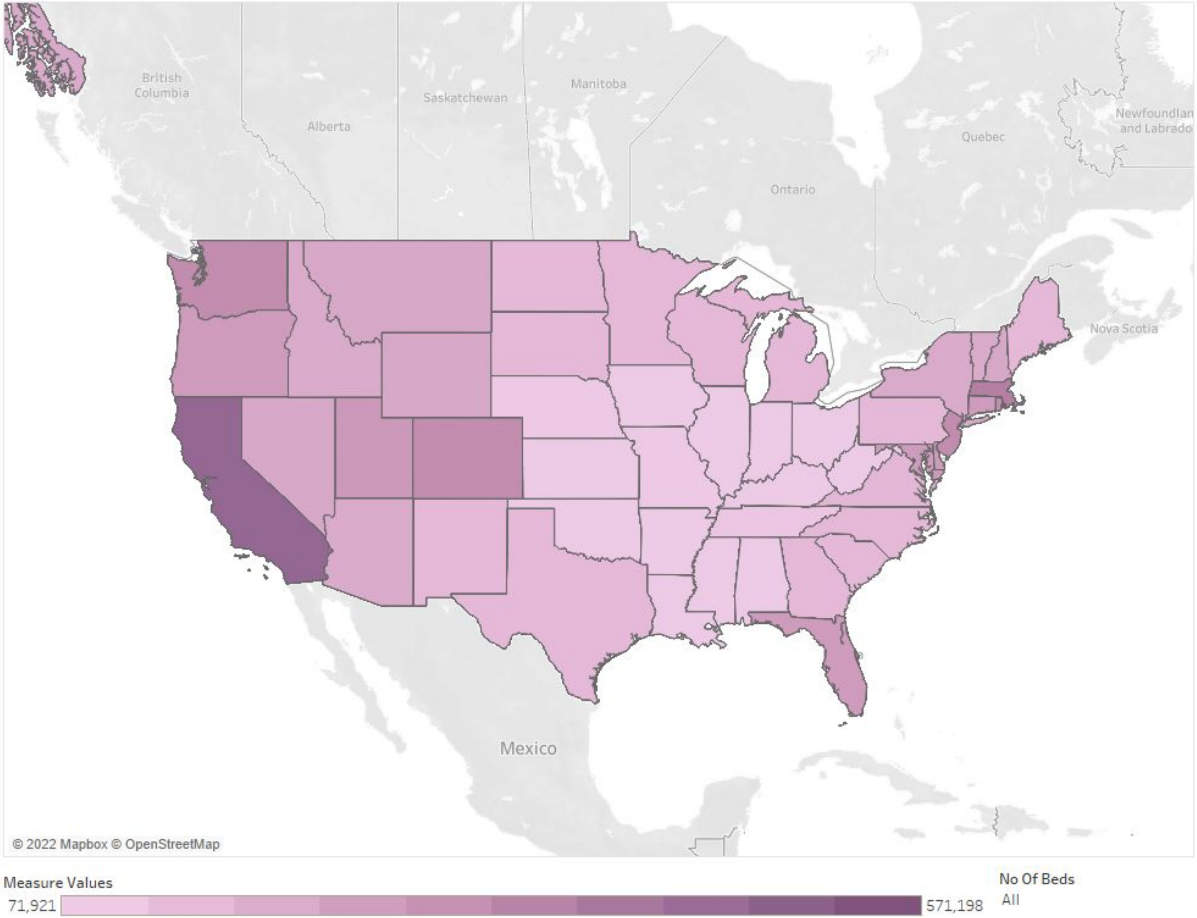
```javascript
window.human = false;

var canvasEl = document.querySelector('.fireworks');
var ctx = canvasEl.getContext('2d');
var numberOfParticules = 30;
var pointerX = 0;
var pointerY = 0;
var tap = ('ontouchstart' in window || navigator.msMaxTouchPoints) ? 'touchstart' : 'mousedown';
var colors = ['#FF1461', '#FFA500', '#FBF38C'];

function setCanvasSize() {
  canvasEl.width = window.innerWidth * 2;
  canvasEl.height = window.innerHeight * 2;
  canvasEl.style.width = window.innerWidth + 'px';
  canvasEl.style.height = window.innerHeight + 'px';
  canvasEl.getContext('2d').scale(2, 2);
}
```

Visualizations

Screenshot example



© 2022 Mapbox © OpenStreetMap