

# **DSCI 551 FINAL PROJECT REPORT**

## **TEAM DETAILS**

**SINGLE PERSON TEAM - GROUP 80**

**NAME:** Vedanvita Gudavalli

**USC ID:** 1741003325

**EMAIL:** vgudaval@usc.edu

## **PROJECT TOPIC**

### **EMULATING FIREBASE**

Firestore is a popular backend-as-a-service (BaaS) that provides a variety of services, including a realtime database. This project aims to build a prototype system to emulate some of the main functionalities of Firestore's Realtime Database using Flask, a micro web framework for Python, and MongoDB, a NoSQL database.

We implemented the following CRUD operations using Flask and MongoDB:

**GET:** Retrieve a listing or all listings with optional filtering and sorting.

**POST:** Create a new listing with a unique identifier.

**PUT:** Update a listing or create a new one if it doesn't exist.

**PATCH:** Update specific fields of a listing.

**DELETE:** Delete a listing by its identifier.

## **WORKFLOW**

- Appropriate Dataset is collected and Pre-processing is implemented.
- The Dataset is loaded into MongoDB using Python with PyMongo Library.
- The proper indexes are set to the MongoDB according to the requests.
- A server is created using Flask with appropriate endpoints and restful libraries that are connected with MongoDB.
- The windows command terminal is set up to support the CURL commands.
- As soon as we enter the command the results are fetched by the restful flask server by calling MongoDB and the results can be seen on the terminal.

## IMPLEMENTATION

- The dataset used for my project is “New York City Airbnb Open Data” which contains Airbnb listings and metrics in NYC for 2019.  
The dataset link :  
<https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data>
- Sufficient time spent on finalizing the dataset so that the required functions can be applied on the dataset suitably. This dataset includes information about listings, such as the name, host, location, room type, price, minimum nights, number of reviews, last review date, reviews per month, calculated host listings count, and availability.
- Data Pre-processing and Data Cleaning is implemented by removing null values and unnecessary columns.
- The dataset is a csv file, so I implemented a python script “csv\_to\_json.py”. csv\_to\_json.py converts the csv data into json format.
- All the required installations are done like Flask, PyMongo and MongoDB on my local system.
- The dataset is loaded into MongoDB using PyMongo library. Created a python script “import\_to\_mongo.py” which implements the data loading in MongoDB. This is done only once.
- The indices are set for the Data. These fields are chosen to create indexes because we can get efficient queries and more filtering options. So I implemented “indices.py” to achieve the same.
- Flask server is implemented as “app.py” and all the CRUD operations are implemented like GET, POST, PUT, PATCH and DELETE.
- The windows command terminal is set up to support CURL commands and environment variables and system variables are set up accordingly to support the commands.

## GET

```
PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings/3831.json"
{
  "_id": {
    "$oid": "64445bb3f3134793d80eecb3"
  },
  "id": 3831,
  "name": "Cozy Entire Floor of Brownstone",
  "host_id": 4869,
  "host_name": "LisaRoxanne",
  "neighbourhood_group": "Brooklyn",
  "neighbourhood": "Clinton Hill",
  "latitude": 40.68514,
  "longitude": -73.95976,
  "room_type": "Entire home/apt",
  "price": 89,
  "minimum_nights": 1,
  "number_of_reviews": 270,
  "last_review": "2019-07-05",
  "reviews_per_month": 4.64,
  "calculated_host_listings_count": 1,
  "availability_365": 194
}
```

GET command used to get the details of listing id : 3831

## airbnb.listings

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

Filter



{ "id": 3831 }

ADD DATA

EXPORT COLLECTION

```
_id: ObjectId('64445bb3f3134793d80eecb3')
id: 3831
name: "Cozy Entire Floor of Brownstone"
host_id: 4869
host_name: "LisaRoxanne"
neighbourhood_group: "Brooklyn"
neighbourhood: "Clinton Hill"
latitude: 40.68514
longitude: -73.95976
room_type: "Entire home/apt"
price: 89
minimum_nights: 1
number_of_reviews: 270
last_review: "2019-07-05"
reviews_per_month: 4.64
calculated_host_listings_count: 1
availability_365: 194
```

## GET

```
Windows PowerShell
PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings.json?orderBy=price&startAt=10000"
[
  {
    "_id": {
      "$oid": "64445bb3f3134793d80f106f"
    },
    "id": 7003697,
    "name": "Furnished room in Astoria apartment",
    "host_id": 20582832,
    "host_name": "Mathrine",
    "neighbourhood_group": "Queens",
    "neighbourhood": "Astoria",
    "latitude": 40.7681,
    "longitude": -73.91651,
    "room_type": "Private room",
    "price": 10000,
    "minimum_nights": 100,
    "number_of_reviews": 2,
    "last_review": "2016-02-13",
    "reviews_per_month": 0.04,
    "calculated_host_listings_count": 1,
    "availability_365": 0
  },
  {
    "_id": {
      "$oid": "64445bb3f3134793d80f31cc"
    },
    "id": 13894339,
    "name": "Luxury 1 bedroom apt. -stunning Manhattan views",
    "host_id": 5143901,
    "host_name": "Erin",
    "neighbourhood_group": "Brooklyn",
    "neighbourhood": "Greenpoint",
    "latitude": 40.7326,
    "longitude": -73.95739,
    "room_type": "Entire home/apt",
    "price": 10000,
    "minimum_nights": 5,
    "number_of_reviews": 5,
    "last_review": "2017-07-27",
    "reviews_per_month": 0.16,
    "calculated_host_listings_count": 1,
    "availability_365": 0
  }
],
```

GET command implementation to get the listing which are ordered by price and starts at 10000

```
Windows PowerShell
PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings.json?orderBy=price&endAt=0"
[
  {
    "_id": {
      "$oid": "64445bb3f3134793d80f4729"
    },
    "id": 18750597,
    "name": "Huge Brooklyn Brownstone Living, Close to it all.",
    "host_id": 8993804,
    "host_name": "Kimberly",
    "neighbourhood_group": "Brooklyn",
    "neighbourhood": "Bedford-Stuyvesant",
    "latitude": 40.69023,
    "longitude": -73.95428,
    "room_type": "Private room",
    "price": 0,
    "minimum_nights": 4,
    "number_of_reviews": 1,
    "last_review": "2018-01-06",
    "reviews_per_month": 0.05,
    "calculated_host_listings_count": 4,
    "availability_365": 28
  },
  {
    "_id": {
      "$oid": "64445bb3f3134793d80f5009"
    },
    "id": 20333471,
    "name": "\u2605Hostel Style Room | Ideal Traveling Buddies\u2605",
    "host_id": 131697576,
    "host_name": "Anisha",
    "neighbourhood_group": "Bronx",
    "neighbourhood": "East Morrisania",
    "latitude": 40.83296,
    "longitude": -73.88668,
    "room_type": "Private room",
    "price": 0,
    "minimum_nights": 2,
    "number_of_reviews": 55,
    "last_review": "2019-06-24",
    "reviews_per_month": 2.56,
    "calculated_host_listings_count": 4,
    "availability_365": 127
  },
]
```

GET command implementation to get the listings which are ordered by price and ends at 0 (price is zero because of coupons)

```
Windows PowerShell
PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings.json?orderBy=price&equalTo=0"
{
  "id": {
    "$oid": "64445bb3f3134793d88f4729"
  },
  "id": 18756597,
  "name": "Huge Brooklyn Brownstone Living, Close to it all.",
  "host_id": 8993884,
  "host_name": "Kimberly",
  "neighbourhood_group": "Brooklyn",
  "neighbourhood": "Bedford-Stuyvesant",
  "latitude": 40.69022,
  "longitude": -73.95428,
  "room_type": "Private room",
  "price": 0,
  "minimum_nights": 4,
  "number_of_reviews": 1,
  "last_review": "2018-01-06",
  "reviews_per_month": 0.05,
  "calculated_host_listings_count": 4,
  "availability_365": 28
},
  "id": {
    "$oid": "64445bb3f3134793d88f5009"
  },
  "id": 20333471,
  "name": "\u2668Hostel Style Room | Ideal Traveling Buddies\u2668",
  "host_id": 131697576,
  "host_name": "Anisha",
  "neighbourhood_group": "Bronx",
  "neighbourhood": "East Morrisania",
  "latitude": 40.83296,
  "longitude": -73.88668,
  "room_type": "Private room",
  "price": 0,
  "minimum_nights": 2,
  "number_of_reviews": 55,
  "last_review": "2019-06-24",
  "reviews_per_month": 2.56,
  "calculated_host_listings_count": 4,
  "availability_365": 127
}
```

**GET command implementation to get the listings which are ordered by price and equals to 0 (price is zero because of coupons)**

```
Windows PowerShell
PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings.json?orderBy=price&limitToFirst=5"
{
  "id": {
    "$oid": "64445bb3f3134793d88f4729"
  },
  "id": 18756597,
  "name": "Huge Brooklyn Brownstone Living, Close to it all.",
  "host_id": 8993884,
  "host_name": "Kimberly",
  "neighbourhood_group": "Brooklyn",
  "neighbourhood": "Bedford-Stuyvesant",
  "latitude": 40.69023,
  "longitude": -73.95428,
  "room_type": "Private room",
  "price": 0,
  "minimum_nights": 4,
  "number_of_reviews": 1,
  "last_review": "2018-01-06",
  "reviews_per_month": 0.05,
  "calculated_host_listings_count": 4,
  "availability_365": 28
},
  "id": {
    "$oid": "64445bb3f3134793d88f5009"
  },
  "id": 20333471,
  "name": "\u2668Hostel Style Room | Ideal Traveling Buddies\u2668",
  "host_id": 131697576,
  "host_name": "Anisha",
  "neighbourhood_group": "Bronx",
  "neighbourhood": "East Morrisania",
  "latitude": 40.83296,
  "longitude": -73.88668,
  "room_type": "Private room",
  "price": 0,
  "minimum_nights": 2,
  "number_of_reviews": 55,
  "last_review": "2019-06-24",
  "reviews_per_month": 2.56,
  "calculated_host_listings_count": 4,
  "availability_365": 127
}
```

**GET command implementation to get the first five listings(The 5 lease priced listings) in ascending order which are ordered by price (price is zero because of coupons).**

```
Windows PowerShell
PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings.json?orderBy=price&limitToLast=5"
[
  {
    "_id": {
      "$oid": "644445bb3f3134793d80f5ee6"
    },
    "id": 22436899,
    "name": "1-BR Lincoln Center",
    "host_id": 72390391,
    "host_name": "Jelena",
    "neighbourhood_group": "Manhattan",
    "neighbourhood": "Upper West Side",
    "latitude": 40.77213,
    "longitude": -73.98665,
    "room_type": "Entire home/apt",
    "price": 10000,
    "minimum_nights": 30,
    "number_of_reviews": 0,
    "last_review": null,
    "reviews_per_month": null,
    "calculated_host_listings_count": 1,
    "availability_365": 83
  },
  {
    "_id": {
      "$oid": "644445bb3f3134793d80f31cc"
    },
    "id": 13894339,
    "name": "Luxury 1 bedroom apt. -stunning Manhattan views",
    "host_id": 5143901,
    "host_name": "Erin",
    "neighbourhood_group": "Brooklyn",
    "neighbourhood": "Greenpoint",
    "latitude": 40.7326,
    "longitude": -73.95739,
    "room_type": "Entire home/apt",
    "price": 10000,
    "minimum_nights": 5,
    "number_of_reviews": 5,
    "last_review": "2017-07-27",
    "reviews_per_month": 0.16,
    "calculated_host_listings_count": 1,
    "availability_365": 0
  }
]
```

GET command implementation to get the last five listings(The top 5 highest price listings) in descending order which are ordered by price (price is zero because of coupons).

## PUT

```
PS C:\Users\vedan\PycharmProject\wc> curl -X PUT -H "Content-Type: application/json" -d '{"name": "DSCI 551 - Spr-23", "price": 150, "id": 2023}' 'http://localhost:5000/listings.json'
{"_id": "2023", "name": "DSCI 551 - Spr-23", "price": 150}
```

airbnb.listings 48.9k 4  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter  Reset Find More Options

ADD DATA EXPORT COLLECTION 1 - 1 of 1

```
{
  "_id": ObjectId('6445e889afb1f081d795e030')
  name: "DSCI 551 - Spr-23"
  price: 150
  id: 2023
}
```

PUT command implementation to insert a new ID which is 2023.

## POST

```
PS C:\Users\vedan\PycharmProject\wc> curl -X POST -H "Content-Type: application/json" -d '{"name": "\Vilas- New Town", "host_id": 12345, "host_name": "\John Doe", "neighbourhood_group": "\Brooklyn", "neighbourhood": "\Williamsburg", "latitude": 40.7145, "longitude": -73.9443, "room_type": "\Private room", "price": 100, "minimum_nights": 1, "number_of_reviews": 0, "last_review": "", "reviews_per_month": 0, "calculated_host_listings_count": 1, "availability_365": 365}' 'http://localhost:5000/listings.json'
```

```
{
  "listing_id": "c51a7480-2434-460a-a99b-e6191176f1f9",
  "message": "Listing created successfully."
}
```

airbnb.listings

48.9k 4  
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter ⓘ ⓘ { "id": "c51a7480-2434-460a-a99b-e6191176f1f9" }

Reset Find </> More Options ▶

ADD DATA EXPORT COLLECTION

1 - 1 of 1 < > ☰ {} 📄

```
_id: ObjectId('6445eb5dfcc2625f8949ba39')
name: "Vilas- New Town"
host_id: 12345
host_name: "John Doe"
neighbourhood_group: "Brooklyn"
neighbourhood: "Williamsburg"
latitude: 40.7145
longitude: -73.9443
room_type: "Private room"
price: 100
minimum_nights: 1
number_of_reviews: 0
last_review: ""
reviews_per_month: 0
calculated_host_listings_count: 1
availability_365: 365
id: "c51a7480-2434-460a-a99b-e6191176f1f9"
```

POST command implementation which creates a new listing id .

## PATCH

airbnb.listings

Documents Aggregations Schema Explain Plan Indexes Validation

Filter ⓘ ⓘ Type a query: { field: 'value' }

ADD DATA EXPORT COLLECTION

```
_id: ObjectId('64445bb3f3134793d80eecebe')
id: 6090
name: "West Village Nest - Superhost"
host_id: 11975
host_name: "Alina"
neighbourhood_group: "Manhattan"
neighbourhood: "West Village"
latitude: 40.7353
longitude: -74.00525
room_type: "Entire home/apt"
price: 120
minimum_nights: 90
number_of_reviews: 27
last_review: "2018-10-31"
reviews_per_month: 0.22
calculated_host_listings_count: 1
availability_365: 0
```

Before updating using PATCH we have a price as 120 and minimum\_nights as 90.

## airbnb.listings

Documents Aggregations Schema Explain Plan Indexes Validation

Filter   {"id": 6090}

 ADD DATA ▾

 EXPORT COLLECTION

```
_id: ObjectId('64445bb3f3134793d80eecbe')
id: 6090
name: "West Village Nest - Superhost"
host_id: 11975
host_name: "Alina"
neighbourhood_group: "Manhattan"
neighbourhood: "West Village"
latitude: 40.7353
longitude: -74.00525
room_type: "Entire home/apt"
price: 200
minimum_nights: 80
number_of_reviews: 27
last_review: "2018-10-31"
reviews_per_month: 0.22
calculated_host_listings_count: 1
availability_365: 0
```

After updating using PATCH we have a price as 200 and minimum\_nights as 80.

## DELETE

### airbnb.listings

Documents Aggregations Schema Explain Plan Indexes

Filter   Type a query: { field: 'value' }

 ADD DATA ▾

 EXPORT COLLECTION

```
_id: ObjectId('64445bb3f3134793d80eeeb8')
id: 5203
name: "Cozy Clean Guest Room - Family Apt"
host_id: 7490
host_name: "MaryEllen"
neighbourhood_group: "Manhattan"
neighbourhood: "Upper West Side"
latitude: 40.80178
longitude: -73.96723
room_type: "Private room"
price: 79
minimum_nights: 2
number_of_reviews: 118
last_review: "2017-07-21"
reviews_per_month: 0.99
calculated_host_listings_count: 1
availability_365: 0
```



```

PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings/5203.json"
{
  "_id": {
    "$oid": "64445bb3f3134793d80eecb8"
  },
  "id": 5203,
  "name": "Cozy Clean Guest Room - Family Apt",
  "host_id": 7490,
  "host_name": "MaryEllen",
  "neighbourhood_group": "Manhattan",
  "neighbourhood": "Upper West Side",
  "latitude": 40.80178,
  "longitude": -73.96723,
  "room_type": "Private room",
  "price": 79,
  "minimum_nights": 2,
  "number_of_reviews": 118,
  "last_review": "2017-07-21",
  "reviews_per_month": 0.99,
  "calculated_host_listings_count": 1,
  "availability_365": 0
}
PS C:\Users\vedan\PycharmProject\wc> curl -X DELETE 'http://localhost:5000/listings/5203.json'
{
  "result": "Listing deleted successfully."
}
PS C:\Users\vedan\PycharmProject\wc> curl -X GET "http://localhost:5000/listings/5203.json"
{
  "error": "Listing not found."
}
PS C:\Users\vedan\PycharmProject\wc>

```

We can see that the listing id 5203 has been deleted.

## LEARNING EXPERIENCES

- Understanding Firebase's Realtime Database

By emulating Firebase's Realtime Database, I gained a deeper understanding of its features, capabilities, and limitations. This knowledge can be useful when working with Firebase or other backend-as-a-service platforms in the future.

- Working with Flask

This project provided hands-on experience with Flask, a popular micro web framework for Python. I learned how to set up a Flask application, define routes, handle HTTP requests and responses, and work with JSON data.

- **MongoDB and NoSQL Databases**

I learned about MongoDB, a widely-used NoSQL database, and how to interact with it using the pymongo library. Explored the benefits of using a document-based database like MongoDB, such as flexibility in data storage, scalability, and fast querying capabilities.

- **Implementing CRUD operations**

Creating, reading, updating, and deleting records are essential operations for most web applications. This project allowed me to implement and understand these CRUD operations using Flask and MongoDB, which can be applied to other web development projects in the future.

- **Implementing Filtering and Sorting**

Gained experience in implementing advanced filtering and sorting functionality in my RESTful API. This skill can be useful when building APIs that require complex data retrieval and manipulation.

- **Error Handling and Validation**

Throughout the project, I encountered various challenges related to error handling and input validation. I learned the importance of validating user input, handling different types of errors, and providing meaningful error messages to users.

- **Testing and Debugging**

To ensure the correctness and robustness of the restful server, I tested its functionality using various curl commands and analyzed its behavior under different conditions. This experience reinforced the importance of thorough testing and debugging in software development.

This process improved my ability to communicate technical concepts clearly and concisely, which is essential for working in teams and presenting our work to others.

## **GOOGLE DRIVE LINK**

[https://drive.google.com/drive/folders/1CJQzEle1WZfsa82C4FNghxP2kJZ9uvwt?usp=share\\_link](https://drive.google.com/drive/folders/1CJQzEle1WZfsa82C4FNghxP2kJZ9uvwt?usp=share_link)

## **YOUTUBE VIDEO LINK**

<https://youtu.be/rokZmiU9Hx8>

