Database Project (Fall 2023)

Homework #6 (50pts, Due date: Nov 8)

Student ID: 2020315798

Student Name: Choi Jin Woo

Instruction: In this homework, we provide a dataset(airbnb-listings-newyork.json), and a jupyter notebook file(DBP_Homework6.ipynb). You should follow the instructions in these documents carefully.

Submission Guide: Submit two files as follows:

-DBP_Homwork6_StudentID.zip

- DBP_Homwork6_StudentID.ipynb
- DBP_Homwork6_StudentID.pdf

Data description:

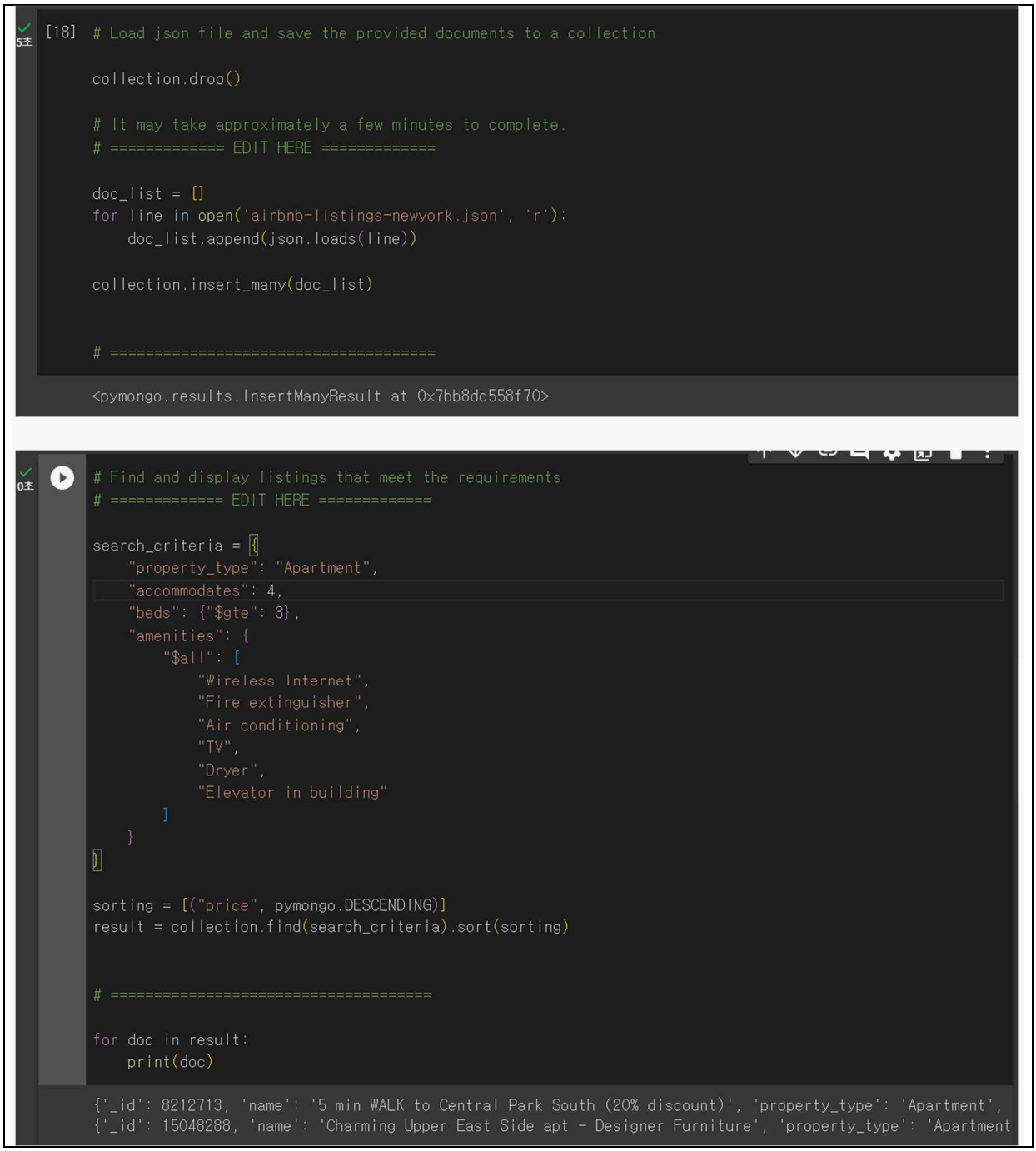
key	Type	description
_id	Int	Airbnb's unique identifier for the listing
name	String	Name of the listing
property_type	String	Property type of the listing
room_type	Int	Room type of the listing
accommodates	Float	The maximum number of people a listing can accommodate
beds	Int	Number of beds
amenities	List	Amenities included in a listing
price	Float	Price per night
text	string	Description about listing

1. [10pts] A user wants to book accommodations using Airbnb. Please find and display listings that meet the following requirements for the user. **Sort the results in descending order** and display the results.

property_type	accommodates	beds	amenities
Apartment	4	At least 3	Wireless Internet Fire extinguisher Air conditioning TV Dryer Elevator in building

[Answer]

Enter your code and result here. You must show your result (captured image).



```
[18] # Load json file and save the provided documents to a collection

     collection.drop()

     # It may take approximately a few minutes to complete.
     # ============= EDIT HERE =============

     doc_list = []
     for line in open('airbnb-listings-newyork.json', 'r'):
         doc_list.append(json.loads(line))

     collection.insert_many(doc_list)


     # ====================================

     <pymongo.results.InsertManyResult at 0x7bb8dc558f70>
```

```
# Find and display listings that meet the requirements
# ============= EDIT HERE =============

search_criteria = {
    "property_type": "Apartment",
    "accommodates": 4,
    "beds": {"$gte": 3},
    "amenities": {
        "$all": [
            "Wireless Internet",
            "Fire extinguisher",
            "Air conditioning",
            "TV",
            "Dryer",
            "Elevator in building"
        ]
    }
}

sorting = [("price", pymongo.DESCENDING)]
result = collection.find(search_criteria).sort(sorting)


# ====================================

for doc in result:
    print(doc)

{'_id': 8212713, 'name': '5 min WALK to Central Park South (20% discount)', 'property_type': 'Apartment',
{'_id': 15048288, 'name': 'Charming Upper East Side apt - Designer Furniture', 'property_type': 'Apartment
```

2. [20pts] Count all elements within the 'amenities' field in the collection, **sort by descending order, and display the top 10 with the highest counts**. The output format should look like this,

[Output]

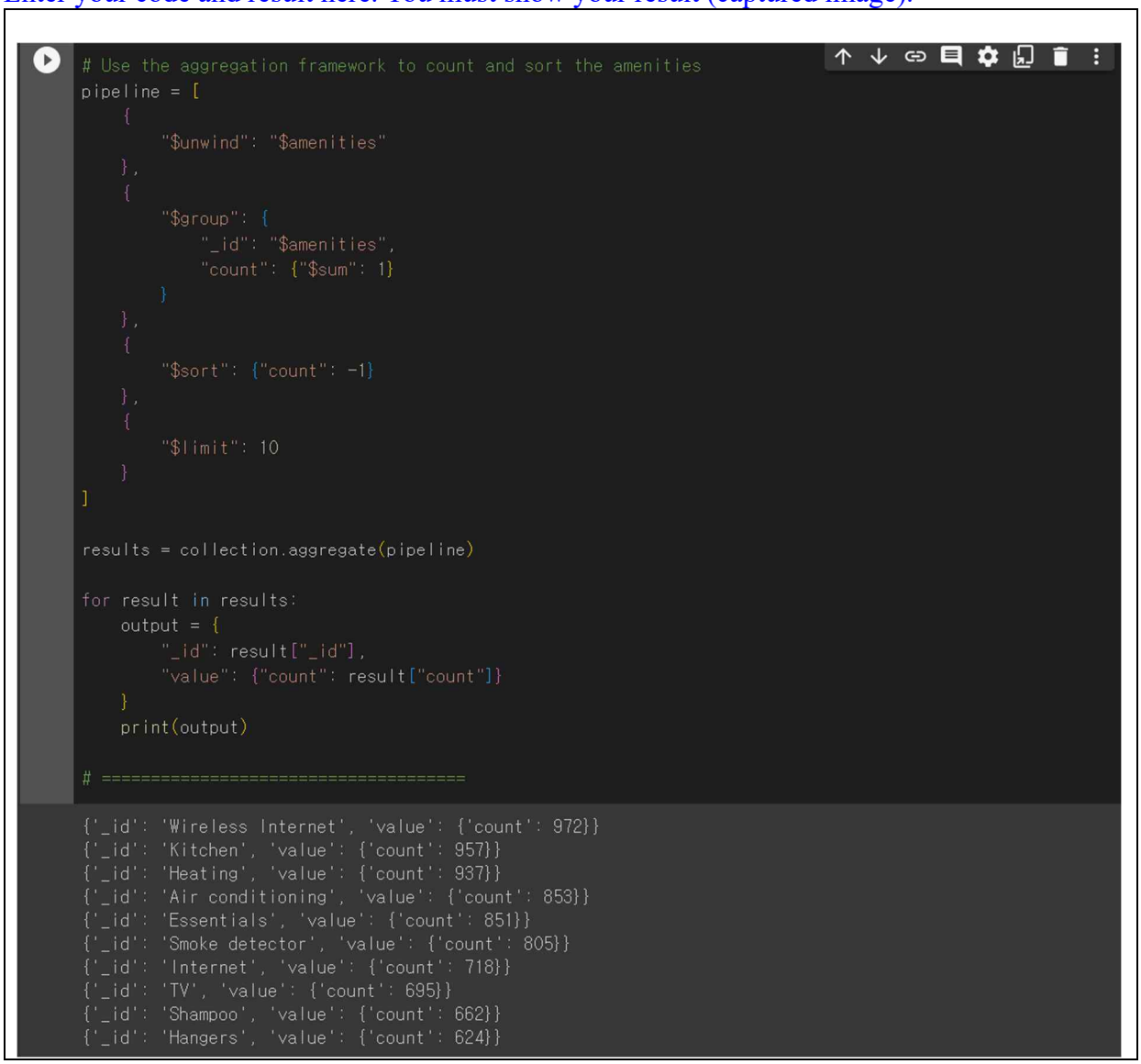
```
{'_id': 'Smoking allowed', 'value': {'count': 40}}
{'_id': 'Indoor fireplace', 'value': {'count': 47}}
{'_id': 'Breakfast', 'value': {'count': 54}}
{'_id': 'Self Check-In', 'value': {'count': 57}}
{'_id': 'Hot tub', 'value': {'count': 60}}
{'_id': 'Gym', 'value': {'count': 76}}
```

Fill in the blank and capture the code and results.

[Answer]

The most amenity	Count
Wireless Internet	972

Enter your code and result here. You must show your result (captured image).



```
# Use the aggregation framework to count and sort the amenities
pipeline = [
    {
        "$unwind": "$amenities"
    },
    {
        "$group": {
            "_id": "$amenities",
            "count": {"$sum": 1}
        }
    },
    {
        "$sort": {"count": -1}
    },
    {
        "$limit": 10
    }
]

results = collection.aggregate(pipeline)

for result in results:
    output = {
        "_id": result["_id"],
        "value": {"count": result["count"]}
    }
    print(output)

# ====================================
```

```
{'_id': 'Wireless Internet', 'value': {'count': 972}}
{'_id': 'Kitchen', 'value': {'count': 957}}
{'_id': 'Heating', 'value': {'count': 937}}
{'_id': 'Air conditioning', 'value': {'count': 853}}
{'_id': 'Essentials', 'value': {'count': 851}}
{'_id': 'Smoke detector', 'value': {'count': 805}}
{'_id': 'Internet', 'value': {'count': 718}}
{'_id': 'TV', 'value': {'count': 695}}
{'_id': 'Shampoo', 'value': {'count': 662}}
{'_id': 'Hangers', 'value': {'count': 624}}
```

3. [20pts] Solve the Word Count using the 'text' field in the collection, **sort by descending order, and display the top 10 results with the most counts.** The output format should look like this,

[Output]

```
{'_id': {'word': 'I'}, 'value': {'count': 112}}
{'_id': {'word': 'love'}, 'value': {'count': 111}}
{'_id': {'word': 'West'}, 'value': {'count': 110}}
{'_id': {'word': 'will'}, 'value': {'count': 109}}
{'_id': {'word': 'restaurants'}, 'value': {'count': 109}}
{'_id': {'word': 'minutes'}, 'value': {'count': 108}}
```

Fill in the blank and capture the code and results.

[Answer]

The most frequent word	Count
and	1989

Enter your code and result here. You must show your result (captured image).

```
[13]  # ============= EDIT HERE =============

      pipeline = [
          {
              '$project': {
                  'words': {
                      '$split': ['$text', ' ']
                   }
               }
          },
          {
              "$unwind": {
                  "path": "$words"
              }
          },
          {
              "$group": {
                  "_id": {
                      "word": "$words"
                  },
                  "count": {"$sum": 1}
              }
          },
          {
              "$sort": {"count": -1}
          },
          {
              "$limit": 10
          }
      ]

      results = collection.aggregate(pipeline)

      for result in results:
          output = {
              "_id": {"word": result["_id"]["word"]},
              "value": {"count": result["count"]}
          }
          print(output)


      # ====================================

      {'_id': {'word': 'and'}, 'value': {'count': 1989}}
      {'_id': {'word': 'the'}, 'value': {'count': 1795}}
      {'_id': {'word': 'a'}, 'value': {'count': 1217}}
      {'_id': {'word': 'in'}, 'value': {'count': 1131}}
      {'_id': {'word': 'to'}, 'value': {'count': 1116}}
      {'_id': {'word': 'is'}, 'value': {'count': 922}}
      {'_id': {'word': 'of'}, 'value': {'count': 919}}
      {'_id': {'word': ''}, 'value': {'count': 610}}
      {'_id': {'word': 'with'}, 'value': {'count': 596}}
      {'_id': {'word': 'for'}, 'value': {'count': 496}}
```