Steve Reiss
HW 10 – sqlalchemy-challenge

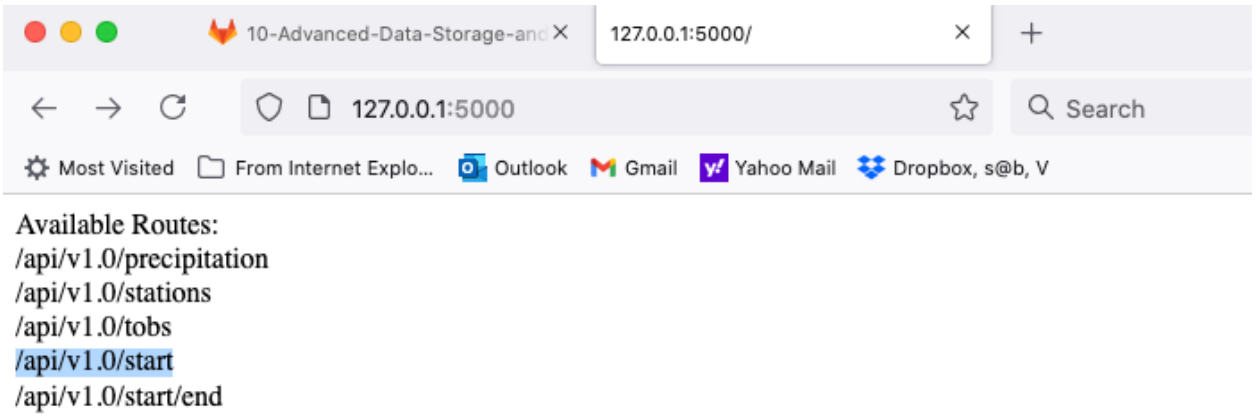
App.py with output images

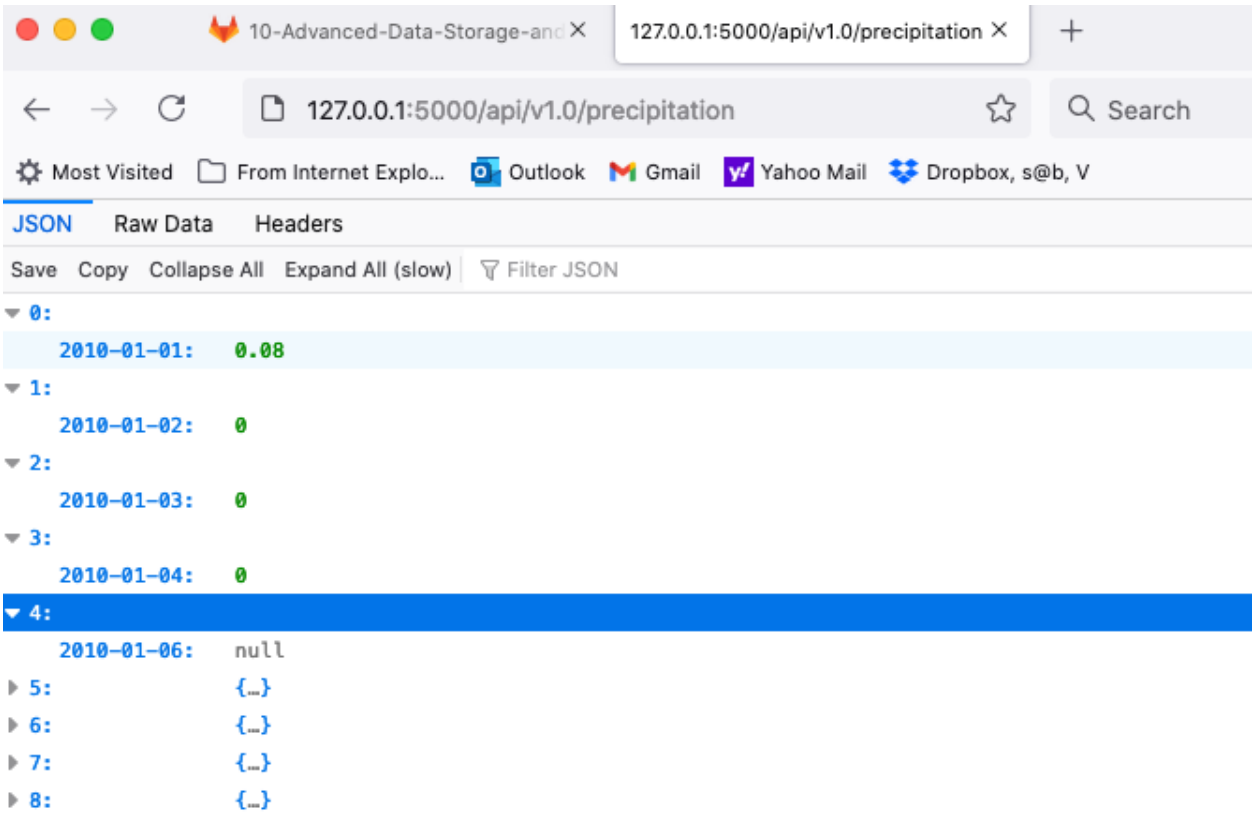
```
#################################################
# Dependencies
#################################################

import numpy as np
import sqlalchemy
from sqlalchemy.ext.automap import automap_base
from sqlalchemy.orm import Session
from sqlalchemy import create_engine, func
from flask import Flask, jsonify
import datetime as dt


#################################################
# Database Setup
#################################################

engine = create_engine("sqlite:///hawaii.sqlite")
# reflect an existing database into a new model
Base = automap_base()
# reflect the tables
Base.prepare(engine, reflect=True)
# Save reference to the table
Measurement = Base.classes.measurement
Station = Base.classes.station
session = Session(engine)


#################################################
# Flask Setup
#################################################
app = Flask(__name__)

#################################################
# Flask Routes
#################################################

@app.route("/")
```

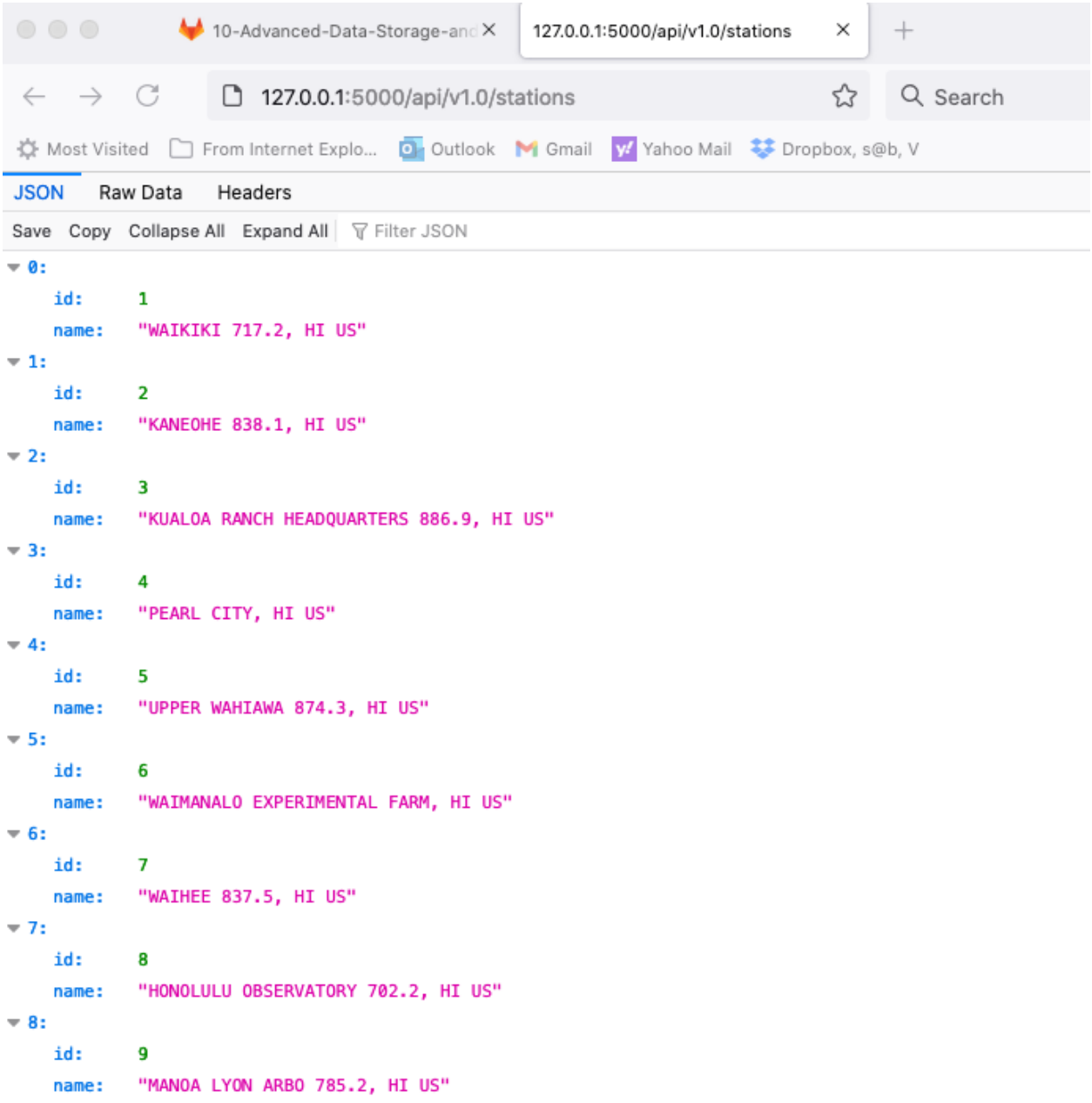
```
def welcome():
    """List all available api routes."""
    return (
        f"Available Routes:<br/>"
        f"/api/v1.0/precipitation<br/>"
        f"/api/v1.0/stations<br/>"
        f"/api/v1.0/tobs<br/>"
        f"/api/v1.0/start<br/>"
        f"/api/v1.0/start/end"
    )

# Create route and link - precipitation
@app.route("/api/v1.0/precipitation")
def precipitation():

    # Convert to a dictionary with date as key and prcp as the value
    results_prcp = session.query(Measurement.date, Measurement.prcp).all()


    all_precipitation = []
    for date, prcp in results_prcp:
        precipitation_dict = {}
        precipitation_dict[date] = prcp
        all_precipitation.append(precipitation_dict)

    return jsonify(all_precipitation)


# Create route and link - stations
@app.route("/api/v1.0/stations")
def stations():
    results_stations = session.query(Station.id, Station.name).all()

    all_stations = []
    for id, name in results_stations:
        stations_dict = {}
        stations_dict["id"] = id
        stations_dict["name"] = name
        all_stations.append(stations_dict)
    return jsonify(all_stations)


# Create route and link - temperatures
@app.route("/api/v1.0/tobs")
def tobs():
```

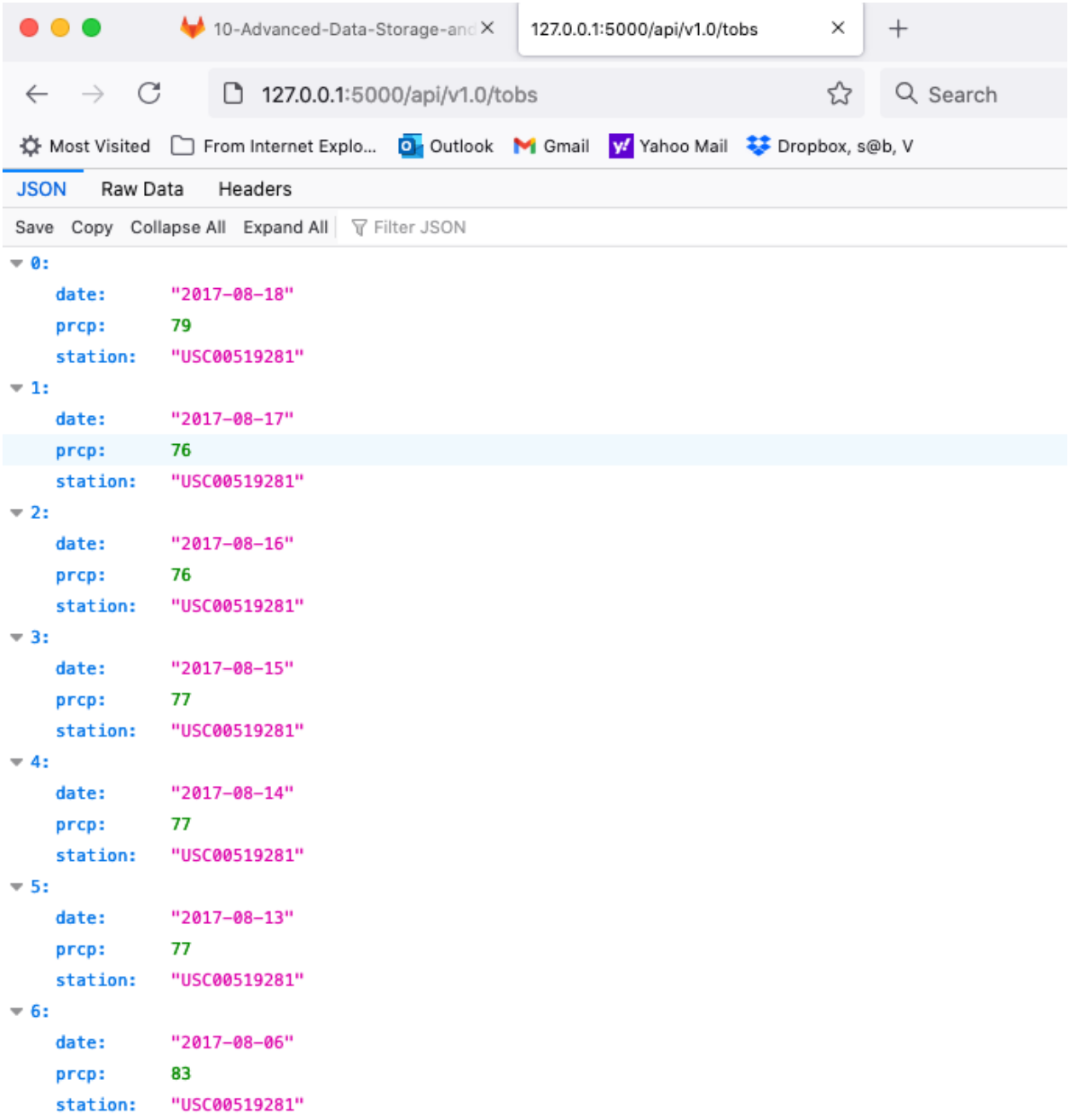
```
  # Query the dates and temperatures for the most active station over the last year of
data
    most_active_station =
session.query(Measurement.station).group_by(Measurement.station).order_by\
    (func.count(Measurement.id).desc()).first()
    latest_date =
session.query(Measurement.date).order_by(Measurement.date.desc()).first()

    latest_year = dt.date(2017,8,23) - dt.timedelta(days=365)

    results_tobs = session.query(Measurement.station, Measurement.date,
Measurement.tobs).filter\
       (Measurement.station == most_active_station[0]).filter(Measurement.date >=
latest_year)\
       .order_by(Measurement.date.desc()).all()

    tobs_stations = []
    for station, date, prcp in results_tobs:
        tobs_dict = {}
        tobs_dict["station"] = station
        tobs_dict["date"] = date
        tobs_dict ["prcp"] = prcp
        tobs_stations.append(tobs_dict)

    return jsonify(tobs_stations)


    #Create route and link for calculations - start


@app.route("/api/v1.0/<start>")
@app.route("/api/v1.0/<start>/<end>")
def start(start = None, end = None):

    sel = [func.min(Measurement.tobs), func.avg(Measurement.tobs),
func.max(Measurement.tobs)]

    if not end:
        tempresults = session.query(*sel).filter(Measurement.date >= start).all()
        tempdates = list(np.ravel(tempresults))
        return jsonify(tempdates)
```

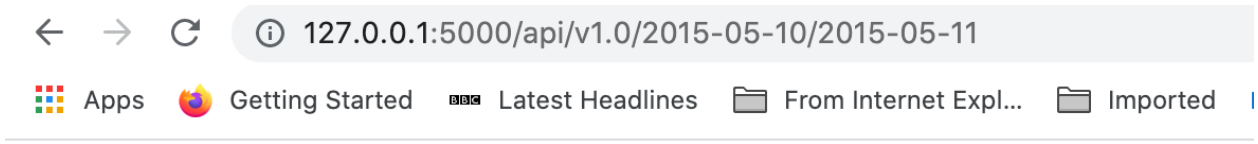
```
    tempresults = session.query(*sel).filter(Measurement.date >=
start).filter(Measurement.date <= end).all()
    tempdates = list(np.ravel(tempresults))
    return jsonify(tempdates=tempdates)


if __name__ == '__main__':
    app.run(debug=True)
```




10-Advanced-Data-Storage-and
127.0.0.1:5000/api/v1.0/precipitation
127.0.0.1:5000/api/v1.0/precipitation
Search
Most Visited
From Internet Explo...
Outlook
Gmail
Yahoo Mail
Dropbox, s@b, V
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All (slow)
Filter JSON
0:
2010-01-01: 0.08
1:
2010-01-02: 0
2:
2010-01-03: 0
3:
2010-01-04: 0
4:
2010-01-06: null
5: {…}
6: {…}
7: {…}
8: {…}


10-Advanced-Data-Storage-and
127.0.0.1:5000/api/v1.0/stations
127.0.0.1:5000/api/v1.0/stations
Search
Most Visited
From Internet Explo...
Outlook
Gmail
Yahoo Mail
Dropbox, s@b, V
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
0:
id: 1
name: "WAIKIKI 717.2, HI US"
1:
id: 2
name: "KANEOHE 838.1, HI US"
2:
id: 3
name: "KUALOA RANCH HEADQUARTERS 886.9, HI US"
3:
id: 4
name: "PEARL CITY, HI US"
4:
id: 5
name: "UPPER WAHIAWA 874.3, HI US"
5:
id: 6
name: "WAIMANALO EXPERIMENTAL FARM, HI US"
6:
id: 7
name: "WAIHEE 837.5, HI US"
7:
id: 8
name: "HONOLULU OBSERVATORY 702.2, HI US"
8:
id: 9
name: "MANOA LYON ARBO 785.2, HI US"


10-Advanced-Data-Storage-and
127.0.0.1:5000/api/v1.0/tobs
127.0.0.1:5000/api/v1.0/tobs
Search
Most Visited
From Internet Explo...
Outlook
Gmail
Yahoo Mail
Dropbox, s@b, V
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
0:
date: "2017-08-18"
prcp: 79
station: "USC00519281"
1:
date: "2017-08-17"
prcp: 76
station: "USC00519281"
2:
date: "2017-08-16"
prcp: 76
station: "USC00519281"
3:
date: "2017-08-15"
prcp: 77
station: "USC00519281"
4:
date: "2017-08-14"
prcp: 77
station: "USC00519281"
5:
date: "2017-08-13"
prcp: 77
station: "USC00519281"
6:
date: "2017-08-06"
prcp: 83
station: "USC00519281"




```
{
  "tempdates": [
    68.0,
    73.84615384615384,
    79.0
  ]
}
```