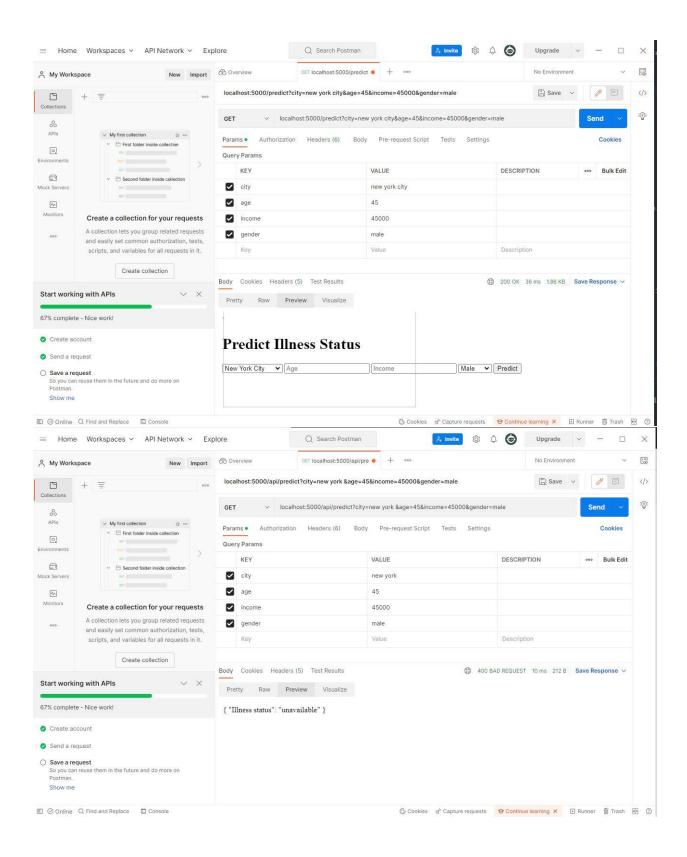
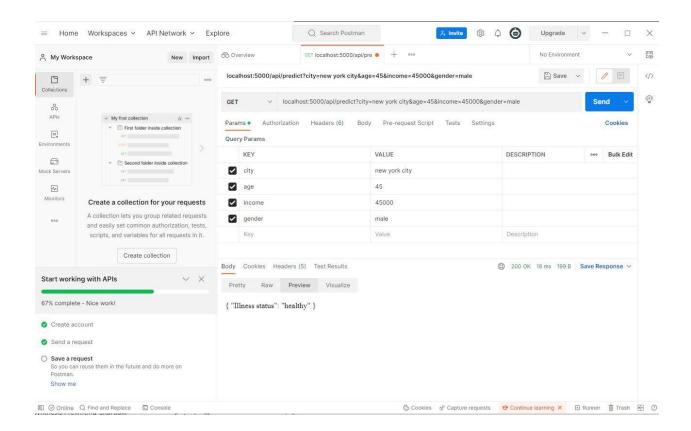
Name: Alyssa Hubiak

Batch Code: LISUM 13:30

Submission Date: 10/05/22

App URL: https://toy-data-api.herokuapp.com/





```
import numpy as np
     from flask import Flask, request, render template, jsonify
     import pickle
     import pandas as pd
     import os
     app = Flask( name )
     model = pickle.load(open('model.pkl', 'rb'))
10
     city dict ={
          'new york city': 1,
         'los angeles': 2,
         'dallas': 3,
         'mountain view': 4,
         'austin': 5,
         'boston': 6,
         'washington d.c.': 7,
         'san diego': 8
     gender dict = {
         'male': 0,
         'female': 1
24
     @app.route('/')
     def home():
         return render_template('index.html')
     @app.route('/predict',methods=['POST'])
     def predict():
         For rendering results on HTML GUI
         int features = [int(x) for x in request.form.values()]
         final features = [np.array(int features)]
         prediction = model.predict(final features)
         output = 'unavailable'
```

```
int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
   if prediction == 0:
    elif prediction == 1:
   return render_template('index.html', prediction_text='Illness status is {}'.format(output))
@app.route('/api/predict')
def ill_predict():
    city_get = request.args.get('city').lower()
    age = request.args.get('age')
    income = request.args.get('income')
    gender_get = request.args.get('gender').lower()
    if city_get in city_dict.keys():
       city = city_dict.get(city_get)
        return jsonify({'Illness status': output1}), 400
    if gender_get in gender_dict.keys():
       gender = gender_dict.get(gender_get)
       return jsonify({'Illness status': output1}), 400
    test_df = pd.DataFrame({'City':[city], 'Age':[age], 'Income':[income], 'Gender':[gender]})
    predicted = model.predict(test df)
    if predicted == 0:
    elif predicted == 1:
```

```
@app.route('/api/predict')
   output1='unavailable'
    city_get = request.args.get('city').lower()
    age = request.args.get('age')
    income = request.args.get('income')
    gender_get = request.args.get('gender').lower()
    if city_get in city_dict.keys():
       city = city_dict.get(city_get)
       return jsonify({'Illness status': output1}), 400
    if gender_get in gender_dict.keys():
       gender = gender_dict.get(gender_get)
       return jsonify({'Illness status': output1}), 400
    test_df = pd.DataFrame({'City':[city], 'Age':[age], 'Income':[income], 'Gender':[gender]})
    predicted = model.predict(test_df)
       output1 = 'healthy'
       output1 = 'sick'
   return jsonify({'Illness status': output1})
port = int(os.environ.get("PORT", 5000))
if __name__ == "__main__":
       app.run(host='0.0.0.0', port=port, debug=True)
```

App connected to GitHub

Code diffs, manual and auto deploys are
available for this app.



Activity Feed





alyssa.hubiak@gmail.com: Deployed b1fc9c40 Today at 10:09 PM - v3





alyssa.hubiak@gmail.com: Build succeeded Today at 10:08 PM - <u>View build log</u>





alyssa.hubiak@gmail.com: Enable Logplex Today at 10:07 PM - v2 - Roll back to here





alyssa.hubiak@gmail.com: Initial release Today at 10:07 PM · v1 · <u>Roll back to here</u>

