NAME: SHURAN FU
BATCH CODE: LISUM10
SUBMISSION DATE: JULY 3, 2022
SUBMITTED TO: WEEK 5 ASSIGNMENT: HEROKU DEPLOYMENT

Step 1: Model building and saving

The toy dataset contains 2 fields, one is years of experience and the other is gender, and we use these 2 features to predict the salary. The dataset is called Salary_Data and the following code in model.py constructs the model and save it to disk.

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
import requests
import json

# Importing the dataset
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 2].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                    test_size = 1/3, random_state = 0)

# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Saving model using pickle
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load( open('model.pkl','rb'))
print(model.predict([[1.8,0.]]))
```

Step 2: Prepare html and css file to show in the web

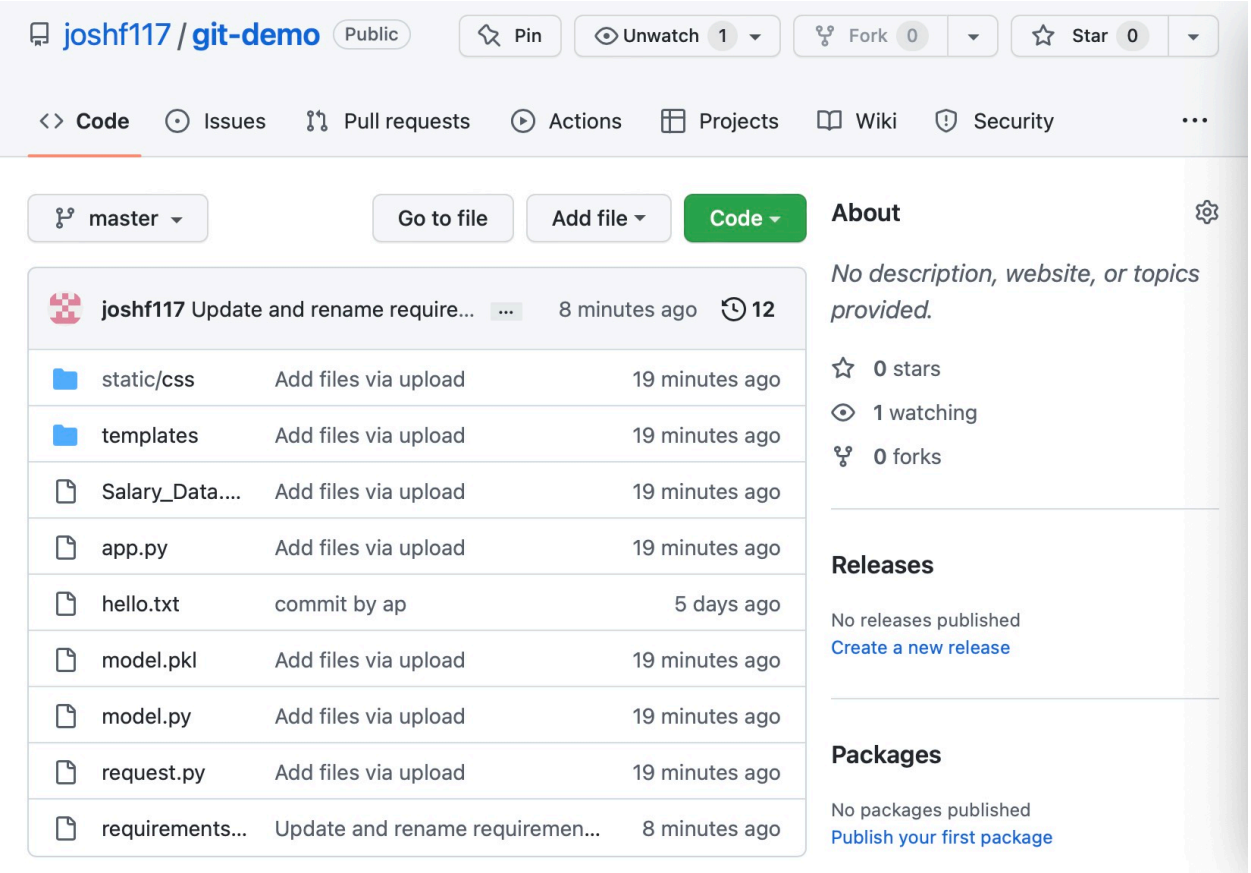
Prepare 2 input boxes so web visitors could enter the information to predict salary.

Step 3: Install Flask and prepare web app framework

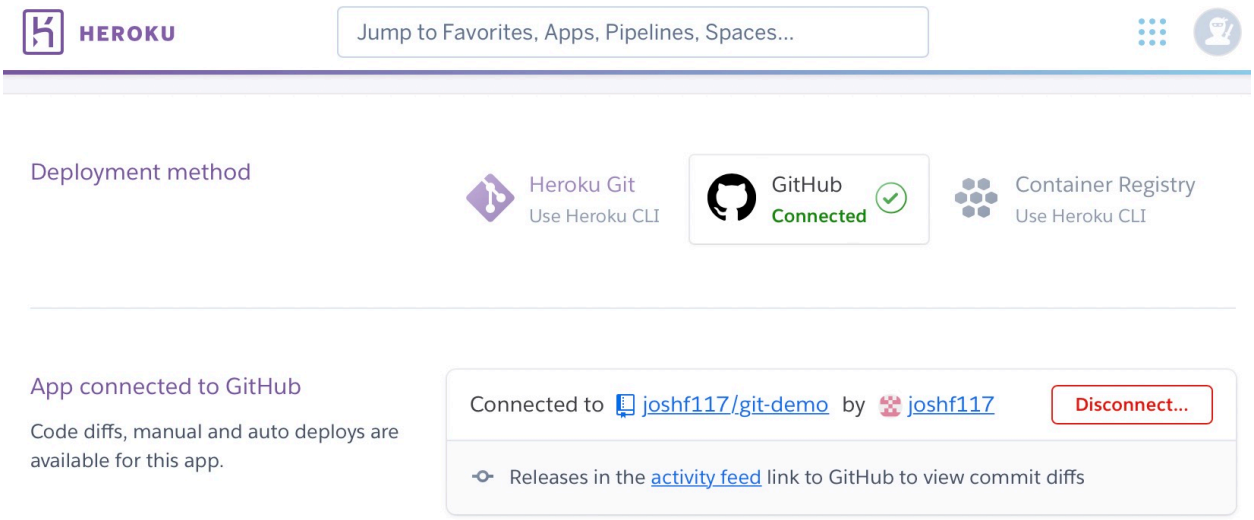
Use an app file called app.py to prepare request handling and processing. It collects the data from the form and address the model to predict the actual result, then it renders on the html file.

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Salary is {}'.format(output))


@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls through request
    '''
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

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

Step 4: Commit the files to github and add a config file called requirements.txt



Step 5: Create a heroku account and connect it to github.



Step 6: Successfully deploy the web app.