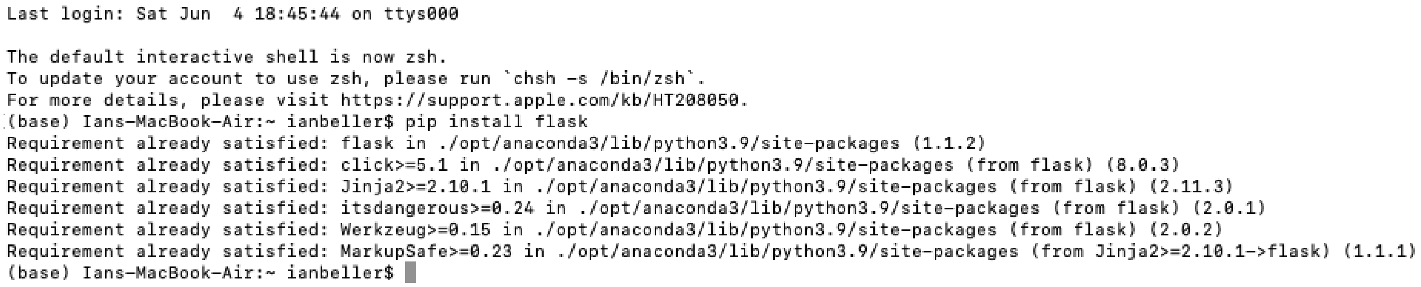
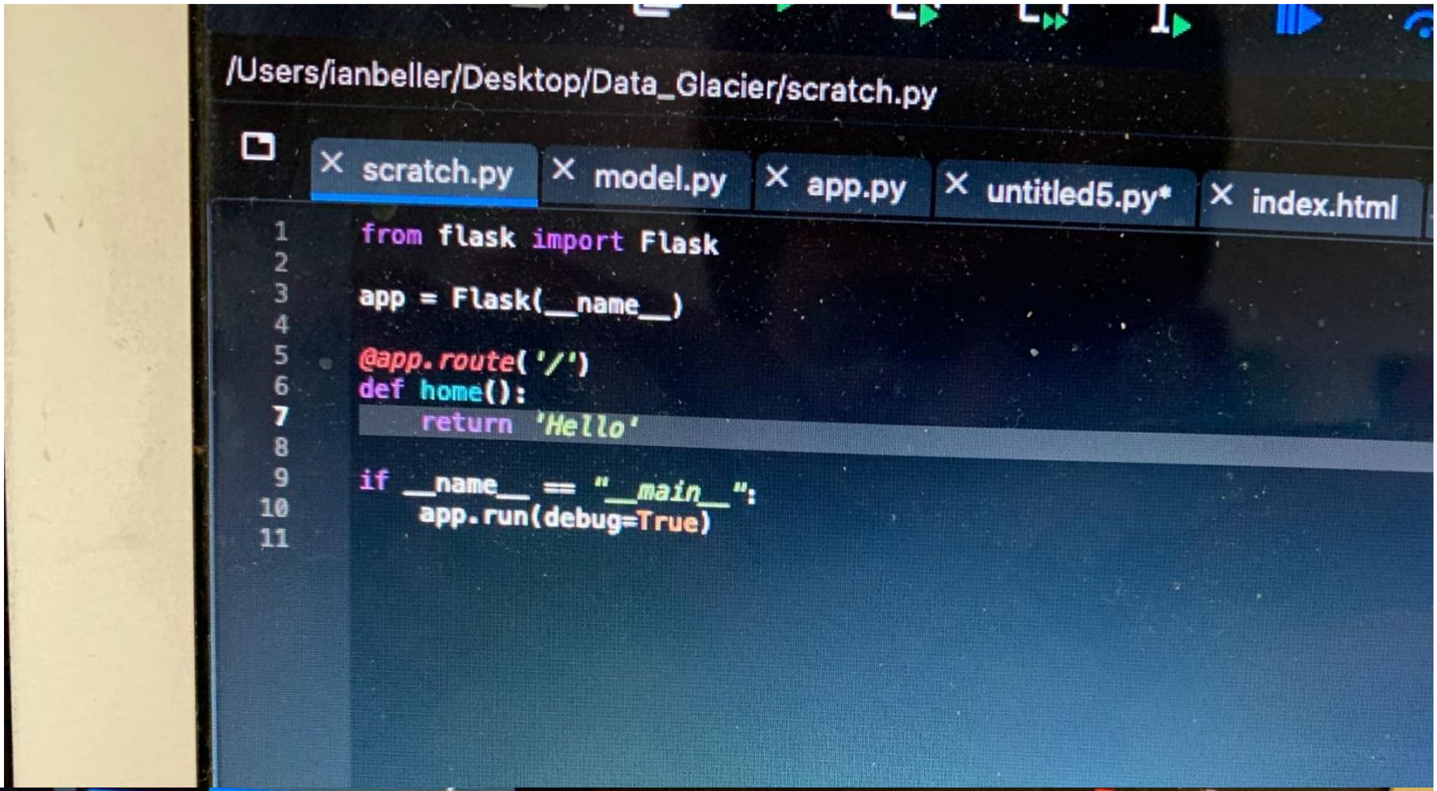
Name: Ian Beller
Batch code: LISMU09
Submission date: June 1st 2022
Submitted to: Gitbub

Installment of flask

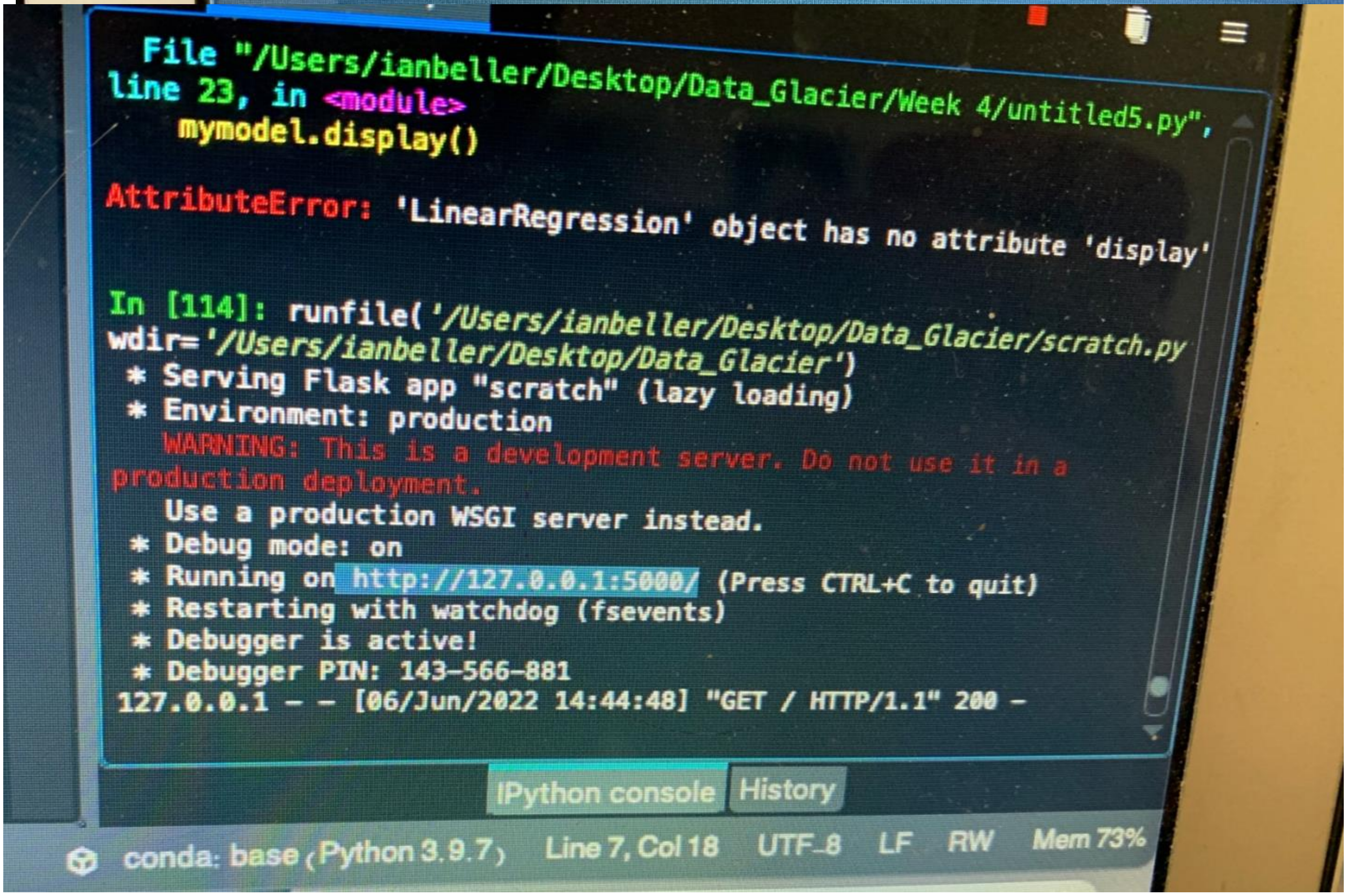


```
Last login: Sat Jun  4 18:45:44 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Ians-MacBook-Air:~ ianbeller$ pip install flask
Requirement already satisfied: flask in ./opt/anaconda3/lib/python3.9/site-packages (1.1.2)
Requirement already satisfied: click>=5.1 in ./opt/anaconda3/lib/python3.9/site-packages (from flask) (8.0.3)
Requirement already satisfied: Jinja2>=2.10.1 in ./opt/anaconda3/lib/python3.9/site-packages (from flask) (2.11.3)
Requirement already satisfied: itsdangerous>=0.24 in ./opt/anaconda3/lib/python3.9/site-packages (from flask) (2.0.1)
Requirement already satisfied: Werkzeug>=0.15 in ./opt/anaconda3/lib/python3.9/site-packages (from flask) (2.0.2)
Requirement already satisfied: MarkupSafe>=0.23 in ./opt/anaconda3/lib/python3.9/site-packages (from Jinja2>=2.10.1->flask) (1.1.1)
(base) Ians-MacBook-Air:~ ianbeller$
```

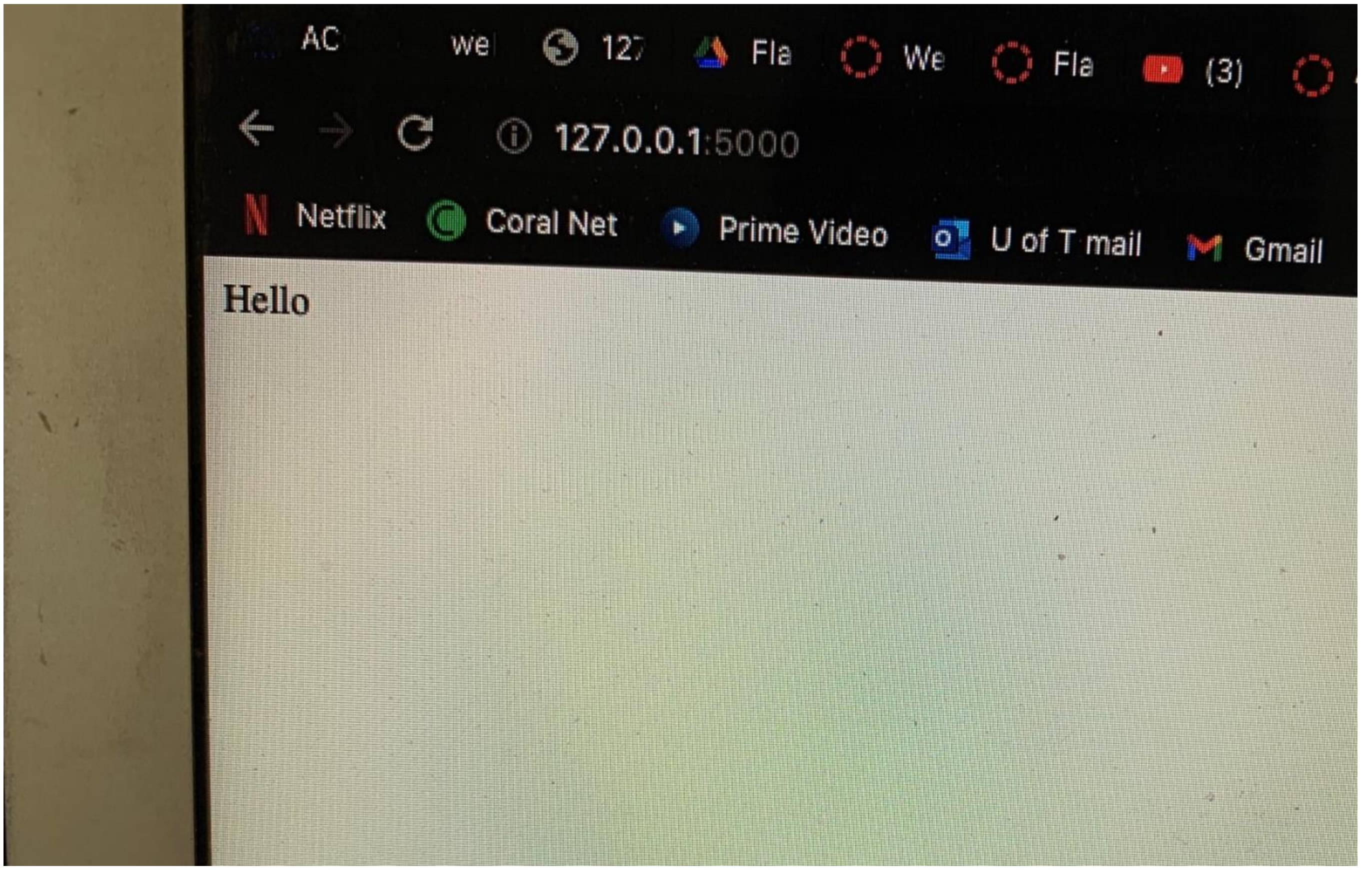
Deploying and setting up flask web app



```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return 'Hello'

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



```
  File "/Users/ianbeller/Desktop/Data_Glacier/Week 4/untitled5.py",
line 23, in <module>
    mymodel.display()

AttributeError: 'LinearRegression' object has no attribute 'display'

In [114]: runfile('/Users/ianbeller/Desktop/Data_Glacier/scratch.py
wdir='/Users/ianbeller/Desktop/Data_Glacier')
 * Serving Flask app "scratch" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a
production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with watchdog (fsevents)
 * Debugger is active!
 * Debugger PIN: 143-566-881
127.0.0.1 - - [06/Jun/2022 14:44:48] "GET / HTTP/1.1" 200 -
```

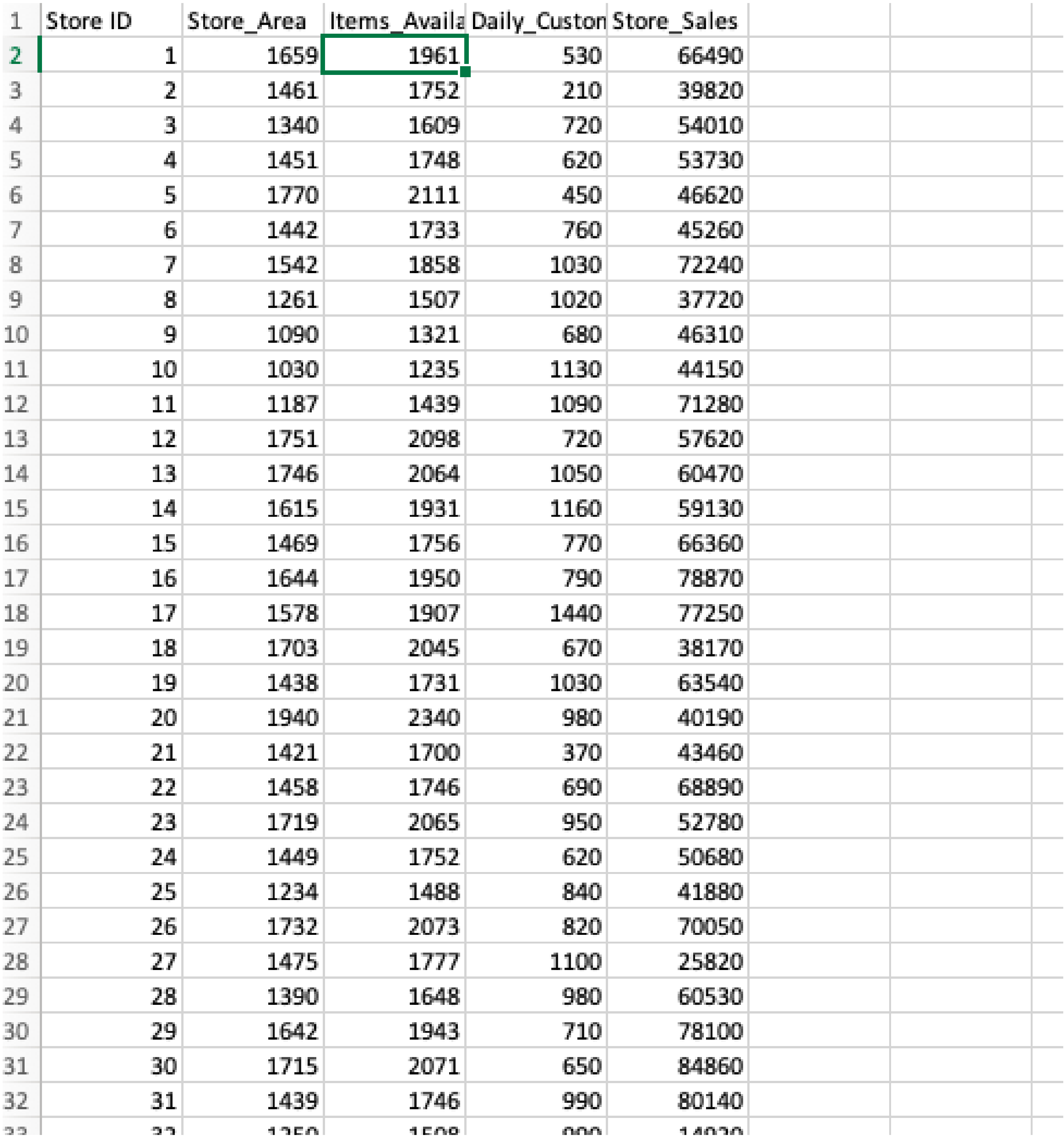


AC
we
127
Fla
We
Fla
(3)
127.0.0.1:5000
Netflix
Coral Net
Prime Video
U of T mail
Gmail

Hello

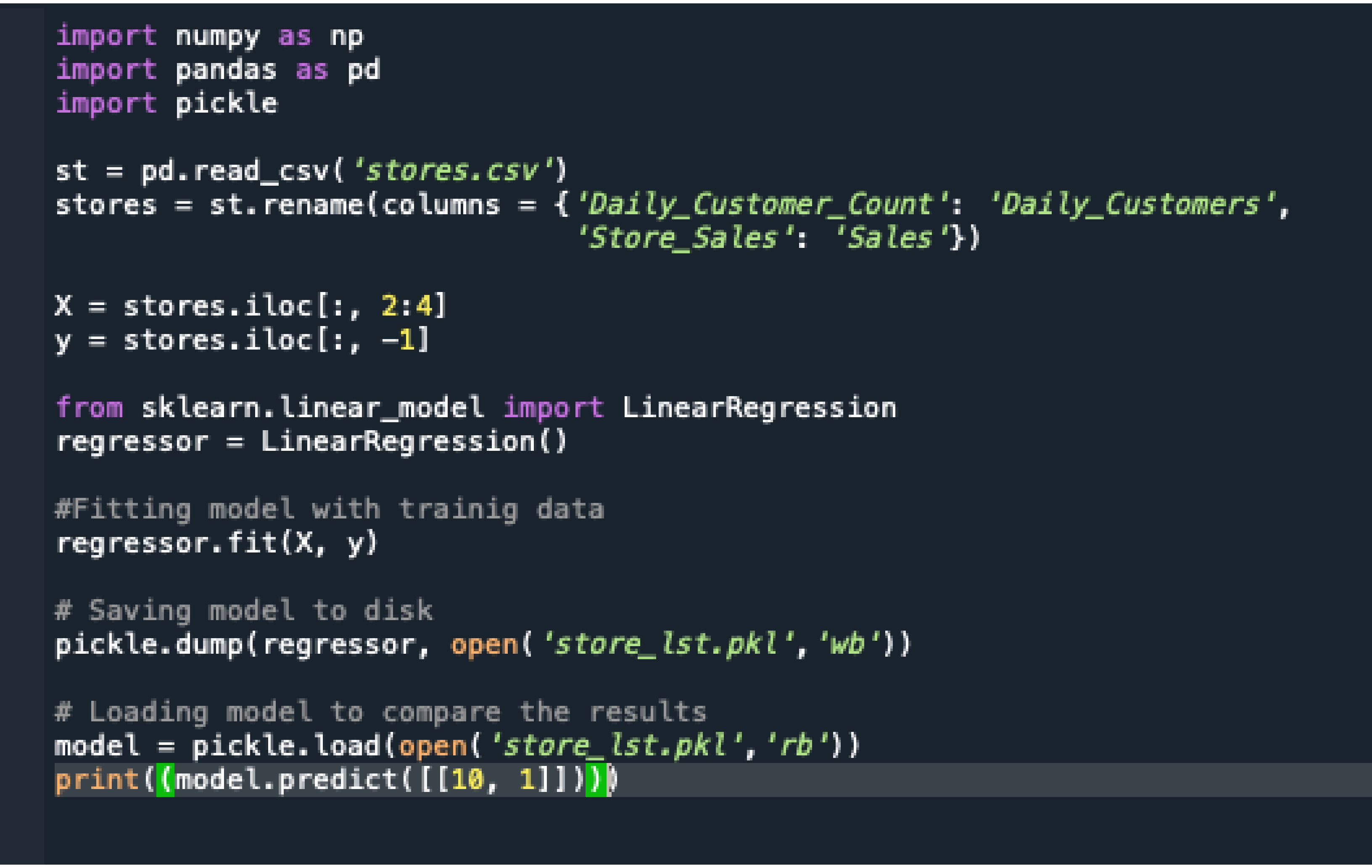
Simple dataset

https://www.kaggle.com/datasets/surajjha101/stores-area-and-sales-data



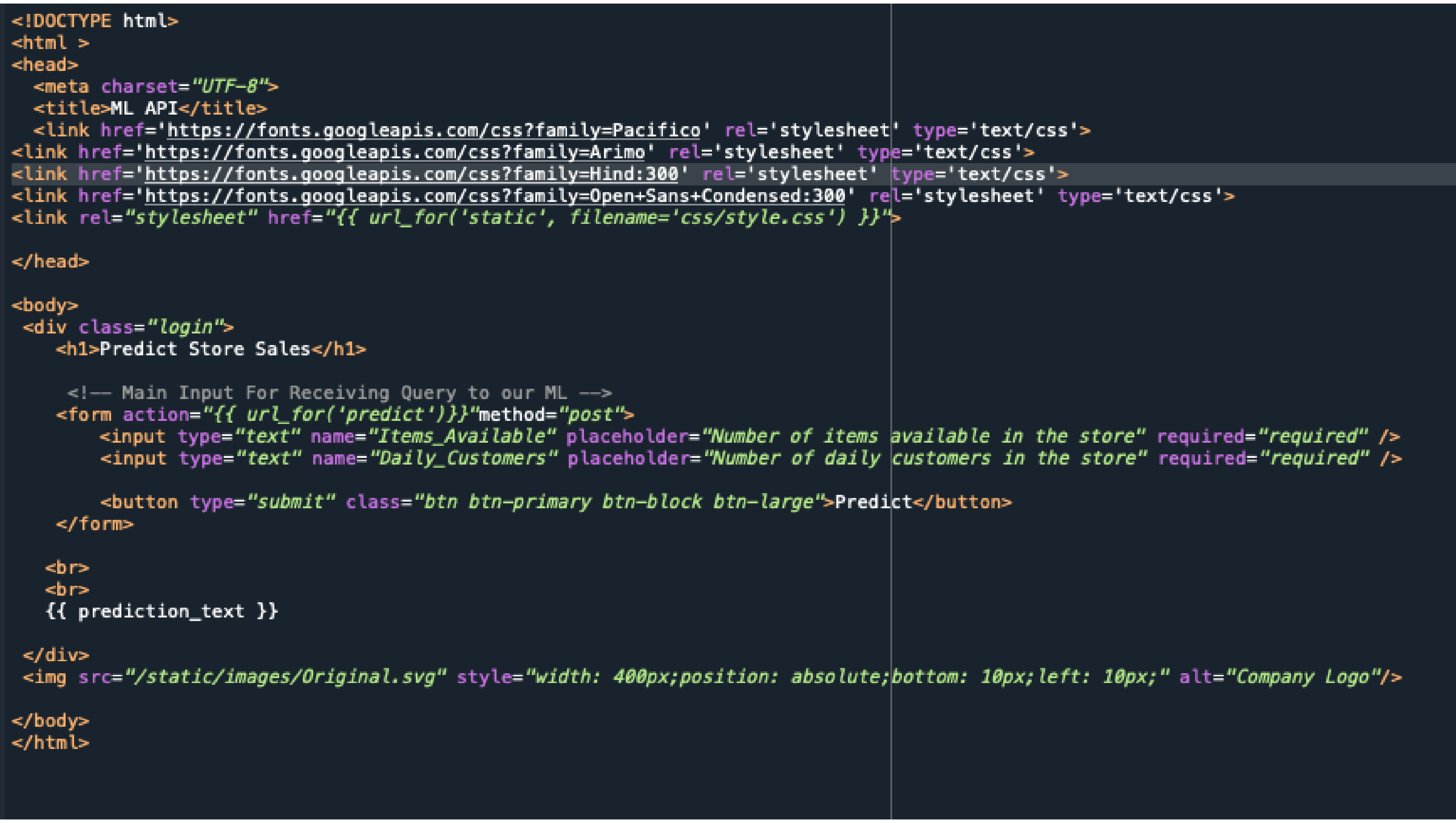
1	Store ID	Store_Area	Items_Availa	Daily_Custon	Store_Sales
2	1	1659	1961	530	66490
3	2	1461	1752	210	39820
4	3	1340	1609	720	54010
5	4	1451	1748	620	53730
6	5	1770	2111	450	46620
7	6	1442	1733	760	45260
8	7	1542	1858	1030	72240
9	8	1261	1507	1020	37720
10	9	1090	1321	680	46310
11	10	1030	1235	1130	44150
12	11	1187	1439	1090	71280
13	12	1751	2098	720	57620
14	13	1746	2064	1050	60470
15	14	1615	1931	1160	59130
16	15	1469	1756	770	66360
17	16	1644	1950	790	78870
18	17	1578	1907	1440	77250
19	18	1703	2045	670	38170
20	19	1438	1731	1030	63540
21	20	1940	2340	980	40190
22	21	1421	1700	370	43460
23	22	1458	1746	690	68890
24	23	1719	2065	950	52780
25	24	1449	1752	620	50680
26	25	1234	1488	840	41880
27	26	1732	2073	820	70050
28	27	1475	1777	1100	25820
29	28	1390	1648	980	60530
30	29	1642	1943	710	78100
31	30	1715	2071	650	84860
32	31	1439	1746	990	80140

Code for Model



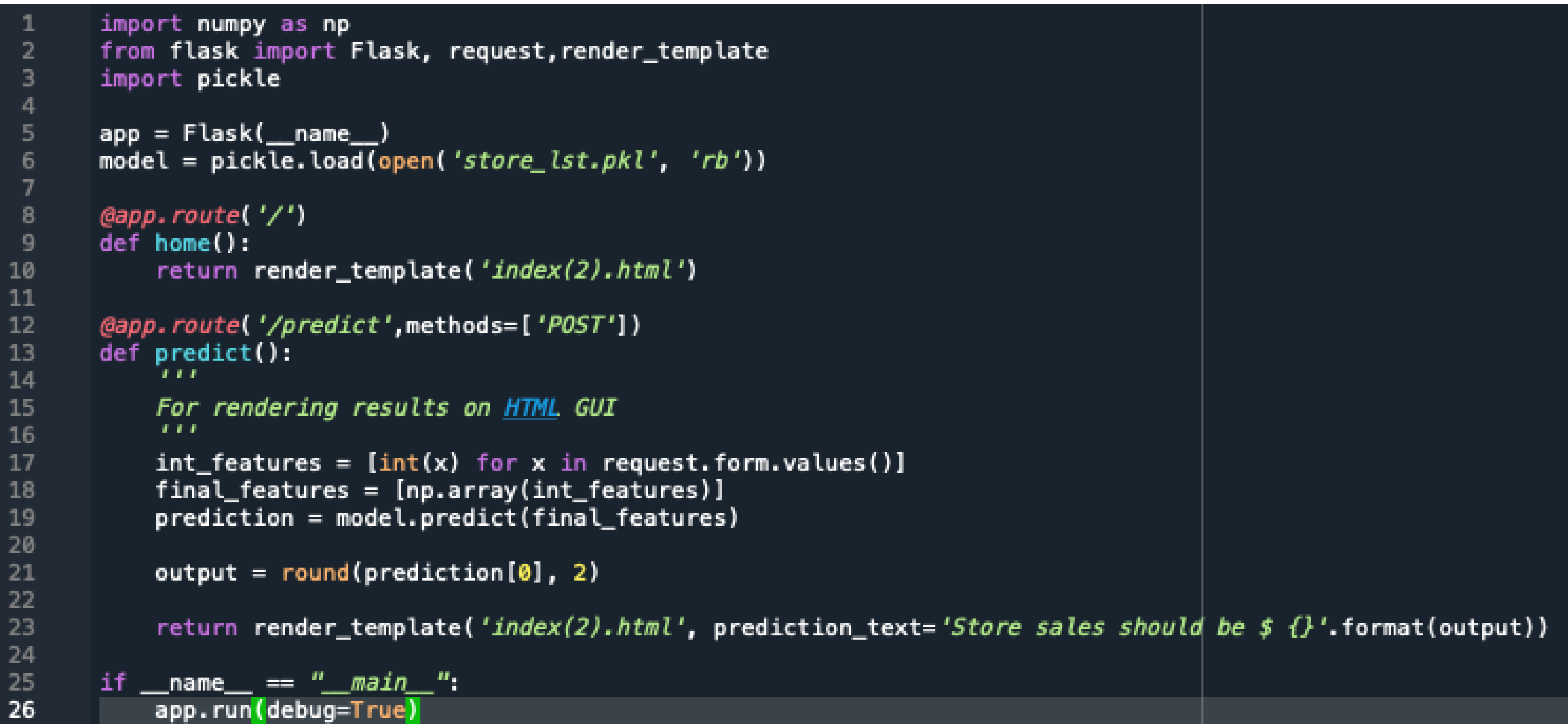
```
import numpy as np
import pandas as pd
import pickle

st = pd.read_csv('stores.csv')
stores = st.rename(columns = {'Daily_Customer_Count': 'Daily_Customers',
                              'Store_Sales': 'Sales'})

X = stores.iloc[:, 2:4]
y = stores.iloc[:, -1]

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with trainig data
regressor.fit(X, y)

# Saving model to disk
pickle.dump(regressor, open('store_lst.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('store_lst.pkl','rb'))
print((model.predict([[10, 1]])))
```

HTML code



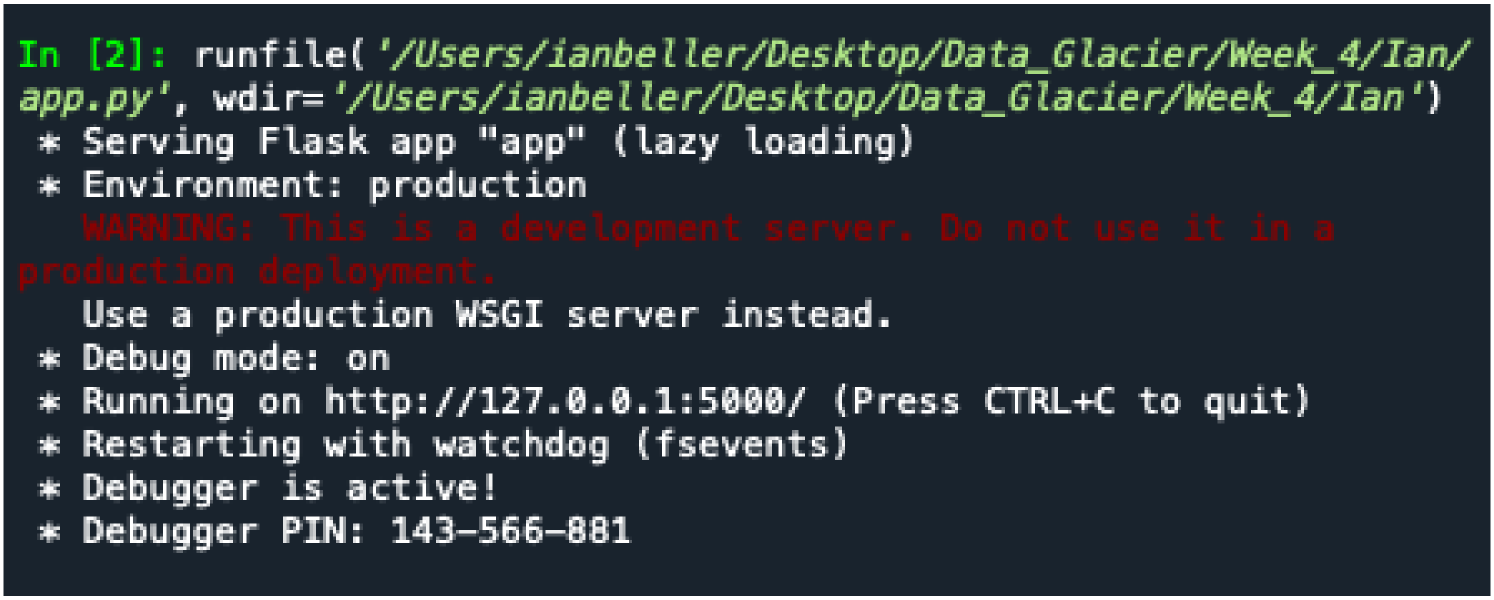
```html
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
    <h1>Predict Store Sales</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="Items_Available" placeholder="Number of items available in the store" required="required" />
        <input type="text" name="Daily_Customers" placeholder="Number of daily customers in the store" required="required" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>
 <img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;" alt="Company Logo"/>

</body>
</html>
```

App code



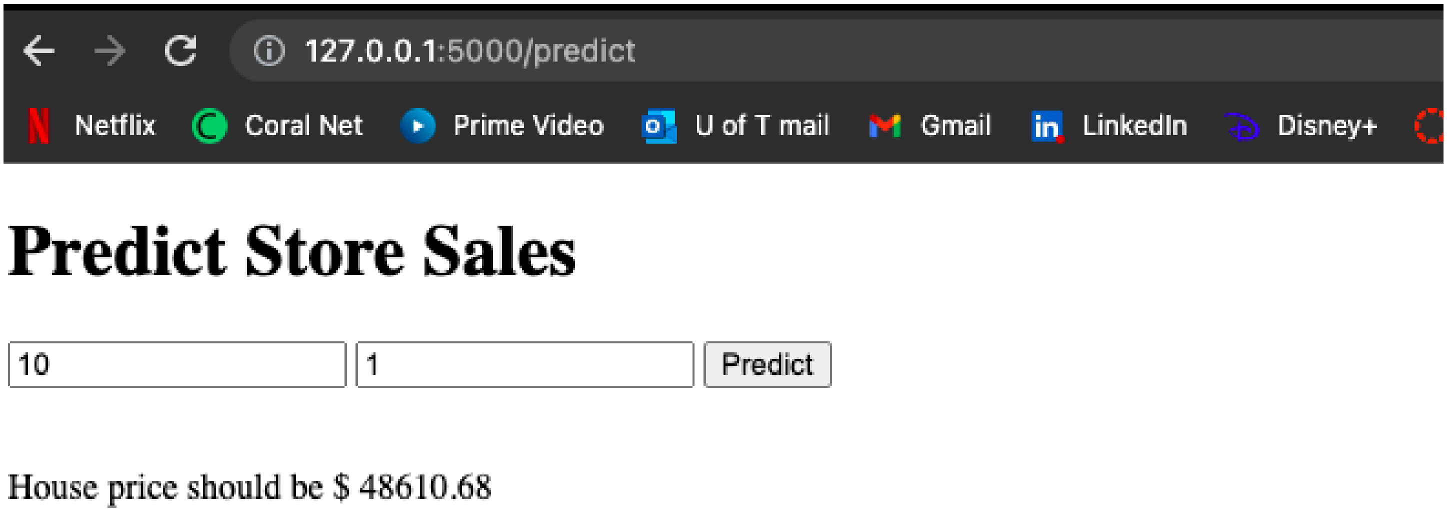
```
import numpy as np
from flask import Flask, request,render_template
import pickle

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

@app.route('/')
def home():
    return render_template('index(2).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 = round(prediction[0], 2)

    return render_template('index(2).html', prediction_text='Store sales should be $ {}'.format(output))

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

Running the app and the web address



```
In [2]: runfile('/Users/ianbeller/Desktop/Data_Glacier/Week_4/Ian/
app.py', wdir='/Users/ianbeller/Desktop/Data_Glacier/Week_4/Ian')
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a
production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with watchdog (fsevents)
 * Debugger is active!
 * Debugger PIN: 143-566-881
```

Web deployment



Predict Store Sales

10

1

House price should be $ 48610.68