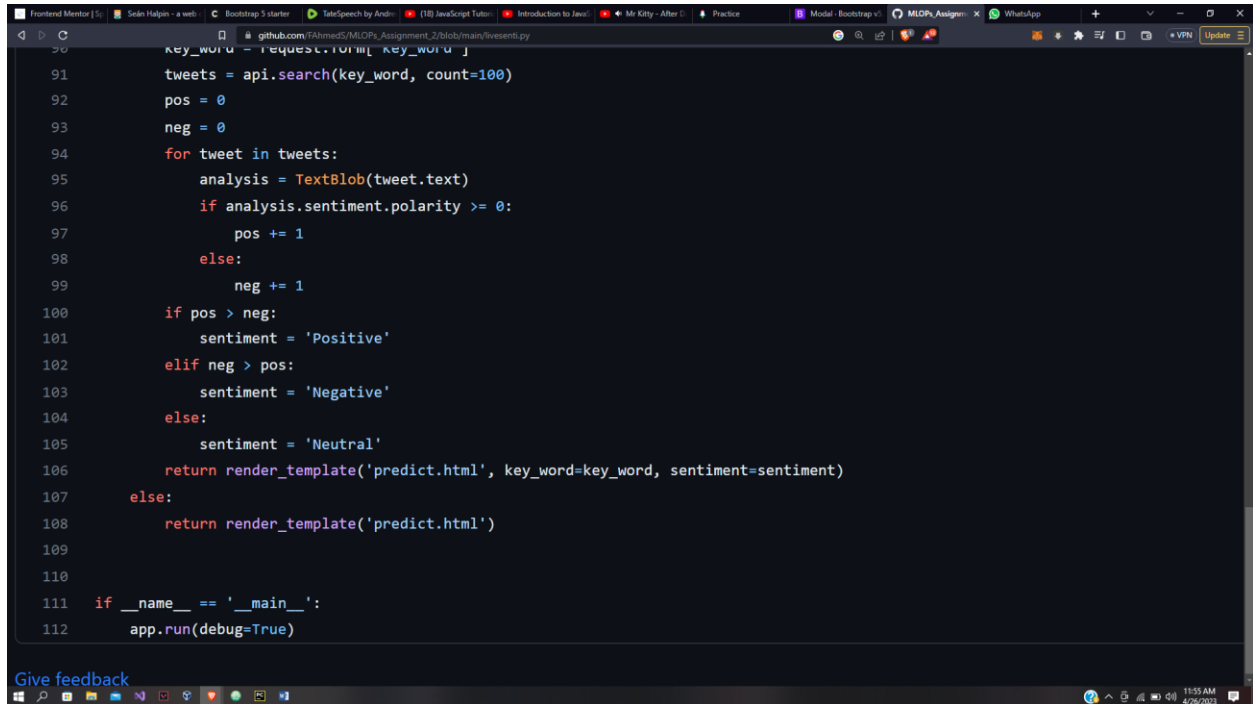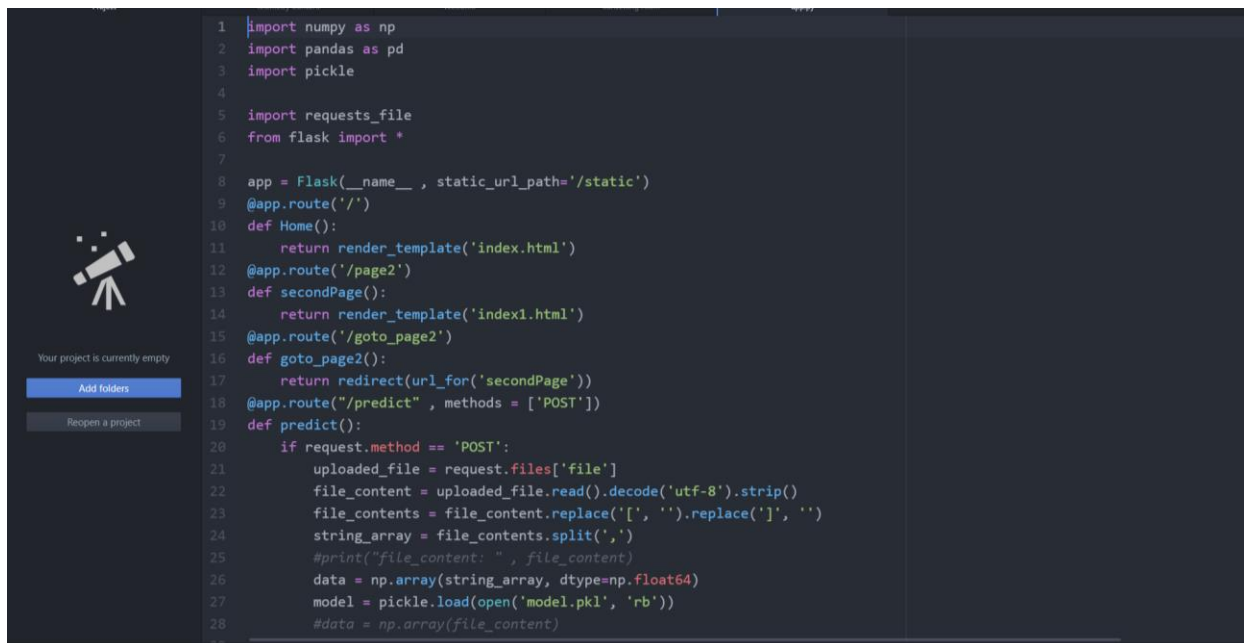# Steps:

1. Created simple sentiment analysis model in python.


```
90      key_word = request.form['key_word']
91      tweets = api.search(key_word, count=100)
92      pos = 0
93      neg = 0
94      for tweet in tweets:
95          analysis = TextBlob(tweet.text)
96          if analysis.sentiment.polarity >= 0:
97              pos += 1
98          else:
99              neg += 1
100     if pos > neg:
101         sentiment = 'Positive'
102     elif neg > pos:
103         sentiment = 'Negative'
104     else:
105         sentiment = 'Neutral'
106     return render_template('predict.html', key_word=key_word, sentiment=sentiment)
107     else:
108         return render_template('predict.html')
109
110
111 if __name__ == '__main__':
112     app.run(debug=True)
```

2. Flask application.


```
1   import numpy as np
2   import pandas as pd
3   import pickle
4
5   import requests_file
6   from flask import *
7
8   app = Flask(__name__ , static_url_path='/static')
9   @app.route('/')
10  def Home():
11      return render_template('index.html')
12  @app.route('/page2')
13  def secondPage():
14      return render_template('index1.html')
15  @app.route('/goto_page2')
16  def goto_page2():
17      return redirect(url_for('secondPage'))
18  @app.route("/predict" , methods = ['POST'])
19  def predict():
20      if request.method == 'POST':
21          uploaded_file = request.files['file']
22          file_content = uploaded_file.read().decode('utf-8').strip()
23          file_contents = file_content.replace('[', '').replace(']', '')
24          string_array = file_contents.split(',')
25          #print("file_content: " , file_content)
26          data = np.array(string_array, dtype=np.float64)
27          model = pickle.load(open('model.pkl', 'rb'))
28          #data = np.array(file_content)
```

3. form.html

Sentiment Analysis - User Input

Enter a keyword:

bad

Analyze

(% if result %)

Keyword: {{ result.keyword }}

Sentiment: {{ result.sentiment }}

(% endif %)

3. Docker file


```
1   # Use an official Python runtime as a parent image
2   FROM python:3.8-slim-buster
3
4   # Set the working directory to /app
5   WORKDIR /app
6
7   # Copy the current directory contents into the container at /app
8   COPY . /app
9
10  # Install any needed packages specified in requirements.txt
11  RUN pip install --trusted-host pypi.python.org -r requirements.txt
12
13  # Make port 80 available to the world outside this container
14  EXPOSE 80
15
16  # Define environment variable
17  ENV NAME World
18
19  # Run app.py when the container launches
20  CMD ["python", "livesenti.py"]
```

4.

4. Connected via Jenkins(webhook ngrok)

5. Bild successful