```
from flask import Flask


import numpy as np
import pandas as pd
import os
import re
import tensorflow as tf
from datetime import datetime, timedelta
from tensorflow import keras
import tensorflow_hub as hub

from flask import request, redirect, url_for, send_from_directory, render_template


b = """
<h1 style="color: #5e9ca0;">NLP Chat Classification</h1>
<p><em>A deployment made by German Baltazar, Anyinssan Nava, and David Lizana.</em></p>
<p>Powered by&nbsp;<img src="https://images.squarespace-cdn.com/content/v1/5d1bb4619672710001
<h2 style="color: #2e6c80;">How does it work?</h2>
<p>Insert a phrase on the text field, and our NLP algorithm will classify it into one of thre
<ol style="list-style: none; font-size: 14px; line-height: 32px; font-weight: bold;">
<li style="clear: both;">Chat<img style="float: left;" src="https://cdn-icons-png.flaticon.co
<li style="clear: both;">Inform<img style="float: left;" src="https://cdn-icons-png.flaticon.
<li style="clear: both;">Question<img style="float: left;" src="https://cdn-icons-png.flatico
</ol>
<p>&nbsp;</p>
<p>After writing your phrase, click on the <span style="background-color: #2b2301; color: #ff
<form action="" enctype="multipart/form-data" method="post"><textarea id="w3review" cols="50"
</form>
        </div>
        <div class="column" >
          <h2 style="color: #2e6c80;"> Our classifications</h2>
            <div class="row">
              <table class="table table-hover" style="margin-left:20px;" >
                  <tr class="table-active">
                  <th style="color: #2e6c80;" scope="col">ANN</th>
                  </tr>
                    <td> {{label}} </td>
                </tr>
            </div>
        </div>
      </div>

    </div>


      </div>
  </body>
"""
```

```
current = os.getcwd()
newdir = 'simpleApi'
path = os.path.join(current, newdir)
print(path)
```

```
C:\Users\Consultant\Desktop\NLP German\simpleApi\simpleApi
```

```
# os.mkdir(path)
# os.chdir(path)
```

```
os.getcwd()
```

```
'C:\\Users\\Consultant\\Desktop\\NLP German\\simpleApi'
```

```
procfile = 'web: gunicorn app:app'
procfiles = open('Procfile','w')
procfiles.write(procfile)
procfiles.close()
```

```
# os.mkdir('templates')
```

```
Html_file = open('templates/index.html', 'w')
```

```
Html_file.write(b)
Html_file.close()
```

```
app = Flask(__name__)
```

```
def load_ann(path, text):
    #Load the model
    loaded_model = keras.models.load_model((path), custom_objects={'KerasLayer': hub.KerasLay
    #Make the prediction
    probs = loaded_model.predict(np.array([text]))
    label = np.argmax(probs)
    prob = probs[0][label]

    return label, prob
```

```
def predict(prediction):
    idx = prediction[0]
    prob = prediction[1]

    if idx == 0:
        output = 'It is a Chat'
    elif idx == 1:
        output = 'It is to Inform'
    elif idx == 2:
        output = 'It is a Question'
    else:
        return "Sorry, we don't know what you're talking about..."

    return output + f" ({prob*100:0.2f}%)"


def process(text): #Modify this one to run your model
    path1 = "C:/Users/Consultant/Desktop/NLP German/ANN_model.h5"
    predANN = load_ann(path1, text)
    prediction1 = predict(predANN)
    return prediction1


@app.route("/")
def template_test():
    return render_template('index.html', label='', imagesource='file://null')

@app.route('/', methods=['POST'])
def upload_file():

        request_out=request.form['w3review']#Info In
        output = process(request_out)#model run
        label = output
#         albert = ' \n '.join(output['ALBERT'])
#         out2 = output["ALBERT"]#Info out
#         Quests=' \n '.join(output['questions'])
#         Notes=' \n '.join(output['notes'])
        return render_template("index.html", label=output)#, label2=albert)


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

```
 * Serving Flask app "__main__" (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: off
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
INFO:werkzeug:127.0.0.1 - - [19/Jun/2022 20:53:41] "GET / HTTP/1.1" 200 -
1/1 [==============================] - 0s 72ms/step
INFO:werkzeug:127.0.0.1 - - [19/Jun/2022 20:53:50] "POST / HTTP/1.1" 200 -
1/1 [==============================] - 0s 73ms/step
INFO:werkzeug:127.0.0.1 - - [19/Jun/2022 20:53:58] "POST / HTTP/1.1" 200 -
1/1 [==============================] - 0s 72ms/step
INFO:werkzeug:127.0.0.1 - - [19/Jun/2022 20:54:11] "POST / HTTP/1.1" 200 -
1/1 [==============================] - 0s 77ms/step
INFO:werkzeug:127.0.0.1 - - [19/Jun/2022 20:54:19] "POST / HTTP/1.1" 200 -
```

● ✕