Project Development Phase

Delivery of Sprint – 4

Date	19 November 2022
Team ID	PNT2022TMID18067
Project Name	Web Phishing Detection

Final product which differentiates phishing and legitimate site:

Source Code:

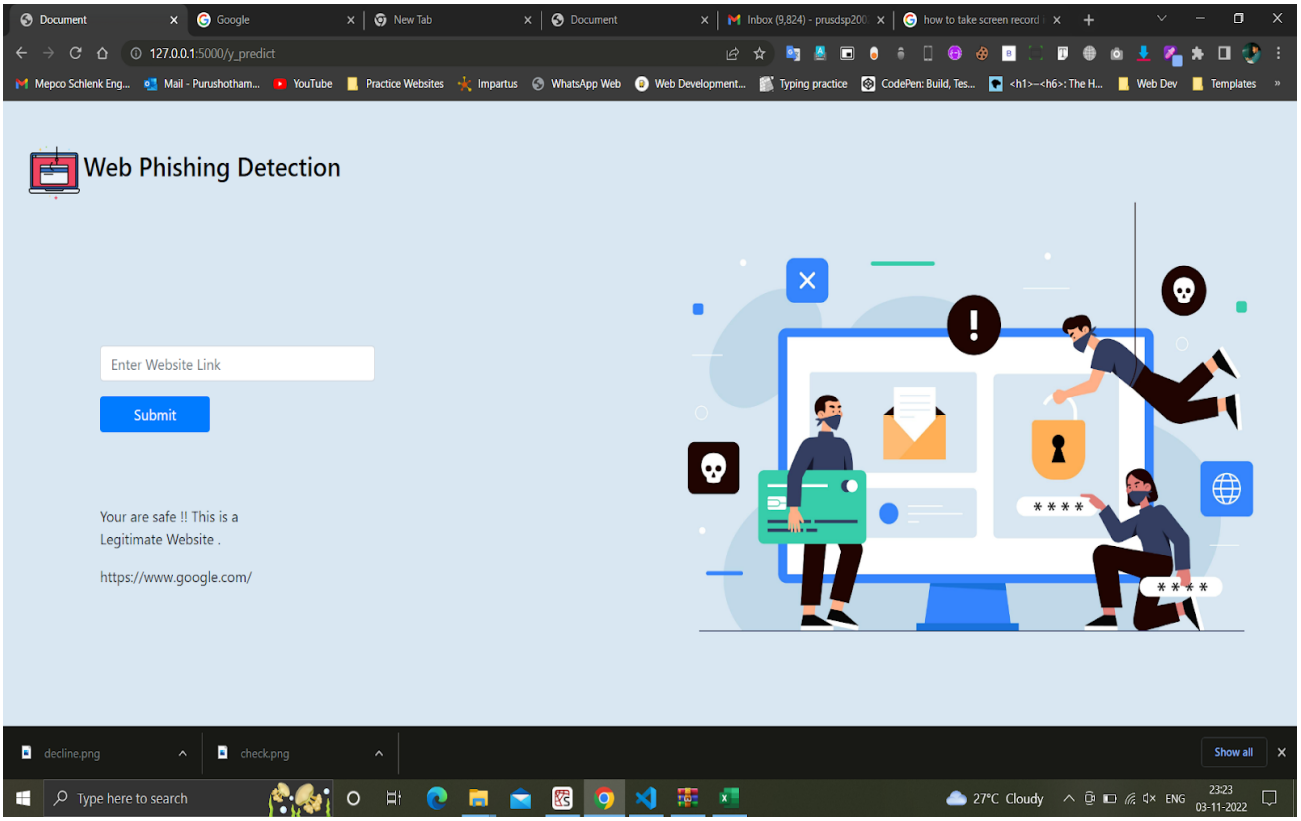
```
import requests
API_KEY = "p0TDrnvV_e4AAWxKBpT5JKScRH_exx1tDXyfMl0dIyEg"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

import new_input
import numpy as np
from flask import Flask , request , jsonify , render_template
import pickle
from feature import FeatureExtraction
app = Flask(__name__,template_folder='template')
model = pickle.load(open('web_phishing_detector.pkl','rb'))
```

```
@app.route ('/predict')
def predict():
   return render_template ('final.html')
#Fetches the URL given by the URL and passes to inputScript
@app.route ( '/y_predict' , methods = ['POST'])
def y_predict ( ) :
   print("In Y_Predict")
   url = request.form [ 'URL' ]
   obj = FeatureExtraction(url)
   x = obj.getFeaturesList()

   payload_scoring = {"input_data": [{"fields": [['having_IPhaving_IP_Address', 'URLURL_Length',
'Shortining_Service',
      'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix',
      'having_Sub_Domain', 'SSLfinal_State', 'Domain_registeration_length',
      'Favicon', 'port', 'HTTPS_token', 'Request_URL', 'URL_of_Anchor',
      'Links_in_tags', 'SFH', 'Submitting_to_email', 'Abnormal_URL',
      'Redirect', 'on_mouseover', 'RightClick', 'popUpWidnow', 'Iframe',
      'age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank',
      'Google_Index', 'Links_pointing_to_page', 'Statistical_report']], "values": [x]}]}

   response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/c1b4c9ee-fecf-4bff-
970c-cfc379ffa2e4/predictions?version=2022-11-03', json=payload_scoring,
   headers={'Authorization': 'Bearer ' + mltoken})
   print("Scoring response")
   pred=response_scoring.json()
   output = pred['predictions'][0]['values'][0][0]
   print(output)
   #prediction = model.predict ( x )
   #print ( prediction )
   #output = prediction [ 0 ]
   print(output)
   if ( output == 1 ) :
      pred = " Your are safe !! This is a Legitimate Website . "
   else :
      pred = " You are on the wrong site . Be cautious ! "
   return render_template ( 'final.html' , prediction_text = pred , url = url )
#Takes the input parameters fetched from the URL by inputScript and returns the predictions
@app.route ( '/predict_api' , methods = [ ' POST ' ] )
def predict_api ( ) :
```

```
    data = request.get_json ( force = True )
    prediction = model.y_predict ( [ np.array ( list ( data.values ( ) ) ) ] )
    output = prediction [ 0 ]
    return jsonify ( output )
if __name__=='__main__':

    app.run ("127.0.0.1",5000)
```

Output:

The system predicts www.google.com as legitimate site.



The system predicts smilesvoegel.servebbs.org/voegol.php as phishing site.