Coding and Solution

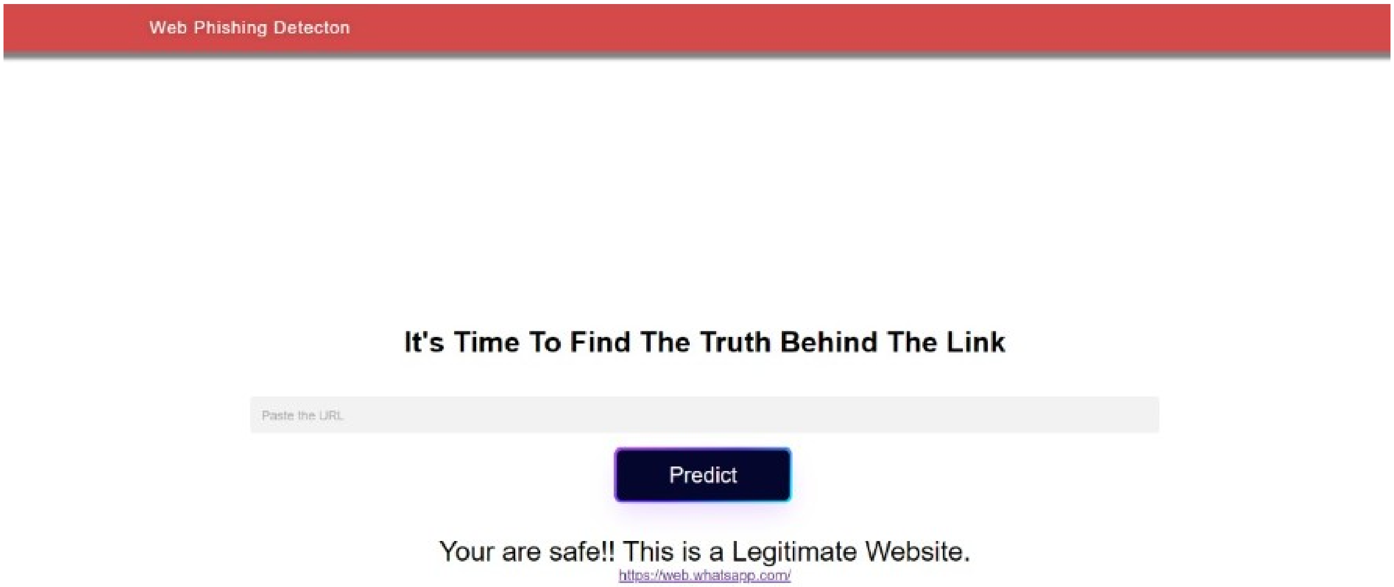
Coding:

```
 import numpy as np

from flask import Flask, request, jsonify, render_template

import pickle

#importing the inputScript file used to analyze the URL

import inputScript

import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM
Cloud account.

API_KEY = "nIRKSVDmk9sXH4oW1LtPgRbeaJMA8x0qJLtH2WFTt24L"

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}

#load model

app = Flask(_name_)

model = pickle.load(open('Phishing_Website.pkl', 'rb'))


@app.route('/')

def predict1():

    return render_template('index.html')

#Redirects to the page to give the user iput URL.
```

```
@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():

    '''

    For rendering results on HTML GUI

    '''

    url = request.form['URL']

    checkprediction = inputScript.main(url)

    scoring = {"input_data": [{"field":
[["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//","PrefixSuffix-
","SubDomains","HTTPS","DomainRegLen","Favicon","NonStdPort","HTTPSDomainURL","
RequestURL","AnchorURL","LinksInScriptTags","ServerFormHandler","InfoEmail","Abnorma
lURL","WebsiteForwarding","StatusBarCust","DisableRightClick","UsingPopupWindow","Ifra
meRedirection","AgeofDomain","DNSRecording","WebsiteTraffic","PageRank","GoogleIndex"
,"LinksPointingToPage","StatsReport"

                                    ]], "values": checkprediction}]}

    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/bf163e78-832a-470d-894f-
f7b8fbe4ac0d/predictions?version=2022-11-18', json=scoring,

    headers={'Authorization': 'Bearer ' + mltoken})

    print("Scoring response")

    predictions = response_scoring.json()
    pred =

    predictions['predictions'][0]['values'][0][0]

    output=pred

    if(pred==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='{}'.format(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():

    '''

    For direct API calls trought

    request '''

    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(debug=True)

if _name_ == '_main_':
```

```
app.run(host='0.0.0.0',

debug=True)
```

Solution:



It's Time To Find The Truth Behind The Link

Your are safe!! This is a Legitimate Website.

https://web.whatsapp.com/