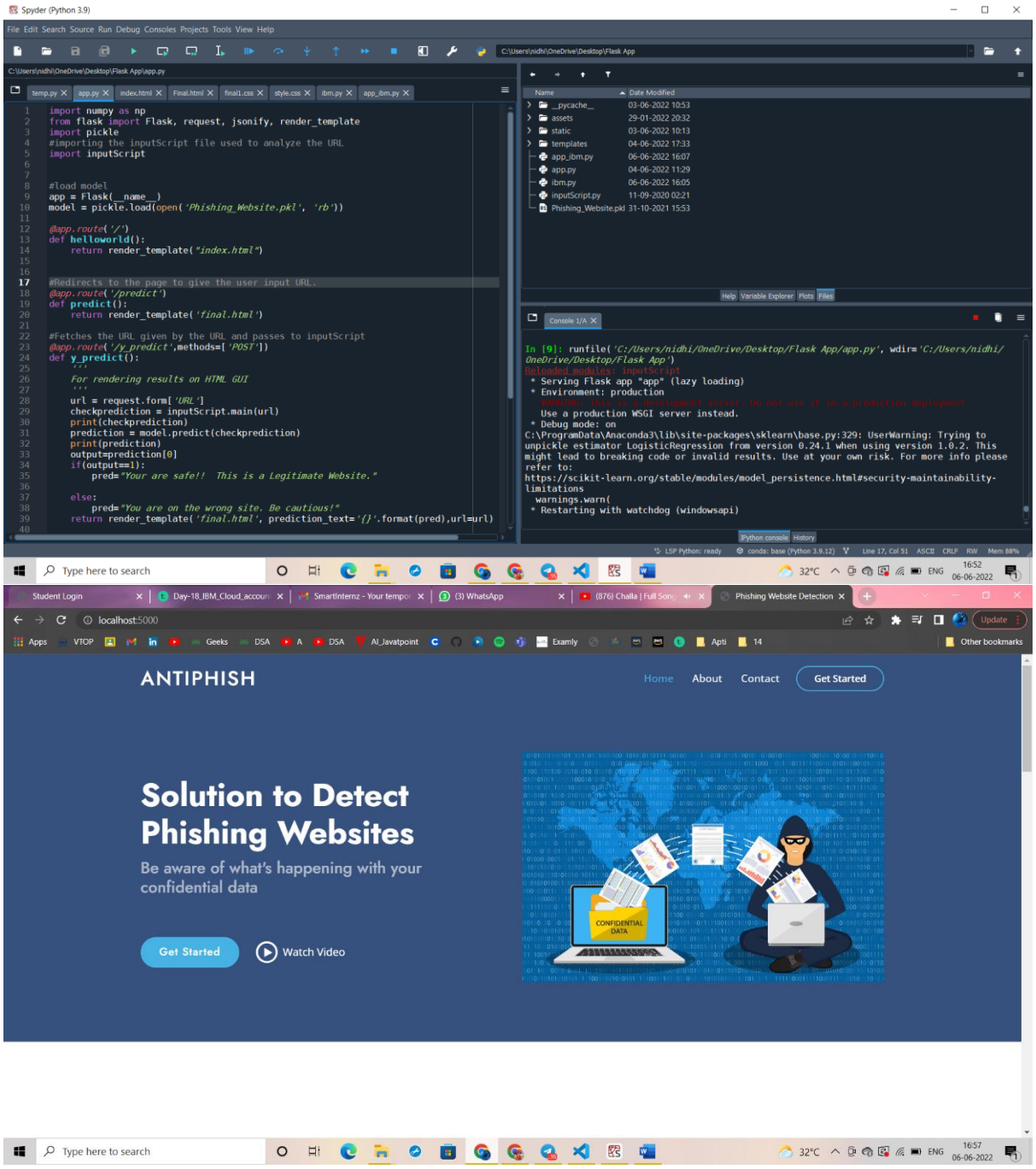
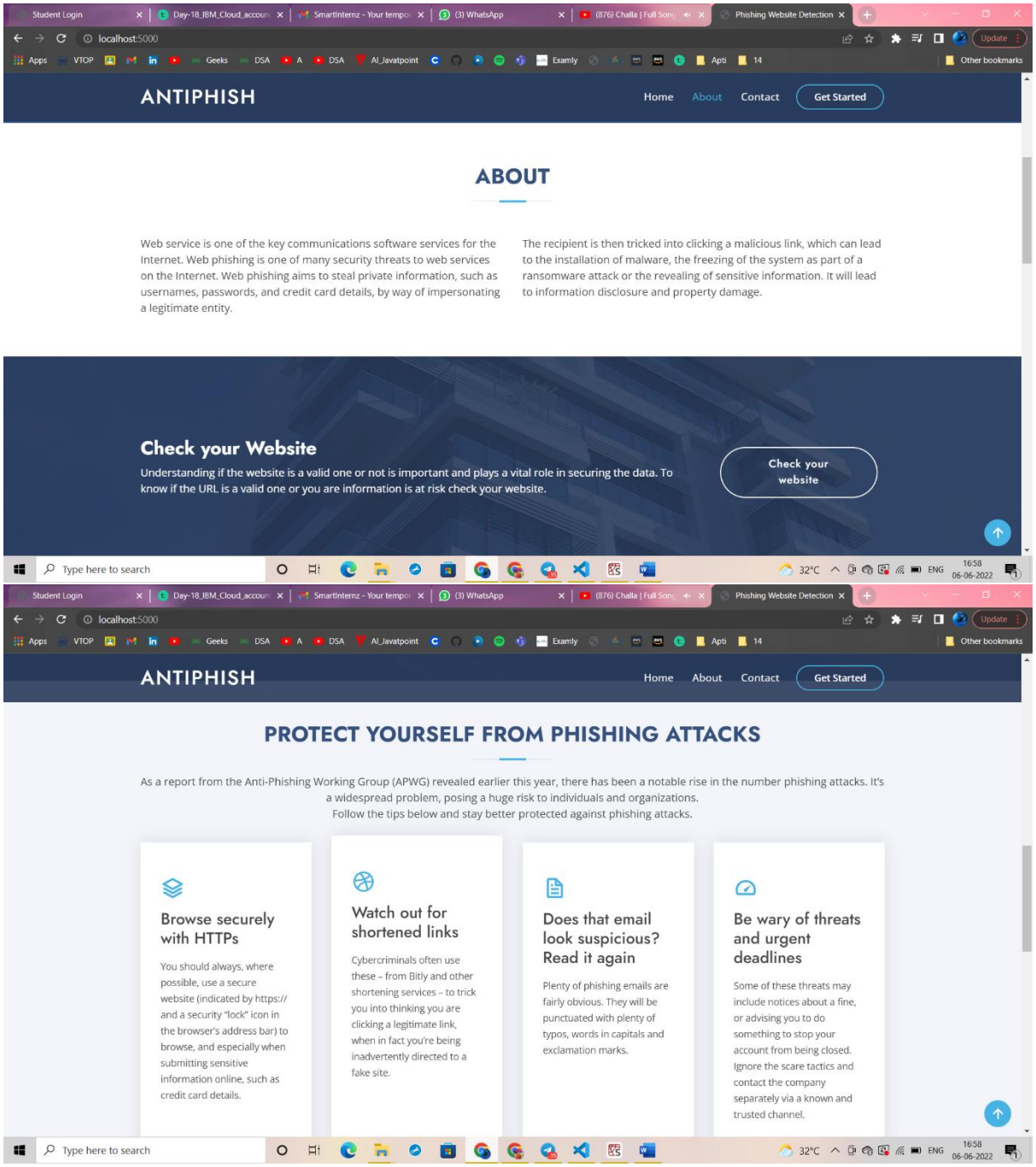
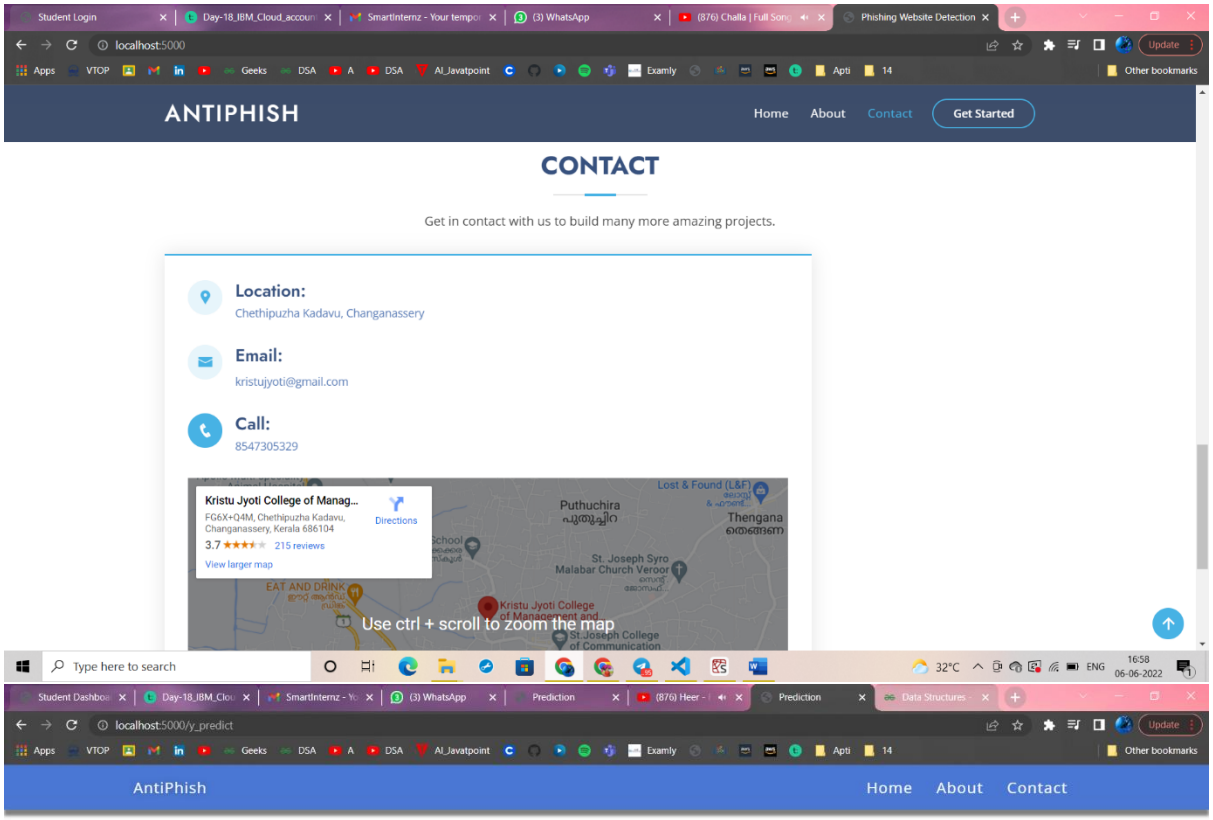
Detection of Phishing Websites from URLs using IBM Watson Studio

Output Screenshots:




Student Login
Phishing Website Detection
localhost:5000
ANTIPHISH
Home
About
Contact
Get Started
ABOUT
Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet. Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity.
The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information. It will lead to information disclosure and property damage.
Check your Website
Understanding if the website is a valid one or not is important and plays a vital role in securing the data. To know if the URL is a valid one or you are information is at risk check your website.
Check your website
Type here to search
32°C
ENG
16:58
06-06-2022

Student Login
Phishing Website Detection
localhost:5000
ANTIPHISH
Home
About
Contact
Get Started
PROTECT YOURSELF FROM PHISHING ATTACKS
As a report from the Anti-Phishing Working Group (APWG) revealed earlier this year, there has been a notable rise in the number phishing attacks. It's a widespread problem, posing a huge risk to individuals and organizations.
Follow the tips below and stay better protected against phishing attacks.
Browse securely with HTTPs
You should always, where possible, use a secure website (indicated by https:// and a security "lock" icon in the browser's address bar) to browse, and especially when submitting sensitive information online, such as credit card details.
Watch out for shortened links
Cybercriminals often use these – from Bitly and other shortening services – to trick you into thinking you are clicking a legitimate link, when in fact you're being inadvertently directed to a fake site.
Does that email look suspicious? Read it again
Plenty of phishing emails are fairly obvious. They will be punctuated with plenty of typos, words in capitals and exclamation marks.
Be wary of threats and urgent deadlines
Some of these threats may include notices about a fine, or advising you to do something to stop your account from being closed. Ignore the scare tactics and contact the company separately via a known and trusted channel.
Type here to search
32°C
ENG
16:58
06-06-2022



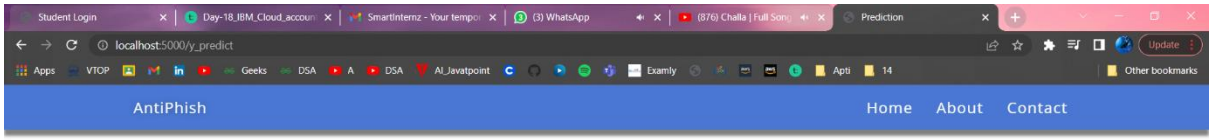
Phishing Website Detection using Machine Learning

Enter the URL to be verified

Predict

Your are safe!! This is a Legitimate Website.

https://www.geeksforgeeks.org/data-structures/?ref=shm



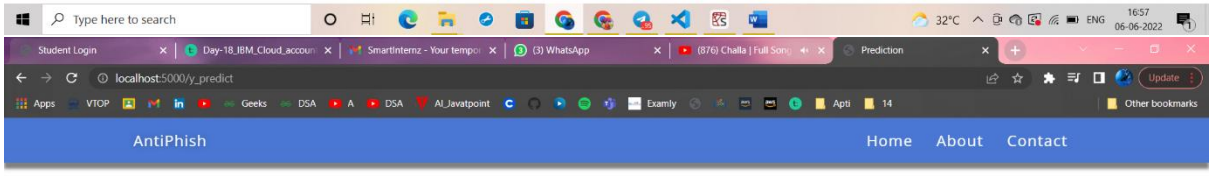
Phishing Website Detection using Machine Learning

Enter the URL to be verified

Predict

Your are safe!! This is a Legitimate Website.

https://smartinternz.com/student-login



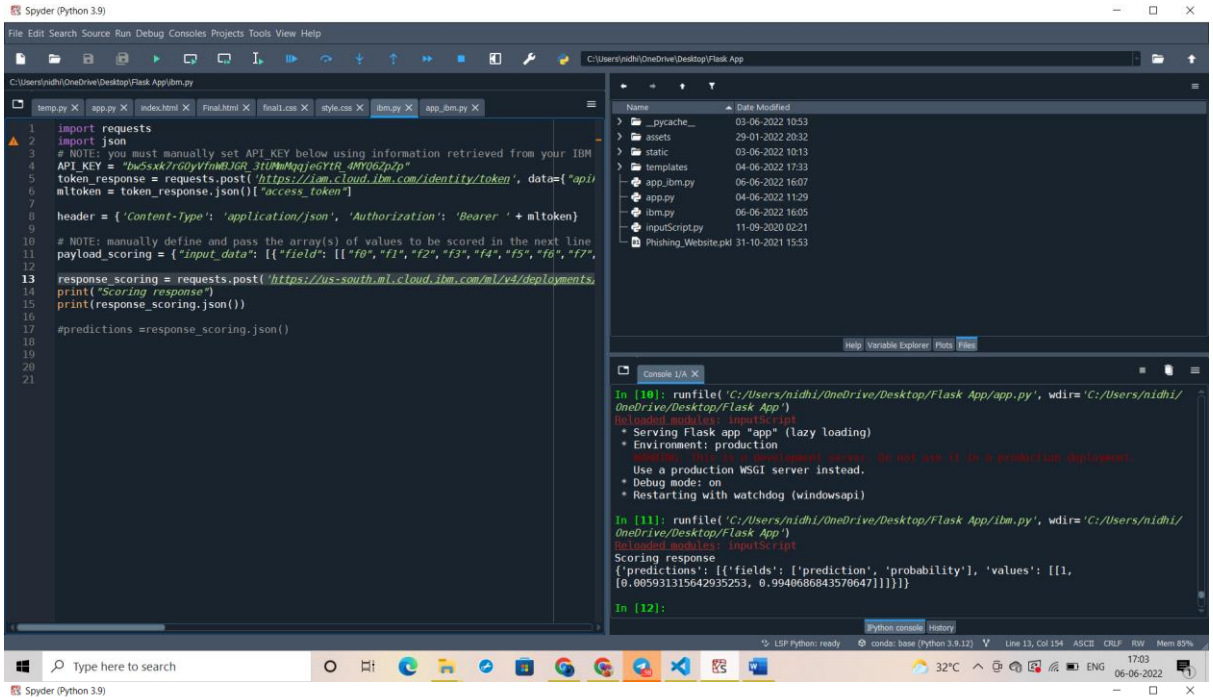
Phishing Website Detection using Machine Learning

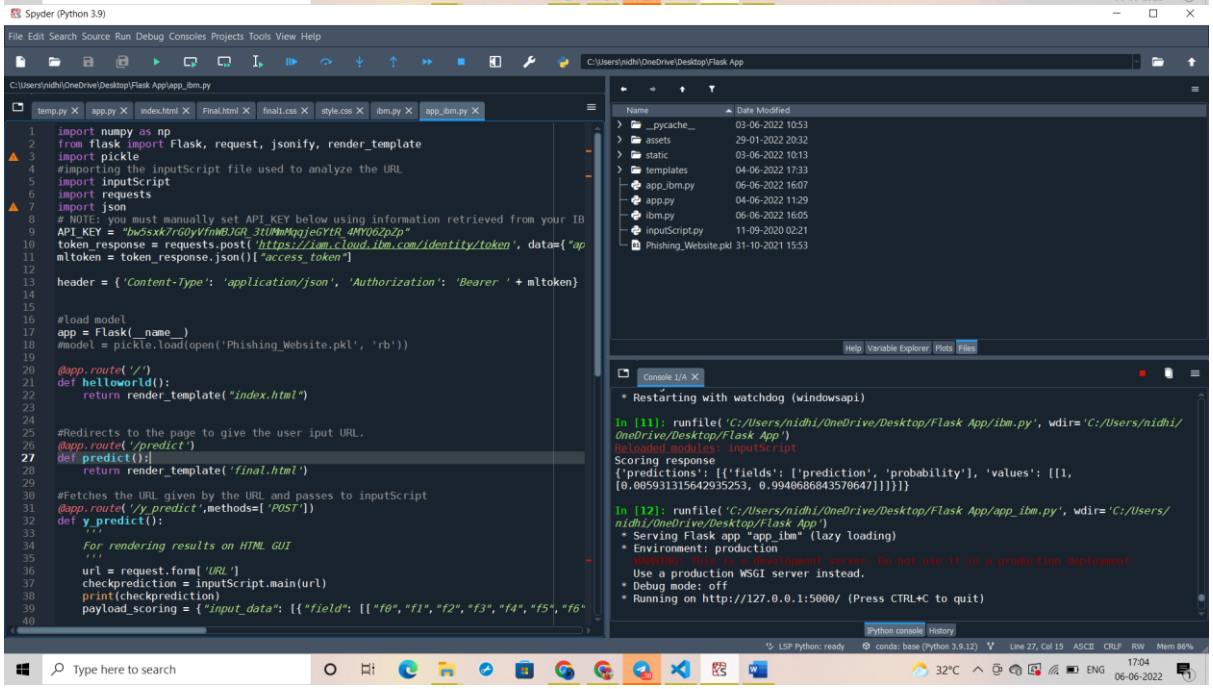
Enter the URL to be verified

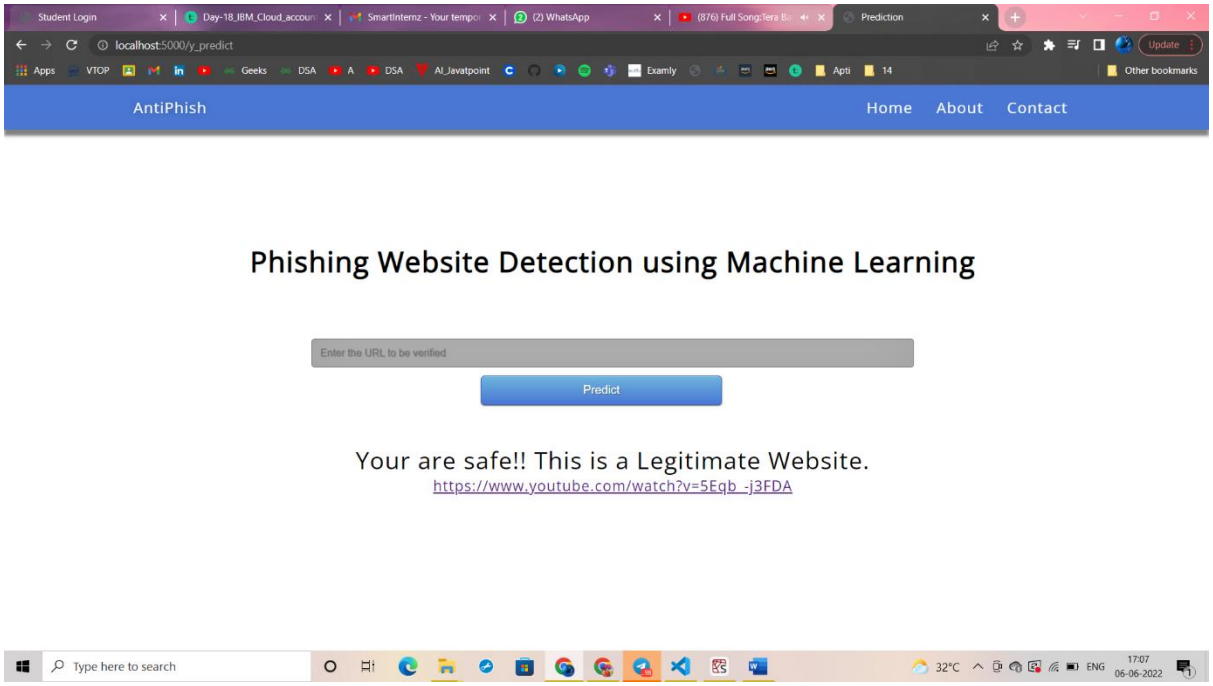
Predict

You are on the wrong site. Be cautious!

https://smartintenz.com/student-login


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\nidhi\OneDrive\Desktop\Flask App
C:\Users\nidhi\OneDrive\Desktop\Flask App\ibm.py
temp.py app.py index.html Final.html final1.css style.css ibm.py app_ibm.py
import requests
import json
# NOTE: you must manually set API_KEY below using information retrieved from your IBM
API_KEY = "bw5sxk7rGOyVfnWBJGR_3tUMmMqqjeGYtR_4MYQ6ZpZp"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apik
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field": [["f0","f1","f2","f3","f4","f5","f6","f7",
response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments,
print("Scoring response")
print(response_scoring.json())
#predictions =response_scoring.json()
Name Date Modified
__pycache__ 03-06-2022 10:53
assets 29-01-2022 20:32
static 03-06-2022 10:13
templates 04-06-2022 17:33
app_ibm.py 06-06-2022 16:07
app.py 04-06-2022 11:29
ibm.py 06-06-2022 16:05
inputScript.py 11-09-2020 02:21
Phishing_Website.pkl 31-10-2021 15:53
Help Variable Explorer Plots Files
Console 1/A
In [10]: runfile('C:/Users/nidhi/OneDrive/Desktop/Flask App/app.py', wdir='C:/Users/nidhi/OneDrive/Desktop/Flask App')
Reloaded modules: inputScript
* 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
* Restarting with watchdog (windowsapi)
In [11]: runfile('C:/Users/nidhi/OneDrive/Desktop/Flask App/ibm.py', wdir='C:/Users/nidhi/OneDrive/Desktop/Flask App')
Reloaded modules: inputScript
Scoring response
{'predictions': [{'fields': ['prediction', 'probability'], 'values': [[1, [0.005931315642935253, 0.9940686843570647]]]}]}
In [12]:
Ipython console History
LSP Python: ready conda: base (Python 3.9.12) Line 13, Col 154 ASCII CRLF RW Mem 85%
Type here to search
32°C ENG 17:03 06-06-2022


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\nidhi\OneDrive\Desktop\Flask App
C:\Users\nidhi\OneDrive\Desktop\Flask App\app_ibm.py
temp.py app.py index.html Final.html final1.css style.css ibm.py app_ibm.py
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
import json
# NOTE: you must manually set API_KEY below using information retrieved from your IB
API_KEY = "bw5sxk7rGOyVfnWBJGR_3tUMmMqqjeGYtR_4MYQ6ZpZp"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"ap
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 helloworld():
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)
print(checkprediction)
payload_scoring = {"input_data": [{"field": [["f0","f1","f2","f3","f4","f5","f6"
Name Date Modified
__pycache__ 03-06-2022 10:53
assets 29-01-2022 20:32
static 03-06-2022 10:13
templates 04-06-2022 17:33
app_ibm.py 06-06-2022 16:07
app.py 04-06-2022 11:29
ibm.py 06-06-2022 16:05
inputScript.py 11-09-2020 02:21
Phishing_Website.pkl 31-10-2021 15:53
Help Variable Explorer Plots Files
Console 1/A
* Restarting with watchdog (windowsapi)
In [11]: runfile('C:/Users/nidhi/OneDrive/Desktop/Flask App/ibm.py', wdir='C:/Users/nidhi/OneDrive/Desktop/Flask App')
Reloaded modules: inputScript
Scoring response
{'predictions': [{'fields': ['prediction', 'probability'], 'values': [[1, [0.005931315642935253, 0.9940686843570647]]]}]}
In [12]: runfile('C:/Users/nidhi/OneDrive/Desktop/Flask App/app_ibm.py', wdir='C:/Users/nidhi/OneDrive/Desktop/Flask App')
* Serving Flask app "app_ibm" (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
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Ipython console History
LSP Python: ready conda: base (Python 3.9.12) Line 27, Col 15 ASCII CRLF RW Mem 86%
Type here to search
32°C ENG 17:04 06-06-2022



IBM Deployment output: