

# Detection of Phishing Websites from URLs using IBM Watson Studio

## Output Screenshots:

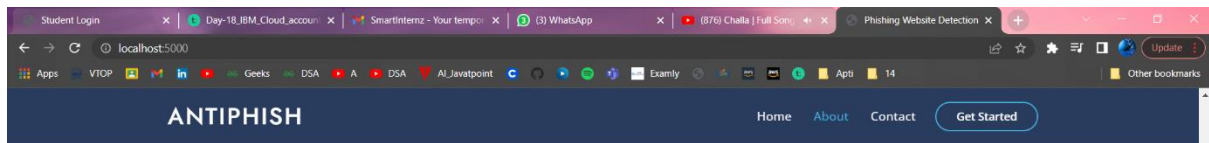
The screenshot displays the Spyder Python IDE interface. The left pane shows the code for a Flask application named 'app.py'. The code imports necessary libraries, loads a pre-trained model from a file named 'Phishing\_Website.pkl', and defines routes for a home page and a prediction endpoint. The prediction endpoint uses the loaded model to predict if a given URL is a phishing website. The right pane shows the file explorer with the project structure. The bottom pane shows the console output, which includes the Flask app running on port 5000 and the prediction results for a test URL.

```
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4 #importing the inputScript file used to analyze the URL
5 import inputScript
6
7
8 #load model
9 app = Flask(__name__)
10 model = pickle.load(open('Phishing_Website.pkl', 'rb'))
11
12 @app.route('/')
13 def helloworld():
14     return render_template("index.html")
15
16
17 #Redirects to the page to give the user input URL.
18 @app.route('/predict')
19 def predict():
20     return render_template('final.html')
21
22 #Fetches the URL given by the URL and passes to inputScript
23 @app.route('/y_predict', methods=['POST'])
24 def y_predict():
25     """
26     For rendering results on HTML GUI
27     """
28     url = request.form['URL']
29     checkprediction = inputScript.main(url)
30     print(checkprediction)
31     prediction = model.predict(checkprediction)
32     print(prediction)
33     output=prediction[0]
34     if(output==1):
35         pred="You are safe!! This is a Legitimate Website."
36     else:
37         pred="You are on the wrong site. Be cautious!"
38     return render_template('final.html', prediction_text='{0}'.format(pred),url=url)
```

The console output shows the following messages:

```
In [9]: runfile('C:/Users/nidhi/OneDrive/Desktop/Flask App/app.py', wdir='C:/Users/nidhi/OneDrive/Desktop/Flask App')
* 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
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator LogisticRegression from version 0.24.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* Restarting with watchdog (windowsapi)
```

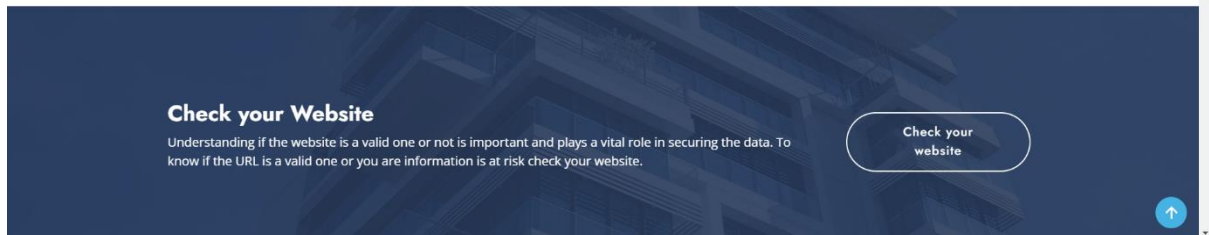
The web browser shows the 'ANTIPHISH' website, which is a solution to detect phishing websites. The page features a large heading 'Solution to Detect Phishing Websites' and a subheading 'Be aware of what's happening with your confidential data'. There are buttons for 'Get Started' and 'Watch Video'. The background of the page is a dark blue with a pattern of binary code and a central illustration of a person in a hoodie sitting at a desk with a laptop, surrounded by floating documents and a 'CONFIDENTIAL DATA' label.



## 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.



## 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

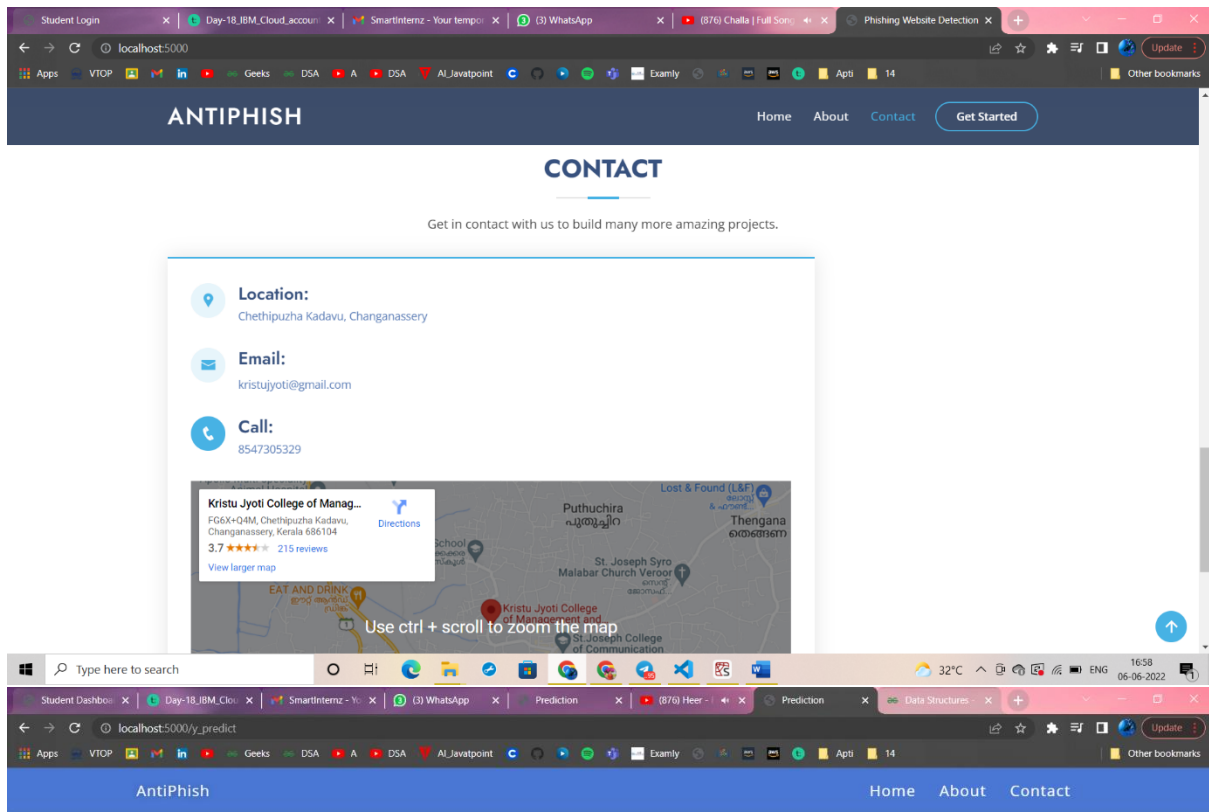
Plenty of phishing emails are fairly obvious. They will be punctuated with plenty of typos, words in capitals and exclamation marks.



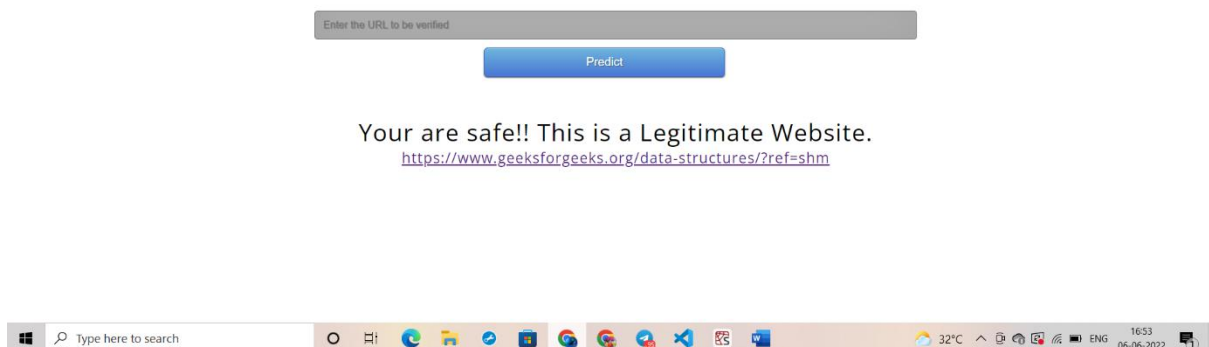
### Be wary of threats and urgent deadlines

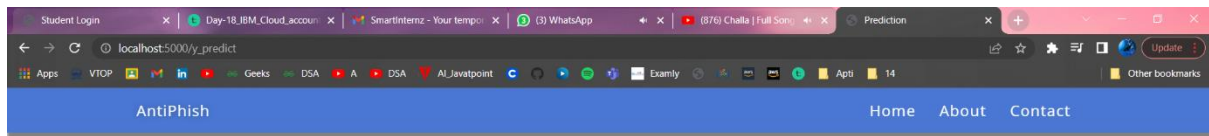
Some of these threats may include notices about a fine, or advising you to do something to stop your account from being closed. Ignore the scare tactics and contact the company separately via a known and trusted channel.





## Phishing Website Detection using Machine Learning





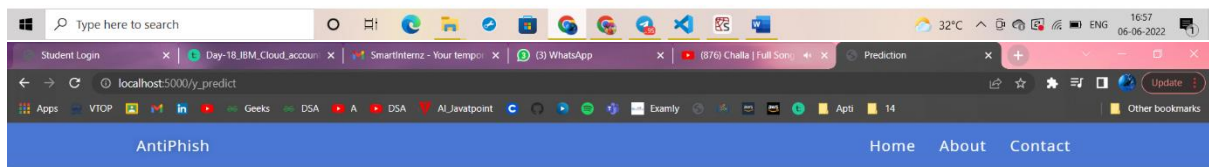
## Phishing Website Detection using Machine Learning

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://smartinternz.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

```
1 import requests
2 import json
3 # NOTE: you must manually set API KEY below using information retrieved from your IBM
4 API_KEY = "bW5sXk7rG0yVfmWBJGR_3tUMmMqjeGYR_4MY06Zp2p"
5 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"api_
6 mtoken = token_response.json()["access_token"]
7
8 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mtoken}
9
10 # NOTE: manually define and pass the array(s) of values to be scored in the next line
11 payload_scoring = {"input_data": [{"field": [{"f0", "f1", "f2", "f3", "f4", "f5", "f6", "f7",
12
13 response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments
14 print('Scoring response')
15 print(response_scoring.json())
16
17 #predictions = response_scoring.json()
18
19
20
21
```

File Explorer

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 I/A X

```
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
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, 0.9940686843578647]]]]}]

In [12]:
```

Python 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 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

```
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4 #importing the inputScript file used to analyze the URL
5 import inputScript
6 import requests
7 import json
8 # NOTE: you must manually set API KEY below using information retrieved from your IB
9 API_KEY = "bW5sXk7rG0yVfmWBJGR_3tUMmMqjeGYR_4MY06Zp2p"
10 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"ap
11 mtoken = token_response.json()["access_token"]
12
13 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mtoken}
14
15
16 #load model
17 app = Flask(__name__)
18 #model = pickle.load(open('Phishing_Website.pkl', 'rb'))
19
20 @app.route('/')
21 def helloworld():
22     return render_template("index.html")
23
24
25 #Redirects to the page to give the user input URL.
26 @app.route('/predict')
27 def predict():
28     return render_template('final.html')
29
30 #Fetches the URL given by the URL and passes to inputScript
31 @app.route('/y_predict', methods=['POST'])
32 def y_predict():
33     """
34     For rendering results on HTML GUI
35     """
36     url = request.form['URL']
37     checkprediction = inputScript.main(url)
38     print(checkprediction)
39     payload_scoring = {"input_data": [{"field": [{"f0", "f1", "f2", "f3", "f4", "f5", "f6"
40
```

File Explorer

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 I/A X

```
* 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, 0.9940686843578647]]]]}]

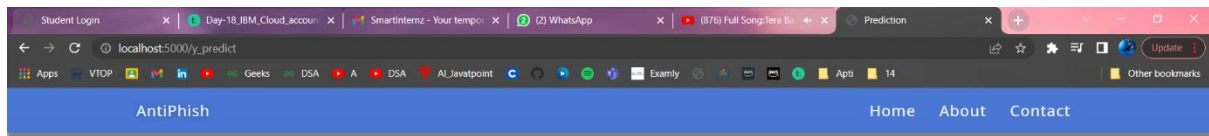
In [12]: runfile('C:/Users/nidhi/OneDrive/Desktop/Flask App/app_ibm.py', wdir='C:/Users/
nidhi/OneDrive/Desktop/Flask App')
* Serving Flask app "app_ibm" (lazy loading)
* Environment: production
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Python 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 17:04 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.youtube.com/watch?v=5Eqb-j3FDA>



## IBM Deployment output:

IBM Watson Studio

Search in your workspaces

Buy

SRUSHTI SHINTRE's Acco...

Deployments / Phishing\_site\_detection\_deploy... / Phishing\_Website\_Detection /

Smart\_Internz\_Proj Deployed Online

API reference **Test**

Enter input data

Body

```
{
  "input_data": [
    {
      "field": [
        "f0", "f1", "f2", "f3", "f4", "f5", "f6", "f7", "f8", "f9", "f10", "f11", "f12", "f13", "f14",
        "f15", "f16", "f17", "f18", "f19", "f20", "f21", "f22", "f23", "f24", "f25", "f26", "f27", "f28", "f29"
      ],
      "values": [
        [1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, 0, -1, -1, 1, 1, 1, 1, 1, -1, 1, 1, -1, 1, 1]
      ]
    }
  ]
}
```

Predict

Result

```
{
  "predictions": [
    {
      "fields": [
        "prediction",
        "probability"
      ],
      "values": [
        [
          1,
          [
            0.005931315642935253,
            0.9940686843570647
          ]
        ]
      ]
    }
  ]
}
```