WEB PHISHING DETECTION

1. INTRODUCTION

1.1 Project overview

Phishing is a common attack on credulous people by making them to disclose their unique information using counterfeit websites. The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions. Phishers use the websites which are visually and semantically similar to those real websites. As technology continues to grow, phishing techniques started to progress rapidly and this needs to be prevented by using anti-phishing mechanisms to detect phishing. Machine learning is a powerful tool used to strive against phishing attacks. This paper surveys the features used for detection and detection techniques using machine learning.

1.2 Purpose

There are a number of users who purchase products online and make payments through various websites. Many websites ask a user to provide sensitive data such as username, password or credit card details etc. often for malicious reasons. This type of website is known as a phishing website. In order to detect and predict phishing websites, we propose an intelligent, flexible and effective system based on a classification Machine Learning algorithm. We implement classification algorithms and techniques to extract the criteria for phishing data sets to classify their legitimacy. The phishing website can be detected based on some important characteristics like URL and Domain Identity, and security and encryption criteria in the final phishing detection rate. Once a user makes an online transaction, he makes payment through the website. Our system will use a Machine Learning algorithm to detect whether the website is a phishing website or not. This application can be used by many E-commerce enterprises in order to make the whole transaction process secure. With the help of this system, users can also purchase products online without any hesitation. Admin can add phishing website URL or fake website URL into system where system could access and scan the phishing website and by using algorithm, it will add new suspicious keywords to uses machine learning technique.

2. LITERATURE SURVEY

2.1 Existing Problem

Phishing is a typical type of social designing assault intended to gather client data, for example, login certifications and Visa data. At the point when a casualty opens an email, text, or instant message subsequent to being hoodwinked into doing as such by a culprit acting like a dependable source, it happens. The beneficiary is in this manner fooled into clicking a hazardous connection, which might introduce malware, lock the framework as a feature of a ransomware assault, or uncover private data.

The foundation subtleties of a casualty's private and expert history might be assembled by phishers utilizing open sources, especially informal communities. The names, occupations, email locations, and interests and diversions of the potential casualty are completely assembled from these sources. When this data is gotten, the phisher can use it to make a reliable fake message.

2.2 Reference

[1]. "Protecting Users Against Phishing Attacks with AntiPhish" Engin Kirda and Christopher Kruegel Technical University of Vienna

[2]. "Learning to Detect Phishing Emails" Ian Fette School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA icf@cs.cmu.edu Norman Sadeh School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA Anthony Tomasic School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA

[3]. Modeling and Preventing Phishing Attacks by Markus Jakobsson, Phishing detection system for e -banking using fuzzy data mining by Aburrous, M. ; Dept. of Comput., Univ. of Bradford, Bradford, UK ; Hossain, M.A. ; Dahal, K. ; Thabatah, F.

[4]. M. Chandrasekaran, et al., "Phishing email detection based on structural properties", in New York State Cyber Security Conference (NYS) , Albany, NY ," 2006

[5]. P. R. a. D. L. Ganger, ”Gone phishing: Evaluating anti-phishing tools for windows. Technical report, ,” September 2006

[6]. M. Bazarganigilani, ”Phishing E-Mail Detection Using Ontology Concept and Nave Bayes Algorithm,” International Journal of Research and Reviews in Computer Science, vol. 2,no.2, 2011.

[7]. M. Chandrasekaran, et al., ”Phoney: Mimicking user response to detect phishing attacks,” in In: Symposium on World of Wireless, Mobile and Multimedia Networks, IEEE Computer Society, 2006, pp. 668-672

[8]. I. Fette, et al., ”Learning to detect phishing emails,” in Proc. 16th International World Wide Web Conference (WWW 2007), ACM Press, New York, NY, USA, May 2007, pp. 649-656

[9]. A. Bergholz, et al., ”Improved phishing detection using model-based features,” in Proc. Conference on Email and Anti-Spam (CEAS). Mountain View Conf, CA, aug 2008

[10]. L. Ma, et al.,”Detecting phishing emails using hybrid features,”IEEE Conf, 2009, pp. 493-497

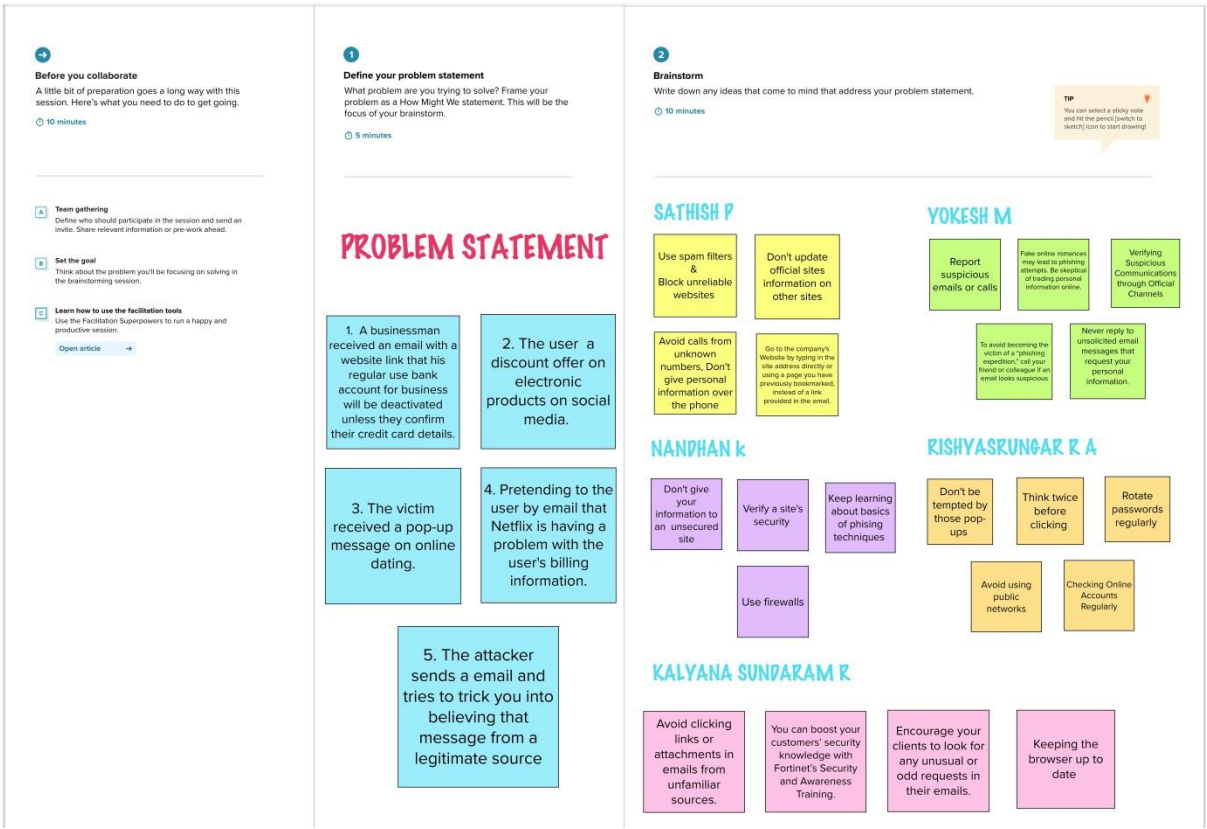
2.3 Problem Statement Definition

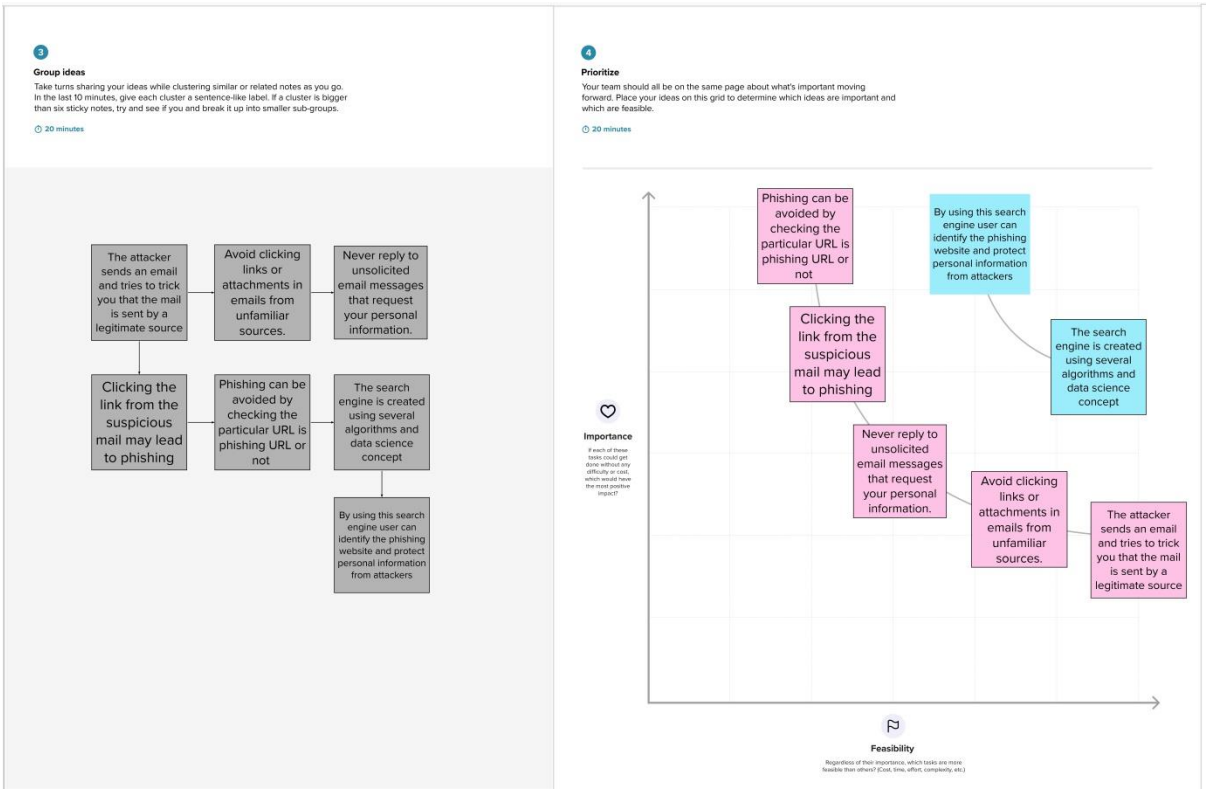
Phishing is a type of computer attack that communicates socially engineered messages to humans via electronic communication channels in order to persuade them to perform certain actions for the attacker’s benefit. For example, the performed action (which the attacker persuades the victim to perform it) for a PayPal user is submitting his/her login credentials to a fake website that looks similar to PayPal. As a perquisite, This also implies that the attack should create a need for the end-user to perform such action, such as informing him that his/her account would be suspended unless he logs in to update certain pieces of information.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map

3.2 Ideation & Brainstorming





3.3 Proposed Solution

Project team shall fill the followinginformation in proposedsolution template.

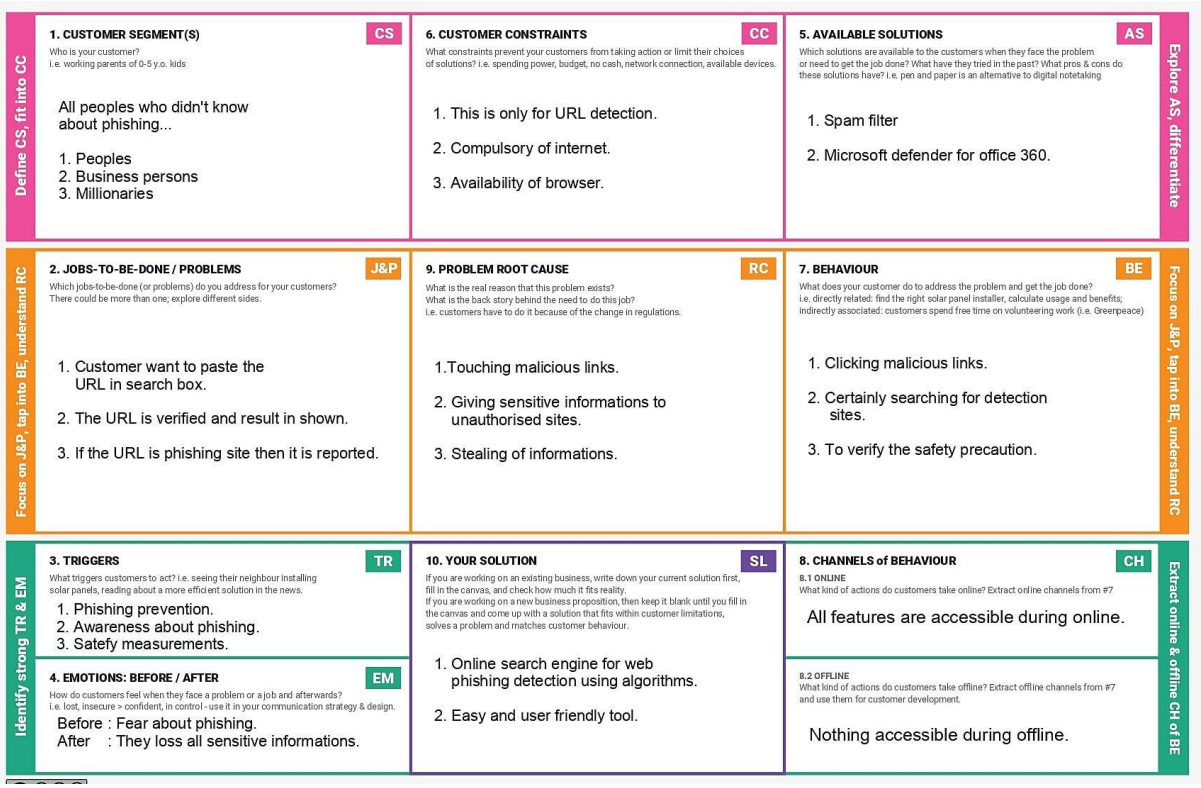
S. No.	Parameter	Description
1.	Problem Statement(Problemto be solved)	There are a number of users who purchase products online andmake payments through e-banking. Thereare e-banking websites that ask usersto provide sensitive data such as username, password & credit card details, etc.,often for malicious reasons. This type of e-banking website is known asa phishing website. Web serviceis one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet.

2.	Idea / Solutio ndescri ption	In order to detect and predict e-banking phishing websites, we proposed an intelligent, flexible and effective system that is based on using classification algorithms. We implemented classification algorithms and techniques to extract the phishingdatasets criteria to classify their legitimacy. The e-banking phishing website can be detected based on some important characteristics like URL and domain identity, and security andencryption criteria in the final phishing detection rate. Once a user makesa transaction onlinewhen he makespayment through an e-banking website our system will use a data mining algorithm to detect whether the e-banking website is a phishing website or not.
3.	Nov elty / Un ique ness	Machine learning technology consists of a many algorithms which requires past data to make a decision or prediction on future data.Using this technique, algorithm will analyzevario us blacklisted and legitimate URLs and their features to accurately detect the phishing websites including zero-hour phishing websites.
4.	Social I mpact/ Custom er Satis faction	Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage t o reputation, and disruption of operational activities. Example: • Facebook and Google.Between 2013 and 2015,Facebook and Google were tricked out of $100 million due to anextended phishing campaign... Customer Satisfaction: • By using our web phishing detection website the usercanch eck their websites by copy and paste the phishing URL. • After knowing the result they can be completely safe from above mentioned impacts.

5.	Business Model (Revenue Model)	As long as phishing websites continue to operate, many more people and companies will suffer privacy leaks or financial losses. Therefore, the demand for fast and accurate phishing website__detection grows stronger. However, theexisting phishing detection methods do not fully analyze the features of phishing, andthe performance and efficien cy of themodels only apply to certain limited datasets and need to be improved to be applied to the real web environment.

6.	Scalability of theSolution	Features are length of an URL, URL has HTTP, URL has suspicious character, prefix/suffix, number of dots, number of slashes, URLhas phishing term,length of subdomain, URL contains IP address

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	Checking URL	If the user suspected any of the URL might be a phishing URL.
FR-2	Copying URL	User can copy the suspected URL and paste in the Search Engine.
FR-3	URL Extraction	After pasting URL in the Search Engine, it can extract the all the information about URL.
FR-4	Data Processing	Search Engine compare the URL with given dataset byusingML algorithms like Logistic Regression and Decision trees.
FR-5	Predicating	Finally the Search Engine predict the result of given URL and showing negative and positive of the URL.

4.2 Non Functional requirement

Following are the non-functional requirements of the proposed solution.

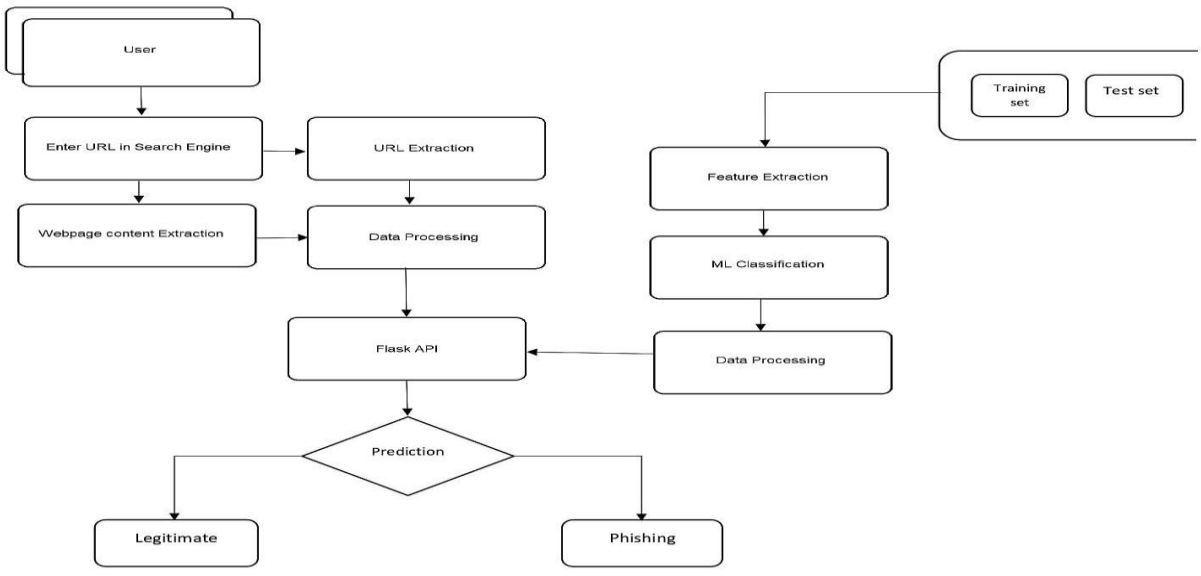
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user can easily understood our website there is no difficulties in finding the Search Engine.

NFR-2	**Security**	Our site is mainly provided for the security process only so there is no possibility for security issues and didn't ask any device permissions.
NFR-3	**Reliability**	All the data processing and prediction are hide to the end users. Showing the positive and negative of the result and it never predict wrongly.
NFR-4	**Performance**	While using dataset with python and ML algorithms it predict faster than using database with python.
NFR-6	**Scalability**	Many URL can be searched at a time by different Users.

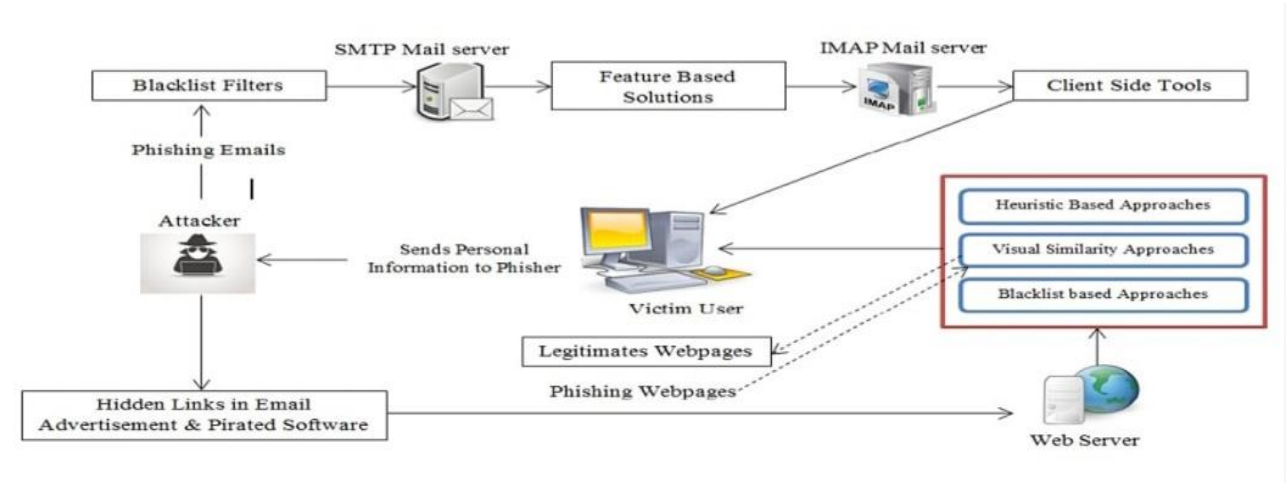
5. PROJECT DESIGN

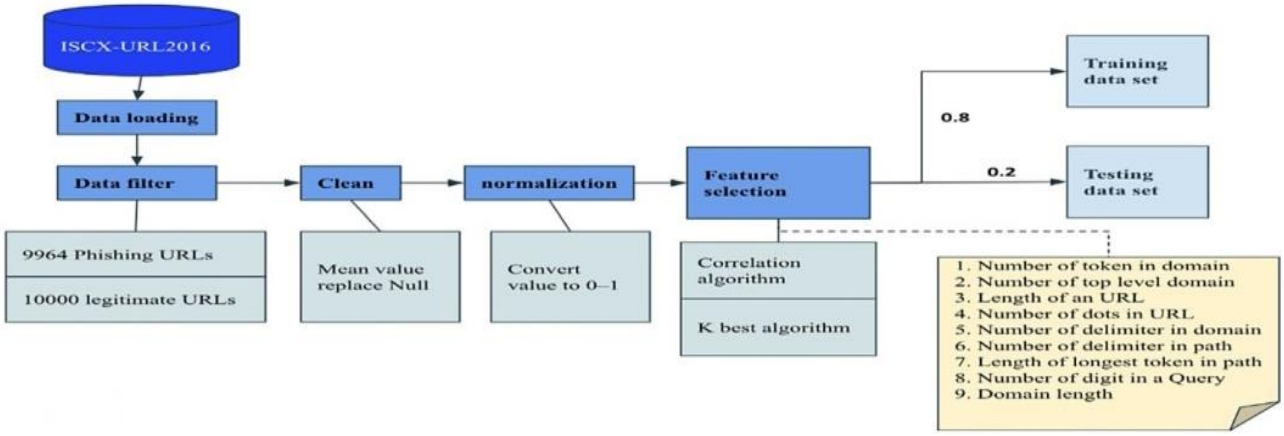
5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored



5.2 Solution & Technical Architecture





5.3 User Stories

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How Attackers distribute emails to user e.g.Messages, Mails etc.	HTML, CSS, JavaScript ,python etc.
2.	Database	Data Type, Configurations etc.	MySQL, No SQL, etc.
3.	Internal network	Data stolen from the user. Feature based solutions	IBM DB2,IBM Cloud ant etc.
4.	Data	Hacked informations stored in storage	IBM Block Storage or Other Storage

Table-2: Application Characteristics

1.	Scalable Architecture	Justify the scalability of architecture (3 – tier,Micro-services)	Technology used
2.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
3.	Performance	Designconsideration for the performance of theapplication (number of requests per sec, use ofCache, use of CDN's)etc.	Technology used

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 22	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 22	05 Nov 2022	20	04 Nov 2022
Sprint-3	20	6 Days	07 Nov 22	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 22	19 Nov 2022	20	18 Nov 2022

Velocity

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) periteration unit (story points per day)

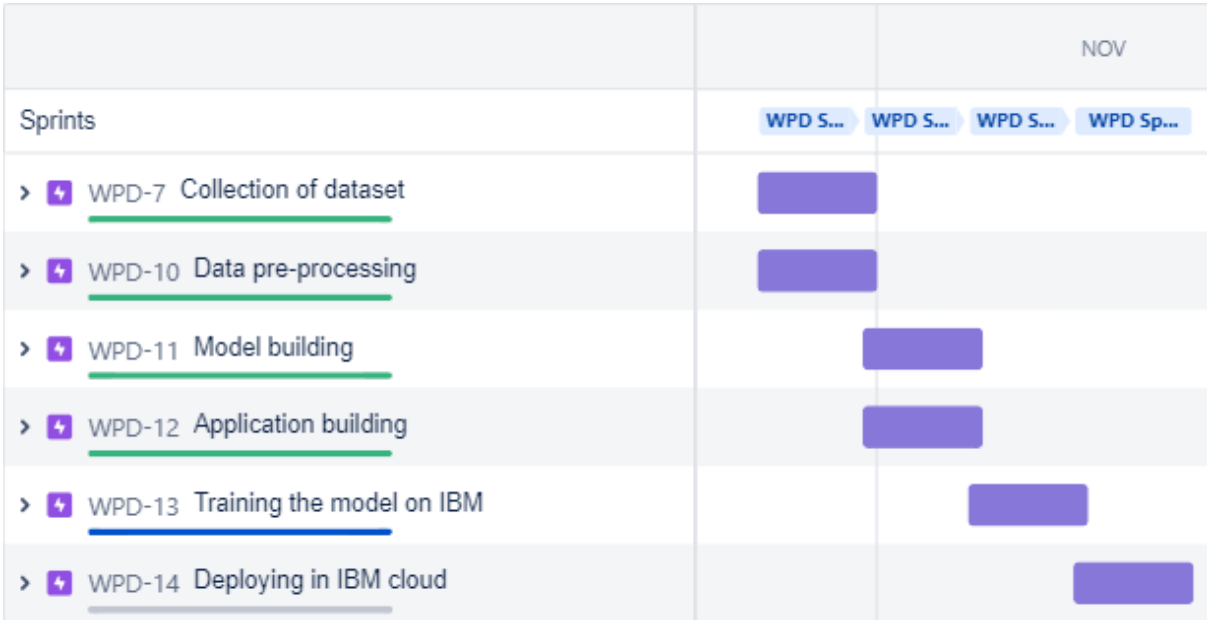
6.2 Sprint Delivery Schedule

Product Backlog, Sprint Schedule,and Estimation (4 Marks)

Use the below templateto create product backlogand sprint schedule

Sprint	Functional Requirement (Epic)	User StoryNumber	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collection of dataset	USN-1	Downloading dataset	1	High	RISHYASRUNGAR R A, NANDHAN K.
Sprint-1	Data pre-processing	USN-2	Data processing	2	Medium	SATHISH P,YOKESH M, KALYANA SUNDARAM R.
Sprint-2	Model building	USN-3	Dataset training andtesting	2	High	SATHISH P, YOKESH M, KALYANA SUNDARAM R.
Sprint-2	Application building	USN-4	Making API	2	Medium	SATHISH P, YOKESH M, KALYANA SUNDARAM R, RISHYASRUNGAR R A, NANDHAN K.
Sprint-3	Training the modelon IBM	USN-5	Predicting	1	High	SATHISH P, YOKESH M, KALYANA SUNDARAM R, RISHYASRUNGAR R A, NANDHAN K.
Sprint-4	Deploying in IBMcloud	USN-	Search Engine	2	High	SATHISH P, YOKESH M, KALYANA SUNDARAM R, RISHYASRUNGAR R A, NANDHAN K.

6.3 Report from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1

We have implemented python program to extract features from URL. Below are the features that we have extracted for detection of phishing URLs.

1) Presence of IP address in URL: If IP address present in URL then the feature is set to 1 else set to 0. Most of the benign sites do not use IP address as an URL to download a webpage. Use of IP address in URL indicates that the attacker is trying to steal sensitive Information.

2) Presence of @ symbol in URL: If @ symbol present in URL then the feature is set to 1 else set to 0. Phishers add a special symbol @ in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.

3) Number of dots in Hostname: Phishing URLs have any dots in URL. For example http://shop.fun.amazon.phishing.com, in this URL phishing.com is an actual domain name, whereas use of "amazon" word is to trick users to click on it. Average number of dots in benign URLs is 3. If the number of dots in URLs is more than 3 then the feature is set to 1 else to 0.

4) Prefix or Suffix separated by (-) to domain: If domain name separated by dash (-) symbol then feature is set to 1 else to 0. The dash symbol is rarely used in legitimate URLs. Phishers add dash symbol (-)to the domain name so that users feel that they are dealing with a legitimate webpage. For example Actual site is http://www.onlineamazon.com but phisher can create another fake website like http://www.online-amazon.com to confuse the innocent users.

5) URL redirection: If "//" presents a URL path then feature is set to 1 else to 0.The existence of "//" within the URL path means that the user will be redirected to another website.

6) HTTPS token in URL: If HTTPS token is present in URL then the feature is set to else to 0. Phishers may add the "HTTPS" token to the domain part of a URL in order to trick users. For example, http://https-www-paypal-it-mpp-home.soft-hair.com.

7) Information submission to Email: Phisher might use "mail()" or "mailto:" functions to redirect the user's information to his personal email[4]. If such functions are present in the URL then feature is set to 1 else to 0.

8) URL Shortening Services "TinyURL" : TinyURL service allows phisher to hide long phishing URL by making it short. The goal is to redirect user to phishing websites. If the URL is crafted using shortening services (like bit.ly) then feature is set to 1 else 0

9) Length of Host name: Average length of the benign URLs is found to be a 25, if URL's length is greater than 25 then the feature is set to 1 else to 0

10) Presence of sensitive words in URL: Phishing sites use sensitive words in its URL so that users feel that they are dealing with a legitimate webpage. Below are the words that foundin many phishing URLs :-'confirm', 'account', 'banking', 'secure', 'ebyisapi', 'webscr', 'signin', 'mail', 'install', 'toolbar', 'backup', 'paypal','password', 'username', etc;

11) Number of slash in URL: The number of slashes in benign URLs is found to be a 5; if number of slashes in URL is greater than 5 then the feature is set to 1 else to 0.

12) Presence of Unicode in URL: Phishers can make a use of Unicode characters in URL to trick users to click on it. For example the domain "xn--

80ak6aa92e.com" is equivalent to" apple.com Visible URL to user is "apple.com" but after clicking on this URL, user will visit "xn--80ak6aa92e.com" which is a phishing site.

13) Age of SSL Certificate: The existence of HTTPS is very important in giving the impression of website legitimacy . But minimum age of the SSL certificate of benign website is between 1 year to 2 years.

14) URL of Anchor: We have extracted this feature by crawling the source code on the URL. URL of the anchor is defined by <a> tag. If the <a>tag has a maximum number of hyperlinks which are from the other domain then the feature is set to 1 else to 0.

15) IFRAME: We have extracted this feature by crawling the source code of the URL. This tag is used to add another web page into existing main webpage. Phishers can make use of the "iframe" tag and make it invisible i.e. without frame orders. Since border of inserted Webpage is invisible, user seems that the inserted web page is also the part of the main web page and can enter sensitive information.

16) Website Rank: We extracted the ranking of websites and compare it with the first One hundred thousand websites of Alexa database. If rank of the website is greater than 10,0000 then feature is set to 1 else to 0.

7.2 Feature 2

The importance to safeguard online users from becoming victims of online fraud, divulging confidential information to an attacker among other effective uses of phishing as an attacker's tool, phishing detection tools play a vital role in ensuring a secure online experience for users. Unfortunately, many of the existing phishing-detection tools, especially those that depend on an existing blacklist, suffer limitations such as low detection accuracy and high false alarm that is often caused by either a delay in blacklist update as a result of human verification process involved in classification or perhaps, it can be attributed to human error in classification which may lead to improper classification of the classes.

These critical issues have drawn many researchers to work on various approaches to improve the detection accuracy of phishing attacks and to minimize false alarm rates. The inconsistent nature of attacks behaviors and continuously

changing URL phish patterns require timely updating of the reference model. Therefore, it requires an effective technique to regulate retraining as to enable machine learning algorithms to actively adapt to the changes in phish patterns.

This study focuses on investigating a better detection approach and to design an ensemble of classifiers suitable to be used in phishing detection. Figure 6.1 summarizes the design and implementation phases leading to the proposed better detection model.

Phase 1

Focuses on dataset gathering, preprocessing, and feature extraction. The objective is to process data for use in Phase 2. The gathering stage is done manually by using Google crawler and Phishtank, each of these data gathering methods were tested to ensure a valid output. The dataset is validated first after gathering, then normalized, features extraction and finally dataset division. Nine features were selected for this project to ensure an optimum result from the classifiers and also, since using a small feature set will invariably speed up processing time for training and for classification of new instances. These features were selected on the basis of the weighted performance of each feature by using an information gain algorithm to ensure that only the best features were selected. This phase focuses on ensuring that the dataset preprocessing is done appropriately to accommodate the models selected.

Phase 2

Focuses on design and implementation of training and validating model using single classifier. A predefined performance metrics is used as a measurement of accuracy, precision, recall, and f-measure. The objective of this phase is to test the performance of individual classifiers in the pool of varying datasets as divided in Chapter 4 and select the most performed of all the reference classifiers. An accuracy of 99.37% was obtained from K-NN which is the highest as compared to other classifiers referenced. Although it was also observed that some of the classifiers like K-NN and C4.5 maintained a close range performance, same cannot be said of the remaining two classifiers that appeared lacking in performance.

The performance of K-NN is not surprising since the dataset used is of a small set and as such K-NN often performs better with a small dataset but the performance

decreases has the size of the dataset increases (Kim and Huh, 2011). Also, since the performance of KNN is primarily determined by the choice of *K*, the best *K* was found by varying it from 1 to 7; and found that KNN performs best when K = 1. This as well, helped in the high accuracy of KNN compared to other classifiers used.

Phase 3

Which corresponds to the third objective is divided into two parts, one is the ensemble design and the other is the comparative study between the best ensemble and the best individual classifier that was selected in Phase 2. To design a good ensemble, only three algorithms are used for each individual ensemble due to the selection of majority voting as the ensemble algorithm, an odd number of algorithms must be used to select the committee of ensembles. For every instance of each ensemble, an ensemble design of three algorithms is being selected until all the algorithms have been combined evenly.

The design ensemble performed very well with an accuracy of 99.31% for the best-performed ensemble and this result is then compared with that obtained in Phase 2. The outcome of the comparison suggests that if K-NN algorithm is removed or if the size of the dataset is increased, the ensemble will most likely perform better than the individual algorithm. This investigation will be considered as part of future work.

8. TESTING

8.1 Test Cases

			18-Nov-22 PNT2022TMID40333 Project - Web Phishing Detection 4 marks					
Feature Type	**Component**	**Test Scenario**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**TC for Automation(Y/N)**	**Executed By**
UI	Index Page	Verify user is able to see the Index	1. Enter URL and click go 2. Type the URL 3. Verify whether it is processing or not.	https://phishing .herokuapp.com/	Displaying Index Page	Working as expected	Pass	N	Sathish P, Yokesh M, Kalyana Sundaram R, Rishyasrungar R A, Nandhan K
Functional	Home Page	Verify user is able to see the Home	1. Enter URL and click go 2. Type or copy paste the URL 3. Check whether the button is responsive or not 4. Reload and Test Simultaneously	https://phishing .herokuapp.com/	Displaying Search Engine for entering the URL	Working as expected	Pass	N	Sathish P, Yokesh M, Kalyana Sundaram R, Rishyasrungar R A, Nandhan K
Functional	Predict Page	Verify user is able to see the Predict	1. Enter URL and click go 2. Type or copy paste the URL 3. Check the website is legitimateor not 4. Observe the results	https://phishing .herokuapp.com/	Displaying Predict Result	Working as expected	Pass	N	Sathish P, Yokesh M, Kalyana Sundaram R, Rishyasrungar R A, Nandhan K

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	3	2	4	18
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	8	2	3	18	31
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	20	11	12	25	68

9. RESULTS

9.1 Performance Metrics

In [15]:
```
print(classification_report(y_test,y_pred1))
```

```
              precision    recall  f1-score   support

          -1       0.92      0.89      0.91      1014
           1       0.91      0.94      0.92      1197

    accuracy                           0.92      2211
   macro avg       0.92      0.91      0.92      2211
weighted avg       0.92      0.92      0.92      2211
```

In [26]:
```
print(confusion_matrix(y_test,y_pred1))
```

```
[[ 905  109]
 [  75 1122]]
```

In [6]:
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
```

Out[6]:
```
LogisticRegression()
```

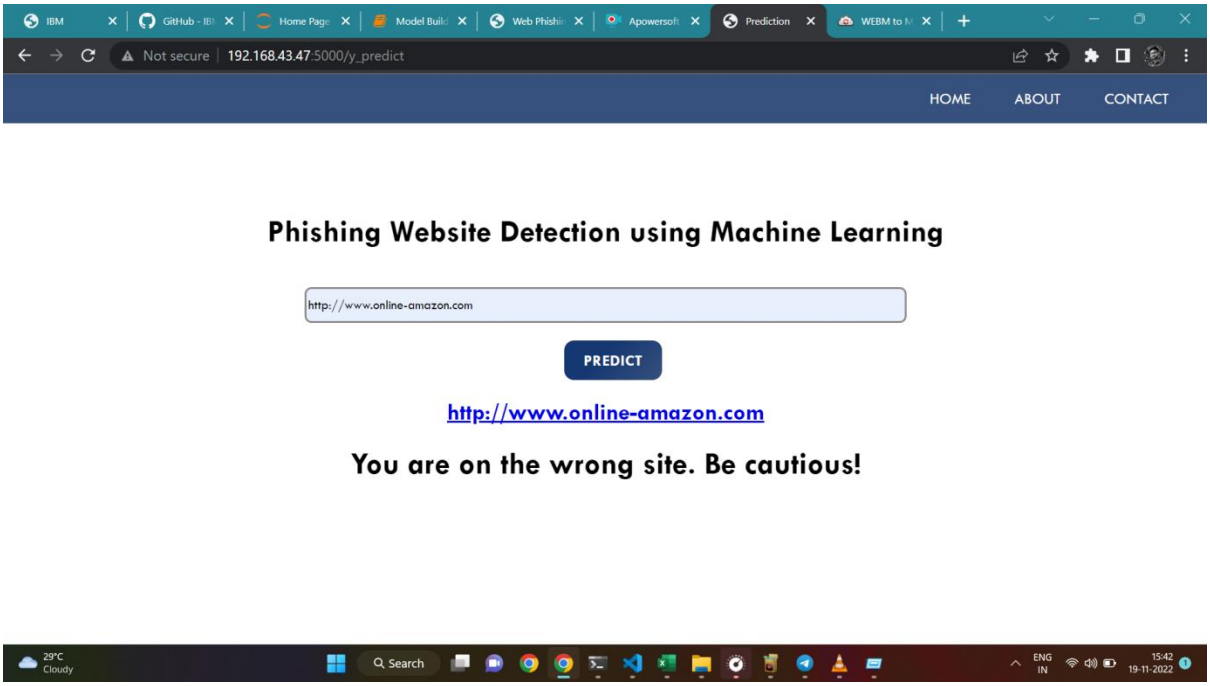
In [9]:
```
y_pred1=lr.predict(x_test)
from sklearn.metrics import accuracy_score
log_reg=accuracy_score (y_test,y_pred1)
log_reg
```

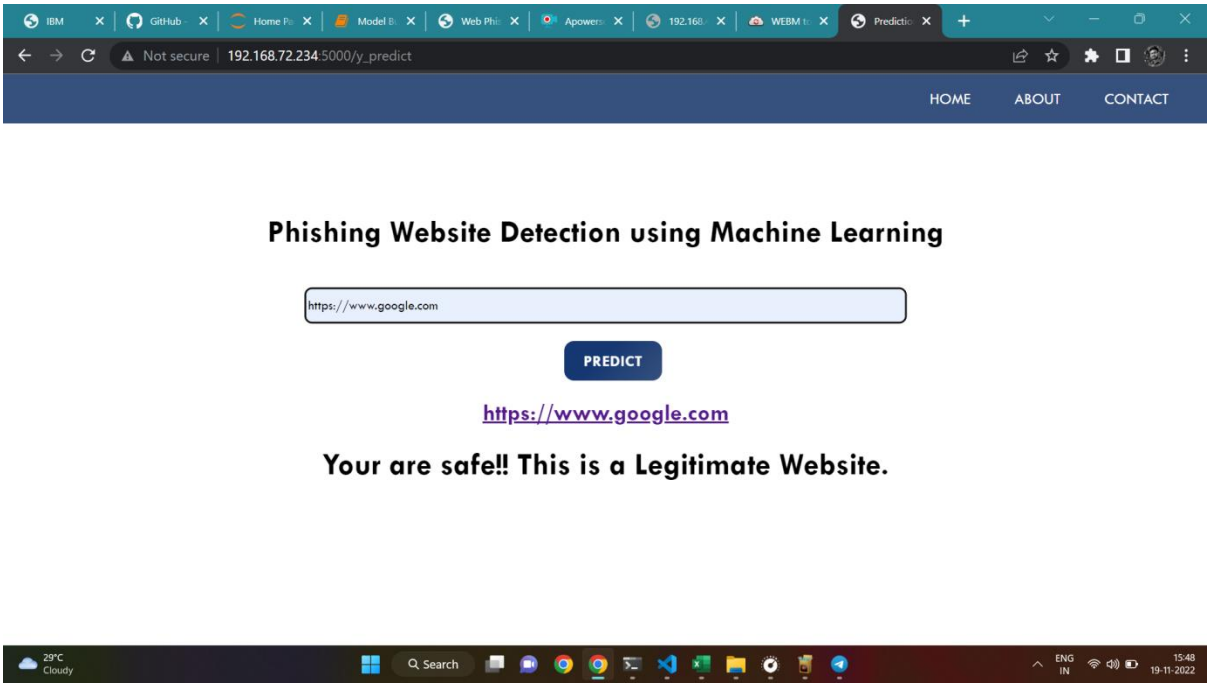
Out[9]:
```
0.9167797376752601
```

In [10]:
```
pd.crosstab(y_test,y_pred1)
```

Out[10]:

col_0	-1	1
row_0		
-1	905	109
1	75	1122





10. ADVANTAGES

- This system can be used by many E-commerce or other websites in order to have a good customer relationship.
- Users can make online payments securely.

- With the help of this system, users can also purchase products online without any hesitation.
- Measure the degrees of corporate and employee vulnerability.
- Eliminate the cyber threat risk level.
- Increase user alertness to phishing risks.
- Machine Learning algorithm used in this system provides better performance as compared to other traditional classification algorithms.

DISADVANTAGES

- All websites related data will be stored in one place.
- Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions.
- If the Internet connection fails, this system won't work.

11. CONCLUTIONS

Phishing is an appalling threat in the web security domain. In this attack, the user inputs his/her personal information to a fake website which looks like a legitimate one. We have presented a survey on phishing detection approaches based on visual similarity. This survey provides a better understanding of phishing website, various solution, and future scope in phishing detection. Many approaches are discussed in this paper for phishing detection; however most of the approaches still have limitations like accuracy, the countermeasure against new phishing websites, failing to detect embedded objects, and so forth. These approaches use various features of a webpage to detect phishing attacks, such as text similarity, font colour, font size, and images present in the webpage. Text based similarity approaches are relatively fast, but they are unable to detect phishing attack if the text is replaced with some image. Image processing-based approaches have a high accuracy rate while they are complex in nature and are time-consuming. Furthermore, most of the work is done offline. These involve data collection and profile-creation phases to be completed first. A comparative table is prepared for easy glancing at the advantages and drawbacks of the available approaches. No single technique is enough for adopting it for phishing detection purposes. Detection of phishing

websites with high accuracy is still an open challenge for further research and development.

12. FUTURE SCOPE

In future if we get structured dataset of phishing we can perform phishing detection much more faster than any other technique.In future we can use a combination of any other two or more classifier to get maximum accuracy. We also plan to explore various phishing techniques that uses Lexical features, Network based features,Content based features, Webpage based features and HTML and JavaScript features of web pages which can improve the performance of the system. In particular, we extract features from URLs and pass it through the various classifiers.

13. APPENDIX

```
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
from flask_cors import CORS
import requests
import flask

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "bLx-wd7zyRvcRj2fC_eiUwXHaiknCIw7ZQaB5d4pAKcF"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app = flask.Flask(__name__, static_url_path='')
CORS(app)


#Redirects to the page to give the user iput URL.
@app.route('/')
@app.route('/index.html')
def home():
    return render_template('index.html')

@app.route('/contact.html')
def contact():
    return render_template('contact.html')

@app.route('/Final.html')
def predict():
    return render_template('Final.html')

#Fetches the URL given by the URL and passes to inputScript
@app.route('/y_predict', methods=['POST','GET'])
def y_predict():

    url = request.form['URL']
```

```python
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


#load model
app = Flask(__name__)
model = pickle.load(open('Phishing_Website.pkl', 'rb'))


#Redirects to the page to give the user iput URL.
@app.route('/')
@app.route('/index.html')
def home():
    return render_template('index.html')

@app.route('/contact.html')
def contact():
    return render_template('contact.html')

@app.route('/Final.html')
def predict():
    return render_template('Final.html')

#Fetches the URL given by the URL and passes to inputScript
@app.route('/y_predict', methods=['POST','GET'])
def y_predict():

    url = request.form['URL']
    checkprediction = inputScript.main(url)
    prediction = model.predict(checkprediction)
    print(prediction)
    output=prediction[0]
    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='{}'.format(pred), url=url)
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Web Phishing Detection</title>
   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/index.css')}}">
</head>
<body>
    <div class="banner">
        <div class="navbar">
            <ul>
                <li><a href='index.html'>Home</a></li>
                <li><a href='index.html#about'>About</a></li>
                <li><a href='contact.html'>Contact</a></li>
                <li><a href='Final.html'>Get Started</a></li>
            </ul>
        </div>
<hr>
        <div class="content">
            <h1><b>Solution to detect<br> Phishing Websites</b></h1>
            <br>
            <p>Be aware of what's happening with your<br> confidential data</p>

        </div>

        <div class="images">
        <image src="{{ url_for('static', filename='css/th.jpeg')}}">
        </div>

        <div style="margin-left:100px; margin-top:-70ch;"class="btn">
            <button class="learn-more">

            <a  href="Final.html">
                <span class="circle" aria-hidden="true">
                    <span class="icon arrow"></span>
                </span>
                <span class="button-text">Get Started</span></a>
            </button>

            <button class="learn-more">
```

```python
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
from flask_cors import CORS
import requests
import flask

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "bLx-wd7zyRvcRj2fC_eiUwXHaiknCIw7ZQaB5d4pAKcF"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app = flask.Flask(__name__, static_url_path='')
CORS(app)


#Redirects to the page to give the user iput URL.
@app.route('/')
@app.route('/index.html')
def home():
    return render_template('index.html')

@app.route('/contact.html')
def contact():
    return render_template('contact.html')

@app.route('/Final.html')
def predict():
    return render_template('Final.html')

#Fetches the URL given by the URL and passes to inputScript
@app.route('/y_predict', methods=['POST','GET'])
def y_predict():

    url = request.form['URL']
```

```
            <button class="learn-more">

                <a  href="https://getcssscan.com/css-buttons-examples">
                    <span class="circle" aria-hidden="true">
                        <span class="icon arrow"></span>
                    </span>
                    <span class="button-text">Watch Video</span></a>
            </button>
            </div>


        </div>
    </div>
    <br><br><br>

    <u><h2 id="about" style="text-align:center;font-size: 35px">About</h2></u>



<div class="text">

    <div style="float:left" class="text1">
        <p style=" margin-left:20px";>
            Web service is one of the key combination software services for the <br>
            Internet. Web phishing is one of many security threats to web services<br>
            on the Internet. Web phishing aims to steal private information, such <br>
            as usernames, passwords, and credit card details, by way of impersonating <br>
            a legitimate entity.
        </p>
    </div>


    <div style="float: right;" class="text2">
        <p style="margin-right:30px;">
            The recipient is then tricked into clicking a malicious link,which can<br>
            lead to the installation of malware,the freezing of the system as part <br>
            of a ransomware attack or the revealing of sensitive information. It <br>
            will lead to information disclosure and property damage.
```

GitHub Link :

https://github.com/IBM-EPBL/IBM-Project-451531660728537

Project Demo Link:

https://www.youtube.com/embed/fq8gGiHkoE0