WEB PHISHING DETECTION

A PROJECT REPORT

SUBMMITED BY

A. PRIYADHARSHINI (923019104302)

A. VANATHI (923019104022)

C.CHINNADURAI (923019104004)

K. CHANDRU (923019104002)

TEAM ID: PNT2022TMID49306

OF

COMPUTER SCIENCE AND ENGINEERING

ARULMURUGAN COLLEGE OF ENGINEERING

KARUR---639 206

ABSTRACT

Phishing URL is a widely used and common technique for cyber security attacks. Phishing is acybercrime that tries to trick the targeted users into exposing their private and sensitive information to the attacker. The motive of the attacker is to gain access to personal informationsuch as usernames, login credentials, passwords, financial account details, social networking data, and personal addresses. These private credentials are then often used for malicious activities such as identity theft, notoriety, financial gain, reputation damage, and many more illegal activities.This paper aims to provide a comprehensive and comparative study of various existing freeservice systems and research based systems used for phishing website detection. The systems in this survey range from different detection techniques and tools used by many researchers. The approach included in these researched papers ranges from Blacklist and Heuristic features to visual and content-based features. The studies presented here use advanced machine learning and deep learning algorithms to achieve better precision and higher accuracy while categorizing websites as phishing or benign. This article would provide a better understanding of the current trends and existing systems in the phishing detection domain.

CHAPTER 1
INTRODUCTION

1.1 Project Overview:

Phisher Find is a website which is used to detect phishing sites to improve the customer's sense of safety whenever he/she attempts to provide any sensitive information to a site. Also, by which people won't access them which will reduce the revenue of malicious site owners.

This application can be accessed online without paying instead, can be accessed via any browser of the customer's choice to detect any site with high accuracy.

This system uses machine learning algorithm which implements classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy.

The design and implementation of a comprehensive web phishing detection system instils a cyber security culture which prevents the need for the deployment of targeted anti-phishing solutions in a corporate to meet industry's compliance obligations.

1.2 Purpose:

Web phishing is a threat in various aspects of security on the internet, which might involve scams and private information disclosure. Some of the common threats of web phishing are:

- Attempt to fraudulently solicit personal information from an individual or organization.
- Attempt to deliver malicious software by posing as a trustworthy organization or entity.
- Installing those malwares infects the data that cause a data breach or even nature's forces that takes down your company's data headquarters, disrupting access.

For this purpose, the objective of our project involves building an efficient and intelligent system to detect such websites by applying a machine-learning algorithm which implements classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy and as a result of which

whenever a user makes a transaction online and makes payment 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.

CHAPTER 2
LITERATURE SURVEY

2.1 Existing problem:

There are phishing detection sites out in the web. But they charge users after a limit of usage. Most of them are builton a clean set of features. We have carefully analysis and identified several factors that could be used to detect a phishing site. These factors fall under the categories of address bar-based features, domain-based features, HTML & JavaScript based features. Using these features, we build an intelligent system which can identify a phishing site with high accuracy and efficiency. It is also an open-source website which will be easily accessible to all users.

2.2 References:

- https://en.wikipedia.org/wiki/Web_service.

[1] Farashazillah Yahya, Ryan Isaac W Mahibol, Chong Kim Ying, Magnus Bin Anai, Sidney Allister Frankie, Eric Ling Nin Wei and Rio Guntur Utomo ”Detection of Phising Websites using Machine Learning Approaches”,2021 International Conference on Data Science and Its Applications (ICoDSA).

[2] Prajakta Patil,Rashmi Rane and Madhuri Bhalekar,”Detecting spam and phishing mails using SVM and obfuscation URL detection algorithm”,2017 International Conference on Inventive Systems and Control (ICISC).

[3] Gaurav Varshney, Manoj Mishra and Pradeep K. Atrey, "A phish detector using lightweight search features", Computers & Security, 2016.

[4] Antonio Hernández Dominguez and Walter Baluja García, "Updated Analysis of Detection Methods for Phishing Attacks", Futuristic Trends in Network and Communication Technologies, vol.1395, pp.56, 2021.

[5] Anggit Ferdita Nugraha and Luthfia Rahman, "Meta-Algorithms for Improving Classification Performance in the Web-phishing Detection Process",

2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), pp.271-275, 2019.

[6] Yoga Pristyanto and Akhmad Dahlan, "Hybrid Resampling for Imbalanced Class Handling on Web Phishing Classification Dataset", 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), pp.401-406, 2019.

[7] Athulya A.A and Praveen K, ”Towards the Detection of Phishing Attacks”,2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)

[8] Miyamoto D, Hazeyama H and Kadobayashi Y,” An evaluation of machine learning-based methods for detection of phishing sites” , International Conference on Neural Information Processing pp. 539-546. Springer, Berlin, Heidelberg. (2008)

[9] KS Swarnalatha, K C Ramchandra, Kaushar Ansari,Love Ojha and Sanjok Subedi Sharma, ”Real-Time Threat Intelligence-Block Phising Attacks”,2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)

[10] Salvi Siddhi Ravindra, Shah Juhi Sanjay, Shaikh Nausheenbanu Ahmed Gulzar and Khodke Pallavi, "Phishing Website Detection Based on URL", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp.589, 2021

2.3 Problem statement definition:

Phishing detection techniques do suffer low detection accuracy and high false alarm especially when novel phishing approaches are introduced. Besides, the most common technique used, blacklist-based method is inefficient in responding to emanating phishing attacks since registering new domain has become easier, no comprehensive blacklist

can ensure a perfect up-to-date database. Furthermore, page content inspection has been used by some strategies to overcome the false negative problems and complement the vulnerabilities of the stale lists. Moreover, page content inspection algorithms each have different approach to phishing website detection with varying degrees of accuracy. Therefore, ensemble can be seen to be a better solution as it can combine the similarity in accuracy and different error-detection rate properties in selected algorithms. Therefore, this study will address a couple of research:

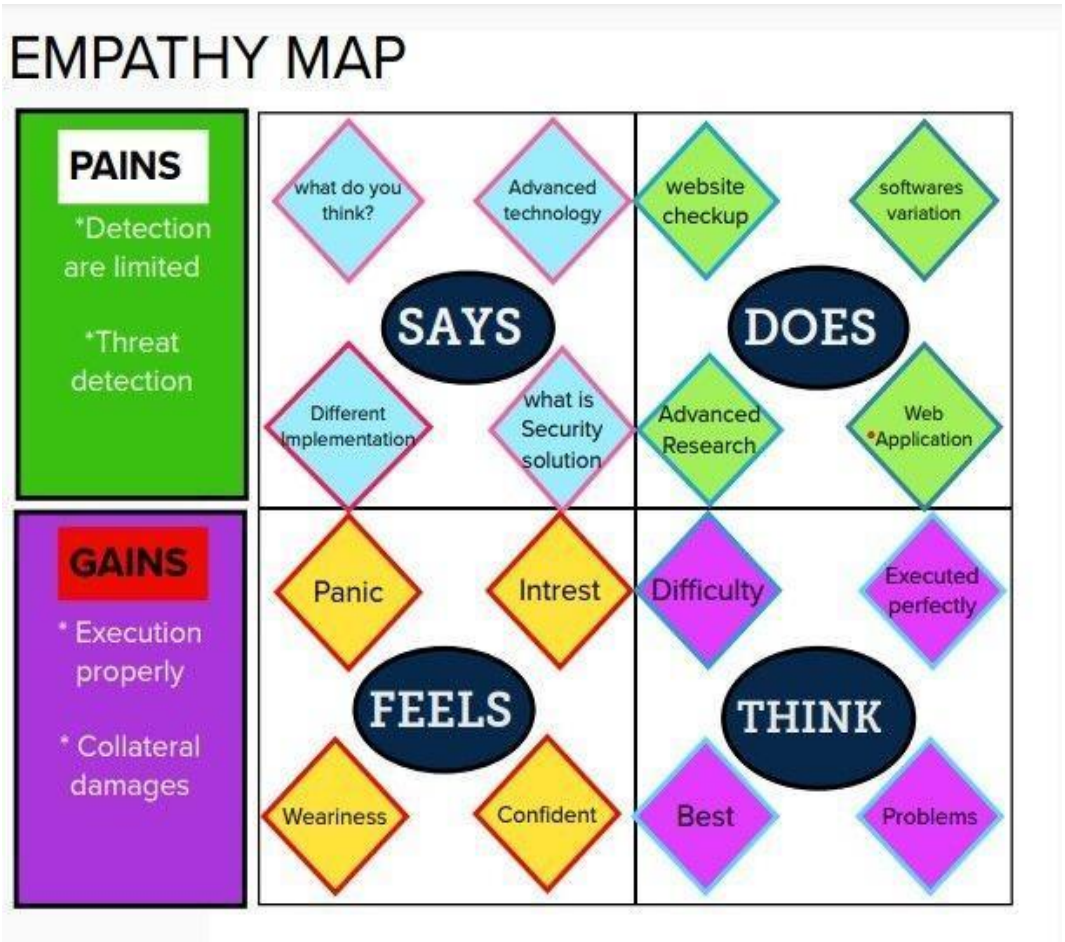
Internet has dominated the world by dragging half of the world's population exponentially into the cyber world. With the booming of internet transactions, cybercrimes rapidly increased and with anonymity presented by the internet, Hackers attempt to trap the end-users through various forms such as phishing, SQL injection, malware, man-in-the-middle, domain name system tunnelling, ransom ware, web Trojan, and so on. Among all these attacks, phishing reports to be the most deceiving attack. Our main aim of this paper is classification of a phishing website with the aid of various machine learning techniques to achieve maximum accuracy and concise model.Nowadays, many people are losing considerable wealth due to online scams. Phishing is one of the means that a scammer can use to deceitfully obtain the victim's personal identification, bank account information, or any other sensitive data. There are a number of anti-phishing techniques and tools in place, but unfortunately phishing still works. One of the reasons is that phishers usually use human behaviour to design and then utilise a new phishing technique. Therefore, identifying the psychological and sociological factors used by scammers could help us to tackle the very root causes of fraudulent phishing attacks

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. Empathy maps should be used throughout any UX process to establish common ground among team members and to understand and prioritize user needs. In user-centered design, empathy maps are best used from the very beginning of the design process.

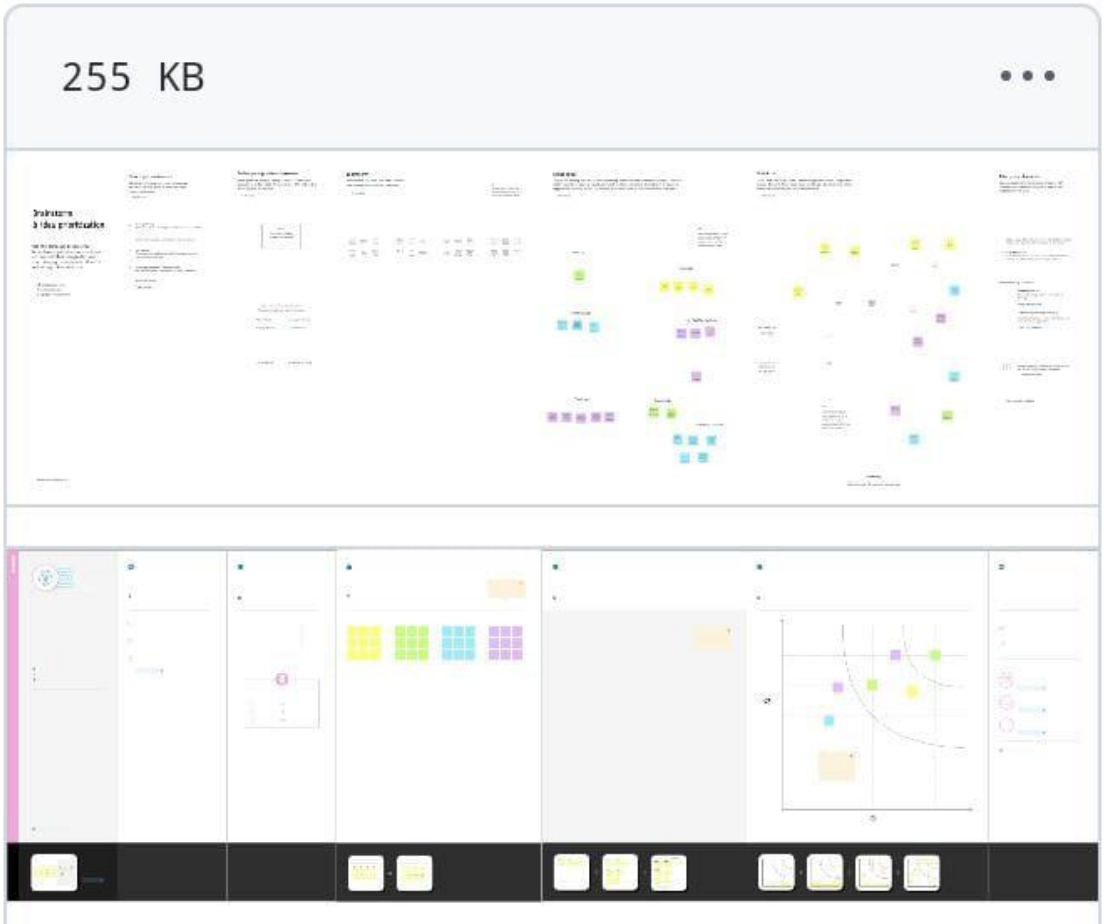


3.2 Ideation & Brainstorming:

Ideation essentially refers to the whole creative process of coming up with and communicating new ideas. Ideation is innovative thinking, typically aimed at solving a problem or providing a more efficient means of doing or accomplishing something.

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas.

A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



3.3 Proposed Solution:

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To reduce the people falling for web phishing scams by creating a sophisticated tool that classifies a website as malicious or safe to use.
2.	Idea / Solution description	Our solution is to build an efficient and intelligent system to detect phishing sites by applying a machine learning algorithm which implements classification algorithmsand techniques to extract the phishing datasets criteria to classify their legitimacy.
3.	Novelty / Uniqueness	Uses an Ensemble model ,Explores weighted features for Neural Network approaches, Extensive feature extraction strategy from the URL Simple and Easy-to-Understand UI.
4.	Social Impact / Customer Satisfaction	By using this application the customer has the sense of safety whenever he attempts toprovide sensitive information to a site.
5.	Business Model (Revenue Model)	This developed model can be used as an enterprise applications by organizations which handles sensitive information and also can be sold to government agencies to prevent the loss of potential important data.
6.	Scalability of the Solution	Solution can use additional hardware resources when the amount of users and activity is increased .The API canensure that multiple requests at the same time are handled in a parallel fashion.

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioural patterns and recognize what would work and why.
Purpose:

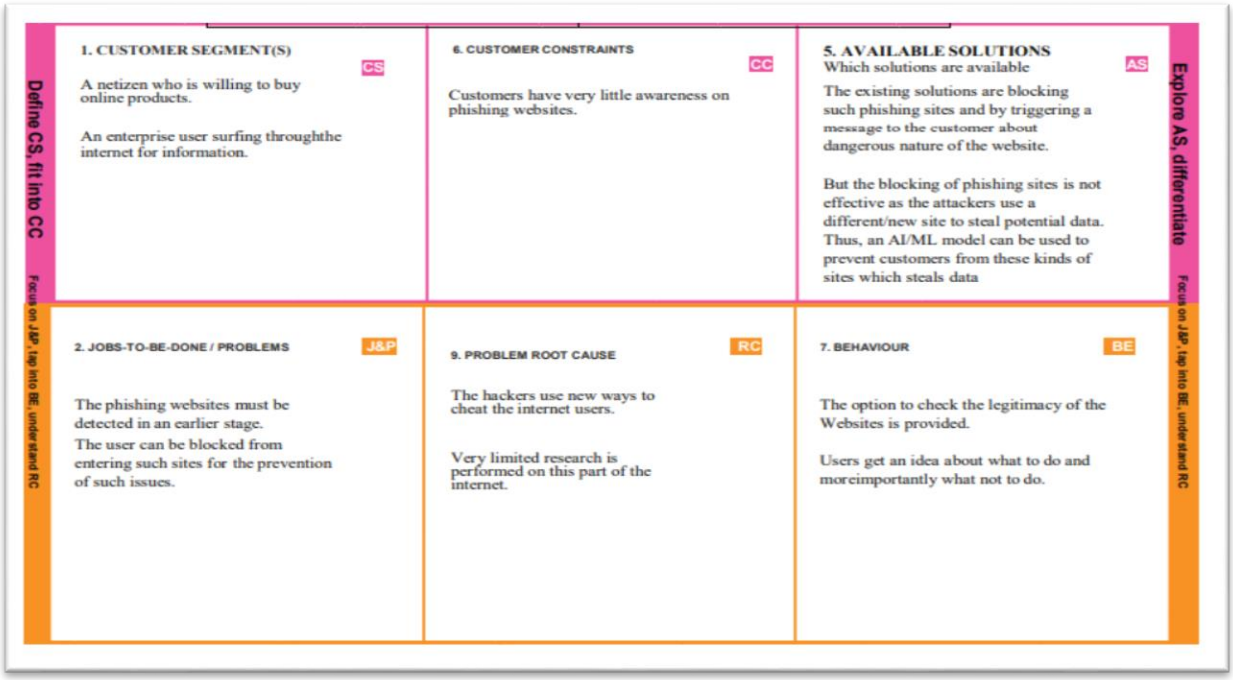
❑ Solve complex problems in a way that fits the state of your customers.

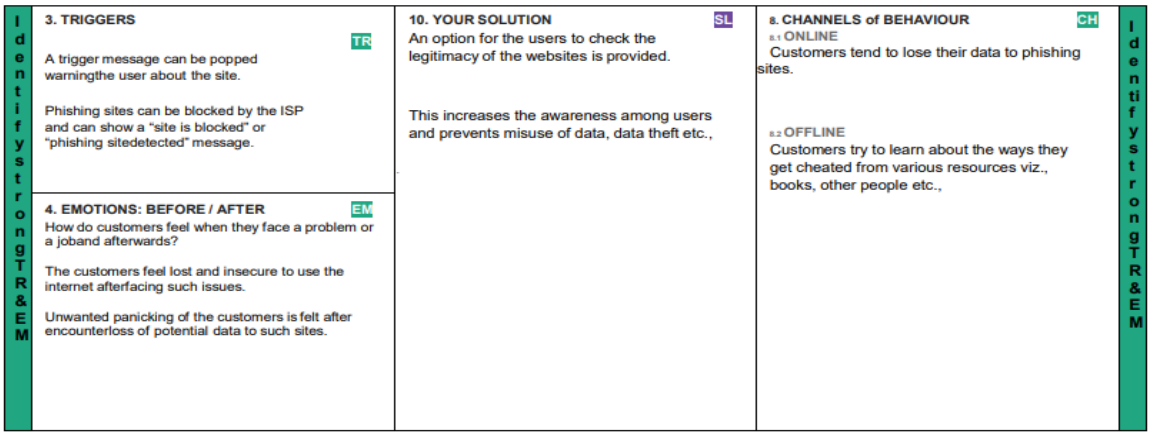
❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behaviour.

❑ Sharpen your communication and marketing strategy with the right triggers and messaging.

❑ Increase touchpoints with your company by finding the right problem-behaviour fit and building trust by solving frequent annoyances, or urgent or costly problems.

❑ Understand the existing situation in order to improve it for your target group.





CHAPTER 4
REQUIREMENT ANALYSIS

4.1 Functional requirements:

FR No.	Functional Requirement(Epic)	Description
FR-1	User Input	User inputs an URL in the form to check whether it is a malicious website.
FR-2	Website comparison	The model compares the given URL with the list of phishing URLs present in the database.
FR-3	Feature Extraction	If it is found none on the comparison it extracts the HTML and domain-based features from the URL.
FR-4	Prediction	The model predicts the URL using machine Learning algorithms such as Random Forest technique.
FR-5	Classifier	Model then sends the output to the classifier and produces the result.
FR-6	Announcement	The model finally displays whether the given URL is phishing or not.

4.2 Non-functional requirements:

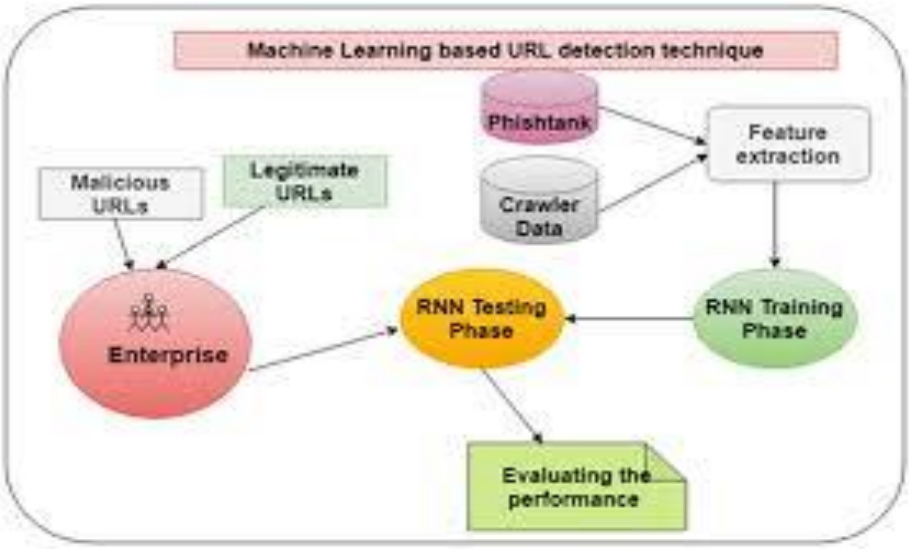
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It is an easy to use and access interface which results in greater efficiency.
NFR-2	Security	It is a secure website which protects the sensitive information of the user and prevents malicious attacks.
NFR-3	Reliability	The system can detect phishing websites with greater accuracy using ML algorithms.
NFR-4	Performance	The system produces responses within seconds and execution is faster.

NFR-5	Availability	Users can access the website via any browser from anywhere at any time.
NFR-6	Scalability	This application can be accessed online without paying. It can detect any web site with high accuracy.

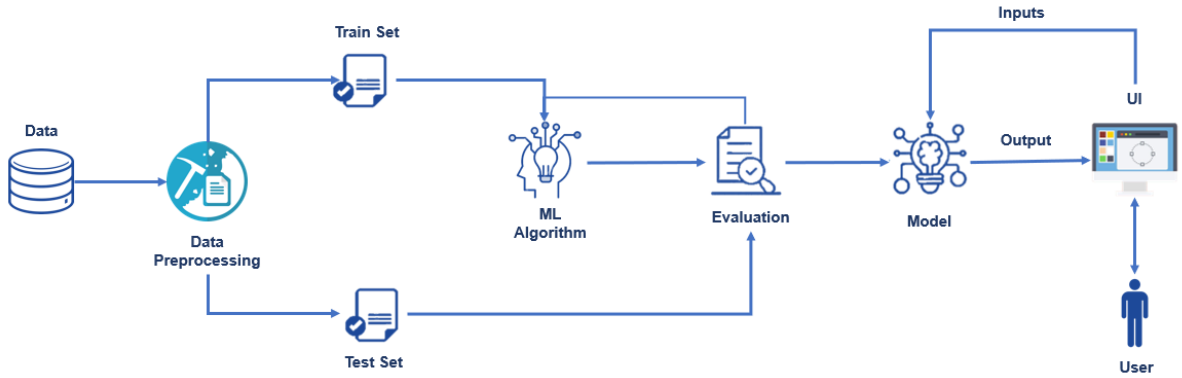
CHAPTER 5
PROJECT DESIGN

5.1 Data Flow diagram:

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 right amount 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:

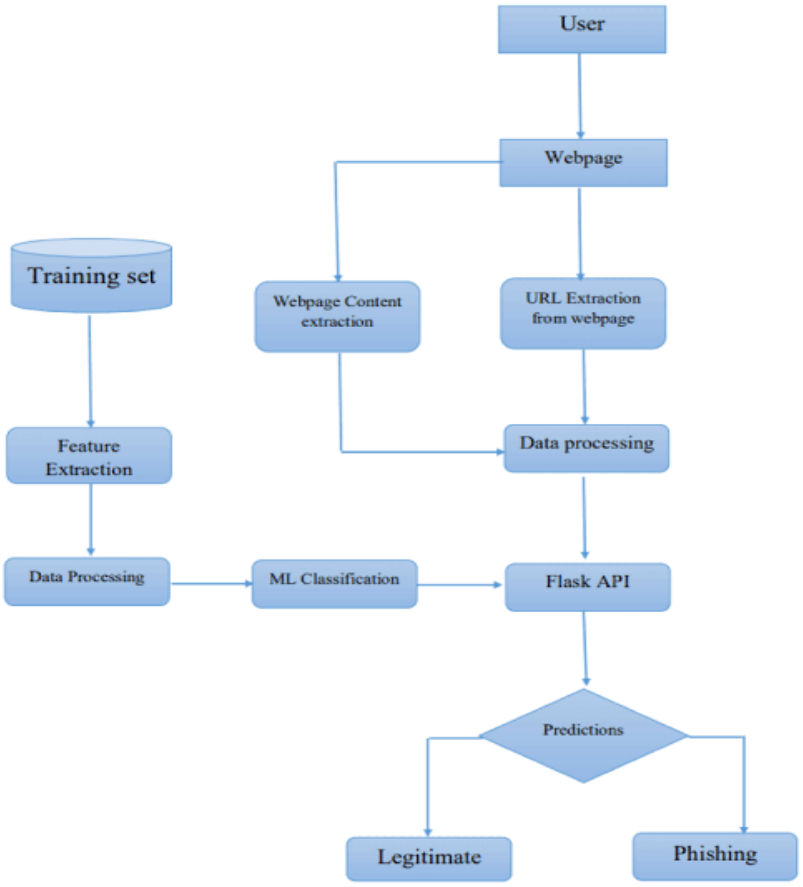


SOLUTION:

Our solution is to build an efficient and intelligent system to detect phishing sites by applying a machine learning algorithm which implements classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy by carefully analysing and identifying various factors that could be used to detect a phishing site. These factors fall under the categories of address bar-based features, domain-based features, HTML & JavaScript based features. Using these features, we can identify a phishing site with high accuracy.

TECHNICAL ARCHITECTURE:

Technical architecture which is also often referred to as application architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components


User
Webpage
Training set
Webpage Content extraction
URL Extraction from webpage
Feature Extraction
Data processing
Data Processing
ML Classification
Flask API
Predictions
Legitimate
Phishing

5.3 User Stories:

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)	User input	USN-1	As a user i can input the particular URL in the required field and waiting for validation.	I can go access the website without any problem	High	Sprint-1
Customer Care Executive	Feature extraction	USN-1	After i compare in case if none found on comparison then we can extract feature using heuristic and visual similarity approach.	As a User i can have comparison between websites for security.	High	Sprint-1
Administrator	Prediction	USN-1	Here the Model will predict the URL websites using Machine Learning algorithms such as Logistic Regression, KNN	In this i can have correct prediction on the particular algorithms	High	Sprint-1
	Classifier	USN-2	Here i will send all the model output to classifier in order to produce final result.	I this i will find the correct classifier for producing the result	Medium	Sprint-2

CHAPTER 6
PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

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

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User input	USN-1	User inputs an URL in the required field tocheck its validation.	1	Medium	Chandru. k
Sprint-1	Website Comparison	USN-2	Model compares the websites using Blacklistand Whitelist approach.	1	High	Priyadharshini. A
Sprint-2	Feature Extraction	USN-3	After comparison, if none found on comparisonthen it extract feature using heuristic and visualsimilarity.	2	High	Priyadharshini. A
Sprint-2	Prediction	USN-4	Model predicts the URL using Machine learning algorithms such as logistic Regression, KNN.	1	Medium	Vanathi. A
Sprint-3	Classifier	USN-5	Model sends all the output to the classifier and produces the final result.	1	Medium	Chinnadurai
Sprint-4	Announcement	USN-6	Model then displays whether the website islegal site or a phishing site.	1	High	Priyadharshini. A
Sprint-4	Events	USN-7	This model needs the capability of retrievingand displaying accurate result for a website.	1	High	Vanathi. A

6.2 Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA:

One part of ensuring the success and smooth operations of your projects in JIRA is reporting. It involves gaining the knowledge about the health, progress and overall status of your JIRA projects through Gadgets, report pages or even third party applications. The goal of this guide is to provide an overview of the tools available to JIRA users today and how they can be used to fulfill the different types of reporting needs that users face today.

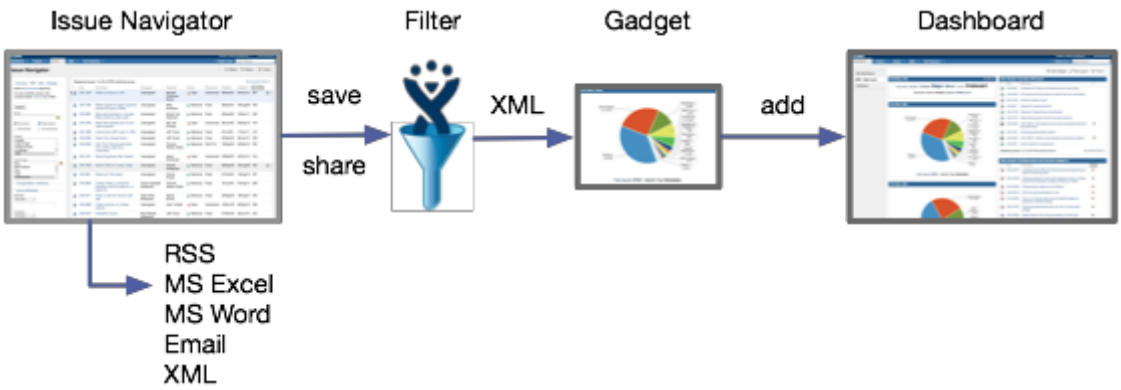
Tools for Reporting

Let us first start with the out-of-the-box tools available, both pre-installed and available through Atlantis Marketplace. We will look at each tool from a technical perspective and in the next chapter, see how they can be applied to the different types of reporting.

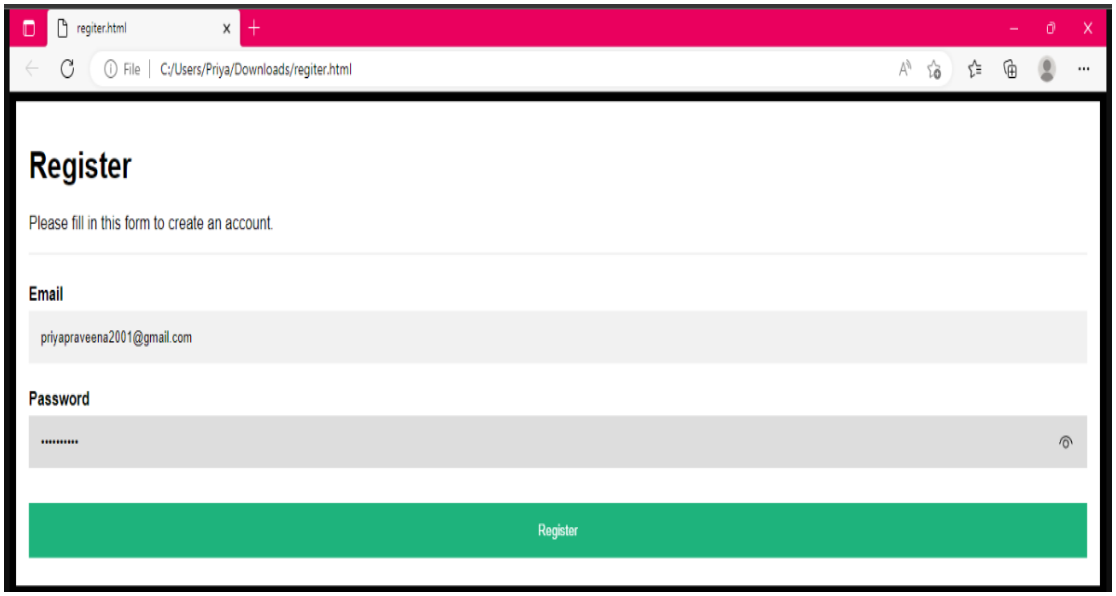
Standard Reports

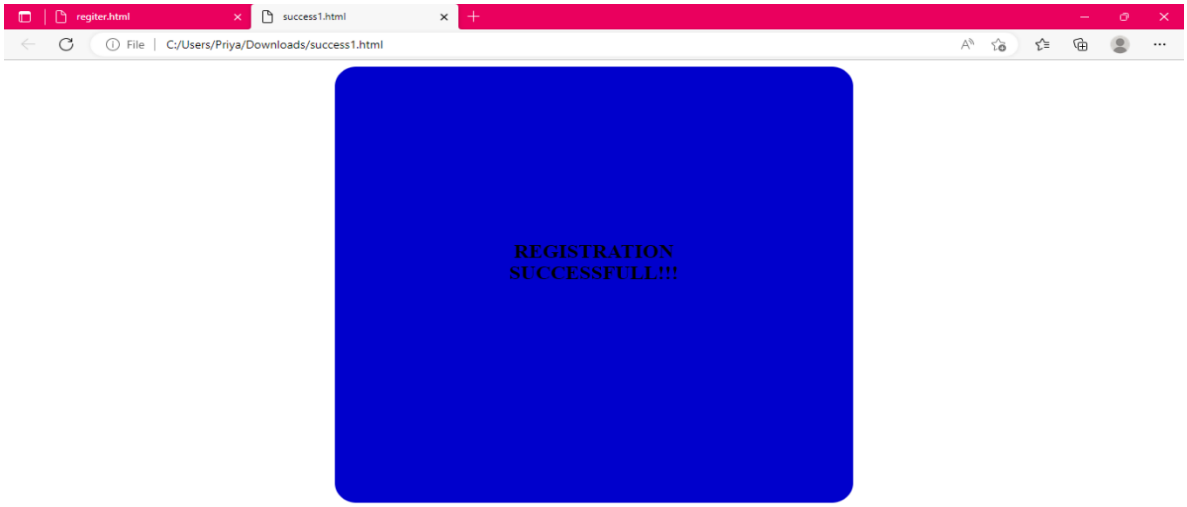
In JIRA, a project will automatically offer standard reports available to the user without any necessary configuration. These standard reports comprise a wide range of reporting applications such as time tracking, workload and also abstract reports like Pie Charts that can be used in various ways.

Custom Gadgets and Dashboards



Sprint 1: REGISTER





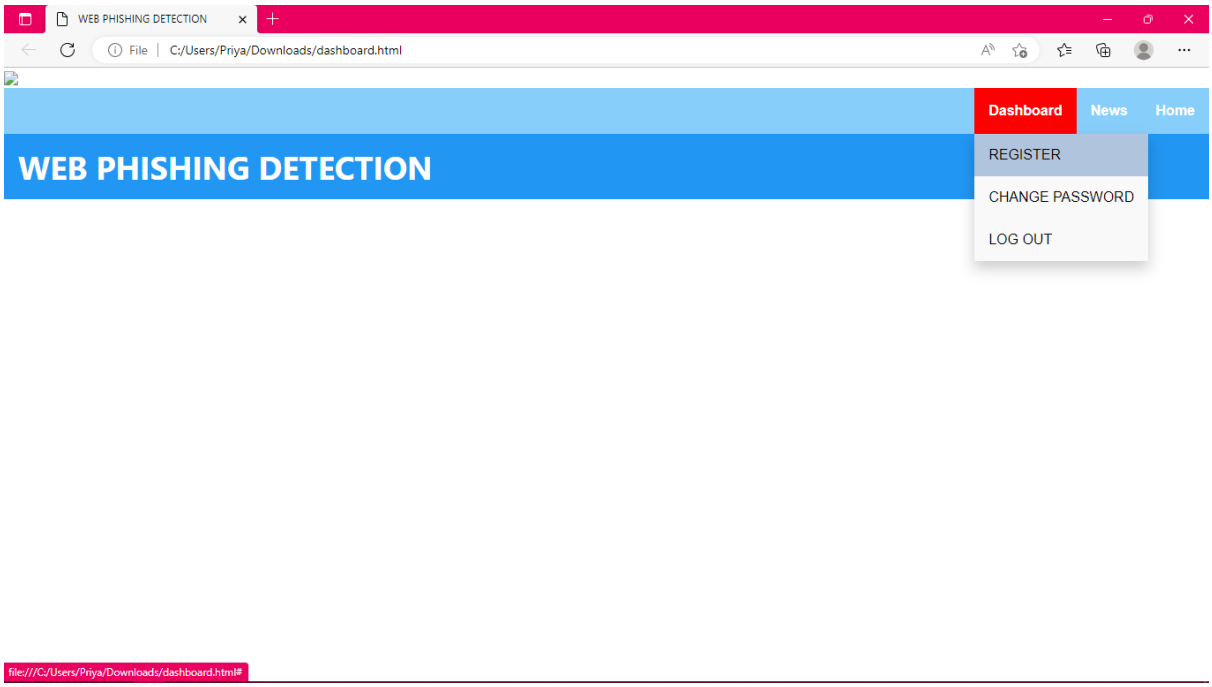
Sprint 2:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
```

```
font-family: Arial, Helvetica, sans-serif;
background-color: black;
}
* {
box-sizing: border-box;
}
/* Add padding to containers */
.container {
padding: 16px;
background-color: white;
}
/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 15px;
margin: 5px 0 22px 0;
display: inline-block;
border: none;
background: #f1f1f1;
}
input[type=text]:focus, input[type=password]:focus {
background-color: #ddd;
outline: none;
}
/* Overwrite default styles of hr */
hr {
border: 1px solid #f1f1f1;
margin-bottom: 25px;
}
/* Set a style for the submit button */
.registerbtn {
background-color: blue;
color: white;
padding: 16px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
```

```
width: 100%;
opacity: 0.9;
}
. registerbtn: hover {
opacity: 1;
}
/* Add a blue text color to links */
a {
color: dodgerblue;
}
/* Set a grey background color and center the text of the "sign in"
section */
.signin {
background-color: #f1f1f1;
text-align: center;
}
</style>
</head>
<body>
<form name=form action='/form_login' method="POST">
<div class="container">
<h1>Login</h1>
<p>Welcome back!!!</p>
<hr>
<label for="email"><b>Email</b></label>
<input type="text" placeholder="Enter Email" name="email"
id="email" required>
<label for="psw"><b>Password</b></label>
<input type="password" placeholder="Enter Password" name="psw"
id="psw" required>
<button type="submit" class="registerbtn">LOGIN</button>
</div>
</form>
<h2><center>{{info}}</center></h2>
</body></html>
```

Sprint 3:



CHAPTER 7
CODING & SOLUTIONING

7.1 Feature 1 – Classification of URL:

The primary feature of this project is to classify the given URL as phishing or benign. Various classification algorithms are used to achieve this.

7.1.1 Methodology:

7.1.1.1 Data collection:

URL features of legitimate websites and phishing websites were collected. The data set consists of total 11,055 URLs which include 6,157 legitimate URLs and 4,898 phishing URLs. Legitimate URLs are labelled as “1” and phishing URLs are labelled as “-1”. The features that are present in the data set include:

- IP Address in URL
- Length of URL

- Using URL Shortening Services
- "@" Symbol in URL
- Redirection "//" in URL
- Prefix or Suffix "-" in Domain
- Having Sub Domain
- Length of Domain Registration
- Favicon
- Port Number
- HTTPS Token
- Request URL
- URL of Anchor
- Links in Tags
- SFH
- Email Submission
- Abnormal URL
- Status Bar Customization (on mouse over)
- Disabling Right Click
- Presence of Popup Window
- IFrame Redirection
- Age of Domain
- DNS Record
- Web Traffic
- Page Rank
- Google Index
- Links pointing to the page
- Statistical Report
- Result

Using IBM Cloud Storage this data is accessed throughout the project. The code written below is used to import the dataset

7.1.1.2 Data pre-processing and Exploratory Data Analysis:

Few plots and graphs were drawn to find how the data is distributed and the how features are related to each other.

Univariate analysis:

Univariate analysis provides an understanding in the characteristics of each feature in the data set. Different characteristics are computed for numerical and categorical data. For the numerical features characteristics are standard deviation, skewness, kurtosis, percentile, interquartile range (IQR) and range. For the categorical features characteristics are count, cardinality, list of unique values, top and freq.

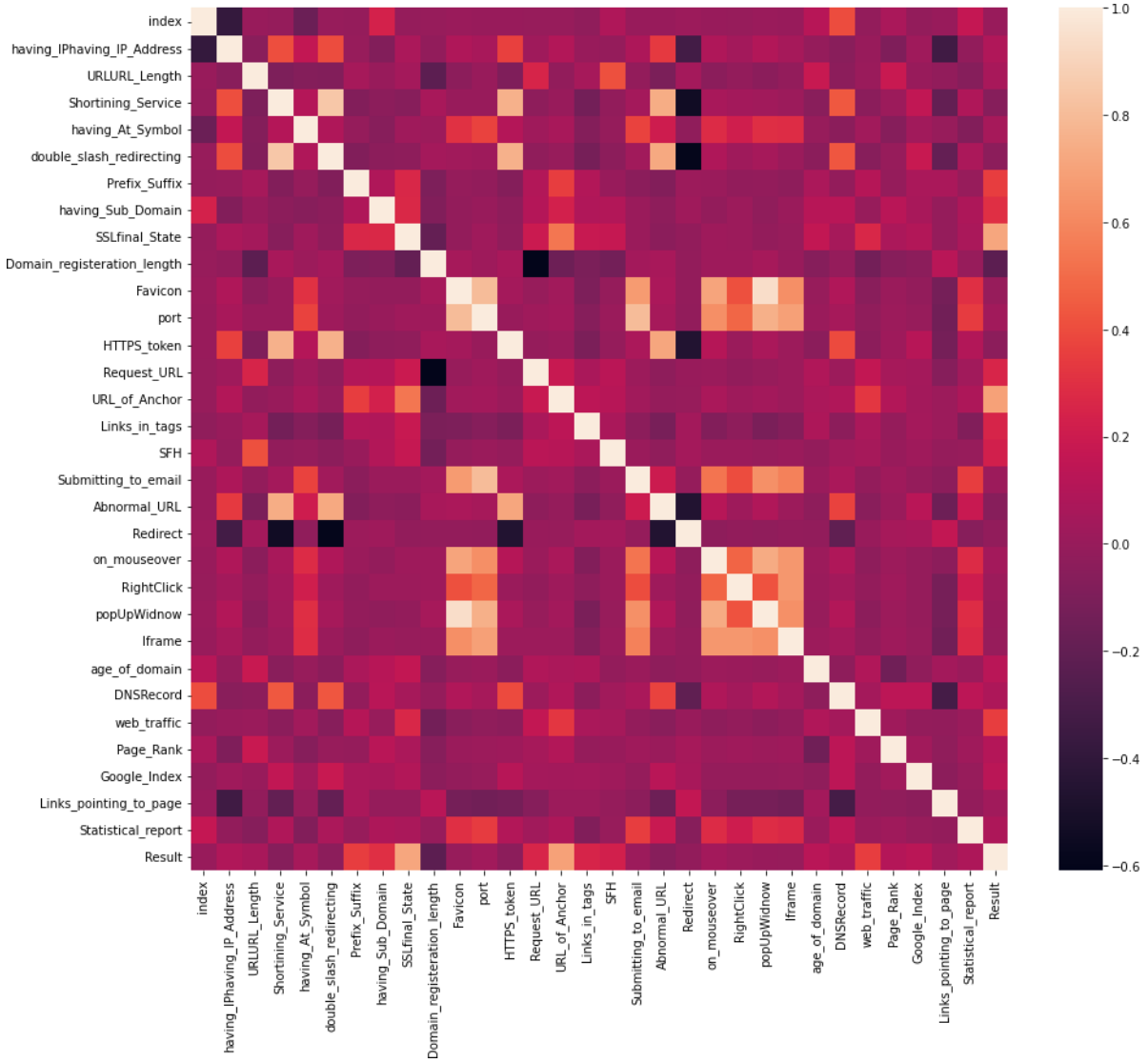
```
data0.describe()
```

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_registeration_length	...	popUpWidnow	Iframe	age_of_domain	DNSRecord	web_traffic	Page_Rank	Google_I
count	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	...	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.000000	11055.00
mean	5528.000000	0.313795	0.633198	0.738761	0.700588	0.741474	0.734962	0.063953	0.250927	-0.336771	...	0.615338	0.816915	0.061719	0.377154	0.287291	-0.483673	0.72
std	3191.447947	0.949534	0.766095	0.673998	0.713598	0.671011	0.678139	0.817518	0.911892	0.941629	...	0.788018	0.576784	0.998168	0.926209	0.827733	0.875289	0.69
min	1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	...	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.00
25%	2764.500000	-1.000000	1.000000	1.000000	1.000000	1.000000	-1.000000	-1.000000	1.000000	-1.000000	...	1.000000	1.000000	-1.000000	-1.000000	0.000000	-1.000000	1.00
50%	5528.000000	1.000000	-1.000000	1.000000	1.000000	1.000000	-1.000000	0.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00
75%	8291.500000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00
max	11055.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00

Bivariate analysis:

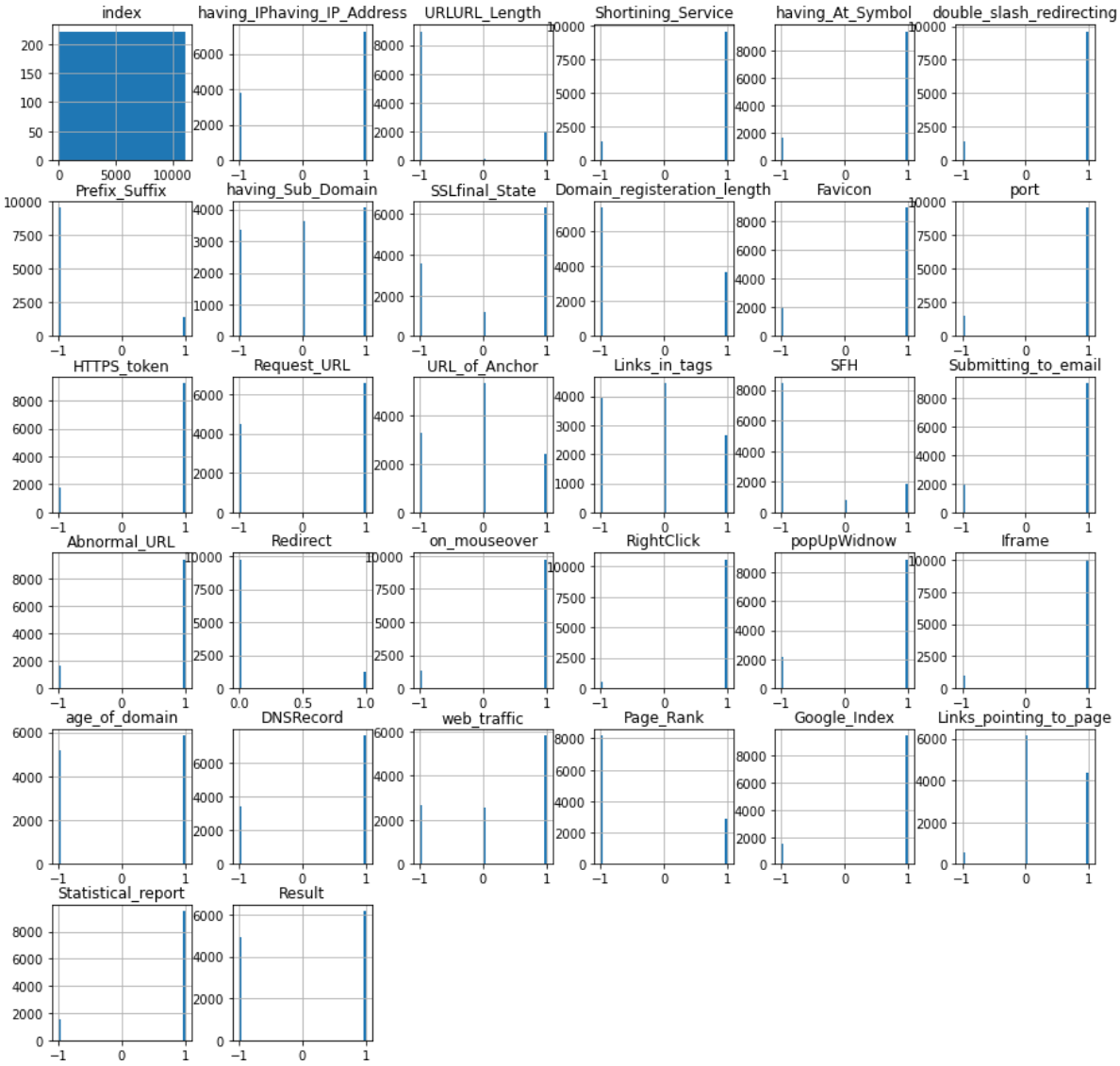
From this correlation matrix, it is evident that there is no correlation with many features. So, it is crucial to eliminate these features.

```
plt.figure(figsize=(15,13))
sns.heatmap(data0.corr())
plt.show()
```



Multivariate analysis:

```
data0.hist(bins = 50,figsize =
(15,15))plt.show()
```



From data distribution graph and correlation matrix, we can conclude that the following features do not have much impact on the result:

- having_Sub_Domain
- Domain_registeration_length
- Favicon
- Request_URL
- URL_of_Anchor
- Links_in_tags
- Submitting_to_email
- Redirect

- web_traffic
- Page_Rank
- Google_Index
- Links_pointing_to_page

```
#Removing the features which do not have much
impact on Result

data=data0.iloc[:,[1,2,3,4,5,6,12,20,21,22,23,24,25,30,31]]

data.head()
```

Checking for null values:

```
#checking the data for null or missing values
data.isnull().sum()
```

```
having_IPhaving_IP_Address    0
URLURL_Length                 0
Shortining_Service            0
having_At_Symbol              0
double_slash_redirecting      0
Prefix_Suffix                 0
HTTPS_token                   0
on_mouseover                  0
RightClick                    0
popUpWidnow                   0
Iframe                        0
age_of_domain                 0
DNSRecord                     0
Statistical_report            0
Result                        0
dtype: int64
```

7.1.1.3 Model building:

From the dataset above, it is clear that this is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

This data set comes under classification problem, as the input URL is classified as phishing (-1) or legitimate (1). The supervised machine learning models (classification) considered to train the dataset in this notebook are:

- XGBoost
- Decision Tree
- Random Forest
- Support Vector Machines

XGBoost:

XGBoost is one of the most popular machine learning algorithms these days. XGBoost stands for eXtreme Gradient Boosting. Regardless of the type of prediction task at hand; regression or classification. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

7.2 Feature 2 – Report:

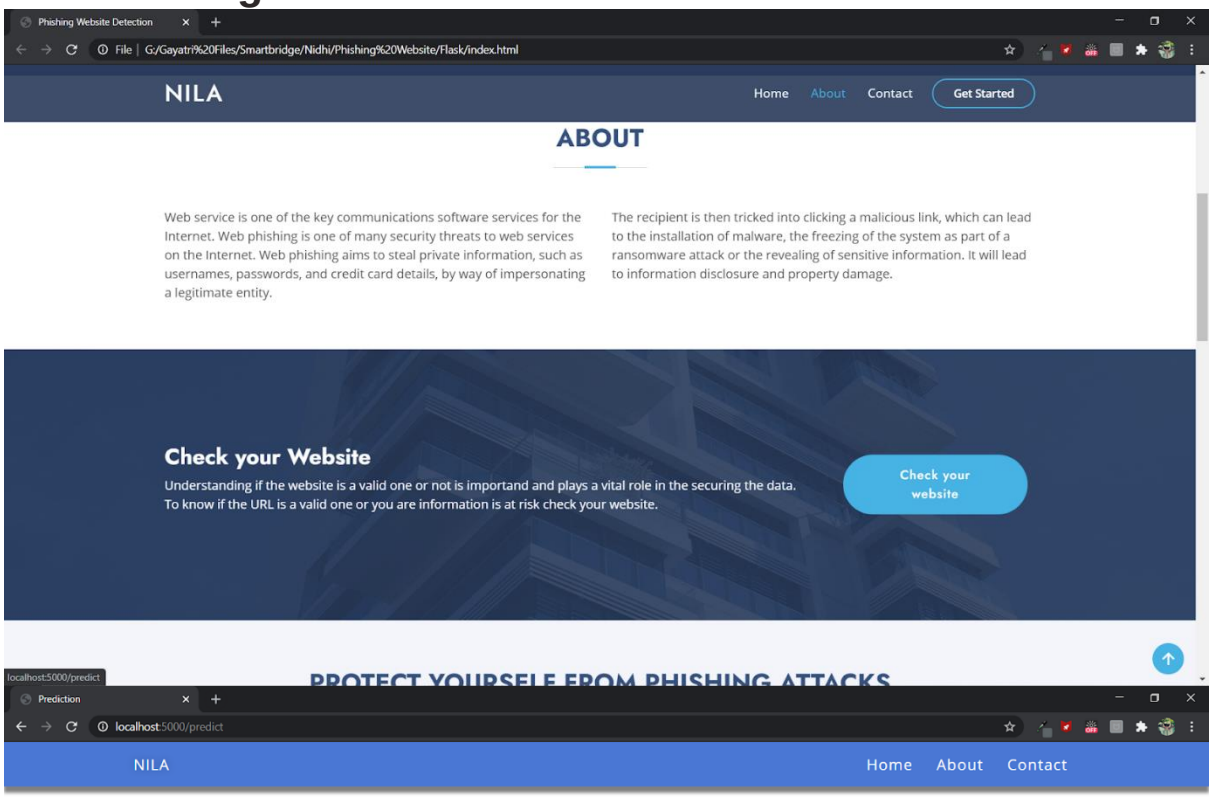
Report page of our site allows users to provide feedback or ask queries to us. It is a platform to connect with the users of our site. The details provided by the user are stored in a database and is accessible by the admin. The report section consists of a basic form with inputs like name, email and query message. After submitting there is a simple response page displayed to the user to confirm their submission.

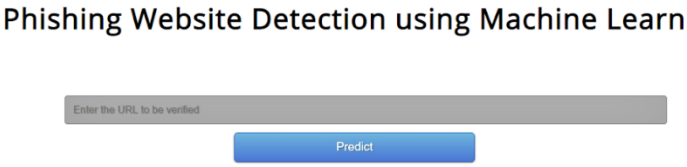
7.3 Database schema:

MySQL is used to create a database to store the inputs from the “Report” page of the website. A table named “responses” is created under the database named “report” with 3 columns named name, email and query. Every time a user submits the report form, the table gets updated with those values.

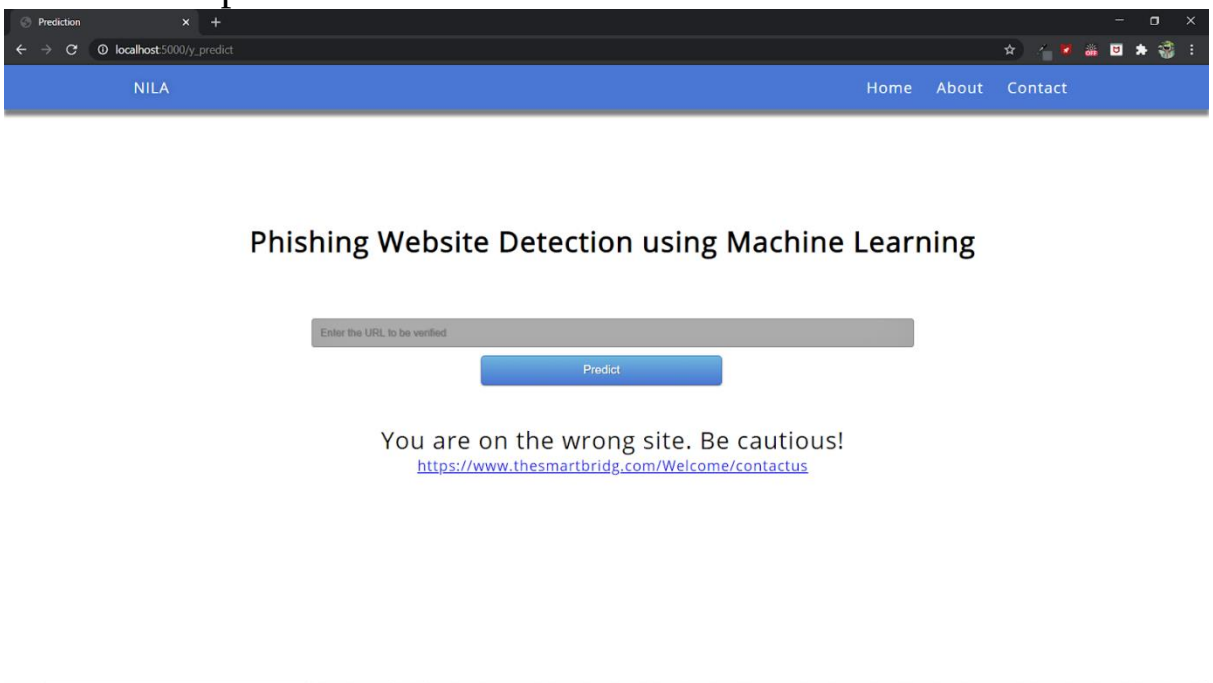
8.TESTING

8.1 Testing Causes





8.2 Final Step



9.RESULTS

9.1 Performance metrics:

The median efficiency is used to assess each categorization model's effectiveness. The final item will appear in the way it was envisioned. Graphical representations are used to depict information during classification. The percentage of predictions made using the testing dataset is used to gauge accuracy. By dividing the entire number of forecasts even by properly predicted estimates, it is simple to calculate. The difference between actual and anticipated output is used to calculate accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP = True Positives, TN = True Negatives, FN = False Negatives and FP = False Positives.

Thus, accuracy for all the four used models were calculated and ranked. XG Boost performed better than other models.

OUTPUT

IBM Project - PNT2022TMID28668

Detect Phishing URLs using Python

PHISHING URL DETECTION

Enter URL

Check here

Still want to Continue

CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- **Increases user alertness to phishing risks** Whenever the user navigates into the website
  and provide the URL of the website that needs to be verified for legitim acy, the system detects phishingsites by applyinga machine learningalg orithm which implements classificationalgorithms and techniques to extract the phishing datasets criteria to classify their
  legitimacy which in turn helps the customers to eliminate the risks of c yber threat and protecttheir valuable corporate or personal data.

- **Users will also be able to pose any query to the admin throughthe re port page designed**Our system is also provided with an option
  for the clients to report to the administrator which helps them to ask their questions significantly improving their experience on our site.

DISADVANTAGES:

- Not a generalized model
- Huge number of rules
- Needs feed continuously

CHAPTER 11
CONCLUSION

Phishing detection is now an area of great interest among the researchers due to its significance in protecting privacy and providing security. There are many methods to perform phishing detection. Our system aims to enhance the detection method to detect phishing websites using machine learning technology. We achieved a high detection accuracy, and the results show that the classifiers givebetter performance when we use more data as training data.
In future, hybrid technology willbe implemented to detect phishing websites more accurately.

CHAPTER 12

FUTURE SCOPE

In the future, optimization can be done in the test units and these units can be made fully automated using Robot-Framework. This is important if more heuristics rules are included in the detection system. If the URL length is very long i.e. more than a million characters, then the system may crash. To prevent this situation, a timeout feature can be added when determining the URL length.

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

BUILDING THE PYTHON FLASK APPLICATION:

In the flask application, the URL is taken from the HTML page and it is scraped to get the different factors or the behavior of the URL. These factors are then given to the model to know if the URL is phishing or safe and is sent back tothe HTML page to notify the user.

SOURCE CODE:

App.py

```
# importing required libraries
From flask import Flask, request, render _template
import requests
import numpy as np
import warnings
import pickle
warnings.filterwarnings('ignore')
from feature import Feature Extraction
file = open("model.pkl","rb")
gbc = pickle.load(file)
file.close()
API_KEY = "UWEsUaH1i-FABXxbCpQ9lcPk5E0jIaivG8i-veVF9zJj"
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
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}
app=Flask(_name_)
@app.route("/", methods=["GET", "POST"])
def index():
if request.method == "POST":
url = request.form["url"]
obj = FeatureExtraction(url)
x = np.array(obj.getFeaturesList()).reshape(1,30)
#1 is safe
#-1 is unsafe
y_pro_phishing = gbc.predict_proba(x)[0,0]
y_pro_non_phishing = gbc.predict_proba(x)[0:,1]
print( y_pro_phishing,y_pro_non_phishing)
# if(y_pred ==1 ):
pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
```

```
payload_scoring = {"input_data": [{"field":
[["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//","PrefixSuffix
-

","SubDomains","HTTPS","DomainRegLen","Favicon","NonStdPort","HTTPS
Domain
URL","RequestURL","AnchorURL","LinksInScriptTags","ServerFormHandler
","Info
Email","AbnormalURL","WebsiteForwarding","StatusBarCust","DisableRight
Click","
UsingPopupWindow","IframeRedirection","AgeofDomain","DNSRecording","
WebsiteT
raffic","PageRank","GoogleIndex","LinksPointingToPage","StatsReport" ]],
"values":obj}]}
response_scoring = requests.post('https://us
south.ml.cloud.ibm.com/ml/v4/deployments/phishing_1/predictions?version=20
22-11-11',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
predictions=response_scoring.json()
print(predictions)
pred=print(predictions['predictions'][0]['values'][0][0])
if(pred != 1):
print("The Website is secure.. Continue")
else:
print("The Website is not Legitimate... BEWARE!!")
return render_template('index.html',xx =round(y_pro_non_phishing,2),url=url )
return
render_template("index.html", xx = -1)
if _name_ == "_main_":
app.run(debug=True)
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Web Phishing Detection</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
```

```
awesome/4.7.0/css/font-awesome.min.css">

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></sc
ript>
<style>
body{
margin: 0;
padding: 0;
font-family:Arial, Helvetica, sans-serif
}.center {
text-align: center;
}
nav{
position:relative;
top: 0;
left: 0;
width: 100%;
height: 70px;
padding: 10px 100px;
box-sizing:border-box;
background:#161616;
}
nav .logo{padding: 15px;
height: 30px;
float: left;
font-size: 25px;
font-weight: bold;
color: #fff;
}
nav ul {
list-style:none;
float: right;
margin: 0;
padding: 0;
display: flex;
font-size: 25px;
}
nav ul li a{
float: right;
display: block;
```

```
color: #f2f2f2;
text-align: center;
padding: 15px;
text-decoration: none;
font-size: 22px;
}
nav ul li a:hover{
background: rgb(200, 212, 200);
border-radius: 6px;
color: rgb(70, 27, 13);
}
nav ul li a.active{
background: #e2472f;
border-radius: 6px;
}
.end {
overflow: hidden;
background-color: rgb(63, 63, 63);
position: fixed;
bottom: 0;
height: 55px;
width: 100%;
}
.continer {align-self:auto;
}
.button1{
appearance: button;
background-color: transparent;
background-image: linear-gradient(to bottom, rgb(160, 245, 174), #37ee65);
border: 0 solid #e5e7eb;
border-radius: .5rem;
box-sizing: border-box;
color: #482307;
column-gap: 1rem;
cursor: pointer;
display: flex;
font-family: ui-sans-serif,system-ui,-apple-system,system-ui,"Segoe
UI",Roboto,"Helvetica
Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI
Emoji","Segoe UI Symbol","Noto
Color Emoji";
font-size: 100%;
font-weight: 700;
```

```
line-height: 24px;
margin: 0;
outline: 2px solid transparent;
padding: 1rem 1.5rem;
text-align: center;
text-transform: none;
transition: all .1s cubic-bezier(.4, 0, .2, 1);
user-select: none;
-webkit-user-select: none;
touch-action: manipulation;
box-shadow: -6px 8px 10px rgba(81,41,10,0.1),0px 2px 2px rgba(81,41,10,0.2);
display: none;
}
.button2{
appearance: button;
background-color: transparent;
background-image: linear-gradient(to bottom, rgb(252, 162, 162), #ee3737);
border: 0 solid #e5e7eb;
border-radius: .5rem;
box-sizing: border-box;
color: #482307;
column-gap: 1rem;
cursor: pointer;
display: flex;
font-family: ui-sans-serif,system-ui,-apple-system,system-ui,"Segoe
UI",Roboto,"Helvetica
Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI
Emoji","Segoe UI Symbol","Noto Color Emoji";
font-size: 100%;
font-weight: 700;
line-height: 24px;
margin: 0;
outline: 2px solid transparent;
padding: 1rem 1.5rem;
text-align: center;
text-transform: none;
transition: all .1s cubic-bezier(.4, 0, .2, 1);
user-select: none;
-webkit-user-select: none;
touch-action: manipulation;
box-shadow: -6px 8px 10px rgba(81,41,10,0.1),0px 2px 2px rgba(81,41,10,0.2);
display: none;
}
```

```
</style>
</head>
<body style="background-image: linear-gradient(to right,#c6ffdd, #fbd786,
#f7797d);">
<div class="center">
<h2 style="font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-
serif;color: rgb(39,
41, 40);">Web Phishing Detection</h2><br>
<form action="/predict" method="post">
<label for="url">Enter The URL:</label>
<input type="text" placeholder="Enter the Suspicious url link" name="url">
<br><br>
<button type="submit" >submit</button>
</form>
<br>
<a href="{{url}}">{{url}}</a>
<br>
<h4 >{{ pred_text }}</h4>
</div>
</body>
</html>
```

Github Link:

IBM-EPBL/**IBM-Project-44757-1660726603**