PROJECT REPORT

WEB PHISHING DETECTION

USING MACHINE LEARNING

SUBMITTED BY:

TEAM ID : PNT2022TMID52463

SOUNDHARAPANDI R	**820519205037**
BALAJI S	**820519205008**
SHEIKDAWOOD S	**820519205034**
ADHILKHAN K	**820519205003**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION:

1.1 PROJECT OVERVIEW:

Web phishing and emails are a routine occurrence for anyone with email in the Internet age. Masking themselves as reputable companies such as Paypal, using devious means such as the use of company logos and standards, malicious actors, again and again, attempt to lure in individuals from a wide range of technical shades. Phishing ranges extensively in sophistication: from mass-produced misspelt requests for overly specific details to sophisticated spear phishing attacks focused on the details of the individual. The ability of phishing attacks to innocuously harvest your private credentials can leave you mercilessly exposed in our data-intensive world. All it takes is for a user to make the critical mistake of clicking on a single malicious link. Through a simple mistake, a user exposes themselves and their data from anything from drive-by downloads, cross-site scripting attacks to the harvesting of their details in an innocuous web form. Phishing has two main delivery vehicles: emails and websites. Emails being the foremost of these, are most classically associated with phishing. These often include malicious URLs to direct users towards maliciously crafted content. By obfuscating the real destination of a URL through a few simple manipulations, users can quickly find themselves on unknown and insecure ground. Therefore it is vital to tackle this massive worldwide problem. In the United Kingdom (UK) alone, phishing is expected to cost the UK economy as much as £280 million per year . This is encouraging companies such as Google to look into the future of Uniform Resource Locators (URLs) themselves . To tackle the problem of phishing, my project has been focused on tackling the malicious URLs included in them as “more than 75% of phishing mails and links include malicious URLs to phishing sites” . Existing techniques to handle URLs involve automated phishing detecting (mainly employing machine learning techniques), user training (the best results of which are gained from embedded training) and automated security indicators (providing information to help the users decide). I aim to create a system which incorporates aspects of these techniques, to inform and protect users from malicious.

1.2 PURPOSE :

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 The objective of this project is **to train machine learning models and deep neural nets on the dataset created to predict phishing websites**. Both phishing and benign URLs of websites are gathered to form a dataset and from them required URL and website content-based features are extracted.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM :

Social media systems use spoofed e-mails from legitimate companies and agencies to enable users to use fake websites to divulge financial details like usernames and passwords. Hackers install malicious software on computers to steal credentials, often using systems to intercept username and passwords of consumers' online accounts. Phisher use multiple methods, including email, Uniform Resource Locators (URL), instant messages, forum postings, telephone calls, and text messages to steal user information. The structure of phishing content is similar to the original content and trick users to access the content in order to obtain their sensitive data. The primary objective of phishing is to gain certain personal information for financial gain or use of identity theft. Phishing attacks are causing severe economic damage around the world. Moreover, Most phishing attacks target financial/payment institutions and webmail, according to the Anti-Phishing. Working Group (APWG) latest Phishing pattern studies.

2.2 References :

1. Akanbi O.A. Amiri I.S. Fazeldehkordi E.: ' *A machine-learning approach to phishing detection and defense* ' (Syngress, 2014, 1st Edition), pp. 1– 8
2. Phishing Activity Trends Reports, 2020. Available at https://apwg.org/trendsreports, accessed on 5 November 2020
3. Goel D. Jain A.K.: 'Mobile phishing attacks and defence mechanisms: state of art and open research challenges ', *Comput. Sec.*, 2018, **73**, pp. 519– 544
4. 4Phishing, 2020. Available at https://en.wikipedia.org/wiki/Phishing#History, accessed on 18 November 2019

5. Stephen Moramarco, Phishing Definition and History. Available at https://resources.infosecinstitute.com/category/enterprise/phishing/phishing-definition-and-history/, accessed on 6 December 2019
6. Gibbs S.: ‘ Facebook and Google were conned out of $100M in phishing scheme ’. Available at https://www.theguardian.com/technology/2017/apr/28/facebook-google-conned-100m-phishing-scheme, accessed on 15 December 2019
7. Trend Micro: ‘ Texas School District loses $2.3 million to phishing scam ’, BEC, 2020. Available at https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/texas-school-district-loses-2-3-million-to-phishing-scam-bec, accessed 15 January 2020
8. Hadley E.: ‘ Vade secure discovers new phishing attack targeting 550 million email users globally ’. Available at https://www.vadesecure.com/en/phishing-attack-targets-550-million/, accessed on 23 November 2019
9. Phishing Activity Trends Reports, 2020. Available at https://docs.apwg.org//reports/apwg_trends_report_h1_2017.pdf, accessed on
10. Phishing Activity Trends Reports, 2020. Available at https://docs.apwg.org//reports/apwg_trends_report_q1_2018.pdf, accessed on 28 November 2020
11. 28Google Safe Browsing, 2020. Available at https://safebrowsing.google.com/, accessed on 6 January 2020

12. PhishTank, 2020. Available at https://www.phishtank.com/, accessed on 17 November 2019

2.3 PROBLEM STATEMENT DEFINITION :

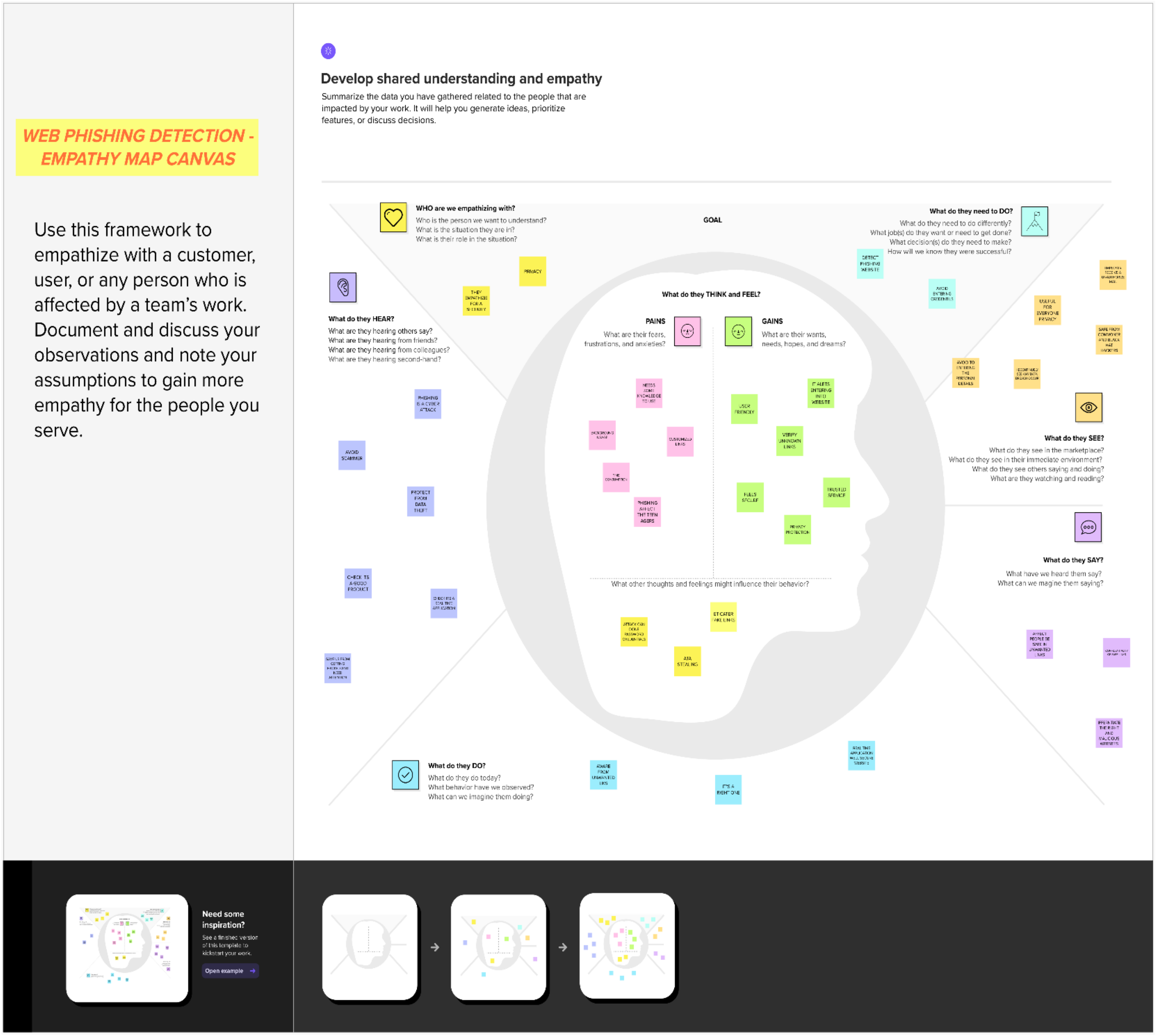
Phishing is a fraudulent technique that uses social and technological tricks to steal customer identification and financial credentials. Social media systems use spoofed e-mails from legitimate companies and agencies to enable users to use fake websites to divulge financial details like usernames and passwords. Hackers install malicious software on computers to steal credentials, often using systems to intercept username and passwords of consumers' online accounts. Phisher use multiple methods, including email, Uniform Resource Locators (URL), instant messages, forum postings, telephone calls, and text messages to steal user information. The structure of phishing content is similar to the original content and trick users to access the content in order to obtain their sensitive data. The primary objective of phishing is to gain certain personal information for financial gain or use of identity theft. Phishing attacks are causing severe economic damage around the world. Moreover, Most phishing attacks target financial/payment institutions and webmail, according to the Anti-Phishing. Working Group (APWG) latest Phishing pattern studies.Phishing is one of the techniques which are used by the intruders to get access to the user credentials or to gain access to the sensitive data. This type of accessing the is done by creating the replica of the websites which looks same as the original websites which we use on our daily basis but when a user click on the link he will see the website and think its original and try to provide his credentials .

To overcome this problem we are using some of the machine learning algorithms in which it will help us to identify the phishing websites based on the features present in the algorithm. By using these algorithm we cam be able to keep the user personal credentials or the sensitive data safe from the intruders.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS :



WEB PHISHING DETECTION - EMPATHY MAP CANVAS

Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document and discuss your observations and note your assumptions to gain more empathy for the people you serve.

Develop shared understanding and empathy

Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.

WHO are we empathizing with?

Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

GOAL

What do they need to DO?

What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

What do they HEAR?

What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

What do they THINK and FEEL?

PAINS

What are their fears, frustrations, and anxieties?

GAINS

What are their wants, needs, hopes, and dreams?

What other thoughts and feelings might influence their behavior?

What do they SEE?

What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

What do they SAY?

What have we heard them say?
What can we magine them saying?

What do they DO?

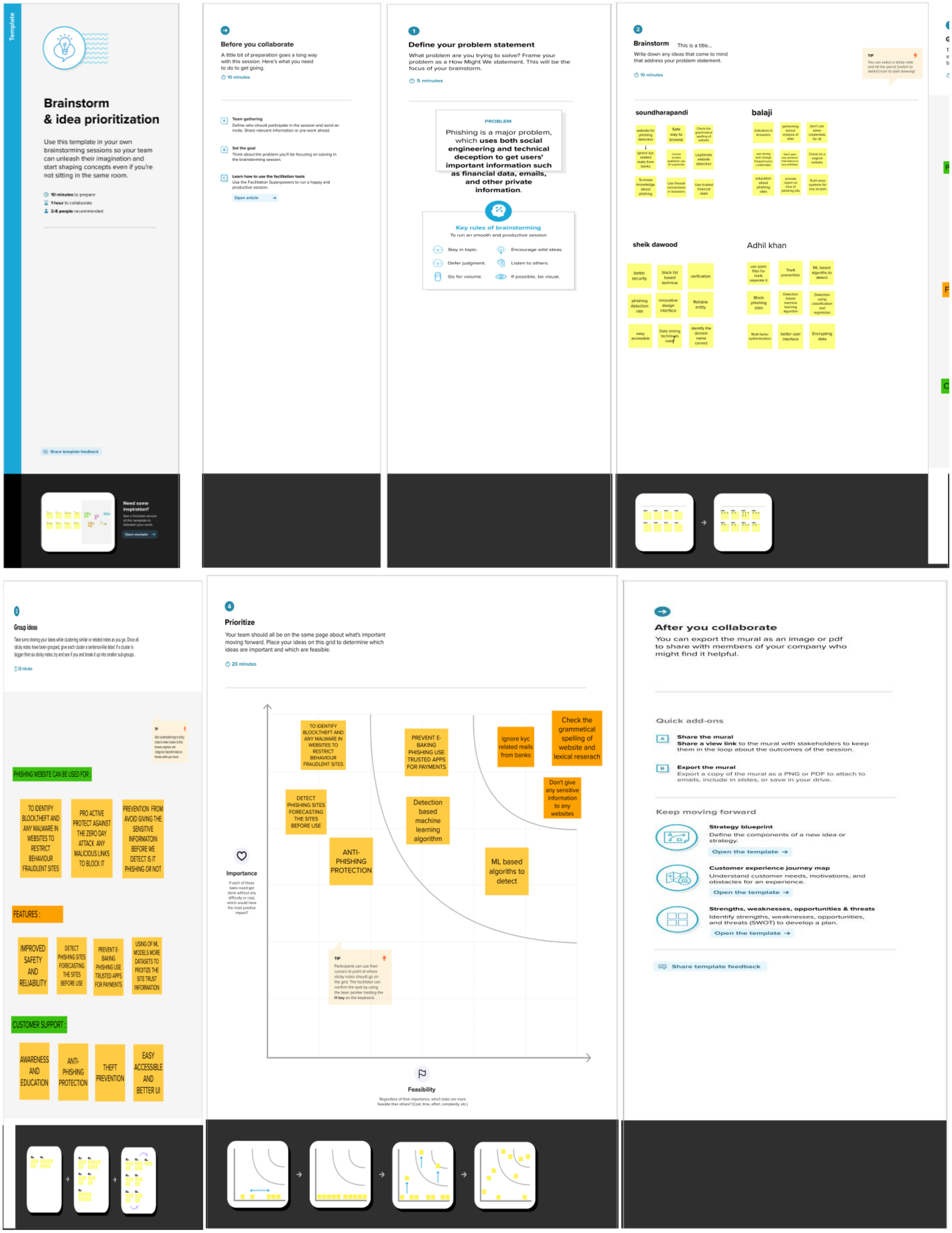
What do they do today?
What behavior have we observed?
What can we imagine them doing?

Need some inspiration?

See a finished version of this template to kickstart your work.

PRIVACY
THEY EMPATHIZE FOR A SECURITY
PHISHING IS A CYBER ATTACK
AVOID SCAMMER
PROTECT FROM DATA THEFT
CHECK TS A GOOD PRODUCT
CHECK ITS A REAL TIME APPLICATION
SAVES US FROM GETTING HACKED AND NEED ATTENTION
NEEDS SOME KNOWLEDGE TO USE
BACKGROUND USAGE
CUSTOMIZED LINKS
TIME CONSUMPTION
PHISHING AFFECT THE TEEN AGERS
USER FRIENDLY
IT ALETS ENTERING INTO WEBSITE
VERIFY UNKNOWN LINKS
FEELS SECURE
TRUSTED SERVICE
PRIVACY PROTECTION
ATTACK CAN DONE PASSWORD CREDENTIALS
ET CATER FAKE LINKS
ATA STEALING
DETECT PHISHING WEBSITE
AVOID ENTERING CREDENTIALS
USEFUL FOR EVERYONE PRIVACY
SAFE FROM COMMONER AND BLACK HAT HACKERS
AVOID TO ENTERING THE PERSONAL DETAILS
AFFECT PEOPLE BE SAFE IN UNWANTED LINKS
IFFERNTIATE THE RIGHT AND MALICIOUS WEBSITES
AWARE FROM UNWANTED LIKS
IT'S A RIGHT ONE
REAL TIME APPLICATION WELL SECURE WEBSITE
Open example

3.2 IDEATION AND BRAINSTORMING



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Need some inspiration?
See a finished version of this template to kickstart your work.
Open example

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.
Open article

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

Phishing is a major problem, which **uses both social engineering and technical deception to get users' important information such as financial data, emails, and other private information.**

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2 **Brainstorm** This is a title...

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

soundharapandi

website for phishing detection | Safe way to browse | Check the grammetical spelling of website
ignore kyc related mails from banks | choose trusted appliation use for payments | Legitimate website detection
To know knowledge about phishing | Use firewall connections in browsers | Use trusted financial apps

balaji

indicators in browsers | performing lexical analysis of data | don't use same credentials for all
use strong and change frequent your credentials | Don't give any sensitive information to any websites | Check its a original website
education about phishing sites | provide report on time of phishing site | Scan yours systems for any viruses

sheik dawood

better security | black list based techniue | verfication
phishing detection rate | innovative design interface | Reliable entity
easy accessible | Data mining techiniques used | identify the domain name correct

Adhil khan

use spam filter for mails seperate it | Theft prevention | ML based algoriths to detect
Block phishing sites | Detection based machine learning algorithm | Detection using classification and regression
Multi factor authentication | better user interface | Encrypting data

3 **Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

PHISHING WEBSITE CAN BE USED FOR :

- TO IDENTIFY BLOCK,THEFT AND ANY MALWARE IN WEBSITES TO RESTRICT BEHAVIOUR FRAUDLENT SITES
- PRO ACTIVE PROTECT AGAINST THE ZERO DAY ATTACK ANY MALICIOUS LINKS TO BLOCK IT
- PREVENTION FROM AVOID GIVING THE SENSITIVE INFORMATOIN BEFORE WE DETECT IS IT PHISHING OR NOT

FEATURES :

- IMPROVED SAFETY AND RELIABILITY
- DETECT PHISHING SITES FORECASTING THE SITES BEFORE USE
- PREVENT E-BAKING PHISHING USE TRUSTED APPS FOR PAYMENTS
- USING OF ML MODELS MORE DATASETS TO PROITIZE THE SITE TRUST INFORMATION

CUSTOMER SUPPORT :

- AWARENESS AND EDUCATION
- ANTI-PHISHING PROTECTION
- THEFT PREVENTION
- EASY ACCESSIBLE AND BETTER UI

4 **Prioritize**

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

- TO IDENTIFY BLOCK,THEFT AND ANY MALWARE IN WEBSITES TO RESTRICT BEHAVIOUR FRAUDLENT SITES
- PREVENT E-BAKING PHISHING USE TRUSTED APPS FOR PAYMENTS
- ignore kyc related mails from banks
- Check the grammetical spelling of website and lexical reserach
- Don't give any sensitive information to any websites
- DETECT PHISHING SITES FORECASTING THE SITES BEFORE USE
- Detection based machine learning algorithm
- ANTI-PHISHING PROTECTION
- ML based algoriths to detect

TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Web phishing tends to steal a lots of information from the user during online transaction like username, password, important documents that has been attached to that websites. There are Multiple Types of Attacks happens in the day to day life, but there is no auto detection Process through Machine Learning is achieved.
2.	Idea / Solution description	Through ML and data mining techniques like classification algorithm user can able to attain a warning signal to notify these phishing websites which helps the user to safeguard identities and their login credentials etc. python is the language that helps to enable these techniques for the online users
3.	Novelty / Uniqueness	This project not only able to identify the malicious websites it also has the ability to automatically block these kind of websites completely in the future when it has been identified and also blocks some various mails ads from these malicious websites
4.	Social Impact / Customer Satisfaction	This web phishing detection project attains the customer satisfaction by discarding various kinds of malicious websites to protect their privacy.This project is not only capable of using by an single individual ,a large social community and a organisation can use this web phishing detection to protect their privacy. This project helps to block various malicious websites simultaneously.
5.	Business Model (Revenue Model)	This developed model can be used as an enterprise applications by organisations 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	This project's performance rate will be high and it also provide many capabilities to the user without reducing its efficiency to detect the malicious websites. Thus the scalability of this project will be high .

3.4 PROBLEM SOLUTION FIT :

1. CUSTOMER SEGMENT(S)

An internet user who is thoughts to buy products online.

An enterprise user surfing throughthe internet for getting more information over the internet

6. CUSTOMER CONSTRAINTS

Customers to have very low awareness onphishing websites.

They don’t know what to do after they losing their data information.

5. AVAILABLE SOLUTIONS

Which solutions are available

The already available solutions are blocking such phishing sites and by triggering a message to the customer about dangerous nature of the

But the blocking of phishing sites are notmore affective as the attackers use a different/new site to steal potential data thus a AI/ML model can be used to prevent customers from these kinds of sites to stealing data

2. JOBS-TO-BE-DONE / PROBLEMS J&P

The phishing websites must bedetected the url have some phishing suspicious in a earlier stage

9. PROBLEM ROOT CAUSE RC

Very limited research on internet

Black hat hacker to use new way of techiniques

7. BEHAVIOUR BE

The option to check the legitimacy of theWebsites is provided.

Users get an idea what to do and moreimportantly what not

Identify strong TR & EM

3. TRIGGERS

TR

A trigger message can be popped warning the user about the site.

Phishing sites can be blocked by the ISP and can show a “site is blocked” or “phishing site detected” message.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?

The customers feel lost and insecure to use the internet after facing such issues.

Unwanted panicking of the customers is felt after encounter loss of potential data to such sites.

10. YOUR SOLUTION

SL

An option for the users to check the legitimacy of the websites is provided.

To increasing the awareness among users and prevents misuse of data, data theft etc.,

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

Customers tend to lose their data to phishingsites.

8.2 OFFLINE

Customers try to learn about the ways theyget cheated from various resources viz., books, other people etc.,

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REUIREMENTS:

A function of software system is defined in functional requirement and the behavior of the system is evaluated when presented with specific inputs or conditions which may include calculations, data manipulation and processing and other specific functionality.

- Our system should be able to load air quality data and preprocess data.
- It should be able to analyze the air quality data.
- It should be able to group data based on hidden patterns.
- It should be able to assign a label based on its data groups.
- It should be able to split data into trainset and testset.
- It should be able to train model using trainset.
- It must validate trained model using testset.
- It should be able to display the trained model accuracy.
- It should be able to accurately predict the air quality on unseen data.

4.2 NON-FUNCTIONAL REQUIREMENTS

Nonfunctional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them. Any attributes required by the customer are described by the specification. We must include only those requirements that are appropriate for our project. Some Non-Functional Requirements are as follows:

- Reliability
- Maintainability
- Performance
- Portability
- Scalability
- Flexibility

Some of the quality attributes are as follows:

ACCESSIBILITY: Availability is a general term used to depict how much an item, gadget, administration, or condition is open by however many individuals as would be prudent. In our venture individuals who have enrolled with the cloud can get to the cloud to store and recover their information with the assistance of a mystery key sent to their email ids. UI is straightforward and productive and simple to utilize.

MAINTAINABILITY: In programming designing, viability is the simplicity with which a product item can be altered so as to: • Correct absconds • Meet new necessities New functionalities can be included in the task based the client necessities just by adding the proper documents to existing venture utilizing ASP.net and C# programming dialects. Since the writing computer programs is extremely straightforward, it is simpler to discover and address the imperfections and to roll out the improvements in the undertaking.

SCALABILITY: Framework is fit for taking care of increment all out throughput under an expanded burden when assets (commonly equipment) are included. Framework can work ordinarily under circumstances, for example, low data transfer capacity and substantial number of clients. 3.2.4

PORTABILITY: Convey ability is one of the key ideas of abnormal state programming. Convenient is the product code base component to have the capacity to reuse the current code as opposed to making new code while moving programming from a domain to another. Venture can be executed under various activity conditions gave it meet its base setups. Just framework records and dependant congregations would need to be designed in such case. The functional requirements for a system describe what the system should do. Those requirments depend on the type of software being developed,the expected users of the software. These are the statement of services the system should provide,how the system should react to particular inputs and how the system should behave in particular situation.

- Extracting data from CSV files

- Cleaning the data

- Vector Representation.

Non-functional requirements is not about functionality or behaviour of system, but rather are used to specify the capacity of a system. They are more related to properties of system such as quality, reliability and quick response time. Non-functional requirements come up via customer needs, • Basic

- Operational Requirement
- Organizational Requirement
- Product Requirement
- User Requirement

➢ Basic Operational Requirement The four primary functions of systems engineering are all performed by the end users, which is the customers.

: Operational requirements which are given by:-

- Mission profile or scenario: It is a map which describes the procedures and leads us to the final goal/ objective. The goal of proposed system is, to predict the crop yield prediction for future year using previous year dataset.

- Performance: It basically gives system parameters to reach our goal. Parameters for the proposed system are accurate predicted value which is compared to the existing system.

- Utilization environments It enlists the different permutations and combinations a system can be reused in many other applications which gives better prediction, as well as gives a new approach to prediction techniques.

• Life cycle: It discuss about the life span of a system. As number of data increases the number of iterations increases, which will give more accuracy to the output.

➢ Organizational Requirement

The Organizational requirement consists of the following types:

• Process Standards: To make sure the system is a quality product, IEEE standards have been used during system development.

• Design Methods: Design is an important step, on which all other steps in the engineering process are based on.

• It takes the project from a theoretical idea to an actual product. It gives us the basis of our solution.Because all the steps after designing are based on the design itself, this step affects the quality of the product and is a major player in how the testing and maintenance of a project take place and how successful they are. Following the design to the 'T' is of utmost importance. ➢ Product Requirement

• Portability: As the system is Python based, it will run on a platform which is supported by ANACONDA.

• Correctness: The system has been put through rigorous testing after it has followed strict guidelines and rules. The testing has validated the data.

• Ease of Use: The user interface allows the user to interact with the system at a very comfortable level with no hassles.

• Modularity: The many different modules in the system are neatly defined for ease of use and to make the product as flexible as possible with different permutations and combinations.

• Robustness: During the development of the system special care is being taken to make sure that the end results are optimized to the highest level and the results are relevant and validated. Python language is used for the development, itself provides robustness to the system and thus makes it highly unlikely to fail. 'System quality' and 'Non-functional requirements' are interchangeable terms. These qualities mainly consist of two things i.e. evolution and execution.

User Requirement :

- The user should able to have User Interface Window .

. • The user should able to configure with neat GUI all the parameters.

Resource Requirement Anaconda 3-5.0.3:

Anaconda is a free and open source distribution of the Python and R programming languages for data science, machine learning and other applications. Anaconda distribution comes with 1400 packages as well as the conda package and virtual environment manager, called Anaconda Navigator. Packages can be made using the conda build command. Anaconda Navigatoris a desktop graphical user interface allows user to manage conda packages. The following applications are available by default in navigator: Jupyter lab, Jupyter netbook, Spyder, Orange, Rstudio etc. conda is an open source, cross platform, language-agnostic package manager and environment management system. It installs, runs and update packages and their dependencies.

1. Jupyter Notebook: The code is fully written in Python language using Jupyter notebook. It is the spin-off projects from the IPyton project, which used to have an IPython Notebook project itself. IPython kernel, which allows you to write your programs in Python. We can install Jupyter Notebook using command $pip installJupyter. It has serveral menus that you can use to interact with your notebook they are listed as:

- File
- Edit
- View
- Insert
- Cell
- Kernel
- Widgets

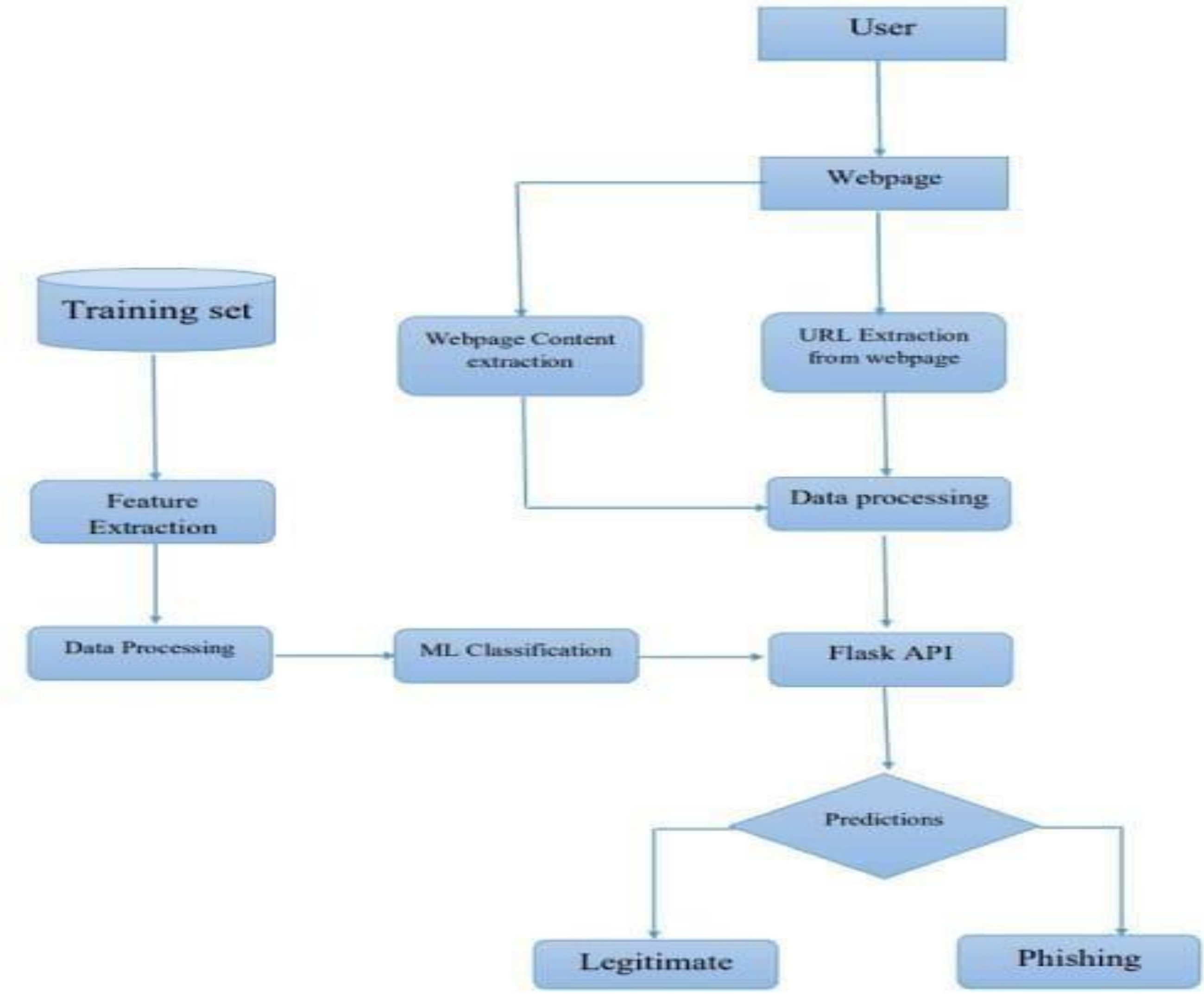
To easily reconnect itself closing it before saving easily recover it.

CHAPTER 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 depictthe right amounto f the systemrequirement graphically. It shows how data entersand leaves the system, what changes the information, and where datais stored.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

SOLUTION ARCHITECTURE:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

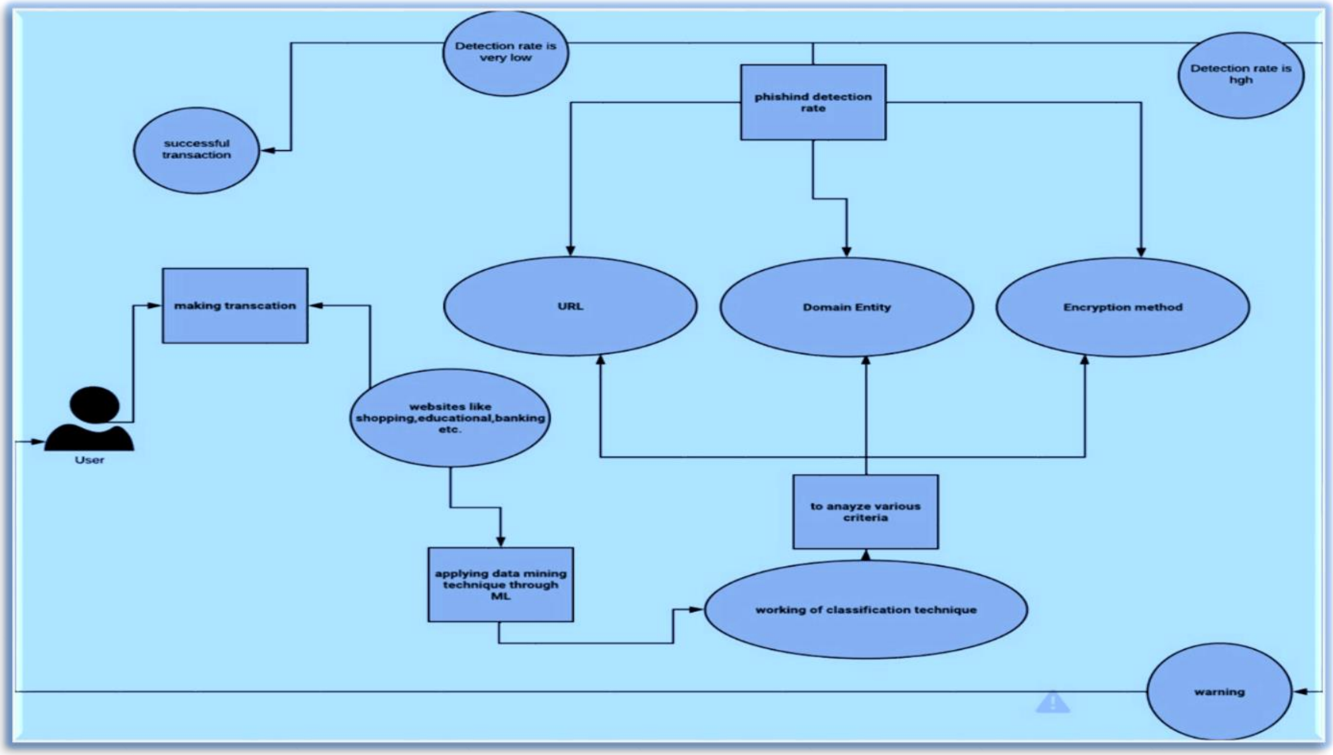
Find the best tech solution to solve existing business problems

.• Describe the structure, characteristics, behavior, and other aspects of the software to

• project stakeholders. Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and

• delivered.


Detection rate is very low
Detection rate is hgh
phishind detection rate
successful transaction
making transcation
URL
Domain Entity
Encryption method
User
websites like shopping,educational,banking etc.
to anayze various criteria
applying data mining technique through ML
working of classification technique
warning

TECHNOLOGY ARCHITECTURE:

Technology architecture deals with the deployment of application components on technology components. A standard set of predefined technology components is provided in order to represent servers, network, workstations, and so on

5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priorit
Customer (Mobile user)		USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High

		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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low
		USN-4	As a user, I can register for the application through Gmail		Medium
		USN-5	As a user, I can log into the application by entering email & password		High
	Dashboard				
Customer (Webuser)	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
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
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
	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

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

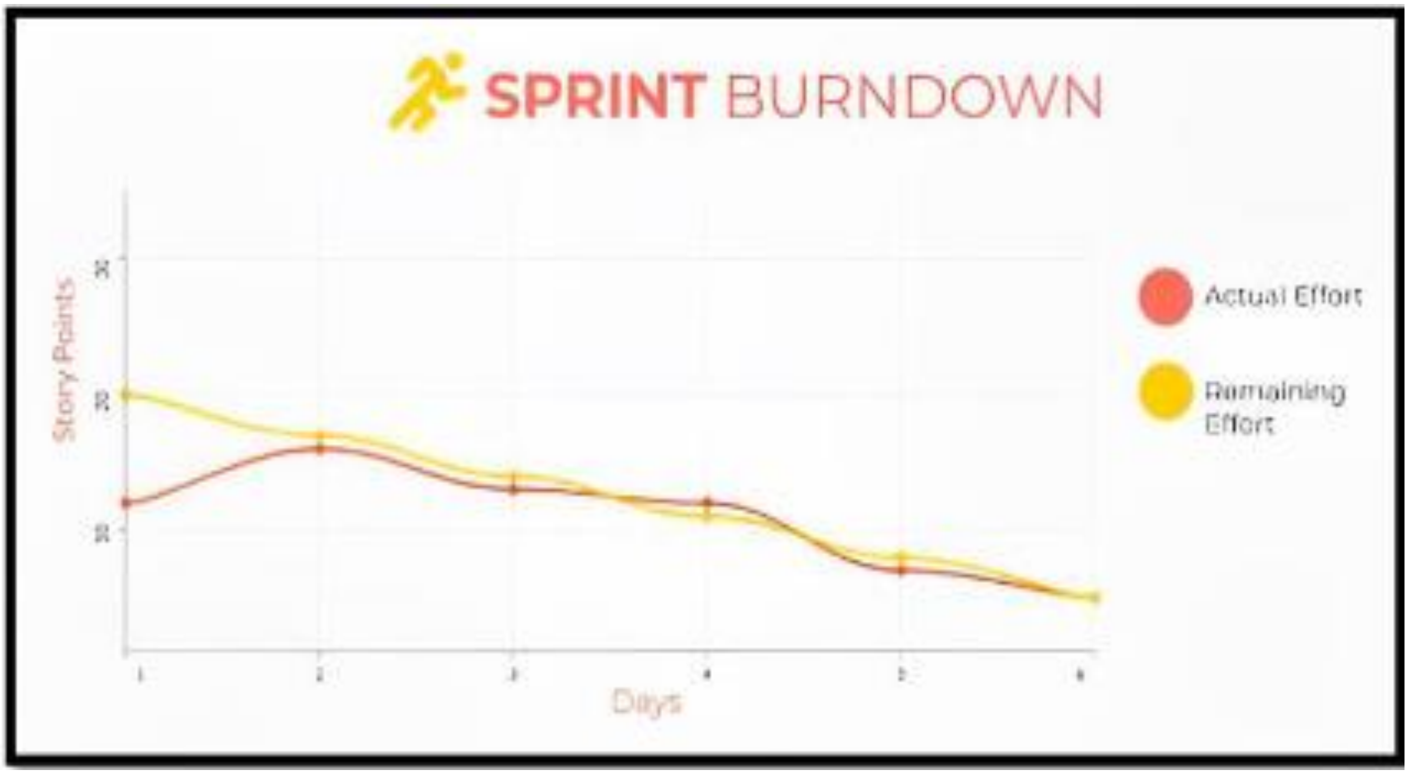
6.1 SPRINT PLANNING AND ESTIMATION:

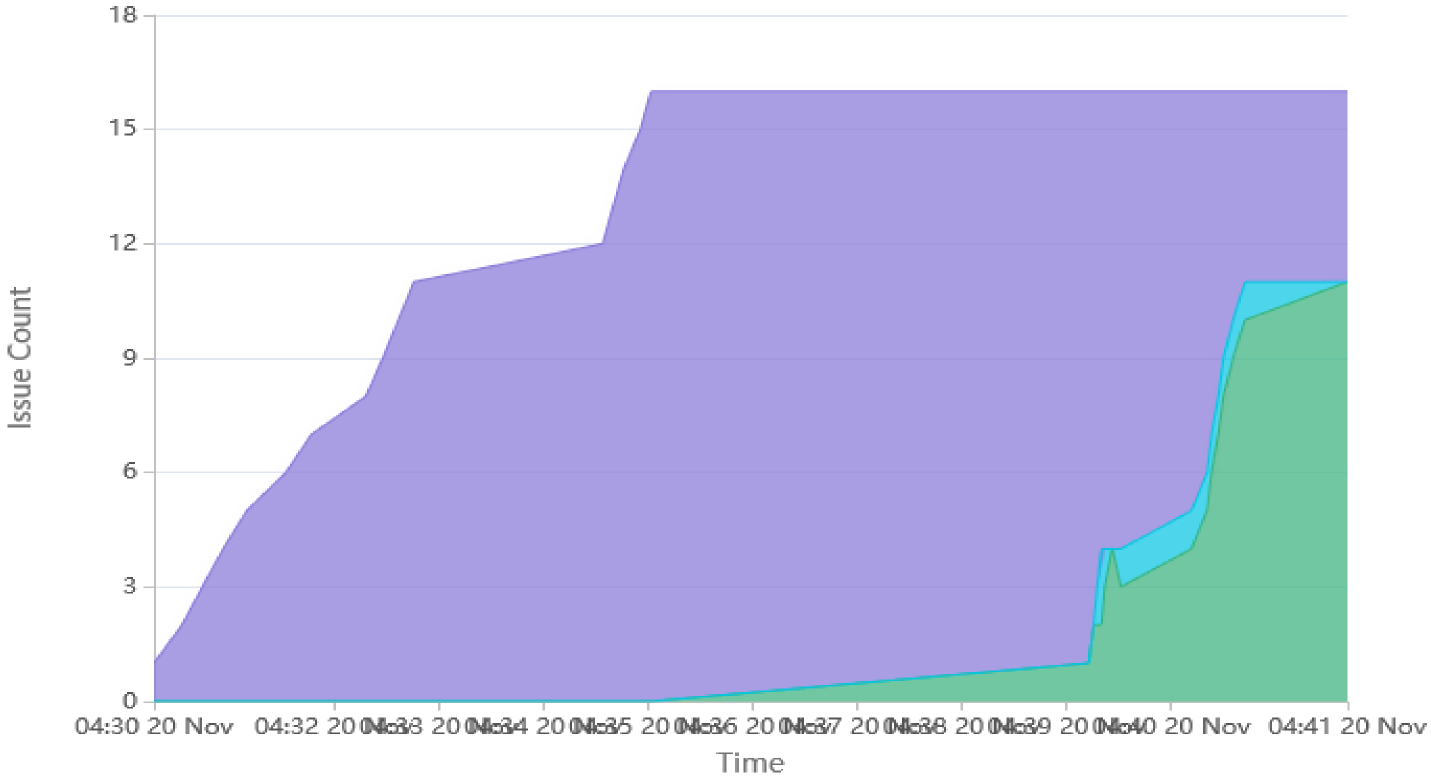
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	User input	USN-1	User inputs an URL in the required field to checkits validation.	1	Medium
Sprint-1	Website Comparison	USN-2	Model compares the websites using Blacklistand Whitelist approach.	1	High
Sprint-2	Feature Extraction	USN-3	After comparison, if none found on comparisonthen it extract feature using heuristic and visual similarity.	2	High
Sprint-2	Prediction	USN-4	Model predicts the URL using Machine learningalgorithms such as logistic Regression, KNN.	1	Medium
Sprint-3	Classifier	USN-5	Model sends all the output to the classifier andproduces the final result.	1	Medium
Sprint-4	Announcement	USN-6	Model then displays whether the website is legal site or a phishing site.	1	High
Sprint-4	Events	USN-7	This model needs the capability of retrieving anddisplaying accurate result for a website.	1	High

6.2 SPRINT DELIVERY SCHEDULE:

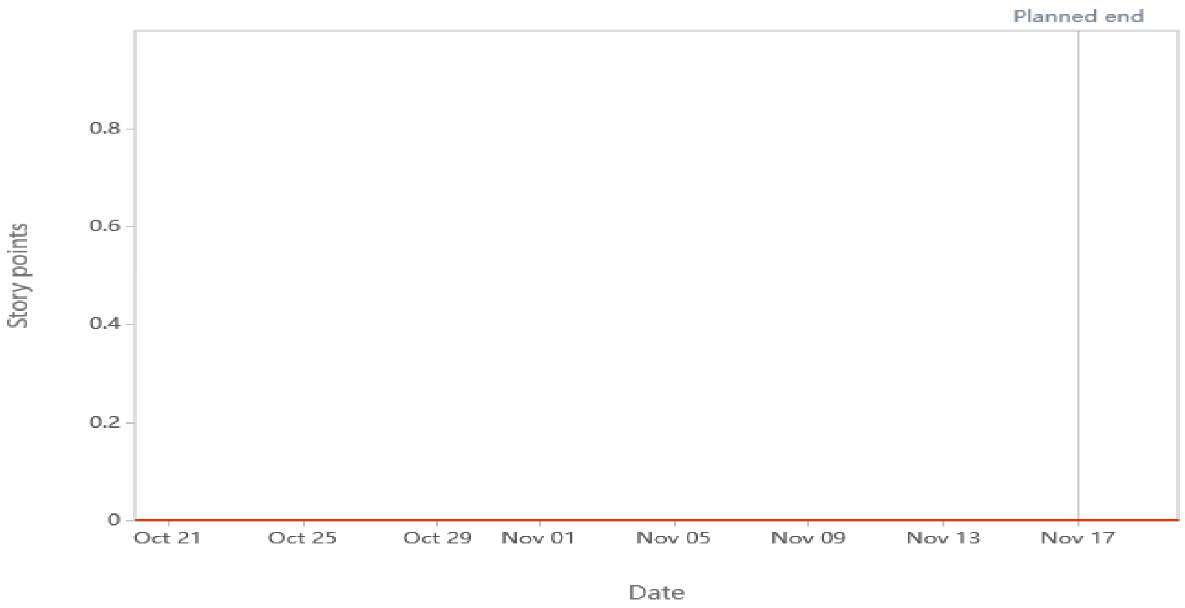
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	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:





18
15
12
9
6
3
0
Issue Count
04:30 20 Nov
04:32 20 Nov
04:41 20 Nov
Time



Planned end
0.8
0.6
0.4
0.2
0
Story points
Oct 21
Oct 25
Oct 29
Nov 01
Nov 05
Nov 09
Nov 13
Nov 17
Date

CHAPTER 7
CODING AND SOLUTIONING

7.1 FEATURE 1:

1. Address Bar based Features
   In this category 9 features are extracted.
2. Domain based Features
   In this category 4 features are extracted.
3. HTML & Javascript based Features
   In this category 4 features are extracted.

Due to the limited search engine and third-party methods discussed in the literature, we extract the particular features from the client side in our approach. We have introduced eleven hyperlink features (F3–F13), two login form features (F14 and F15), character level TF-IDF features (F2), and URL character sequence features (F1). All these features are discussed in the following subsections.

7.2 Feature 2:

URL character sequence features (F1)

The URL stands for Uniform Resource Locator. It is used for providing the location of the resources on the web such as images, files, hypertext, video, etc. URL. Each URL starts with a protocol (http, https, and ftp) used to access the resource requested. In this part, we extract character sequence features from URL. We employ the method used in[35] to process the URL at the character level. More information is contained at the character level. Phishers also imitate the URLs of legitimate websites by changing many unnoticeable characters, e.g., "www.icbc.com" as "www.1cbc.com". Character level URL processing is a solution to the out of vocabulary problem. Character level sequences identify substantial information from specific groups of characters that appear together which could be a symptom of phishing. In general, a URL is a string of characters or words where some words have little semantic meanings. Character sequences help find this sensitive information and improve the efficiency of phishing URL detection. During the learning task, machine learning techniques can be applied directly using the extracted character sequence features without the expert intervention. The main processes of character sequences generating include: preparing the character vocabulary, creating a tokenizer object using Keras preprocessing package (https://Keras.io) to process URLs in char level and add a "UNK" token to the vocabulary after the max value of chars dictionary, transforming text of URLs to sequence of tokens, and padding the sequence of URLs to ensure equal length vectors. The description of URL features extraction is shown in Algorithm 1.

HTML features

The webpage source code is the programming behind any webpage, or software. In case of websites, this code can be viewed by anyone using various tools, even in the web browser itself. In this section, we extract the textual and hyperlink features existing in the HTML source code of the webpage.

Textual content-based features (F2)

Script, CSS, img, and anchor files (F3, F4, F5, and F6)

Empty hyperlinks (F7 and F8)

Total hyperlinks feature (F9)

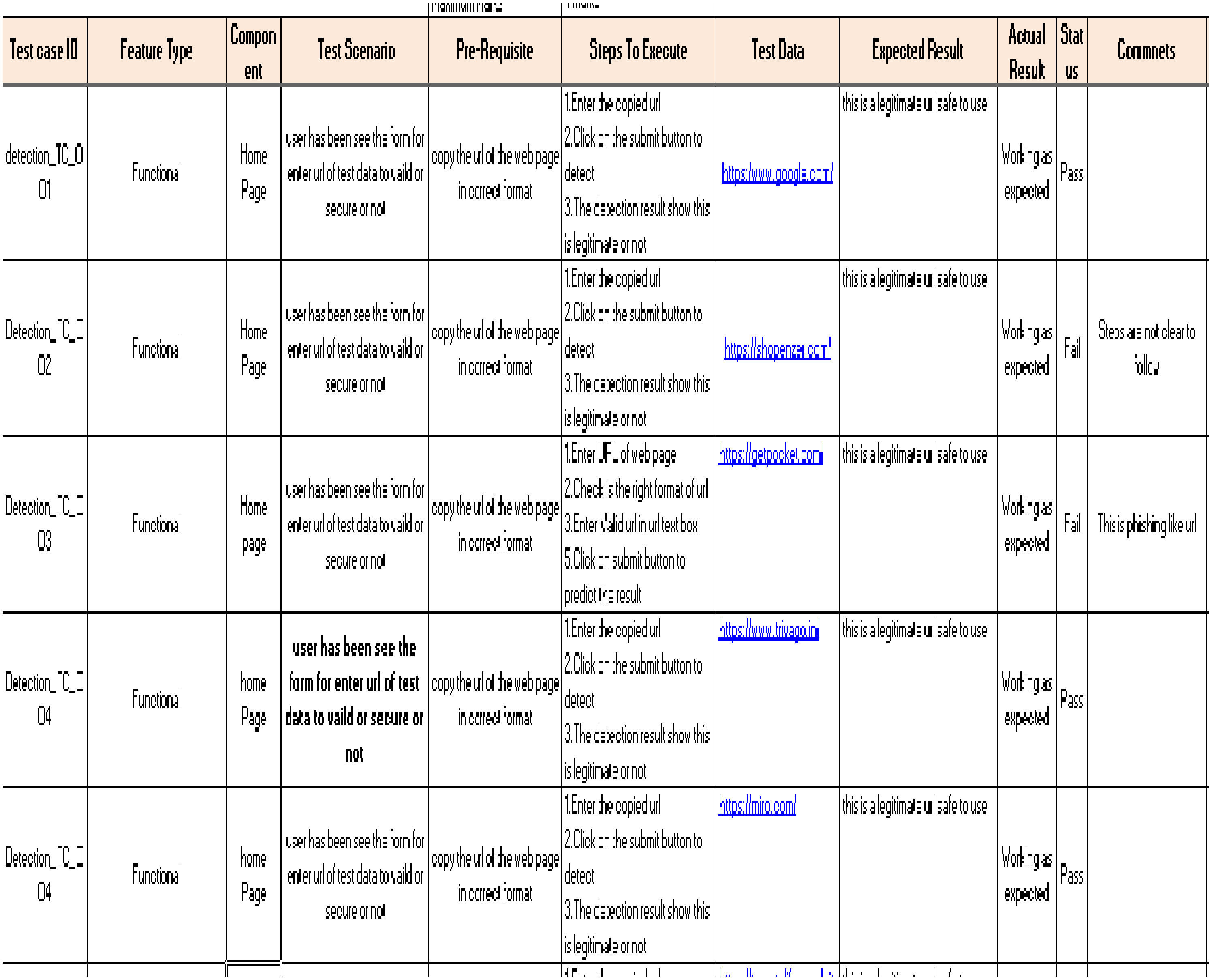
Internal and external hyperlinks (F10, F11, and F12)

Error in hyperlinks (F13)

Login form features (F14 and F15)

CHAPTER 8
TESTING

8.1 TEST CASES:



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets
detection_TC_O01	Functional	Home Page	user has been see the form for enter url of test data to vaild or secure or not	copy the url of the web page in ccrrect format	1.Enter the copied url 2.Click on the submit button to detect 3.The detection result show this is legitimate or not	https://www.google.com/	this is a legitimate url safe to use	Working as expected	Pass	
Detection_TC_O02	Functional	Home Page	user has been see the form for enter url of test data to vaild or secure or not	copy the url of the web page in ccrrect format	1.Enter the copied url 2.Click on the submit button to detect 3.The detection result show this is legitimate or not	https://shopenzer.com/	this is a legitimate url safe to use	Working as expected	Fail	Steps are not clear to follow
Detection_TC_O03	Functional	Home page	user has been see the form for enter url of test data to vaild or secure or not	copy the url of the web page in ccrrect format	1.Enter URL of web page 2.Check is the right format of url 3.Enter Valid url in url text box 5.Click on submit button to predict the result	https://getpocket.com/	this is a legitimate url safe to use	Working as expected	Fail	This is phishing like url
Detection_TC_O04	Functional	home Page	**user has been see the form for enter url of test data to vaild or secure or not**	copy the url of the web page in ccrrect format	1.Enter the copied url 2.Click on the submit button to detect 3.The detection result show this is legitimate or not	https://www.trivago.in/	this is a legitimate url safe to use	Working as expected	Pass	
Detection_TC_O04	Functional	home Page	user has been see the form for enter url of test data to vaild or secure or not	copy the url of the web page in ccrrect format	1.Enter the copied url 2.Click on the submit button to detect 3.The detection result show this is legitimate or not	https://miro.com/	this is a legitimate url safe to use	Working as expected	Pass	

8.2 USER ACCEPTANCE TESTING:

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	5	2	1	8
Totals	24	14	13	26	77

TestCaseAnalysis

This report shows the number of test cases that have passed ,failed ,and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	50	0	0	50
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	5	0	0	5

Version Control	3	0	0	3

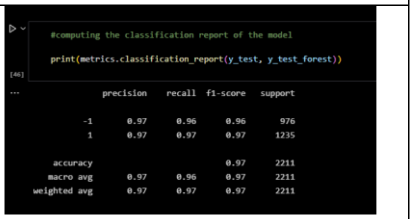
CHAPTER 9

RESULT

9.1 PERFORMANCE METRICS:

Web phishing detection project team’s performance testing using Random forest classification.

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Metrics</td><td><b>Classification Model:</b>
<b>Random forest classification</b>
Accuracy Score=96.6%</td><td>

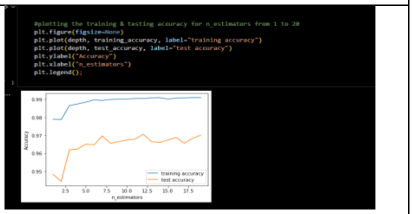


```
#computing the classification report of the model

print(metrics.classification_report(y_test, y_test_forest))
```

[46]

```
              precision    recall  f1-score   support

          -1       0.97      0.96      0.96       976
           1       0.97      0.97      0.97      1235

    accuracy                           0.97      2211
   macro avg       0.97      0.96      0.97      2211
weighted avg       0.97      0.97      0.97      2211
```

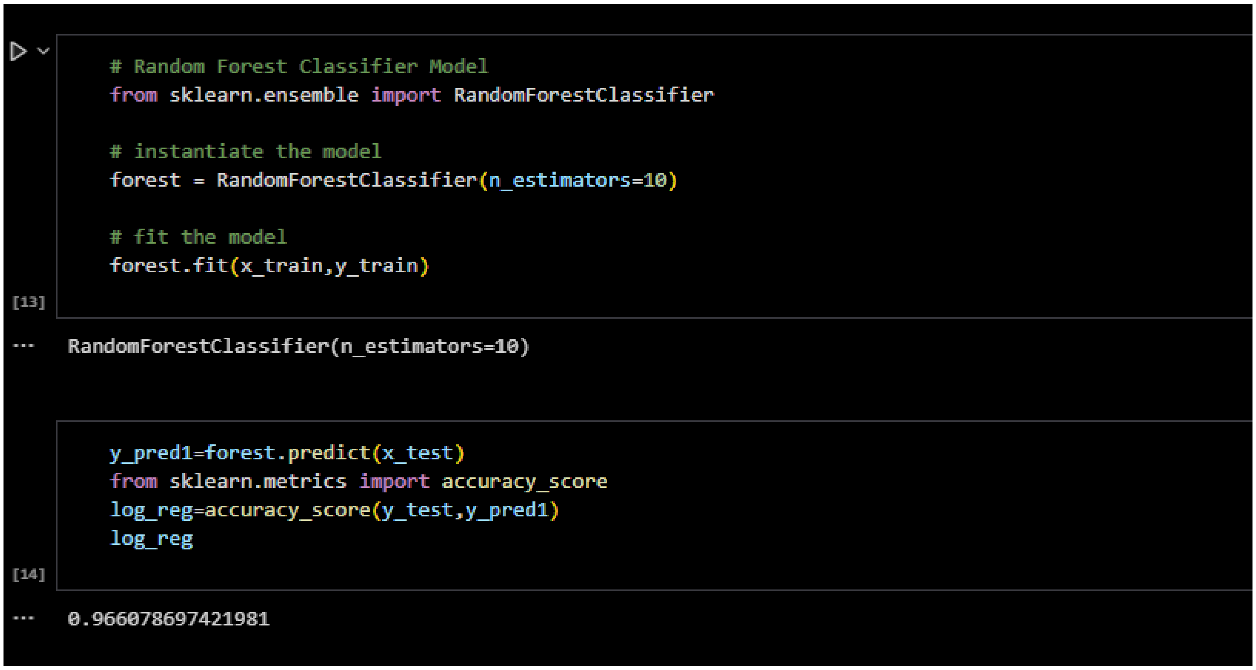
</td></tr>
<tr><td>2.</td><td>Tune the Model</td><td>Hyperparameter Tuning – 96%
Validation Method – forest&cross validation</td><td>



```
#plotting the training & testing accuracy for n_estimators from 1 to 20
plt.figure(figsize=None)
plt.plot(depth, training_accuracy, label="training accuracy")
plt.plot(depth, test_accuracy, label="test accuracy")
plt.ylabel("Accuracy")
plt.xlabel("n_estimators")
plt.legend();
```

</td></tr>
</table>

METRICS CLASSIFICATION :



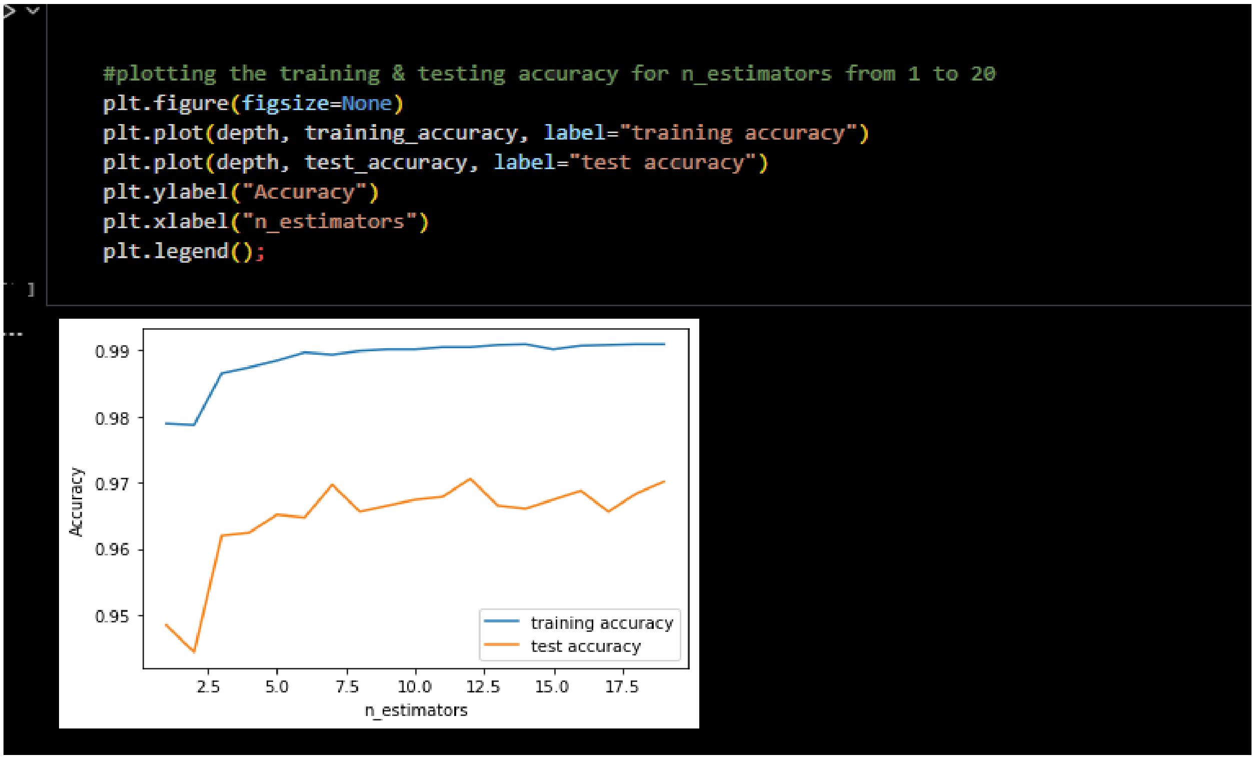
```
# Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest = RandomForestClassifier(n_estimators=10)

# fit the model
forest.fit(x_train,y_train)
```

```
RandomForestClassifier(n_estimators=10)
```

```
y_pred1=forest.predict(x_test)
from sklearn.metrics import accuracy_score
log_reg=accuracy_score(y_test,y_pred1)
log_reg
```

```
0.966078697421981
```

PERFORMANCE:



```
#plotting the training & testing accuracy for n_estimators from 1 to 20
plt.figure(figsize=None)
plt.plot(depth, training_accuracy, label="training accuracy")
plt.plot(depth, test_accuracy, label="test accuracy")
plt.ylabel("Accuracy")
plt.xlabel("n_estimators")
plt.legend();
```

CHAPTER 10

ADVANTAGES AND DIS-ADVANTAGES

ADVANTAGES:

- This system can be used by many E-commerce or other websites in order to have good customer relationship.
- User can make online payment securely.
- Data mining algorithm used in this system provides better performance as compared to other traditional classifications algorithms.

- With the help of this system user can also purchase products online without any hesitation

DISADVANTAGES:

- If Internet connection fails, this system won’t work.
- All websites related data will be stored in one place.

CHAPTER 11

CONCLUSION

CONCLUSION:

It is found that phishing attacks is very crucial and it is important for us to get a mechanism to detect it. As very important and personal information of the user can be leaked through phishing websites, it becomes more critical to take care of this issue. This problem can be easily solved by using any of the machine learning algorithm with the classifier. We already have classifiers which gives good prediction rate of the phishing beside, but after our survey that it will be better to use a hybrid approach for the prediction and further improve the accuracy prediction rate of phishing websites. We have seen that existing system gives less accuracy so we proposed a new phishing method that employs URL based features and also we generated classifiers through several machine learning.This apparatus ought to be improved continually through consistent retraining. As a matter of fact, the accessibility of crisp and cutting-edge preparing dataset which may gained utilizing our very own device [30, 32] will help us to retrain our model consistently and handle any adjustments in the highlights, which are influential in deciding the site class. Albeit neural system demonstrates its capacity to tackle a wide assortment of classification issues, the procedure of finding the ideal structure is very difficult, and much of the time, this structure is controlled by experimentation .Our model takes care of this issue via computerizing the way toward organizing a neural system conspire; hence, on the off chance that we construct an enemy of phishing model and for any reasons we have to refresh it, at that point our model will encourage this procedure, that is, since our model will mechanize the organizing procedure and will request scarcely any client defined parameters.

CHAPTER 12

FUTURE SCOPE

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.

APPENDIX

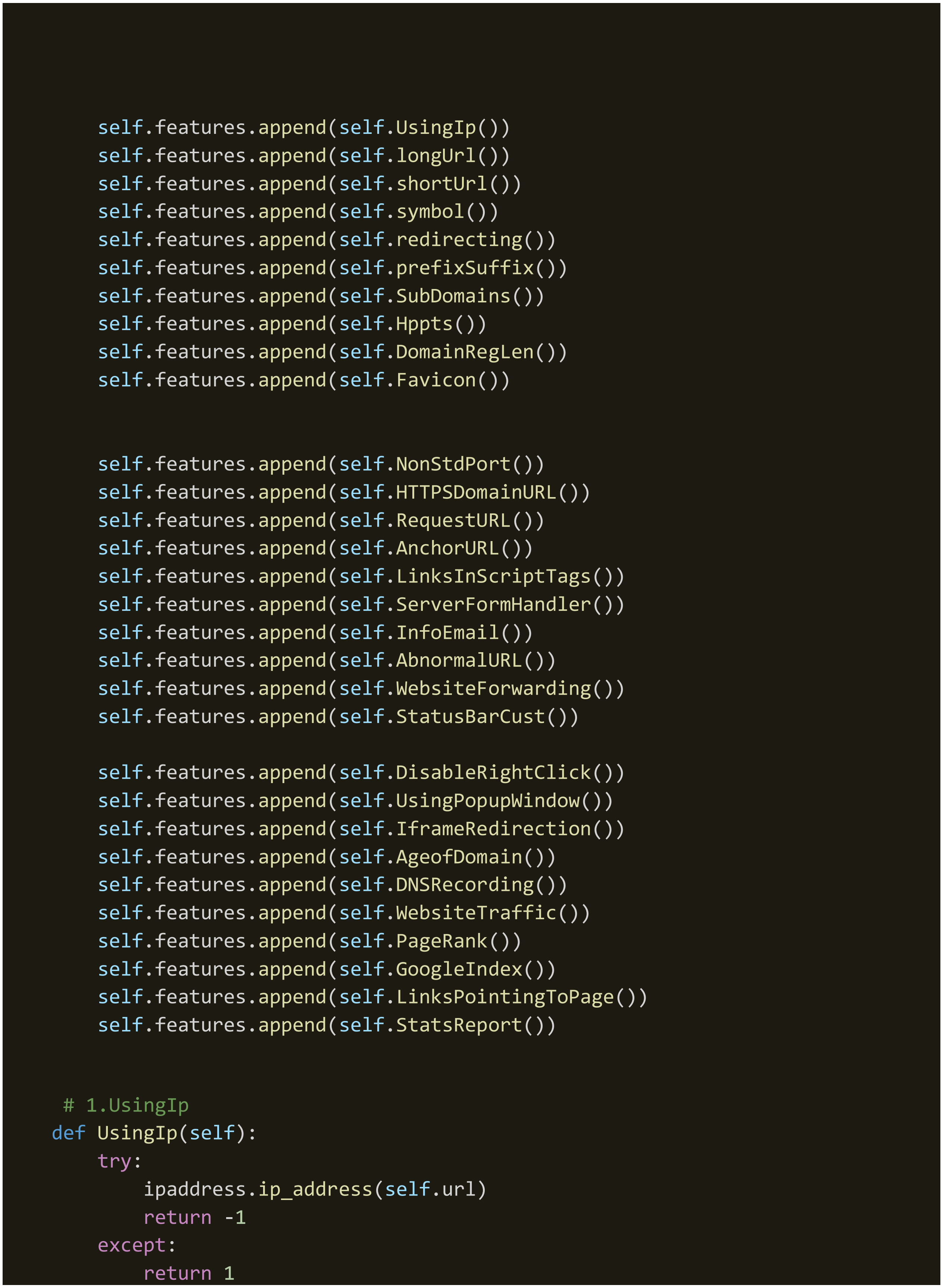
SOURCE CODE:

App.py:

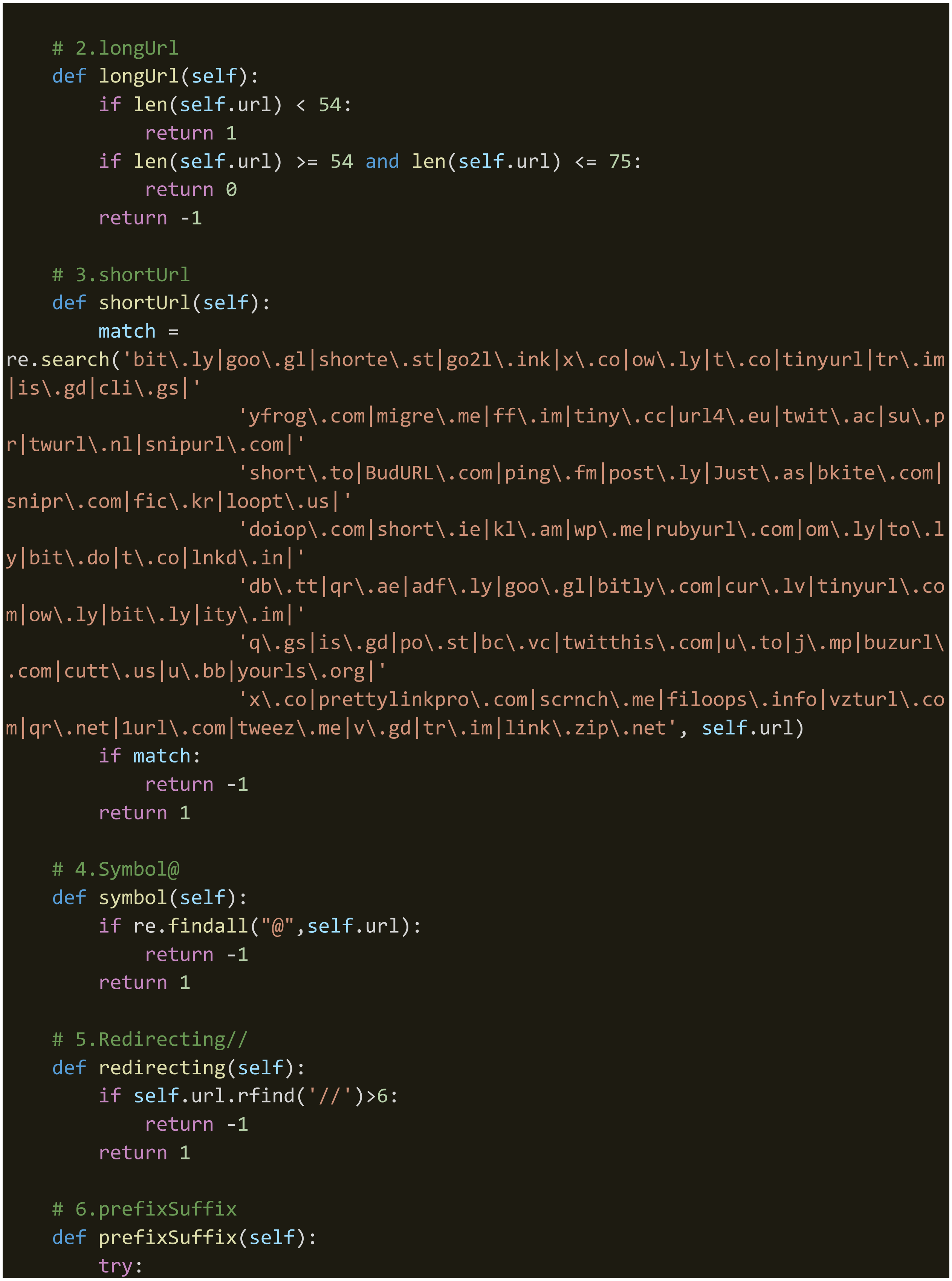
```
    #importing required libraries


from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings('ignore')
from feature import FeatureExtraction

file = open("phishing_website.pkl","rb")
forest= pickle.load(file)
file.close()


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)
        y_pred =forest.predict(x)[0]
        #1 is safe
        #-1 is unsafe
        y_pro_phishing = forest.predict_proba(x)[0,0]
        y_pro_non_phishing = forest.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
        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)
```

feature.py :

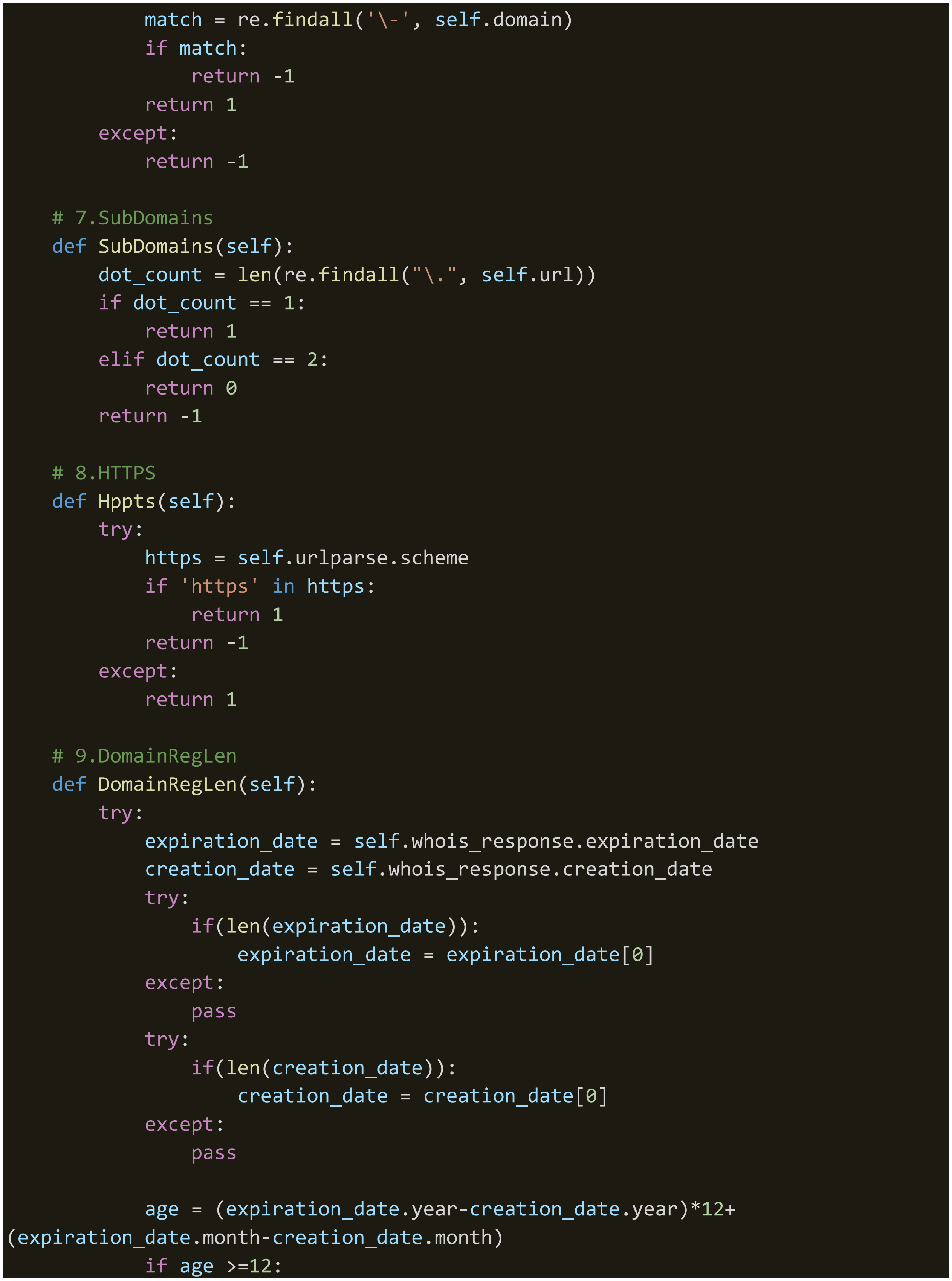
```
   import ipaddress

import re
import urllib.request
from bs4 import BeautifulSoup
import socket
import requests
from googlesearch import search
import whois
from datetime import date, datetime
import time
from dateutil.parser import parse as date_parse
from urllib.parse import urlparse

class FeatureExtraction:
    features = []
    def __init__(self,url):
        self.features = []
        self.url = url
        self.domain = ""
        self.whois_response = ""
        self.urlparse = ""
        self.response = ""
        self.soup = ""

        try:
            self.response = requests.get(url)
            self.soup = BeautifulSoup(response.text, 'html.parser')
        except:
            pass

        try:
            self.urlparse = urlparse(url)
            self.domain = self.urlparse.netloc
        except:
            pass

        try:
            self.whois_response = whois.whois(self.domain)
        except:
            pass
```



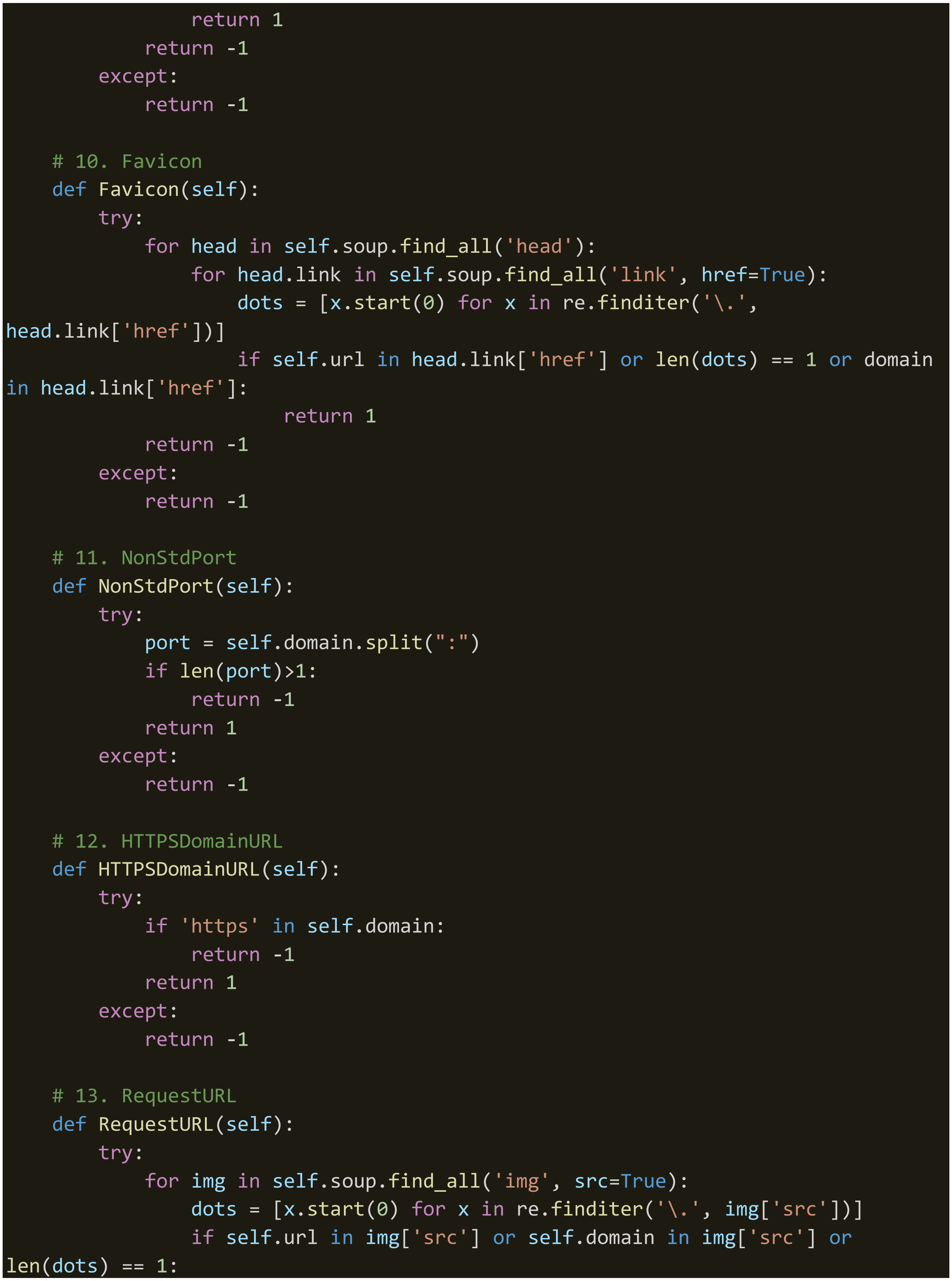
```python
    self.features.append(self.UsingIp())
    self.features.append(self.longUrl())
    self.features.append(self.shortUrl())
    self.features.append(self.symbol())
    self.features.append(self.redirecting())
    self.features.append(self.prefixSuffix())
    self.features.append(self.SubDomains())
    self.features.append(self.Hppts())
    self.features.append(self.DomainRegLen())
    self.features.append(self.Favicon())


    self.features.append(self.NonStdPort())
    self.features.append(self.HTTPSDomainURL())
    self.features.append(self.RequestURL())
    self.features.append(self.AnchorURL())
    self.features.append(self.LinksInScriptTags())
    self.features.append(self.ServerFormHandler())
    self.features.append(self.InfoEmail())
    self.features.append(self.AbnormalURL())
    self.features.append(self.WebsiteForwarding())
    self.features.append(self.StatusBarCust())

    self.features.append(self.DisableRightClick())
    self.features.append(self.UsingPopupWindow())
    self.features.append(self.IframeRedirection())
    self.features.append(self.AgeofDomain())
    self.features.append(self.DNSRecording())
    self.features.append(self.WebsiteTraffic())
    self.features.append(self.PageRank())
    self.features.append(self.GoogleIndex())
    self.features.append(self.LinksPointingToPage())
    self.features.append(self.StatsReport())


 # 1.UsingIp
def UsingIp(self):
    try:
        ipaddress.ip_address(self.url)
        return -1
    except:
        return 1
```



```
    # 2.longUrl
    def longUrl(self):
        if len(self.url) < 54:
            return 1
        if len(self.url) >= 54 and len(self.url) <= 75:
            return 0
        return -1

    # 3.shortUrl
    def shortUrl(self):
        match =
re.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im
|is\.gd|cli\.gs|'
                    'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.p
r|twurl\.nl|snipurl\.com|'
                    'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|
snipr\.com|fic\.kr|loopt\.us|'
                    'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.l
y|bit\.do|t\.co|lnkd\.in|'
                    'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.co
m|ow\.ly|bit\.ly|ity\.im|'
                    'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\
.com|cutt\.us|u\.bb|yourls\.org|'
                    'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.co
m|qr\.net|1url\.com|tweez\.me|v\.gd|tr\.im|link\.zip\.net', self.url)
        if match:
            return -1
        return 1

    # 4.Symbol@
    def symbol(self):
        if re.findall("@",self.url):
            return -1
        return 1

    # 5.Redirecting//
    def redirecting(self):
        if self.url.rfind('//')>6:
            return -1
        return 1

    # 6.prefixSuffix
    def prefixSuffix(self):
        try:
```

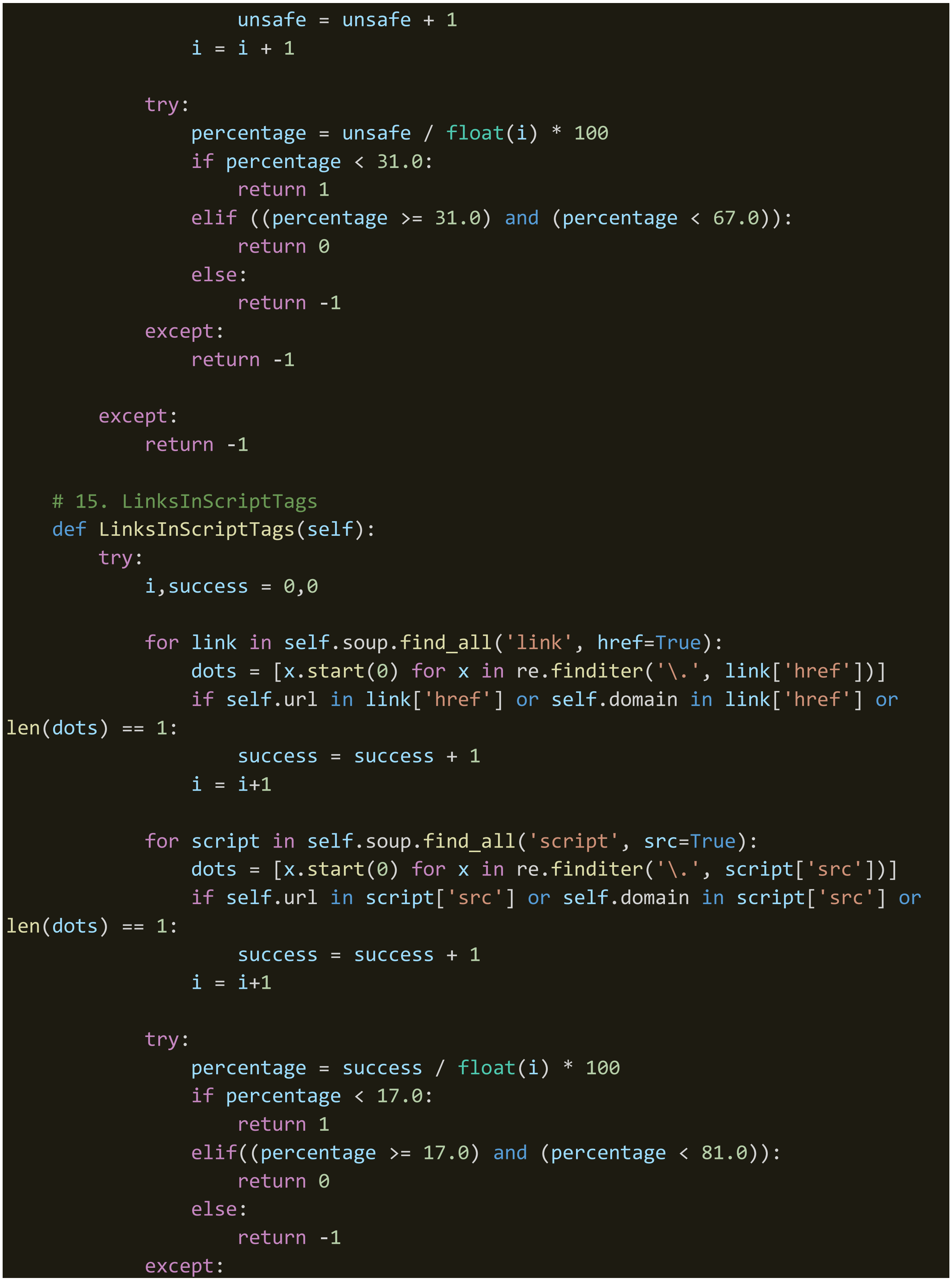


```
            match = re.findall('\-', self.domain)
            if match:
                return -1
            return 1
        except:
            return -1

    # 7.SubDomains
    def SubDomains(self):
        dot_count = len(re.findall("\.", self.url))
        if dot_count == 1:
            return 1
        elif dot_count == 2:
            return 0
        return -1

    # 8.HTTPS
    def Hppts(self):
        try:
            https = self.urlparse.scheme
            if 'https' in https:
                return 1
            return -1
        except:
            return 1

    # 9.DomainRegLen
    def DomainRegLen(self):
        try:
            expiration_date = self.whois_response.expiration_date
            creation_date = self.whois_response.creation_date
            try:
                if(len(expiration_date)):
                    expiration_date = expiration_date[0]
            except:
                pass
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            age = (expiration_date.year-creation_date.year)*12+
(expiration_date.month-creation_date.month)
            if age >=12:
```



```
                return 1
            return -1
        except:
            return -1

    # 10. Favicon
    def Favicon(self):
        try:
            for head in self.soup.find_all('head'):
                for head.link in self.soup.find_all('link', href=True):
                    dots = [x.start(0) for x in re.finditer('\.',
head.link['href'])]
                    if self.url in head.link['href'] or len(dots) == 1 or domain
in head.link['href']:
                        return 1
            return -1
        except:
            return -1

    # 11. NonStdPort
    def NonStdPort(self):
        try:
            port = self.domain.split(":")
            if len(port)>1:
                return -1
            return 1
        except:
            return -1

    # 12. HTTPSDomainURL
    def HTTPSDomainURL(self):
        try:
            if 'https' in self.domain:
                return -1
            return 1
        except:
            return -1

    # 13. RequestURL
    def RequestURL(self):
        try:
            for img in self.soup.find_all('img', src=True):
                dots = [x.start(0) for x in re.finditer('\.', img['src'])]
                if self.url in img['src'] or self.domain in img['src'] or
len(dots) == 1:
```

```
                    success = success + 1
                i = i+1

            for audio in self.soup.find_all('audio', src=True):
                dots = [x.start(0) for x in re.finditer('\.', audio['src'])]
                if self.url in audio['src'] or self.domain in audio['src'] or
len(dots) == 1:
                    success = success + 1
                i = i+1

            for embed in self.soup.find_all('embed', src=True):
                dots = [x.start(0) for x in re.finditer('\.', embed['src'])]
                if self.url in embed['src'] or self.domain in embed['src'] or
len(dots) == 1:
                    success = success + 1
                i = i+1

            for iframe in self.soup.find_all('iframe', src=True):
                dots = [x.start(0) for x in re.finditer('\.', iframe['src'])]
                if self.url in iframe['src'] or self.domain in iframe['src'] or
len(dots) == 1:
                    success = success + 1
                i = i+1

            try:
                percentage = success/float(i) * 100
                if percentage < 22.0:
                    return 1
                elif((percentage >= 22.0) and (percentage < 61.0)):
                    return 0
                else:
                    return -1
            except:
                return 0
        except:
            return -1

    # 14. AnchorURL
    def AnchorURL(self):
        try:
            i,unsafe = 0,0
            for a in self.soup.find_all('a', href=True):
                if "#" in a['href'] or "javascript" in a['href'].lower() or
"mailto" in a['href'].lower() or not (url in a['href'] or self.domain in
a['href']):
```

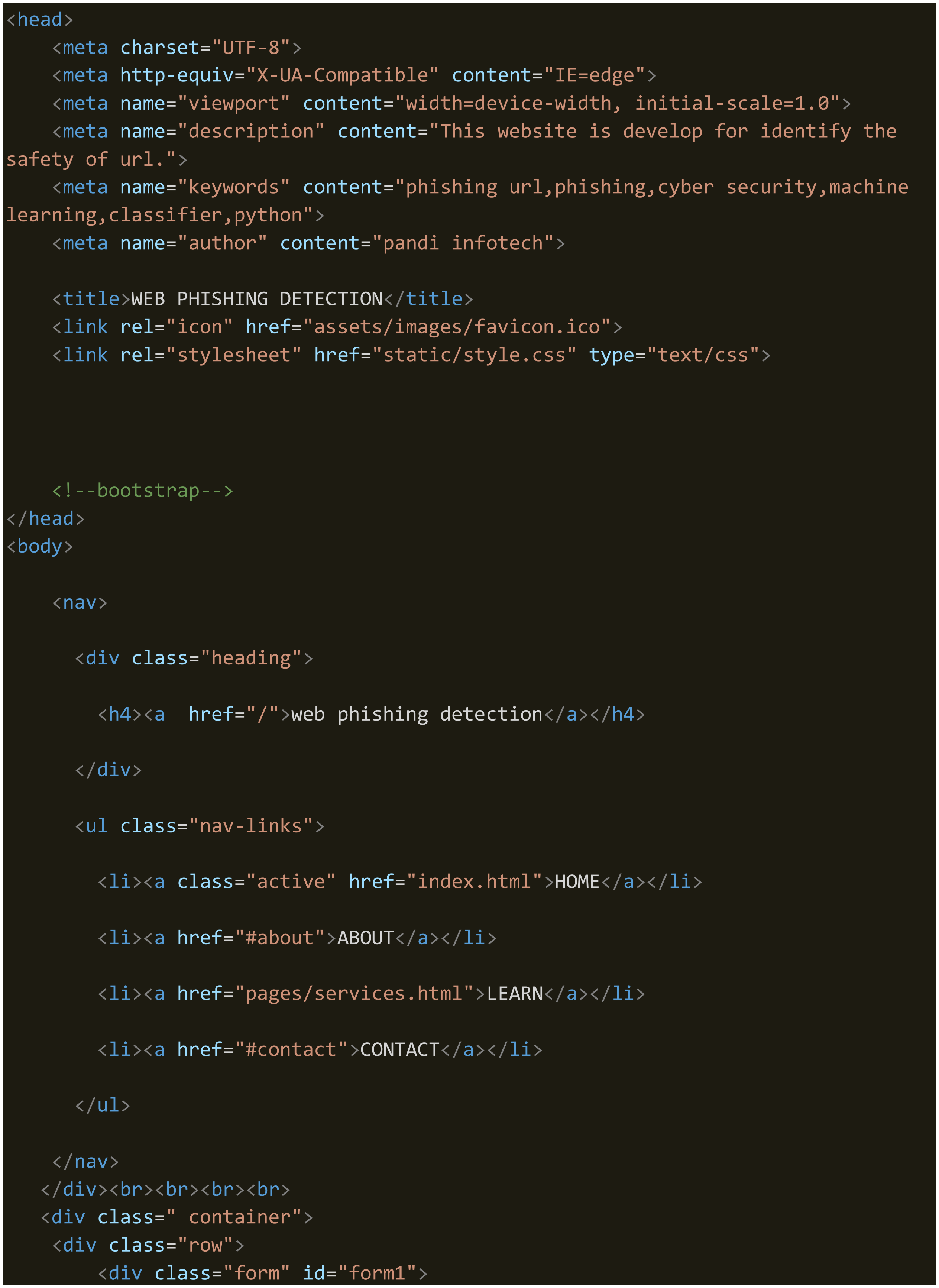


```
                    unsafe = unsafe + 1
                i = i + 1

            try:
                percentage = unsafe / float(i) * 100
                if percentage < 31.0:
                    return 1
                elif ((percentage >= 31.0) and (percentage < 67.0)):
                    return 0
                else:
                    return -1
            except:
                return -1

        except:
            return -1

    # 15. LinksInScriptTags
    def LinksInScriptTags(self):
        try:
            i,success = 0,0

            for link in self.soup.find_all('link', href=True):
                dots = [x.start(0) for x in re.finditer('\.', link['href'])]
                if self.url in link['href'] or self.domain in link['href'] or
len(dots) == 1:
                    success = success + 1
                i = i+1

            for script in self.soup.find_all('script', src=True):
                dots = [x.start(0) for x in re.finditer('\.', script['src'])]
                if self.url in script['src'] or self.domain in script['src'] or
len(dots) == 1:
                    success = success + 1
                i = i+1

            try:
                percentage = success / float(i) * 100
                if percentage < 17.0:
                    return 1
                elif((percentage >= 17.0) and (percentage < 81.0)):
                    return 0
                else:
                    return -1
            except:
```

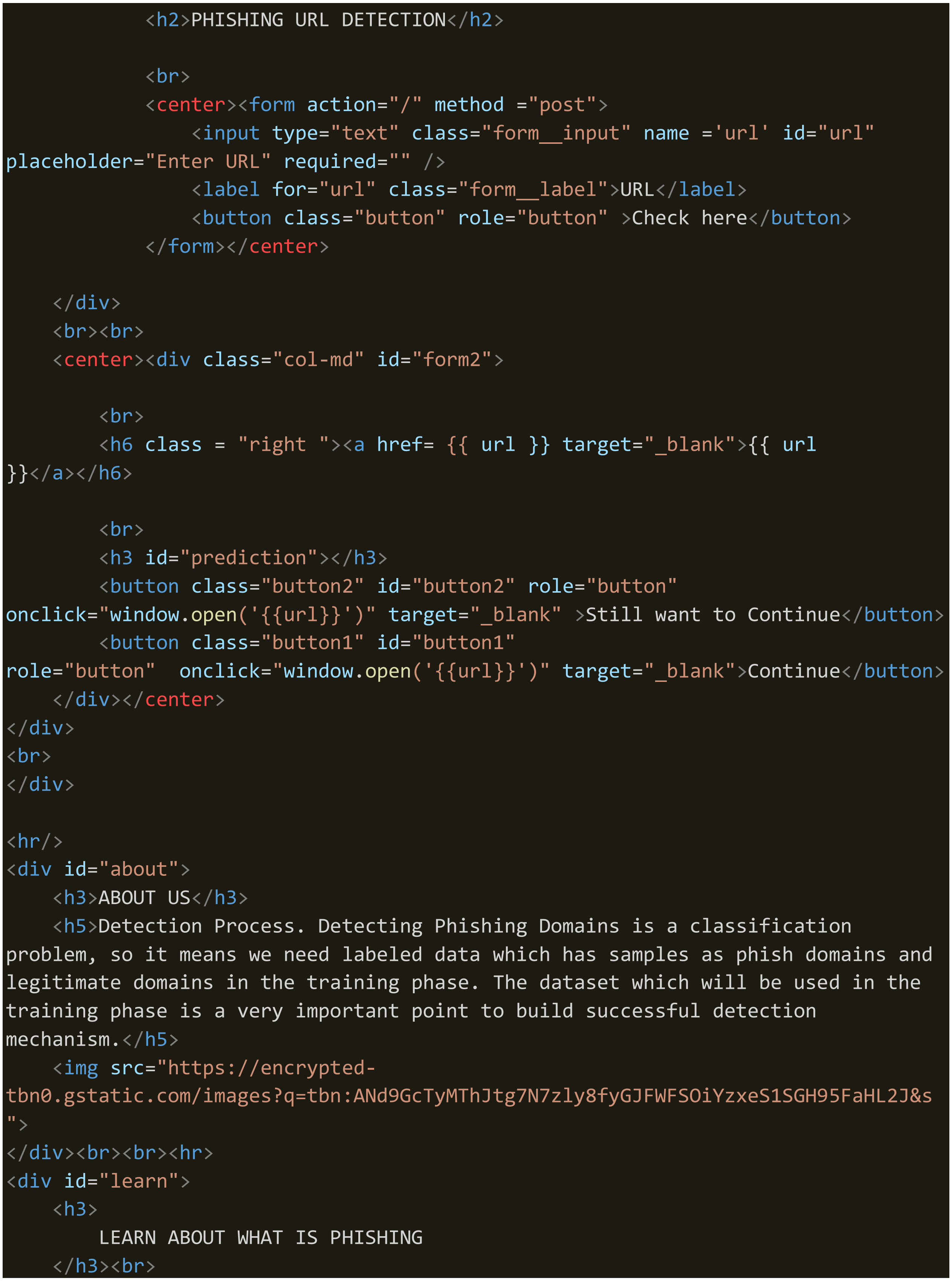
```python
                return 0
        except:
            return -1

    # 16. ServerFormHandler
    def ServerFormHandler(self):
        try:
            if len(self.soup.find_all('form', action=True))==0:
                return 1
            else :
                for form in self.soup.find_all('form', action=True):
                    if form['action'] == "" or form['action'] == "about:blank":
                        return -1
                    elif self.url not in form['action'] and self.domain not in form['action']:
                        return 0
                    else:
                        return 1
        except:
            return -1

    # 17. InfoEmail
    def InfoEmail(self):
        try:
            if re.findall(r"[mail\(\)|mailto:?]", self.soap):
                return -1
            else:
                return 1
        except:
            return -1

    # 18. AbnormalURL
    def AbnormalURL(self):
        try:
            if self.response.text == self.whois_response:
                return 1
            else:
                return -1
        except:
            return -1

    # 19. WebsiteForwarding
    def WebsiteForwarding(self):
        try:
            if len(self.response.history) <= 1:
                return 1
            elif len(self.response.history) <= 4:
                return 0
            else:
                return -1
        except:
             return -1

    # 20. StatusBarCust
    def StatusBarCust(self):
        try:
            if re.findall("<script>.+onmouseover.+</script>",
self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 21. DisableRightClick
    def DisableRightClick(self):
        try:
            if re.findall(r"event.button ?== ?2", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 22. UsingPopupWindow
    def UsingPopupWindow(self):
        try:
            if re.findall(r"alert\(", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 23. IframeRedirection
    def IframeRedirection(self):
        try:
            if re.findall(r"[<iframe>|<frameBorder>]", self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

    # 24. AgeofDomain
    def AgeofDomain(self):
        try:
            creation_date = self.whois_response.creation_date
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-
creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 25. DNSRecording
    def DNSRecording(self):
        try:
            creation_date = self.whois_response.creation_date
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-
creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 26. WebsiteTraffic
    def WebsiteTraffic(self):
        try:
            rank = 
BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=10&dat=s&url
=" + url).read(), "xml").find("REACH")['RANK']
            if (int(rank) < 100000):
                return 1
            return 0
        except :
            return -1

    # 27. PageRank
    def PageRank(self):
        try:
            prank_checker_response = 
requests.post("https://www.checkpagerank.net/index.php", {"name": self.domain})

            global_rank = int(re.findall(r"Global Rank: ([0-9]+)", 
rank_checker_response.text)[0])
            if global_rank > 0 and global_rank < 100000:
                return 1
            return -1
        except:
            return -1


    # 28. GoogleIndex
    def GoogleIndex(self):
        try:
            site = search(self.url, 5)
            if site:
                return 1
            else:
                return -1
        except:
            return 1

    # 29. LinksPointingToPage
    def LinksPointingToPage(self):
        try:
            number_of_links = len(re.findall(r"<a href=", self.response.text))
            if number_of_links == 0:
                return 1
            elif number_of_links <= 2:
                return 0
            else:
                return -1
        except:
            return -1

    # 30. StatsReport
    def StatsReport(self):
        try:
            url_match = re.search(
        'at\.ua|usa\.cc|baltazarpresentes\.com\.br|pe\.hu|esy\.es|hol\.es|sweddy\.com|myjino\.ru|96\.lt|ow\.ly', url)
            ip_address = socket.gethostbyname(self.domain)
            ip_match = re.search('146\.112\.61\.108|213\.174\.157\.151|121\.50\.168\.88|192\.185\.217\.116|78\.46\.211\.158|181\.174\.165\.13|46\.242\.145\.103|121\.50\.168\.40|83\.125\.22\.219|46\.242\.145\.98|'
                                '107\.151\.148\.44|107\.151\.148\.107|64\.70\.19\.203|199\.184\.144\.27|107\.151\.148\.108|107\.151\.148\.109|119\.28\.52\.61|54\.83\.43\.69|52\.69\.166\.231|216\.58\.192\.225|'
                                '118\.184\.25\.86|67\.208\.74\.71|23\.253\.126\.58|104\.239\.157\.210|175\.126\.123\.219|141\.8\.224\.221|10\.10\.10\.10|43\.229\.108\.32|103\.232\.215\.140|69\.172\.201\.153|'
                                '216\.218\.185\.162|54\.225\.104\.146|103\.243\.24\.98|199\.59\.243\.120|31\.170\.160\.61|213\.19\.128\.77|62\.113\.226\.131|208\.100\.26\.234|195\.16\.127\.102|195\.16\.127\.157|'
                                '34\.196\.13\.28|103\.224\.212\.222|172\.217\.4\.225|54\.72\.9\.51|192\.64\.147\.141|198\.200\.56\.183|23\.253\.164\.103|52\.48\.191\.26|52\.214\.197\.72|87\.98\.255\.18|209\.99\.17\.27|'
                                '216\.38\.62\.18|104\.130\.124\.96|47\.89\.58\.141|78\.46\.211\.158|54\.86\.225\.156|54\.82\.156\.19|37\.157\.192\.102|204\.11\.56\.48|110\.34\.231\.42', ip_address)
            if url_match:
                return -1
            elif ip_match:
                return -1
            return 1
        except:
            return 1

    def getFeaturesList(self):
        return self.features
```

index.html :

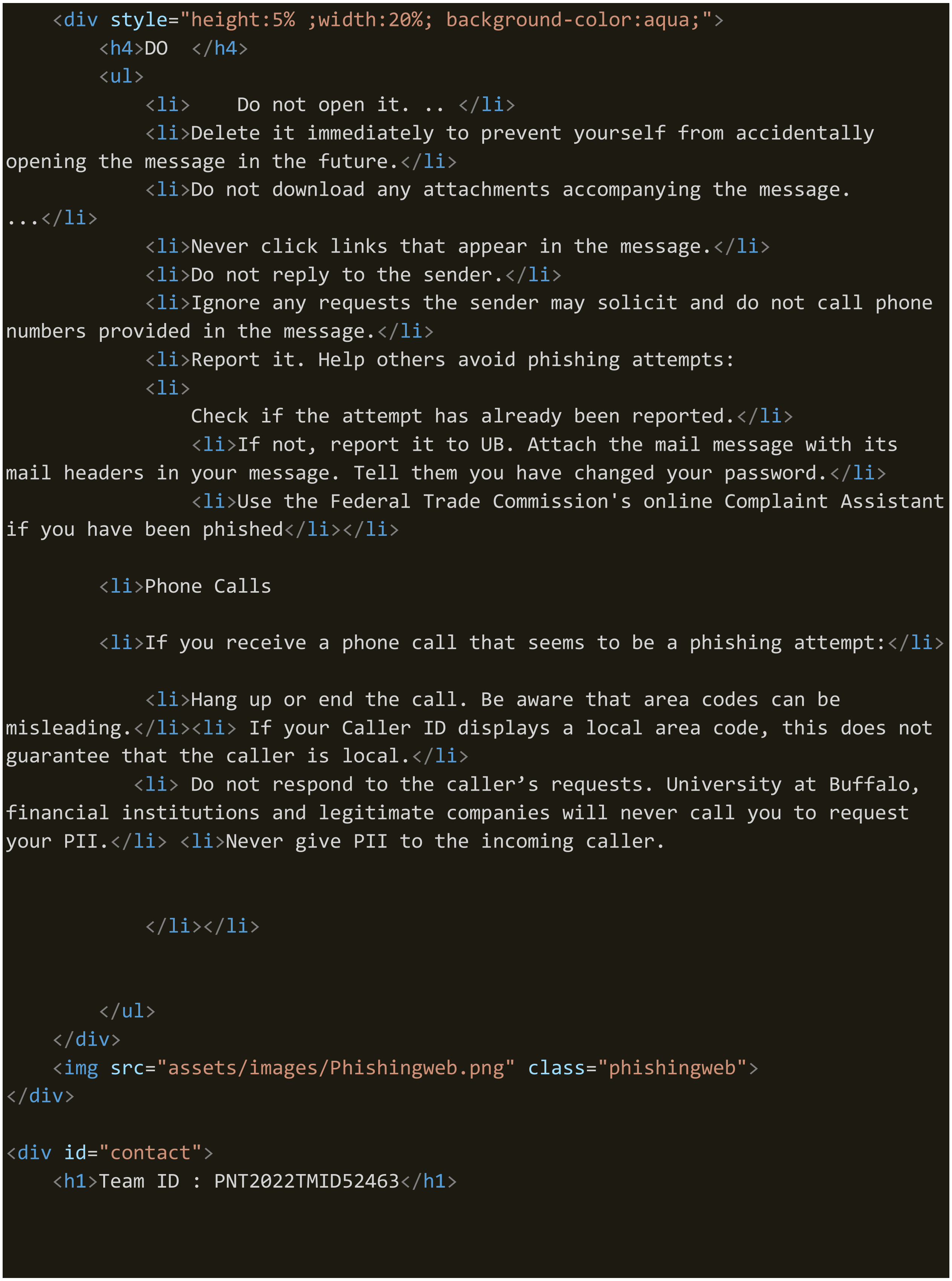
```
  <!DOCTYPE html>

<html lang="en">
```



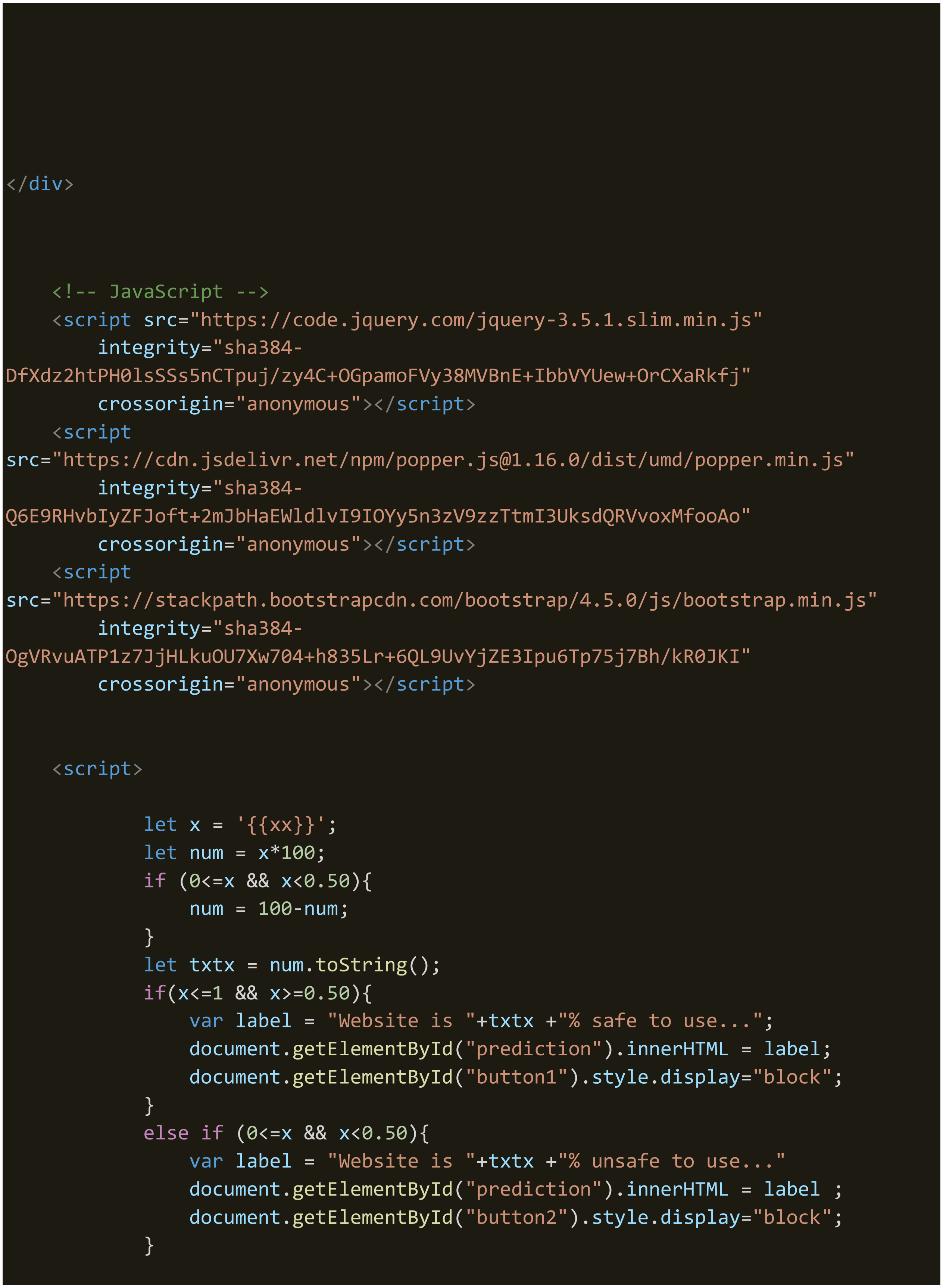
```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="This website is develop for identify the
safety of url.">
    <meta name="keywords" content="phishing url,phishing,cyber security,machine
learning,classifier,python">
    <meta name="author" content="pandi infotech">

    <title>WEB PHISHING DETECTION</title>
    <link rel="icon" href="assets/images/favicon.ico">
    <link rel="stylesheet" href="static/style.css" type="text/css">




    <!--bootstrap-->
</head>
<body>

    <nav>

      <div class="heading">

        <h4><a  href="/">web phishing detection</a></h4>

      </div>

      <ul class="nav-links">

        <li><a class="active" href="index.html">HOME</a></li>

        <li><a href="#about">ABOUT</a></li>

        <li><a href="pages/services.html">LEARN</a></li>

        <li><a href="#contact">CONTACT</a></li>

      </ul>

    </nav>
   </div><br><br><br><br>
   <div class=" container">
    <div class="row">
        <div class="form" id="form1">
```



```
            <h2>PHISHING URL DETECTION</h2>

            <br>
            <center><form action="/" method ="post">
                <input type="text" class="form__input" name ='url' id="url"
placeholder="Enter URL" required="" />
                <label for="url" class="form__label">URL</label>
                <button class="button" role="button" >Check here</button>
            </form></center>

    </div>
    <br><br>
    <center><div class="col-md" id="form2">

        <br>
        <h6 class = "right "><a href= {{ url }} target="_blank">{{ url
}}</a></h6>

        <br>
        <h3 id="prediction"></h3>
        <button class="button2" id="button2" role="button"
onclick="window.open('{{url}}')" target="_blank" >Still want to Continue</button>
        <button class="button1" id="button1"
role="button"  onclick="window.open('{{url}}')" target="_blank">Continue</button>
    </div></center>
</div>
<br>
</div>

<hr/>
<div id="about">
    <h3>ABOUT US</h3>
    <h5>Detection Process. Detecting Phishing Domains is a classification
problem, so it means we need labeled data which has samples as phish domains and
legitimate domains in the training phase. The dataset which will be used in the
training phase is a very important point to build successful detection
mechanism.</h5>
    <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTyMThJtg7N7zly8fyGJFWFSOiYzxeS1SGH95FaHL2J&s
">
</div><br><br><hr>
<div id="learn">
    <h3>
        LEARN ABOUT WHAT IS PHISHING
    </h3><br>
```



```
    <div style="height:5% ;width:20%; background-color:aqua;">
        <h4>DO  </h4>
        <ul>
            <li>    Do not open it. .. </li>
            <li>Delete it immediately to prevent yourself from accidentally
opening the message in the future.</li>
            <li>Do not download any attachments accompanying the message.
...</li>
            <li>Never click links that appear in the message.</li>
            <li>Do not reply to the sender.</li>
            <li>Ignore any requests the sender may solicit and do not call phone
numbers provided in the message.</li>
            <li>Report it. Help others avoid phishing attempts:
            <li>
                Check if the attempt has already been reported.</li>
                <li>If not, report it to UB. Attach the mail message with its
mail headers in your message. Tell them you have changed your password.</li>
                <li>Use the Federal Trade Commission's online Complaint Assistant
if you have been phished</li></li>

        <li>Phone Calls

        <li>If you receive a phone call that seems to be a phishing attempt:</li>

            <li>Hang up or end the call. Be aware that area codes can be
misleading.</li><li> If your Caller ID displays a local area code, this does not
guarantee that the caller is local.</li>
           <li> Do not respond to the caller’s requests. University at Buffalo,
financial institutions and legitimate companies will never call you to request
your PII.</li> <li>Never give PII to the incoming caller.


            </li></li>


        </ul>
    </div>
    <img src="assets/images/Phishingweb.png" class="phishingweb">
</div>

<div id="contact">
    <h1>Team ID : PNT2022TMID52463</h1>
```

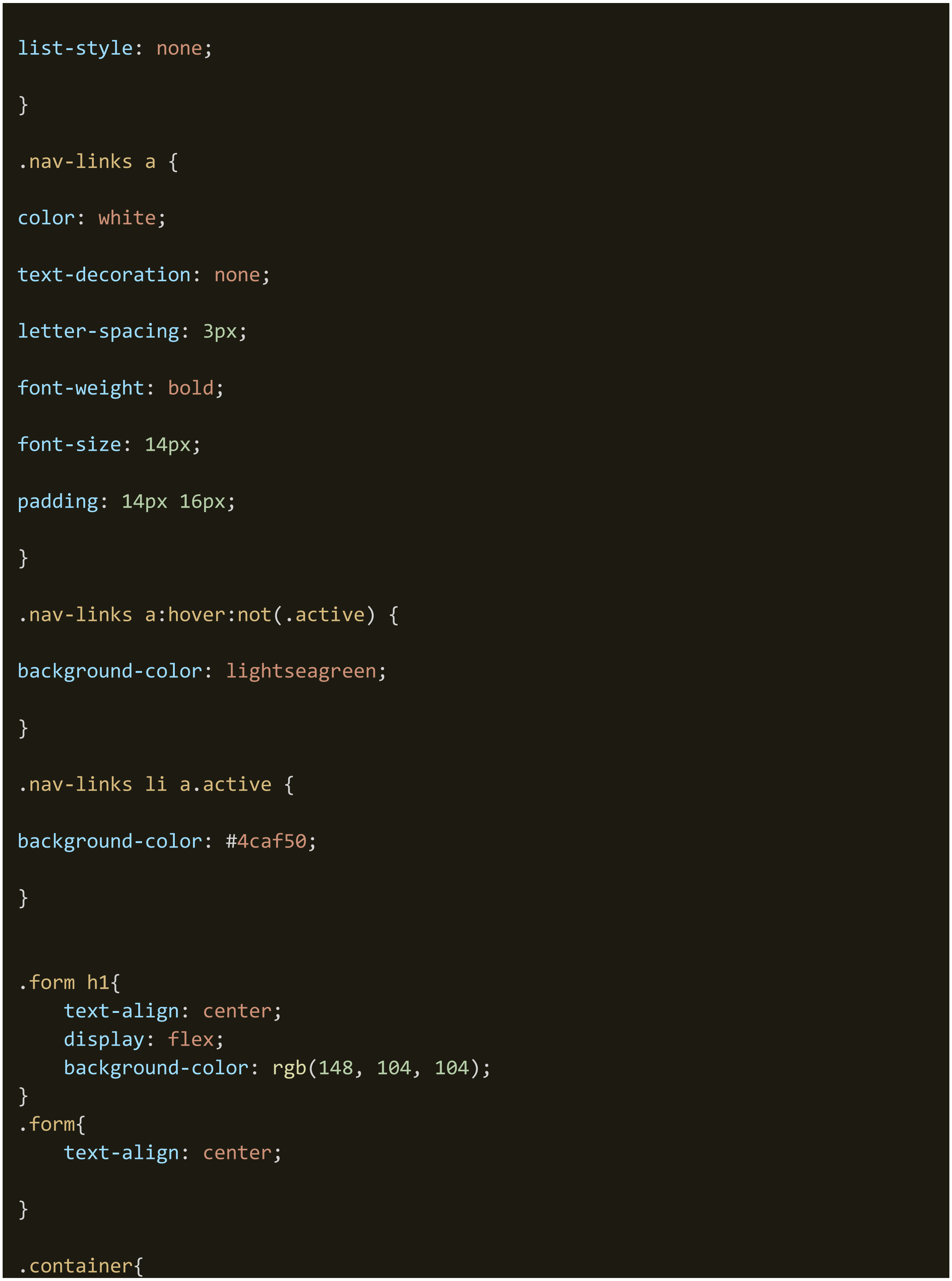


```
</div>



    <!-- JavaScript -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
        integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
        integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
        crossorigin="anonymous"></script>


    <script>

            let x = '{{xx}}';
            let num = x*100;
            if (0<=x && x<0.50){
                num = 100-num;
            }
            let txtx = num.toString();
            if(x<=1 && x>=0.50){
                var label = "Website is "+txtx +"% safe to use...";
                document.getElementById("prediction").innerHTML = label;
                document.getElementById("button1").style.display="block";
            }
            else if (0<=x && x<0.50){
                var label = "Website is "+txtx +"% unsafe to use..."
                document.getElementById("prediction").innerHTML = label ;
                document.getElementById("button2").style.display="block";
            }
    </script>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><hr><div>
   <footer><center><p>ABS PHISHING DETECTOR © 2022 PANDI INFOTECH</p></center></footer></div>
</body>
</html>
```

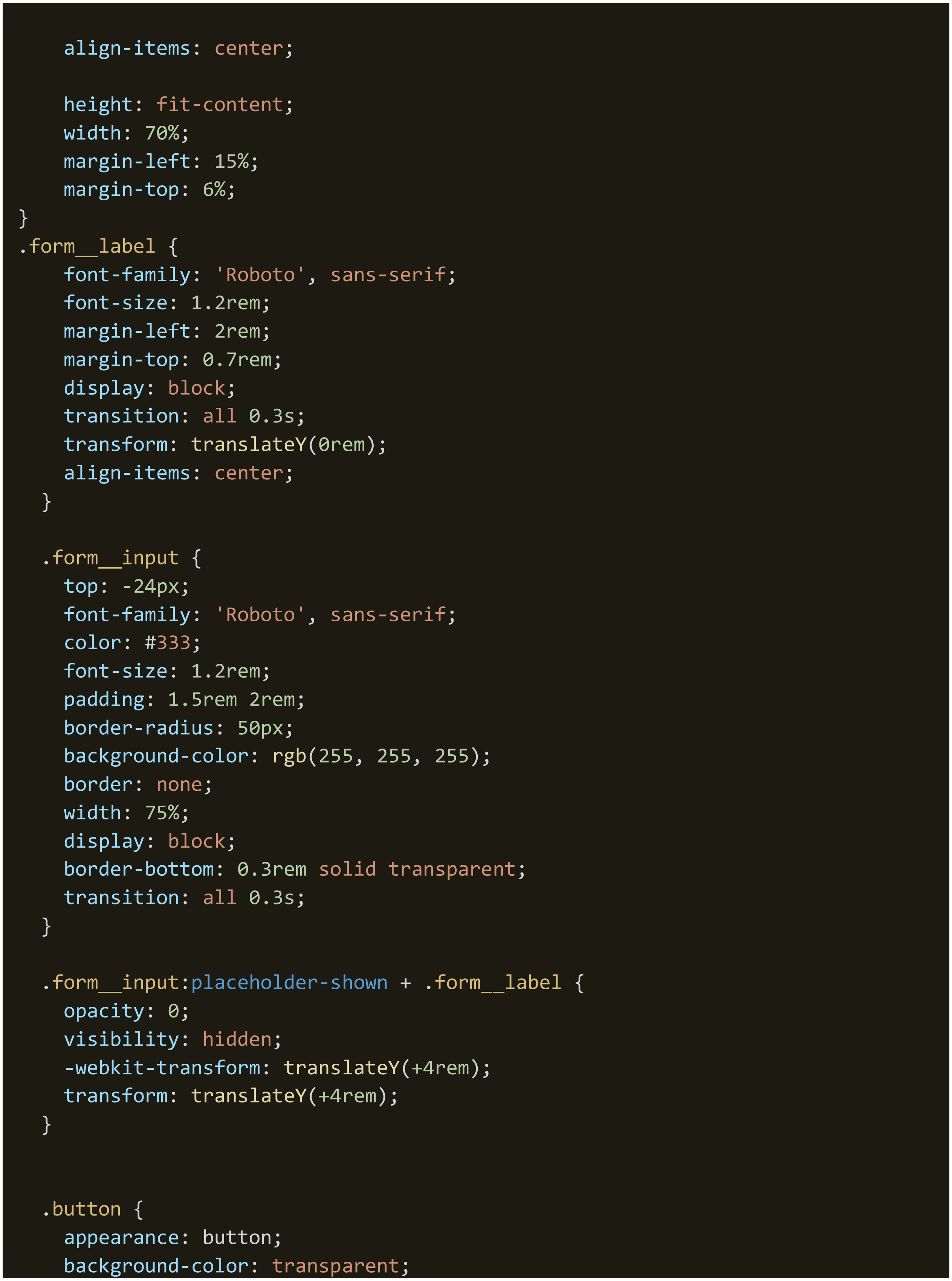
Style.css :

```
body{
  background-color: rgb(23, 153, 170);
  background-repeat: no-repeat;
  background-size: 100%;
  background-attachment:fixed;

}
    * {

 margin: 0px;

 padding: 0px;

 box-sizing: border-box;

 }

 .body-text {

 display: flex;

 font-family: "Montserrat", sans-serif;

 align-items: center;

 justify-content: center;

 margin-top: 250px;
```

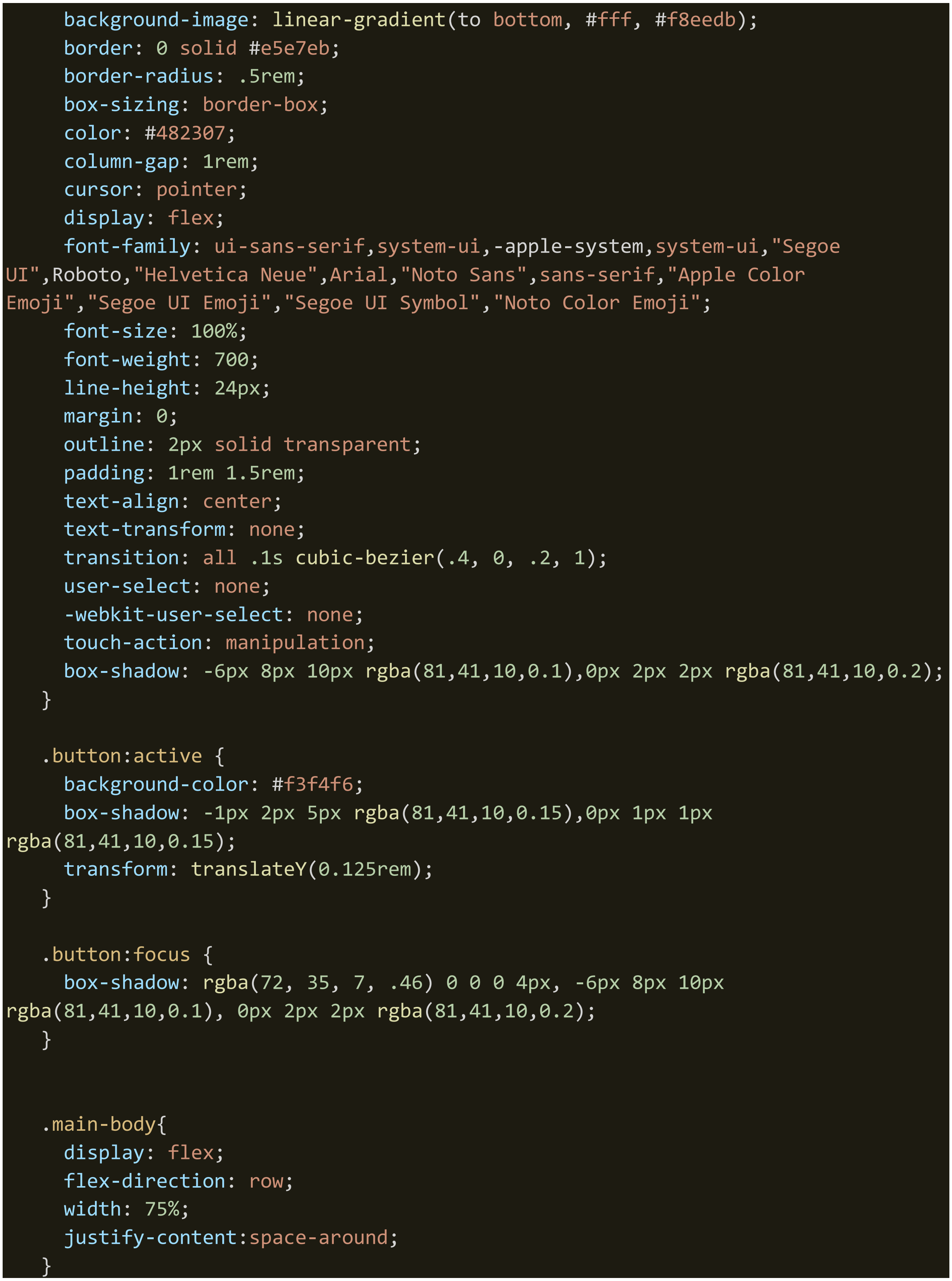
```
}

nav {

display: flex;

justify-content: space-around;

align-items: center;

min-height: 8vh;

background-color: teal;

font-family: "Montserrat", sans-serif;

}

.heading {

color: white;

text-transform: uppercase;

letter-spacing: 5px;

font-size: 20px;

}
.heading h4 a{
    text-decoration: none;
    color: #d8cabf;
}

.nav-links {

display: flex;

justify-content: space-around;

width: 30%;

}

.nav-links li {
```



```
list-style: none;

}

.nav-links a {

color: white;

text-decoration: none;

letter-spacing: 3px;

font-weight: bold;

font-size: 14px;

padding: 14px 16px;

}

.nav-links a:hover:not(.active) {

background-color: lightseagreen;

}

.nav-links li a.active {

background-color: #4caf50;

}


.form h1{
    text-align: center;
    display: flex;
    background-color: rgb(148, 104, 104);
}
.form{
    text-align: center;

}

.container{
```

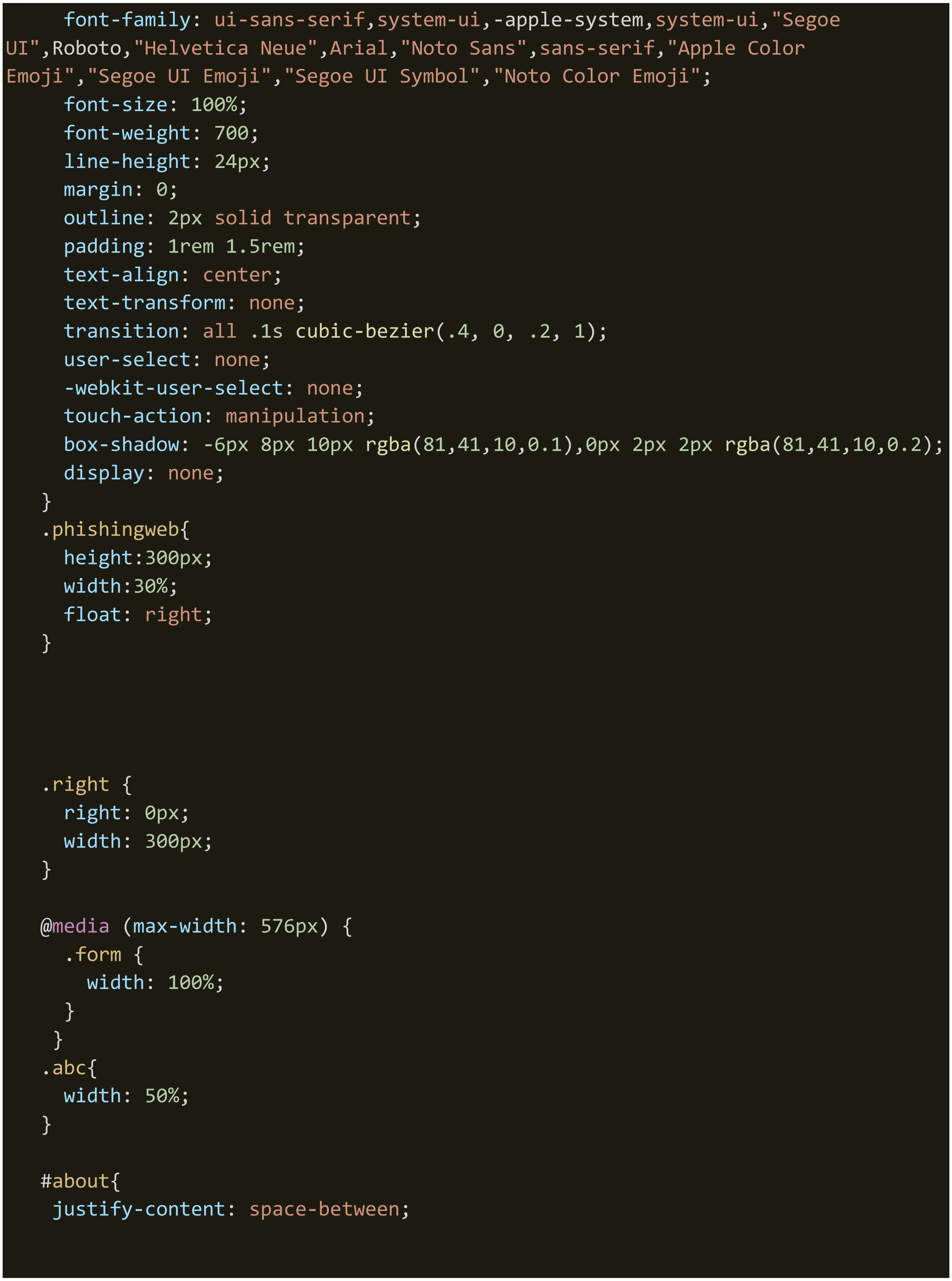


```
    align-items: center;

    height: fit-content;
    width: 70%;
    margin-left: 15%;
    margin-top: 6%;
}
.form__label {
    font-family: 'Roboto', sans-serif;
    font-size: 1.2rem;
    margin-left: 2rem;
    margin-top: 0.7rem;
    display: block;
    transition: all 0.3s;
    transform: translateY(0rem);
    align-items: center;
  }

  .form__input {
    top: -24px;
    font-family: 'Roboto', sans-serif;
    color: #333;
    font-size: 1.2rem;
    padding: 1.5rem 2rem;
    border-radius: 50px;
    background-color: rgb(255, 255, 255);
    border: none;
    width: 75%;
    display: block;
    border-bottom: 0.3rem solid transparent;
    transition: all 0.3s;
  }

  .form__input:placeholder-shown + .form__label {
    opacity: 0;
    visibility: hidden;
    -webkit-transform: translateY(+4rem);
    transform: translateY(+4rem);
  }


  .button {
    appearance: button;
    background-color: transparent;
```



```
     background-image: linear-gradient(to bottom, #fff, #f8eedb);
     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);
   }

   .button:active {
     background-color: #f3f4f6;
     box-shadow: -1px 2px 5px rgba(81,41,10,0.15),0px 1px 1px
rgba(81,41,10,0.15);
     transform: translateY(0.125rem);
   }

   .button:focus {
     box-shadow: rgba(72, 35, 7, .46) 0 0 0 4px, -6px 8px 10px
rgba(81,41,10,0.1), 0px 2px 2px rgba(81,41,10,0.2);
   }


   .main-body{
     display: flex;
     flex-direction: row;
     width: 75%;
     justify-content:space-around;
   }
```

```
 .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;
   }
   .phishingweb{
     height:300px;
     width:30%;
     float: right;
   }




   .right {
     right: 0px;
     width: 300px;
   }

   @media (max-width: 576px) {
     .form {
       width: 100%;
     }
    }
   .abc{
     width: 50%;
   }

   #about{
    justify-content: space-between;
   }
 footer{
     text-align:end;
 }
```

Ibm_app.py integrating and scoring endpoint :

```
  #importing required libraries


from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings('ignore')
from feature import FeatureExtraction



import requests

# NOTE: you must manually set API_KEY below using information retrieved from your
IBM Cloud account.
API_KEY = "kUBJ2G0ca5P26hfhcYTiu1nAK9cheAMdRgqygGwAFDzP"
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}

""" NOTE: manually define and pass the array(s) of values to be scored in the
next line
```

```
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/dca11834-060f-424c-8263-
7a42e6b430cf/predictions?version=2022-11-18', json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
"""
""" file = open("phishing_website.pkl","rb")
forest= pickle.load(file)
file.close() """
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)
        y_pred =forest.predict(x)[0]


        payload_scoring = {"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','f
28','f29']], "values": [[-1,1,1,1,-1,-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]]}]}

        response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/dca11834-060f-424c-8263-
7a42e6b430cf/predictions?version=2022-11-18', json=payload_scoring,
        headers={'Authorization': 'Bearer ' + mltoken})
        print("Scoring response")
        print(response_scoring.json())


        #1 is safe
        #-1 is unsafe
        y_pro_phishing = forest.predict_proba(x)[0,0]
        y_pro_non_phishing = forest.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
        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)
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-47377-1660798749

PROJECT DEMO LINK:

https://youtu.be/Yw7oRQQ9Hxc