WEB PHISING DETECTION

TEAM ID:PNT2022TMID21307

TEAM MEMBERS

Varshini S 917719C111

Ruchitaa Raj N R 917719C083

Ramprasad R 917719C078

Deepak RM 917719C014

1. INTRODUCTION

1.1 Project Overview

There are a number of users who purchase products online and make payments through e-banking. There are e-banking websites that ask users to provide sensitive data such as username, password & credit card details, etc., often for malicious reasons. This type of e-banking website is known as a phishing website. Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet.

Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity. It will lead to information disclosure and property damage. Large organizations may get trapped in different kinds of scams.

We have come up with a solution to detect if a website is safe or not by using machine learning for prediction. This helps the user to predict the legitimacy of a website beforehand and thus prevents the user from entering their personal information.

1.2 Purpose

Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information.

An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft.

2. LITERATURE SURVEY

2.1 Existing problem

Junaid Rashid et al[1] used machine learning based phishes detection gadget relies upon efficiently on the aspects of accuracy. The most of antiphishers researchers center of attention on optimizing new feature proposals or classification algorithms, where developing proper features analysis and selection techniques is not the important plan. The paper involved phishingenabled, reaching an effective positive rate of 97% and a false positive rate of 4%. The features are obtained by META tagging, web pages content, URLs, hyperlinks, TF-IDF, and more.

Ping Yi[2] et al used deep learning frameworks to detect web phishing. This paper mainly focuses on applying a deep learning framework to detect phishing websites. This paper first designs two types of features for web phishing: original features and interaction features. A detection model based on Deep Belief Networks (DBN) is then presented. The test using real IP flows from ISP (Internet Service Provider) shows that the detecting model based on DBN can achieve an approximately 90% true positive rate and 0.6% false positive rate.

Jain, A.K. et al [3] described anti-phishing technology that removes 19 features on the buyer's side to determine phishing websites from approved sites using machine learning. They used 2,141 phishing pages as well as the famous Alexa website, some online debit gateways, and some great banking websites.

Chiew[4] et al proposed to use probability minimization standard and Monte Carlo algorithm using a new neural network-based classification technique for detecting phishing net pages. The thirty points were used to categorize the four main areas, especially around the bar-based, anomaly-based, HTML and JavaScript.

Zhang, W., et al[5] extract features towards URL, text, and web content and utilize Extreme Machine Learning (ELM) technology. The first step in this method is to write the text content of the classifier to determine the content of the label text through ELM. In this case, OCR software is used to retrieve the text from the image. It is a second-stage-based hybrid that combines text and other function classifiers

2.2 References

[1]Junaid Rashid;Toqeer Mahmood;Muhammad Wasif Nisar;Tahira Nazir; (2020). Phishing Detection Using Machine Learning Technique . 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), (), –. doi:10.1109/smart-tech49988.2020.00026

[2] Yi, Ping; Guan, Yuxiang; Zou, Futai; Yao, Yao; Wang, Wei; Zhu, Ting (2018). Web Phishing Detection Using a Deep Learning Framework. Wireless Communications and Mobile Computing, 2018(), 1–9. doi:10.1155/2018/4678746

[3] Jain, A.K. and B.B. Gupta, Towards detection of phishing websites on clientside using machine learning based approach. Telecommunication Systems, 2018. 68(4): p. 687-700.

[4] Chiew, K.L., et al., Utilisation of website logo for phishing detection. Computers & Security, 2015. 54: p. 16-26.

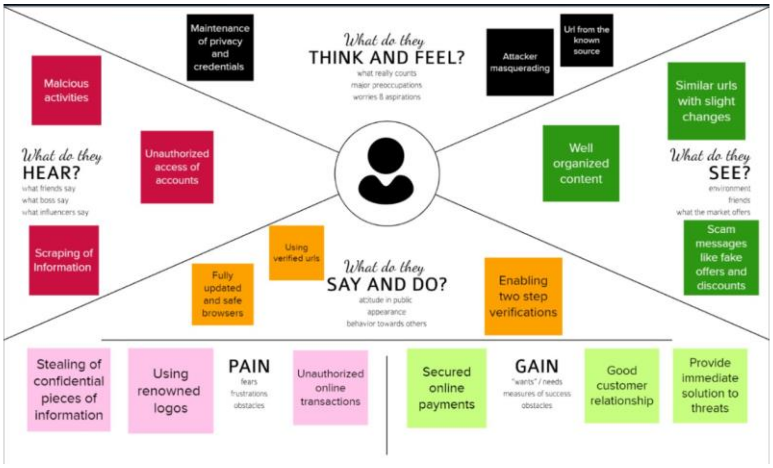
[5] Zhang, W., et al., Two-stage ELM for phishing Web pages detection using hybrid features. World Wide Web, 2017. 20(4): p. 797-813

2.3 Problem Statement Definition

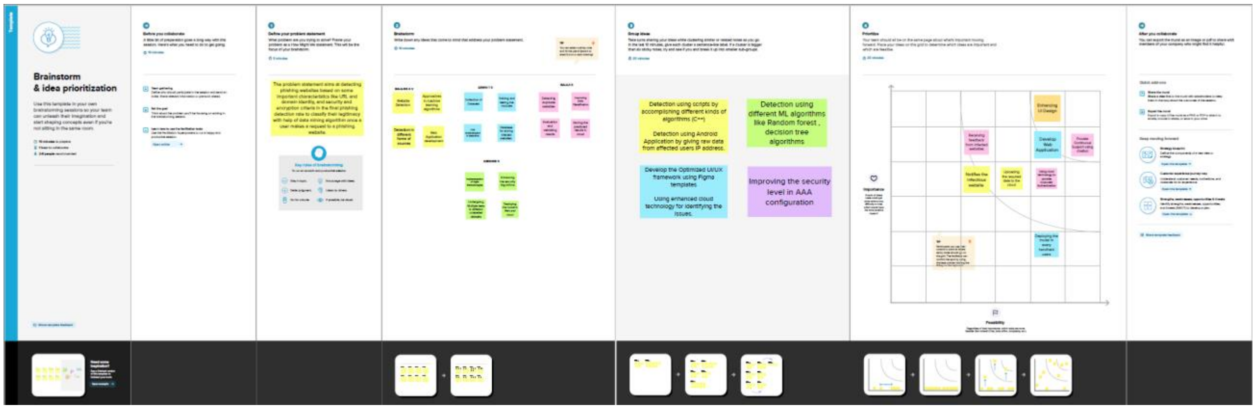
To detect and predict the website URL and check if the website is safe or unsafe to use using machine learning algorithm.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming


Brainstorm
& idea prioritization
Before you collaborate
Define your problem statement
Brainstorm
Group ideas
Prioritize
After you collaborate
The problem statement aims at detecting phishing websites based on some important characteristics like URL and domain identity, and security and encryption criteria in the final phishing detection rate to classify their legitimacy with help of data mining algorithm once a user makes a request to a phishing website.
Key rules of brainstorming
Detection using scripts by accomplishing different kinds of algorithms (C++)
Detection using Android Application by giving raw data from affected users IP address.
Detection using different ML algorithms like Random forest , decision tree algorithms
Develop the Optimized UI/UX framework using Figma templates
Using enhanced cloud technology for identifying the issues.
Improving the security level in AAA configuration
Importance
Feasibility

3.3 Proposed Solution

IDEA / SOLUTION DESCRIPTION:

The main objective of the project is applying a machine-learning algorithm to detect Phishing websites. It is a web application to detect good and malicious URLs i.e web phishing sites. We employ HTML, CSS, and JavaScript for the website and the web application is deployed using Flask framework. Different ML models are run and the one with the high accuracy will be selected for final model. The dataset is downloaded and then run-on notebooks. Necessary preprocessing techniques are implemented by applying various statistical methods and encodings are done. Then it is split to train test model and different algorithms are tried.

UNIQUENESS/NOVELTY:

In order to detect and predict e-banking phishing websites, we proposed an intelligent, flexible and effective system that is based on using classification algorithms. We implemented classification algorithms like logistic regression, SVM, KNN and techniques to extract the phishing datasets criteria to classify their legitimacy. The e-banking phishing website can be detected based on some important characteristics like URL and domain identity, and security and encryption criteria in the final phishing detection rate. Once a user makes a transaction online when he 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.

SOCIAL IMPACT/ CUSTOMER SATISFACTION:

It helps customers to reduce threats that happen during e-banking. Customer can enter the correct URL and all the passwords, usernames, private information, credit card details are kept safe. It will prevent from information disclosure and property damage. It will increase customer satisfaction without fearing of malicious sites.

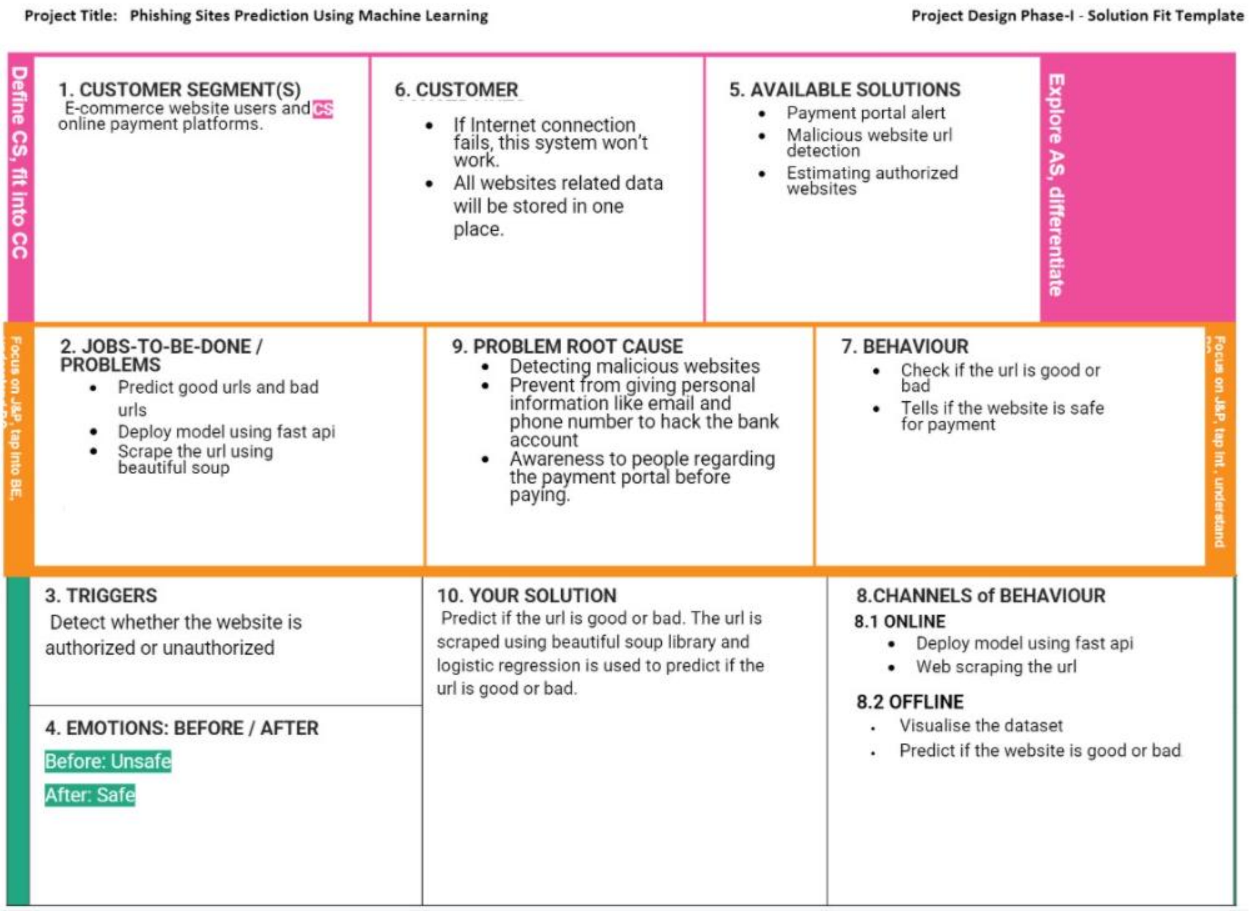
BUSINESS MODEL (FINANCIAL BENEFIT):

Phishing detection helps in preventing from falling for malicious websites and traps. Thus ensuring the safety of one's personal data and other private information. Doing this beforehand by detecting through ML models can save time. The proposed solution is also a low-cost model and the customers are not charged for the service they receive.

SCALABILITY OF SOLUTION:

The model’s performance is increased by building it more accurate model with the use of several classification algorithms and selecting the best accurate model among the different models run. Also, through integration of these models, an optimized hybrid model can be obtained in order to result in more scalability. Deploying the ML model into cloud also makes it easy for enterprises to experiment with the model capabilities and scale up. Placing a finished flight prediction model into a live environment can be used for its intended purpose and it is integrated with Flask, so that they can be accessed by end users.

3.4 Problem Solution fit



Project Title: Phishing Sites Prediction Using Machine Learning

Project Design Phase-I - Solution Fit Template

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

E-commerce website users and CS online payment platforms.

6. CUSTOMER

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

5. AVAILABLE SOLUTIONS

- Payment portal alert
- Malicious website url detection
- Estimating authorized websites

Explore AS, differentiate

Focus on J&P, tap into BE,

2. JOBS-TO-BE-DONE / PROBLEMS

- Predict good urls and bad urls
- Deploy model using fast api
- Scrape the url using beautiful soup

9. PROBLEM ROOT CAUSE

- Detecting malicious websites
- Prevent from giving personal information like email and phone number to hack the bank account
- Awareness to people regarding the payment portal before paying.

7. BEHAVIOUR

- Check if the url is good or bad
- Tells if the website is safe for payment

Focus on J&P, tap int , understand

3. TRIGGERS

Detect whether the website is authorized or unauthorized

4. EMOTIONS: BEFORE / AFTER

Before: Unsafe

After: Safe

10. YOUR SOLUTION

Predict if the url is good or bad. The url is scraped using beautiful soup library and logistic regression is used to predict if the url is good or bad.

8.CHANNELS of BEHAVIOUR

8.1 ONLINE

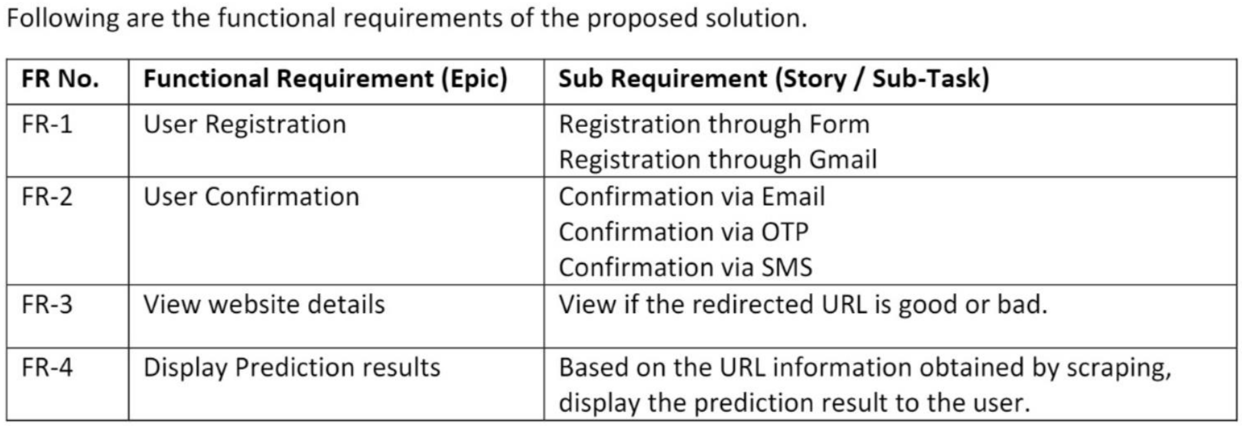
- Deploy model using fast api
- Web scraping the url

8.2 OFFLINE

- Visualise the dataset
- Predict if the website is good or bad

4. REQUIREMENT ANALYSIS

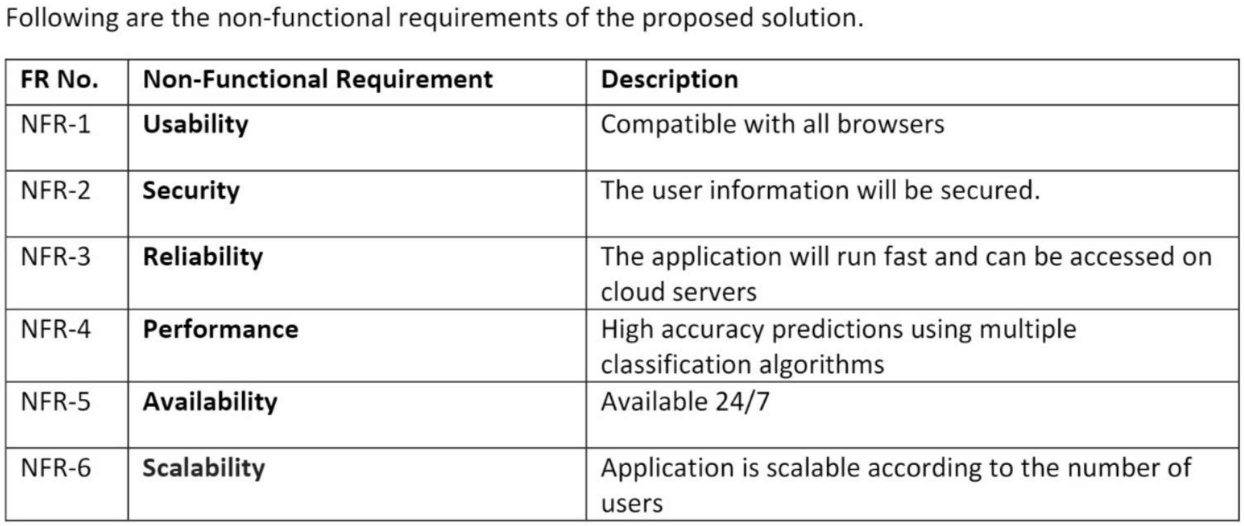
4.1 Functional requirement



Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Confirmation via SMS
FR-3	View website details	View if the redirected URL is good or bad.
FR-4	Display Prediction results	Based on the URL information obtained by scraping, display the prediction result to the user.

4.2 Non-Functional requirements

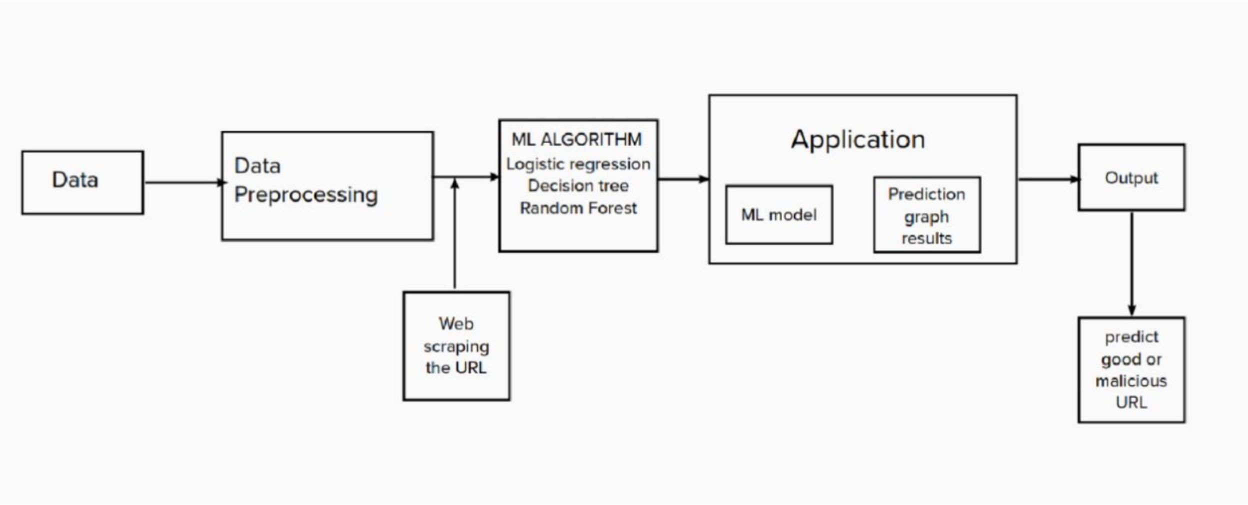


Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Compatible with all browsers
NFR-2	**Security**	The user information will be secured.
NFR-3	**Reliability**	The application will run fast and can be accessed on cloud servers
NFR-4	**Performance**	High accuracy predictions using multiple classification algorithms
NFR-5	**Availability**	Available 24/7
NFR-6	**Scalability**	Application is scalable according to the number of users

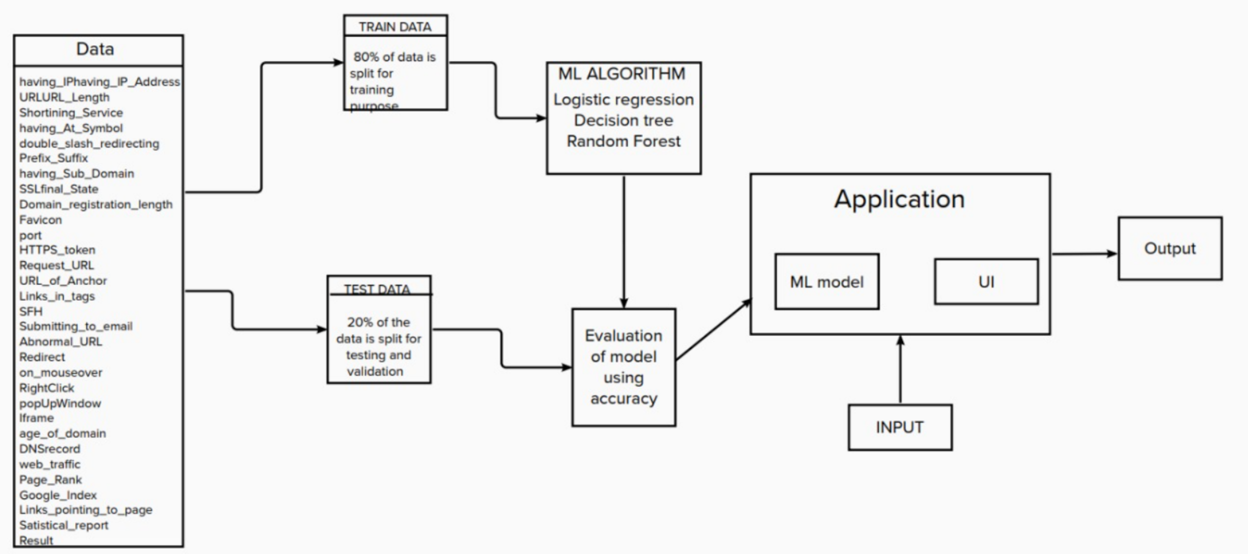
5. PROJECT DESIGN

5.1 Data Flow Diagrams

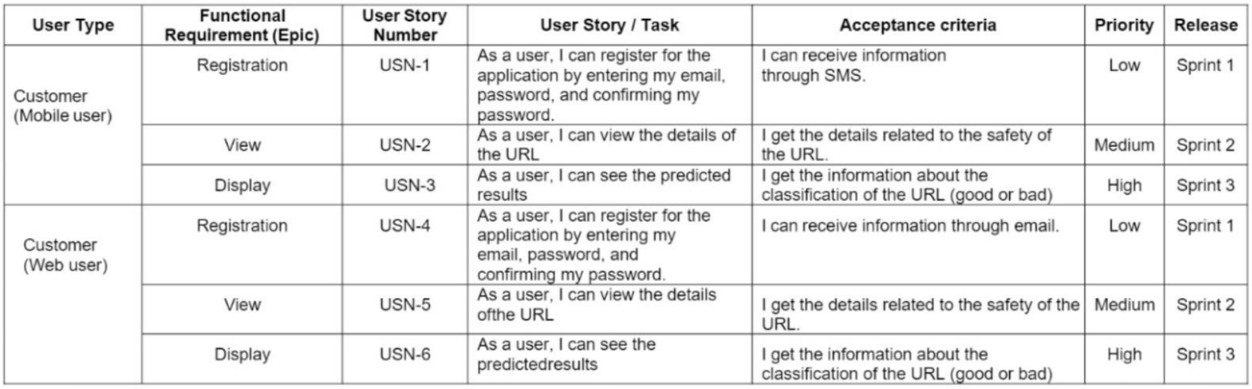


5.2 Solution & Technical Architecture

The below diagram illustrates the architecture of the solution proposed. The programming language that'll be used to analyses data and build the machine language is Python as it is easier to work on and has several built-in functions that is user friendly and fast enough to give the results. The application will be built using Flask as it is a simple Python framework that can be used to build web applications. It easily encapsulates the trained machine learning model. The user interface will be built using HTML, CSS and JavaScript. The attributes mentioned are the actual attributes that'll be taken into account for prediction in case of input data. The percentage of train and test data split will depend on the performance of the model during testing.



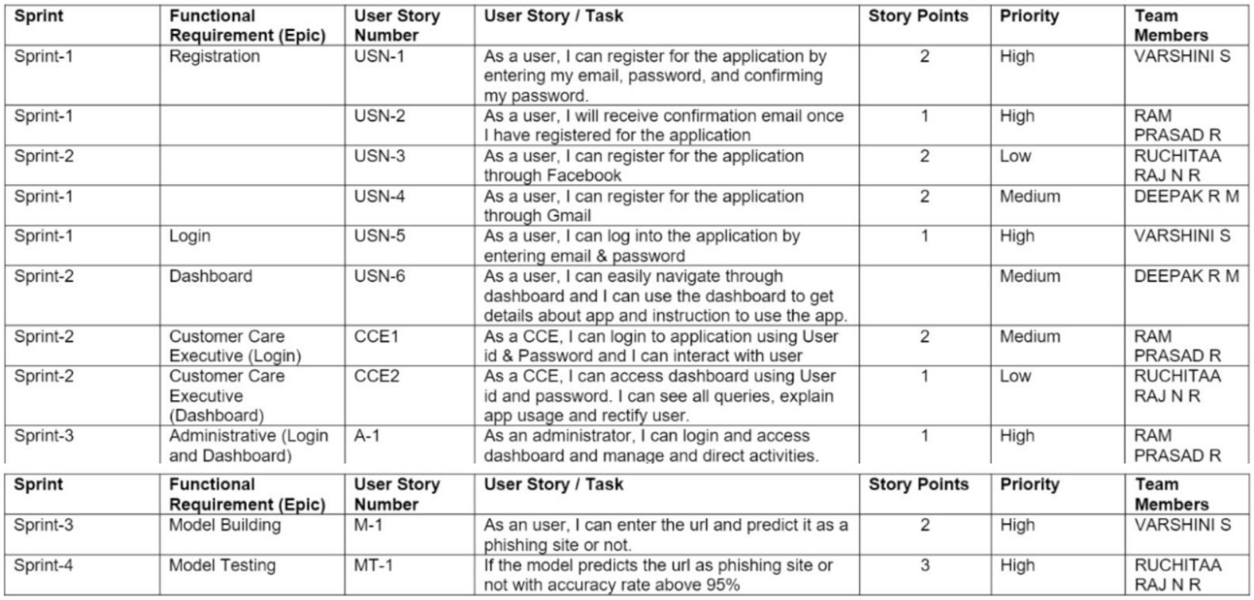
5.3 User Stories



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 confirming my password.	I can receive information through SMS.	Low	Sprint 1
	View	USN-2	As a user, I can view the details of the URL	I get the details related to the safety of the URL.	Medium	Sprint 2
	Display	USN-3	As a user, I can see the predicted results	I get the information about the classification of the URL (good or bad)	High	Sprint 3
Customer (Web user)	Registration	USN-4	As a user, I can register for the application by entering my email, password, and confirming my password.	I can receive information through email.	Low	Sprint 1
	View	USN-5	As a user, I can view the details ofthe URL	I get the details related to the safety of the URL.	Medium	Sprint 2
	Display	USN-6	As a user, I can see the predictedresults	I get the information about the classification of the URL (good or bad)	High	Sprint 3

6. PROJECT PLANNING & SCHEDULING

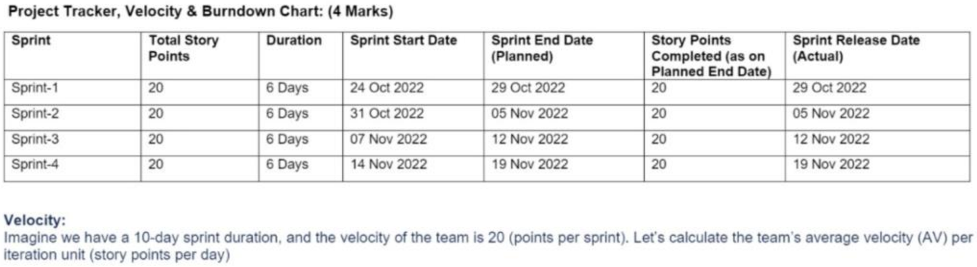
6.1 Sprint Planning & Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	VARSHINI S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	RAM PRASAD R
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	RUCHITAA RAJ N R
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	DEEPAK R M
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	VARSHINI S
Sprint-2	Dashboard	USN-6	As a user, I can easily navigate through dashboard and I can use the dashboard to get details about app and instruction to use the app.		Medium	DEEPAK R M
Sprint-2	Customer Care Executive (Login)	CCE1	As a CCE, I can login to application using User id & Password and I can interact with user	2	Medium	RAM PRASAD R
Sprint-2	Customer Care Executive (Dashboard)	CCE2	As a CCE, I can access dashboard using User id and password. I can see all queries, explain app usage and rectify user.	1	Low	RUCHITAA RAJ N R
Sprint-3	Administrative (Login and Dashboard)	A-1	As an administrator, I can login and access dashboard and manage and direct activities.	1	High	RAM PRASAD R

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Model Building	M-1	As an user, I can enter the url and predict it as a phishing site or not.	2	High	VARSHINI S
Sprint-4	Model Testing	MT-1	If the model predicts the url as phishing site or not with accuracy rate above 95%	3	High	RUCHITAA RAJ N R

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

Velocity:

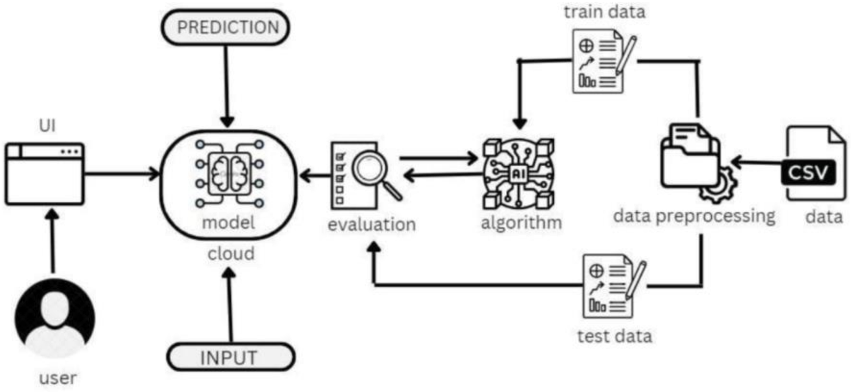
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

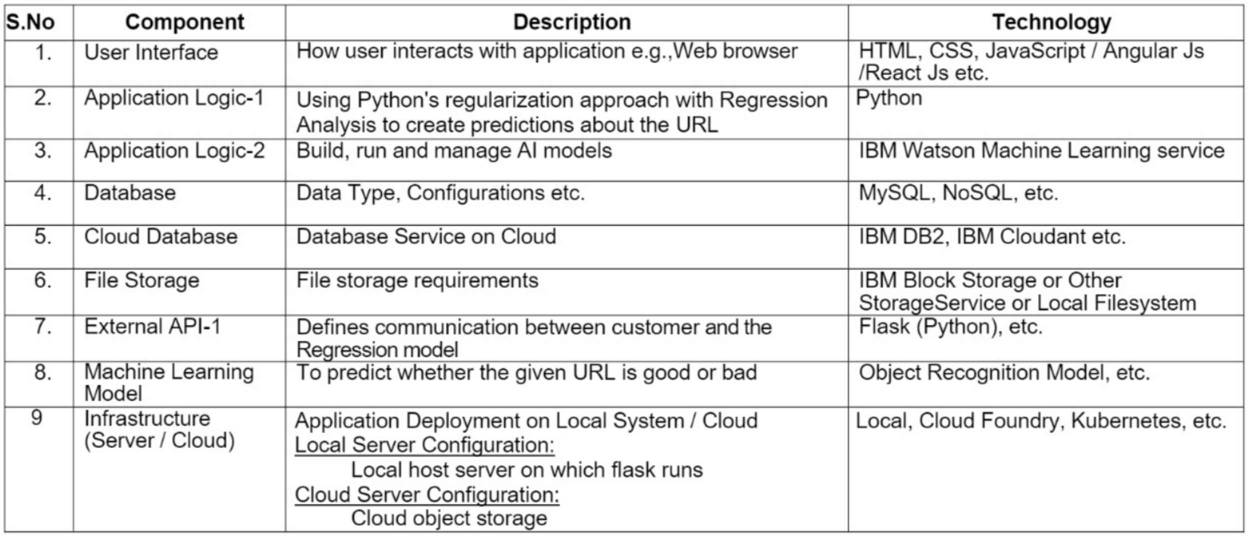
Av = 20/6 = 3.3

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

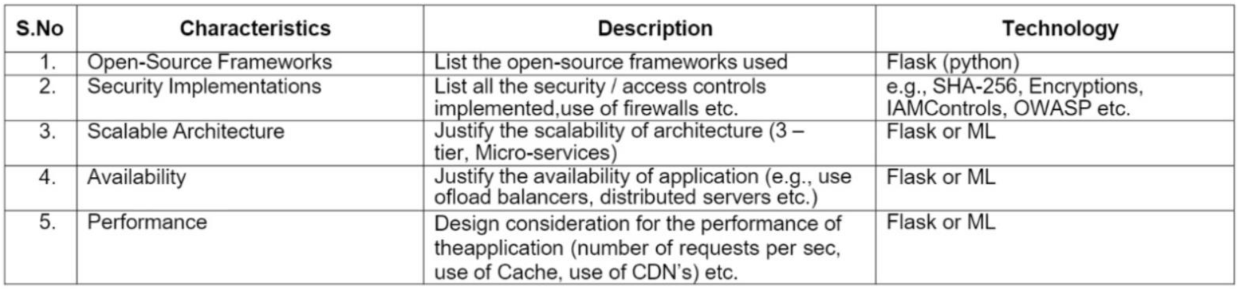


Technological Stack



S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.,Web browser	HTML, CSS, JavaScript / Angular Js /React Js etc.
2.	Application Logic-1	Using Python's regularization approach with Regression Analysis to create predictions about the URL	Python
3.	Application Logic-2	Build, run and manage AI models	IBM Watson Machine Learning service
4.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
5.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
6.	File Storage	File storage requirements	IBM Block Storage or Other StorageService or Local Filesystem
7.	External API-1	Defines communication between customer and the Regression model	Flask (Python), etc.
8.	Machine Learning Model	To predict whether the given URL is good or bad	Object Recognition Model, etc.
9	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Local host server on which flask runs Cloud Server Configuration: Cloud object storage	Local, Cloud Foundry, Kubernetes, etc.

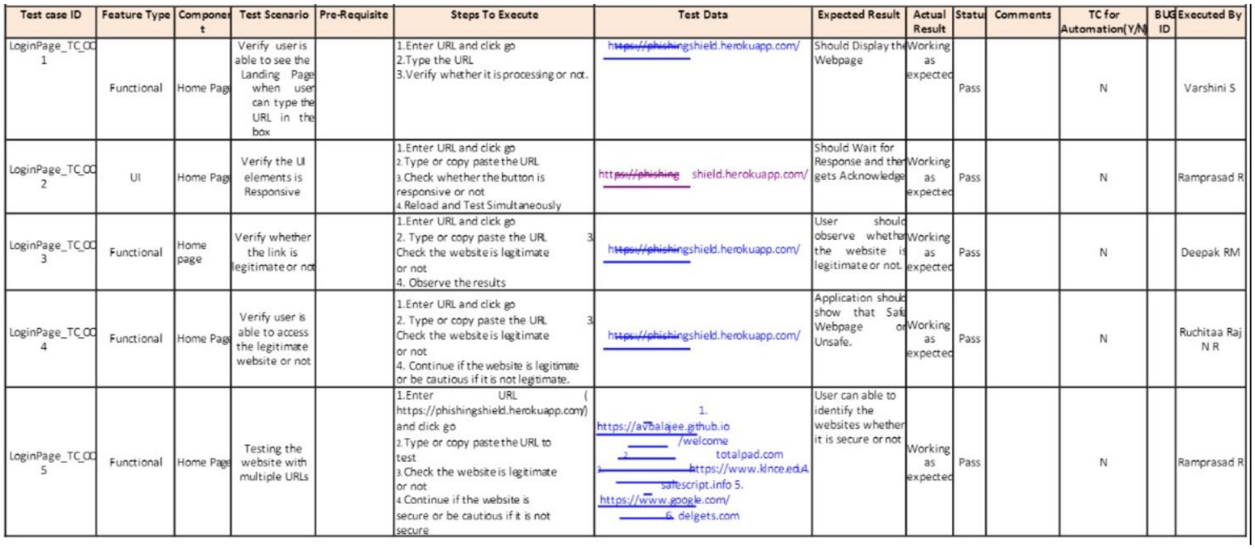
7.2 Feature 2



S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Flask (python)
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	e.g., SHA-256, Encryptions, IAMControls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Flask or ML
4.	Availability	Justify the availability of application (e.g., use ofload balancers, distributed servers etc.)	Flask or ML
5.	Performance	Design consideration for the performance of theapplication (number of requests per sec, use of Cache, use of CDN's) etc.	Flask or ML

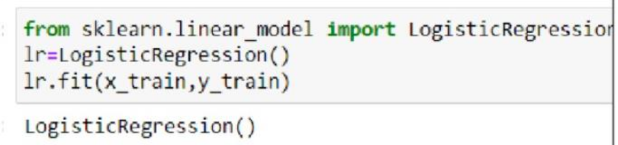
8. TESTING

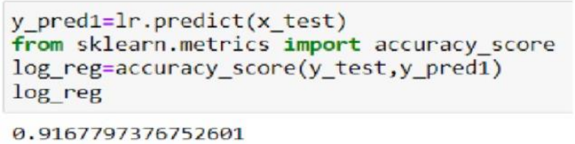
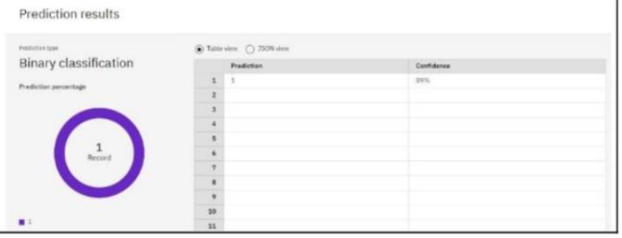
8.1 Test Cases



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Landing Page when user can type the URL in the box		1.Enter URL and click go 2.Type the URL 3.Verify whether it is processing or not.	https://phishingshield.herokuapp.com/	Should Display the Webpage	Working as expected	Pass		N		Varshini S
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements is Responsive		1.Enter URL and click go 2.Type or copy paste the URL 3.Check whether the button is responsive or not 4.Reload and Test Simultaneously	https://phishing shield.herokuapp.com/	Should Wait for Response and then gets Acknowledge	Working as expected	Pass		N		Ramprasad R
LoginPage_TC_OO3	Functional	Home page	Verify whether the link is legitimate or not		1.Enter URL and click go 2. Type or copy paste the URL 3. Check the website is legitimate or not 4. Observe the results	https://phishingshield.herokuapp.com/	User should observe whether the website is legitimate or not.	Working as expected	Pass		N		Deepak RM
LoginPage_TC_OO4	Functional	Home Page	Verify user is able to access the legitimate website or not		1.Enter URL and click go 2. Type or copy paste the URL 3. Check the website is legitimate or not 4. Continue if the website is legitimate or be cautious if it is not legitimate.	https://phishingshield.herokuapp.com/	Application should show that Safe Webpage or Unsafe.	Working as expected	Pass		N		Ruchitaa Raj N R
LoginPage_TC_OO5	Functional	Home Page	Testing the website with multiple URLs		1.Enter URL (https://phishingshield.herokuapp.com/) and click go 2.Type or copy paste the URL to test 3.Check the website is legitimate or not 4.Continue if the website is secure or be cautious if it is not secure	1. https://avbalajee.github.io/welcome 2. totalpad.com 3. https://www.klnce.edu 4. safescript.info 5. https://www.google.com/ 6. delgets.com	User can able to identify the websites whether it is secure or not	Working as expected	Pass		N		Ramprasad R

S.No.	Parameter	Values	Screenshot

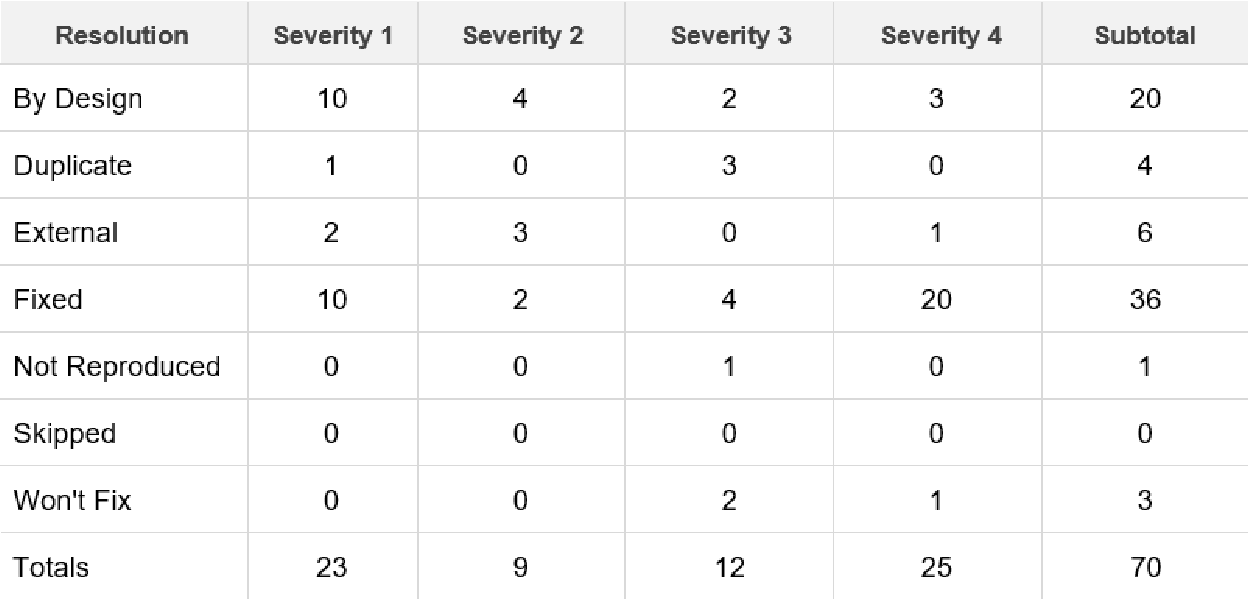
1.	Model- Logistic Regression	-	 from sklearn.linear_model import LogisticRegressior
			lr=LogisticRegression()
			lr.fit(x_train,y_train)
			LogisticRegression()

2.	Accuracy	Accuracy Score – 91.67%	 y_pred1=lr.predict(x_test) from sklearn.metrics import accuracy_score log_reg=accuracy_score(y_test,y_pred1) log_reg 0.9167797376752601
3.	Confidence Score	Class Detected - 1 Confidence Score – 89%	 Prediction results Prediction type Binary classification Prediction percentage 1 Record ● Table view ○ JSON view Prediction: 1, Confidence: 89%

8.2 User Acceptance Testing

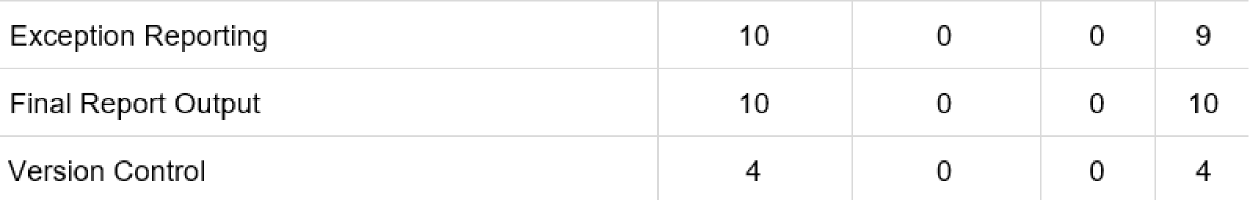
Defect Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	5	0	0	4
Outsource Shipping	3	0	0	3



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	10	2	4	20	36
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0
Won't Fix	0	0	2	1	3
Totals	23	9	12	25	70

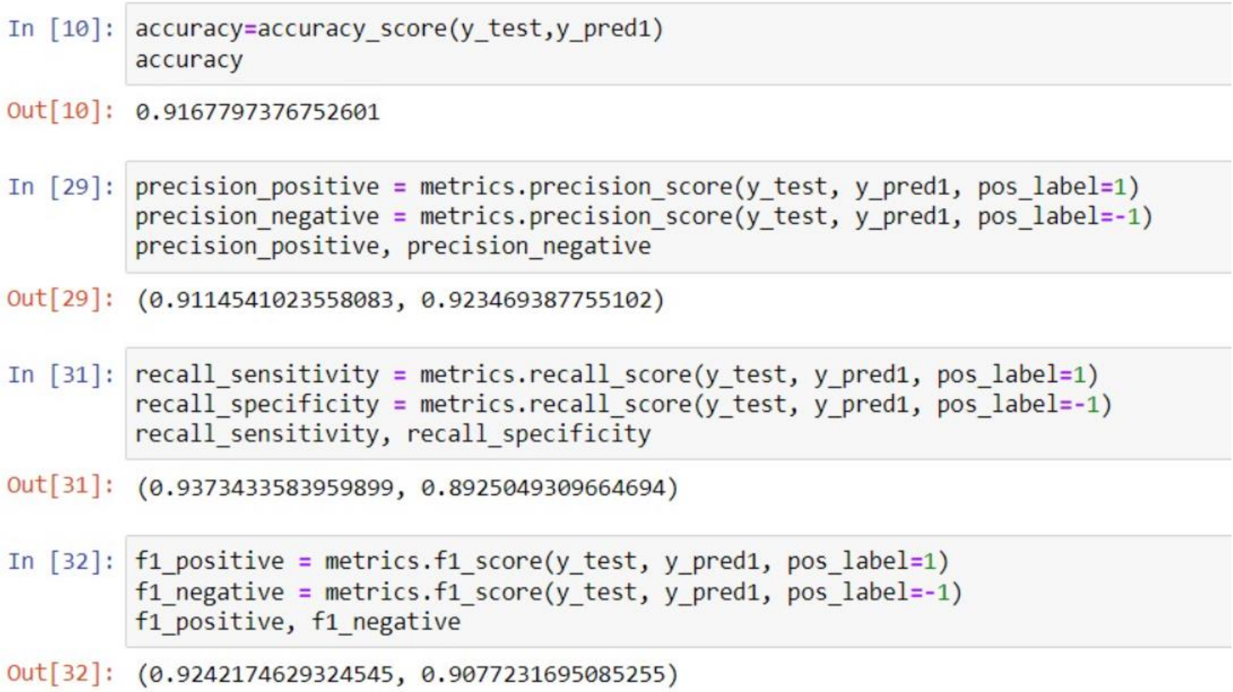
Test Case Analysis



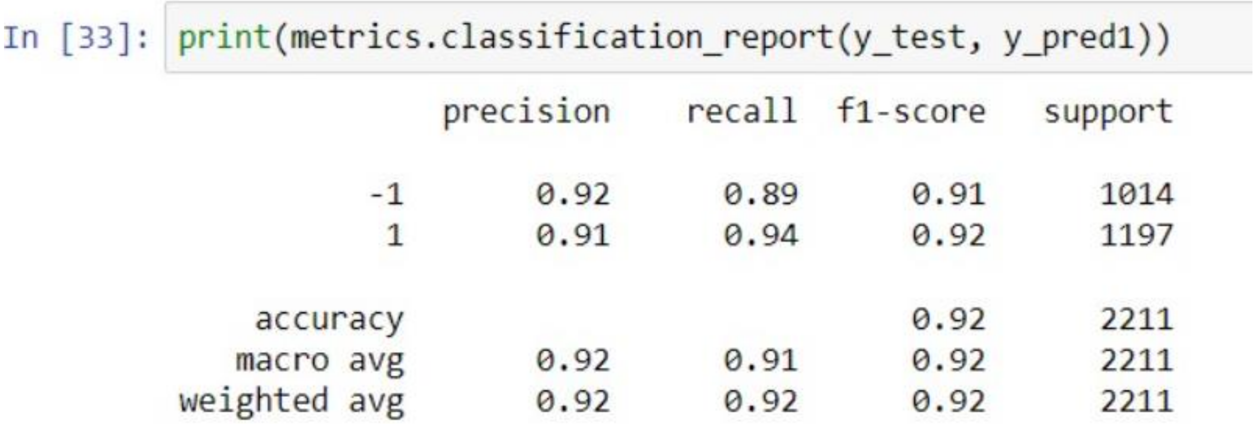
Exception Reporting	10	0	0	9
Final Report Output	10	0	0	10
Version Control	4	0	0	4

9. RESULTS

9.1 Performance Metrics



```
In [10]: accuracy=accuracy_score(y_test,y_pred1)
         accuracy

Out[10]: 0.9167797376752601

In [29]: precision_positive = metrics.precision_score(y_test, y_pred1, pos_label=1)
         precision_negative = metrics.precision_score(y_test, y_pred1, pos_label=-1)
         precision_positive, precision_negative

Out[29]: (0.9114541023558083, 0.923469387755102)

In [31]: recall_sensitivity = metrics.recall_score(y_test, y_pred1, pos_label=1)
         recall_specificity = metrics.recall_score(y_test, y_pred1, pos_label=-1)
         recall_sensitivity, recall_specificity

Out[31]: (0.9373433583959899, 0.8925049309664694)

In [32]: f1_positive = metrics.f1_score(y_test, y_pred1, pos_label=1)
         f1_negative = metrics.f1_score(y_test, y_pred1, pos_label=-1)
         f1_positive, f1_negative

Out[32]: (0.9242174629324545, 0.9077231695085255)
```



```
In [33]: print(metrics.classification_report(y_test, y_pred1))

              precision    recall  f1-score   support

          -1       0.92      0.89      0.91      1014
           1       0.91      0.94      0.92      1197

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

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. Build secure connection between uses male transfer agent (MTA) in male user agent (MUA)
2. Provide clear idea about the effective levels of each classifier on phishing email detection
3. High level of accuracy may take the advantages of many classifiers
4. High level of accuracy
5. Create new type of features like Markov features
6. Fast in classification process

7. Fast, less consuming memory, high accuracy, evolving with time and online working

Disadvantages:

1. Time consuming, huge number of features and consumes memory
2. Non standard classifier
3. Time consuming because this technique has many layers to make the final result.
4. Huge number of features, many algorithms for classification which mean time consuming.
5. Expensive and need large mail server and high memory requirement.
6. Less accuracy because it depends on unsupervised learning and needs feed continuously.

11. CONCLUSION

The system designed is used to prevent valuable information from leak out, produce better control mechanism and alerts user to keep the private information safe. Like any other program, there are improvements which could be made into the system. The proposed system has been identified and chosen to address the web phishing. The application is designed to show awareness, features that can be displayed, safety of the website. Its unique features such as capturing blacklisted URLs from the browser directly to verify the validity of the website, notifying the user on unsafe websites by entering the URL and checking its safety. In our project, we used machine learning classification algorithm to differentiate whether the site is safe or unsafe. With the higher accuracy of the model, a web application was build using the flask framework. It was also deployed in IBM cloud as an extension.

12. FUTURE SCOPE

Like any other program, there are improvements which could be made into the system. Based on the capabilities which the current system processes, a pop up could be displayed when accessing the phishing site. Further notification through email can also be sent to assist the used to be alerted when they are trying to access a blacklisted website. A text message integration would be a grater recommendation that could improve the program in future. The future version of the application could also implement an option to directly notify the phishing website with a text message. The program could be made to access the list as an attachment. This text message integration would further enhance the usability of the application. This could be further improved to be added as a chrome extension.

13. APPENDIX

Code:

i. model.ipynb

```
import pandas as pd
import numpy as np
```

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix,accuracy_score
# Reading the dataset
# Importing Dataset
ds = pd.read_csv("data_website.csv")
ds.head()
# Handling null values
ds.info()
ds.isnull().any()
# Splitting the data
# removing index column in independent dataset
x = ds.iloc[:,1:31].values
y = ds.iloc[:,-1].values
print(x,y)
# splitting data into train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)
# Model Building
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
y_pred1=lr.predict(x_test)
from sklearn.metrics import accuracy_score
log_reg=accuracy_score(y_test,y_pred1)
log_reg
import pickle
pickle.dump(lr,open('Phishing_Website.pkl','wb'))
```

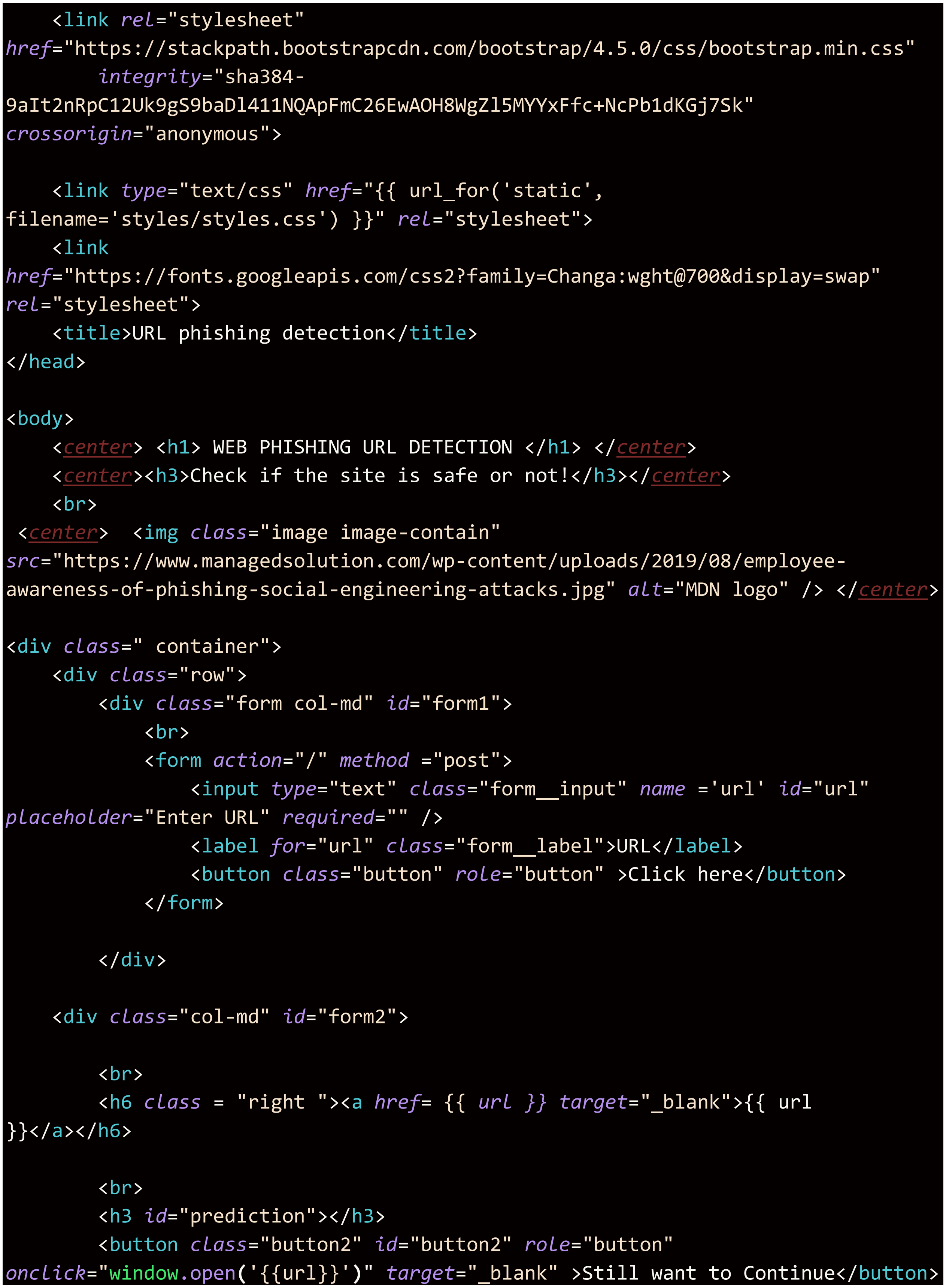
ii. app.py

```
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 math

file = open("model.pkl","rb")
gbc = pickle.load(file)
file.close()
```

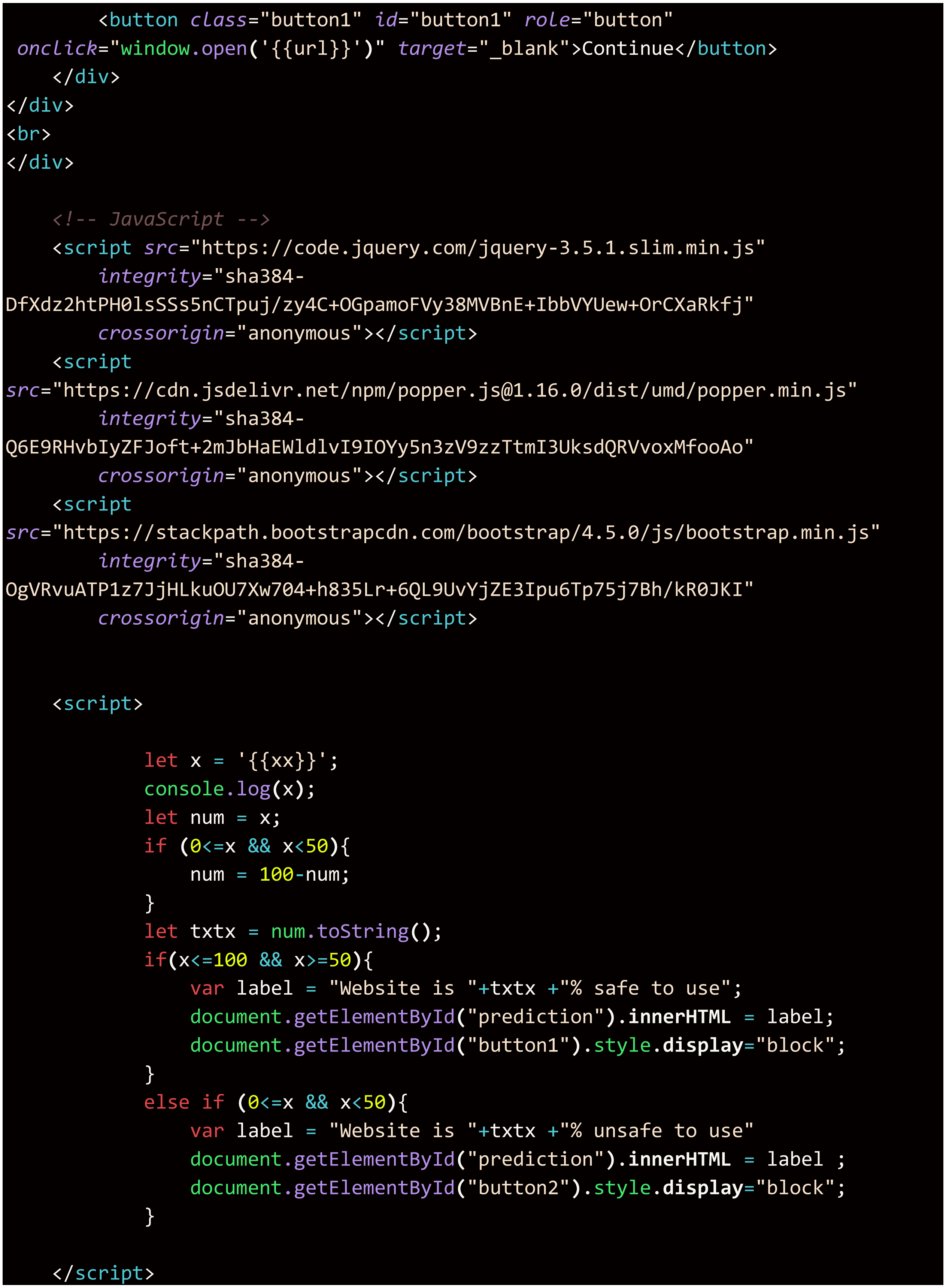
```
app = Flask(__name__,template_folder="templates")

@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 =gbc.predict(x)[0]
        #0 - unsafe
        #1 - safe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0,1]
        print("phi ",y_pro_phishing)
        print("non phi ",y_pro_non_phishing)
        x = math.floor(y_pro_non_phishing*1000)/10
        print(x)
        return render_template('index.html',xx =x,url=url )

    #home page render
    return render_template("index.html", xx =-1)


if __name__ == "__main__":
    app.run(debug=True,port=2002)
```

iii. 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 developed to identify the
safety of url.">
    <meta name="keywords" content="phishing url,phishing,cyber security,machine
learning,classifier,python">



    <!-- BootStrap -->
```



```
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
        integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">

    <link type="text/css" href="{{ url_for('static',
filename='styles/styles.css') }}" rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css2?family=Changa:wght@700&display=swap"
rel="stylesheet">
    <title>URL phishing detection</title>
</head>

<body>
    <center> <h1> WEB PHISHING URL DETECTION </h1> </center>
    <center><h3>Check if the site is safe or not!</h3></center>
    <br>
 <center>  <img class="image image-contain"
src="https://www.managedsolution.com/wp-content/uploads/2019/08/employee-
awareness-of-phishing-social-engineering-attacks.jpg" alt="MDN logo" /> </center>

<div class=" container">
    <div class="row">
        <div class="form col-md" id="form1">
            <br>
            <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" >Click here</button>
            </form>

        </div>

    <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>
```

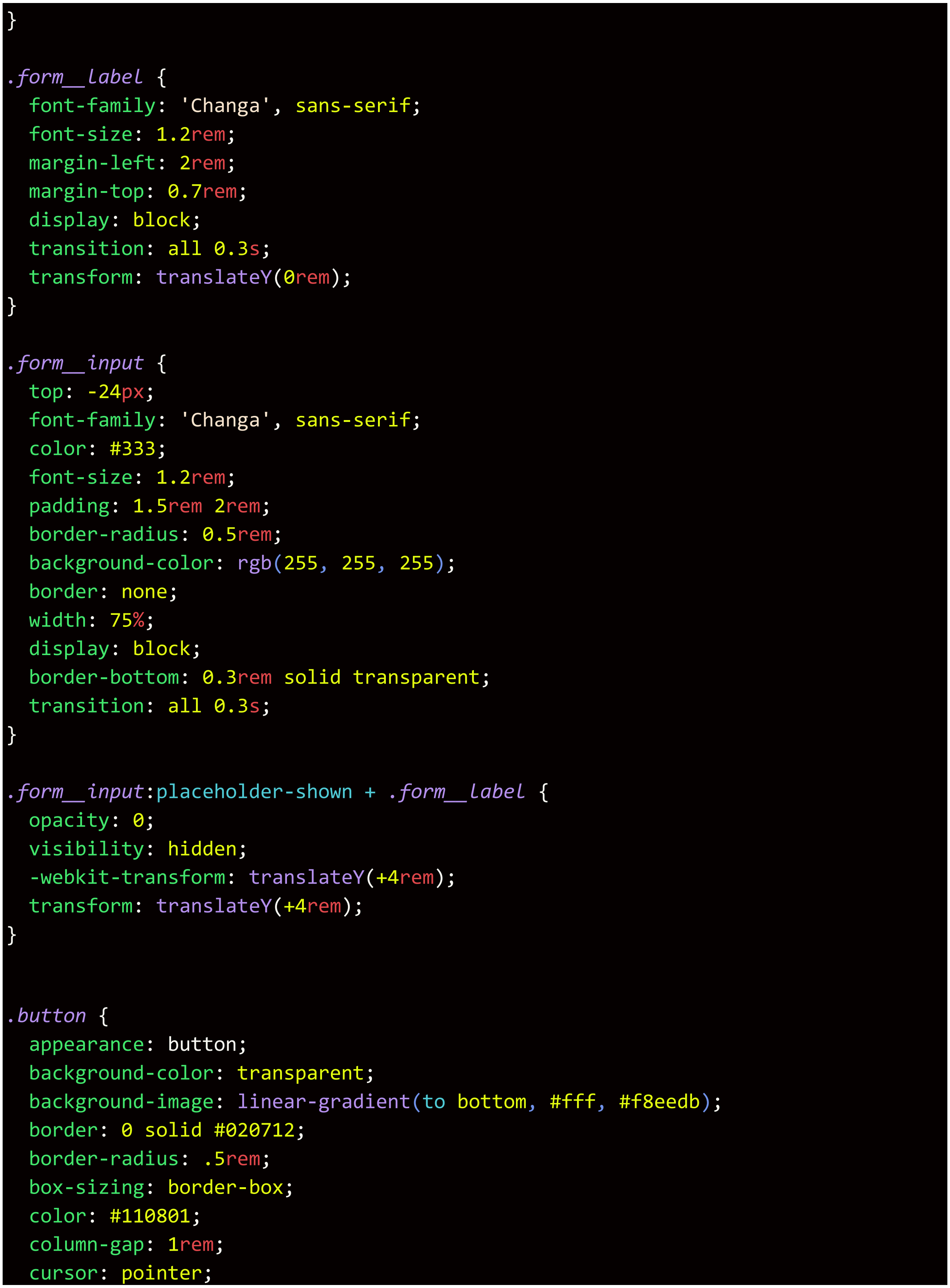


```
        <button class="button1" id="button1" role="button"
 onclick="window.open('{{url}}')" target="_blank">Continue</button>
    </div>
</div>
<br>
</div>

    <!-- 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}}';
            console.log(x);
            let num = x;
            if (0<=x && x<50){
                num = 100-num;
            }
            let txtx = num.toString();
            if(x<=100 && x>=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<50){
                var label = "Website is "+txtx +"% unsafe to use"
                document.getElementById("prediction").innerHTML = label ;
                document.getElementById("button2").style.display="block";
            }

    </script>
```

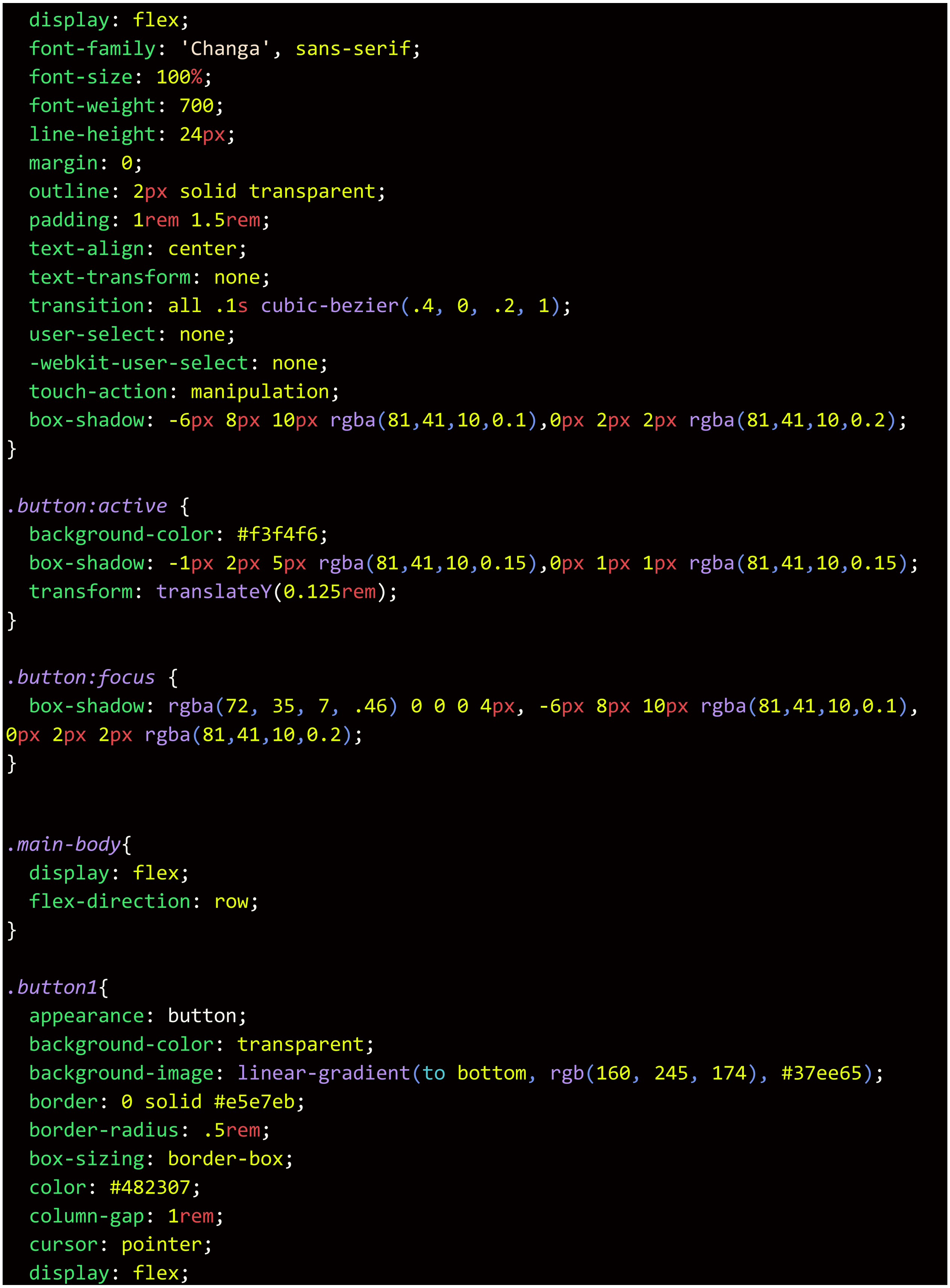
```

</body>

</html>

```

iv. styles.css

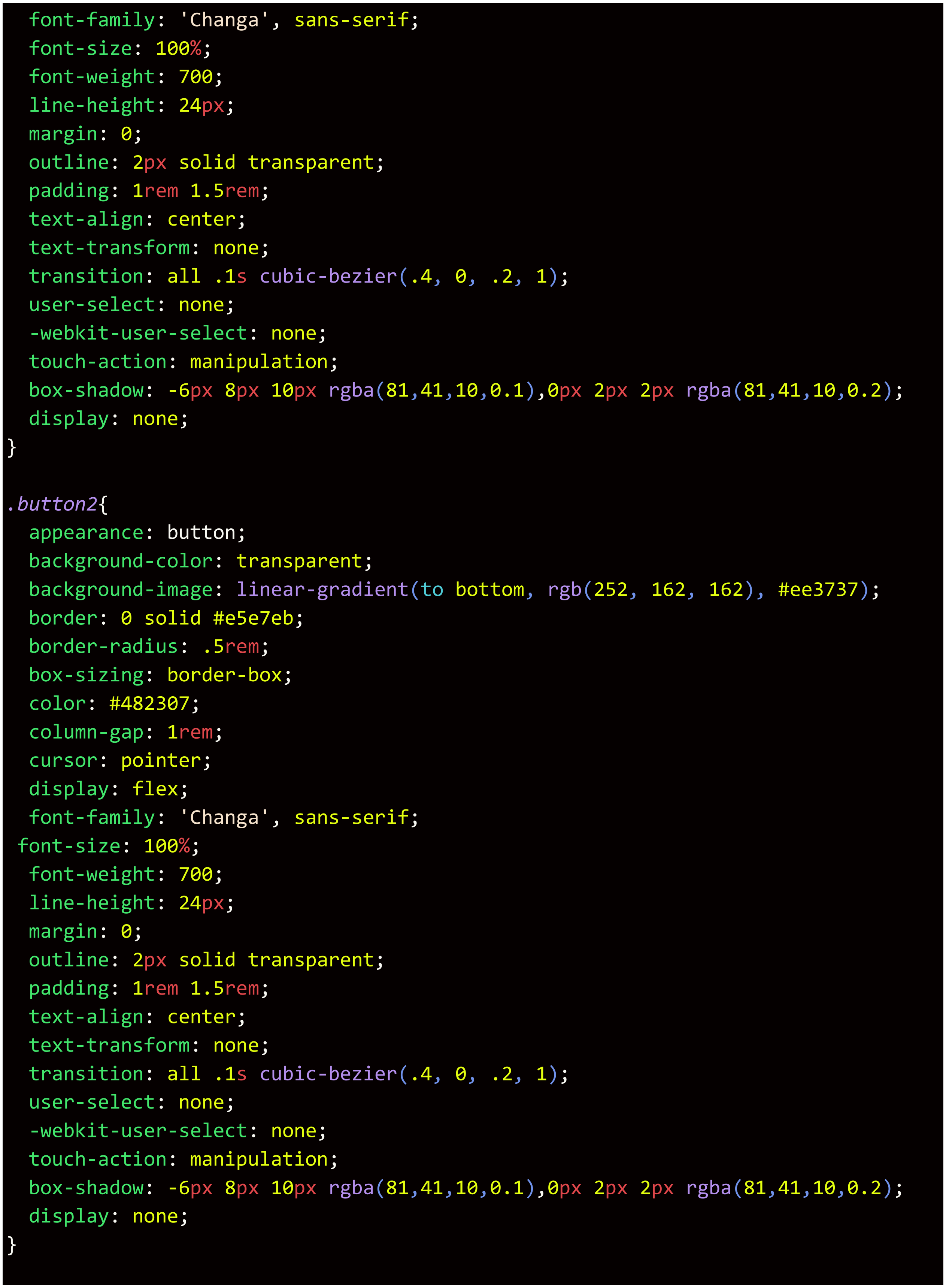
```
*,
*::after,
*::before {
  margin: 0;
  padding: 0;
  box-sizing: inherit;
  font-size: 62,5%;
}
.image {
  width: 1000px;
  height: 500px;
  border-radius: 50%;
}

h1 {
  font-family: 'Changa', sans-serif;
}

.image-contain {
  object-fit: contain;
  object-position: center;

}

.image-cover {
  object-fit: cover;
  object-position: center;
}
body {
  padding: 5% 5%;
  background: #0f2027;
  background: linear-gradient(to right,#b2ca71, #a6ce7d, #1d210e);
  justify-content: center;
  align-items: center;
  height: 100vh;
  color: #fff;
```



```css
}

.form__label {
  font-family: 'Changa', sans-serif;
  font-size: 1.2rem;
  margin-left: 2rem;
  margin-top: 0.7rem;
  display: block;
  transition: all 0.3s;
  transform: translateY(0rem);
}

.form__input {
  top: -24px;
  font-family: 'Changa', sans-serif;
  color: #333;
  font-size: 1.2rem;
  padding: 1.5rem 2rem;
  border-radius: 0.5rem;
  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 #020712;
  border-radius: .5rem;
  box-sizing: border-box;
  color: #110801;
  column-gap: 1rem;
  cursor: pointer;
```



```
  display: flex;
  font-family: 'Changa', sans-serif;
  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;
}

.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: 'Changa', sans-serif;
  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: 'Changa', sans-serif;
 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;
}
```

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

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

v. Scoring_Endpoint.py

```
from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
import requests
warnings.filterwarnings('ignore')
from feature import FeatureExtraction
import math
file = open("model.pkl","rb")
gbc = pickle.load(file)
file.close()
API_KEY = "<YOUR_API_KEY>"
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}
app = Flask(__name__, template_folder="templates")
@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 =gbc.predict(x)[0]
#0 - unsafe
#1 - safe
y_pro_phishing = gbc.predict_proba(x)[0,0]
y_pro_non_phishing = gbc.predict_proba(x)[0,1]
# if(y_pred ==1 ):
pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
# payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}
payload_scoring = {"input_data": [{"fields":
["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//","PrefixSuffix-
","SubDomains","HTTPS","DomainRegLen","Favicon","NonStdPort","HTTPSDomainURL","Re
questURL","AnchorURL","LinksInScriptTags","ServerFormHandler","InfoEmail","Abnorm
alURL","WebsiteForwarding","StatusBarCust","DisableRightClick","UsingPopupWindow"
,"IframeRedirection","AgeofDomain","DNSRecording","WebsiteTraffic","PageRank","Go
ogleIndex","LinksPointingToPage","StatsReport"
], "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,-
1,1,0,1]}]}
response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/27c47874-fd3f-4c1c-aefa-
afa3d1738374/predictions?version=2022-11-17', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response for given input")
print(response_scoring.json())
predictions=response_scoring.json()
x = math.floor(y_pro_non_phishing*1000)/10
pred=print(predictions['predictions'][0]['values'][0][0])
if(pred == -1):
print("The Website is unsafe")
else:
print("The Website is safe")
return render_template('index.html',xx =x,url=url )
return render_template("index.html", xx =-1)
if __name__ == "__main__":
app.run(debug=True,port=2020)
```

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-25913-1659977253

Project Demo link:

https://drive.google.com/file/d/1Vp811IKarZCXBiiv26JEQXH0oLJZsO3d/view