WEB PHISHING DETECTION

PROJECT REPORT

team id	PNT2022TMID21631
team members	R.Hari Ganesh S.Failur Rahuman A.Ghoushick E.Eber Sheckel
Department	CSE

Project Report

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

    a. Feature 1

    b. Feature 2

    c. Database Schema (if Applicable)

8. **TESTING**

    a. Test Cases

    b. User Acceptance Testing

9. **RESULTS**

    a. Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

1.Introduction

1.1 Project Overview

This project deals with the detection of web phishing which helps a lot of web users to find out the illegitimate websites which has main goal of stealing the users privacy for their own good. This project that we implement helps the user to easily detect the malicious website through the ML algorithm that has been used in this system. The efficiency of this system's performance and scalability is high due to the concept of data mining that has been implemented in this system. When a user tends to login in to a website, this system provides a pop message of warning if this type of website matches with the pattern that has been generated here to find the difference between malicious and legitimate user. This system is able to analyze various kinds of malicious websites using classification technique used in data mining and also tends to produce the accurate final output for web phishing detection.

1.2 Purpose

There are various kinds of users who will be doing some activity of file uploading, file downloading, searching data as well as doing transaction of various documents and money transactions in various kinds of websites so if the websites that the user has been using tends to be malicious then these kinds of websites able to steal their privacy details like their login credentials, as well as their bank account details.The websites that causes these kinds of illegal activities are called as web Phishing, To avoid this above scenarios ,This project has been implemented with the main purpose to detect these malicious websites by analyzing as well giving necessary warning to the users about the websites that they have been using which helps the user to safeguard their important and privacy details.

2. Liteerature Survey

2.1 Existing Problem

Nowadays Web Phishing is trending among public because of the huge growth in the number of various malicious websites which has the main goal of stealing their users’s important details like account information, login credentials etc. This problem not only affects the particular user it also affects the people who are contacted to that user as well thus Web phishing tends to affect the whole community by gathering all the details of the users and the person who are responsible for this kind of activity use this gathered information to generate money for their own good. According to the huge development of technology , the web Phishing tends to occur in more faster rate and the people who are working in the cyber security team finds it difficult to detect these kinds of malicious websites. The system that tends to detect these in a fast and efficient manner has not been introduced to the world yet.

2.2 References

[1] Higashino, M., et al. An Anti-phishing Training System for Security Awareness and Education Considering Prevention of Information Leakage. in 2019 5th International Conference on Information Management (ICIM). 2019.

[2] H. Bleau, Global Fraud and Cybercrime Forecast,. 2017.

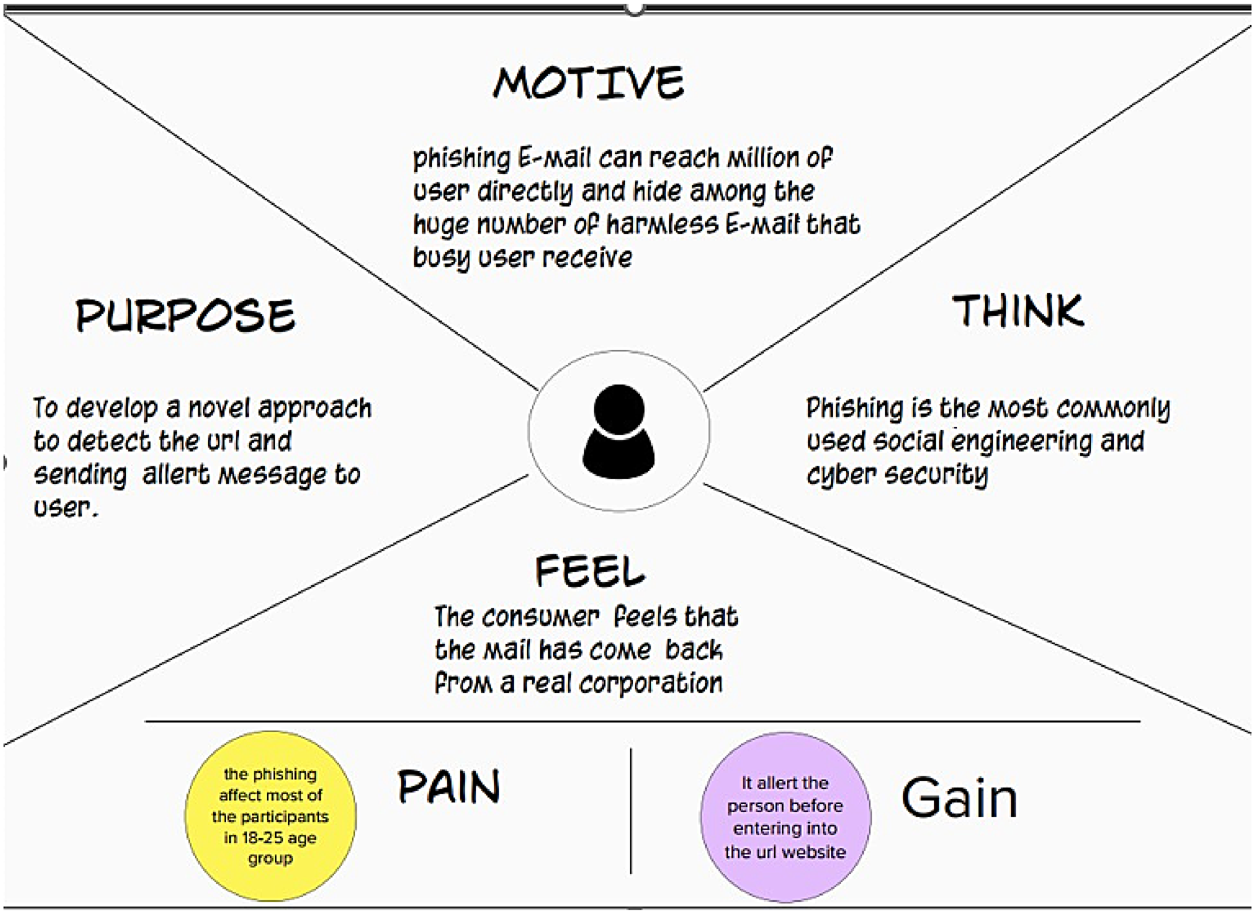
[3] Michel Lange, V., et al., Planning and production of grammatical and lexical verbs in multi-word messages. PloS one, 2017. 12(11): p. e0186685-e018668

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. Phishers 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.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



phishing E-mail can reach million of user directly and hide among the huge number of harmless E-mail that busy user receive

To develop a novel approach to detect the url and sending allert message to user.

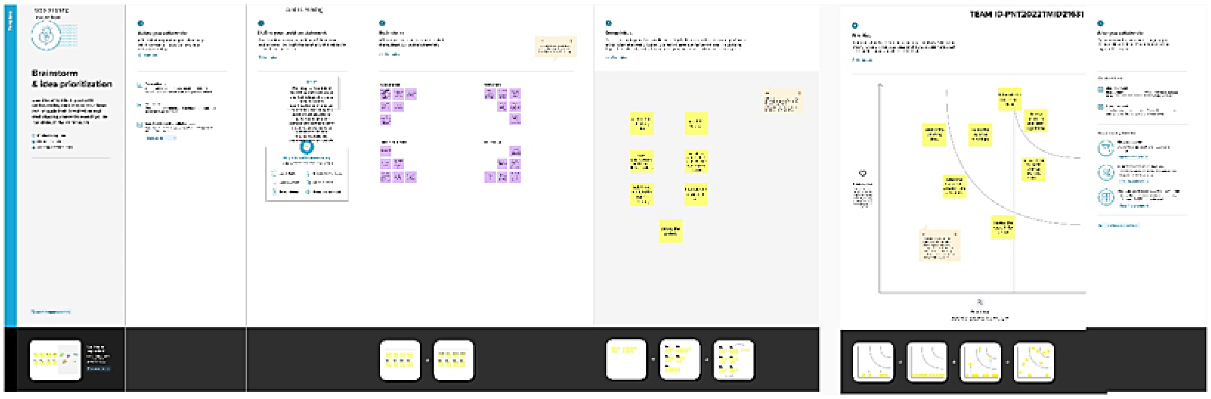
Phishing is the most commonly used social engineering and cyber security

The consumer feels that the mail has come back from a real corporation

the phishing affect most of the participants in 18-25 age group

It allert the person before entering into the url website

3.2 Ideation & Brainstorming



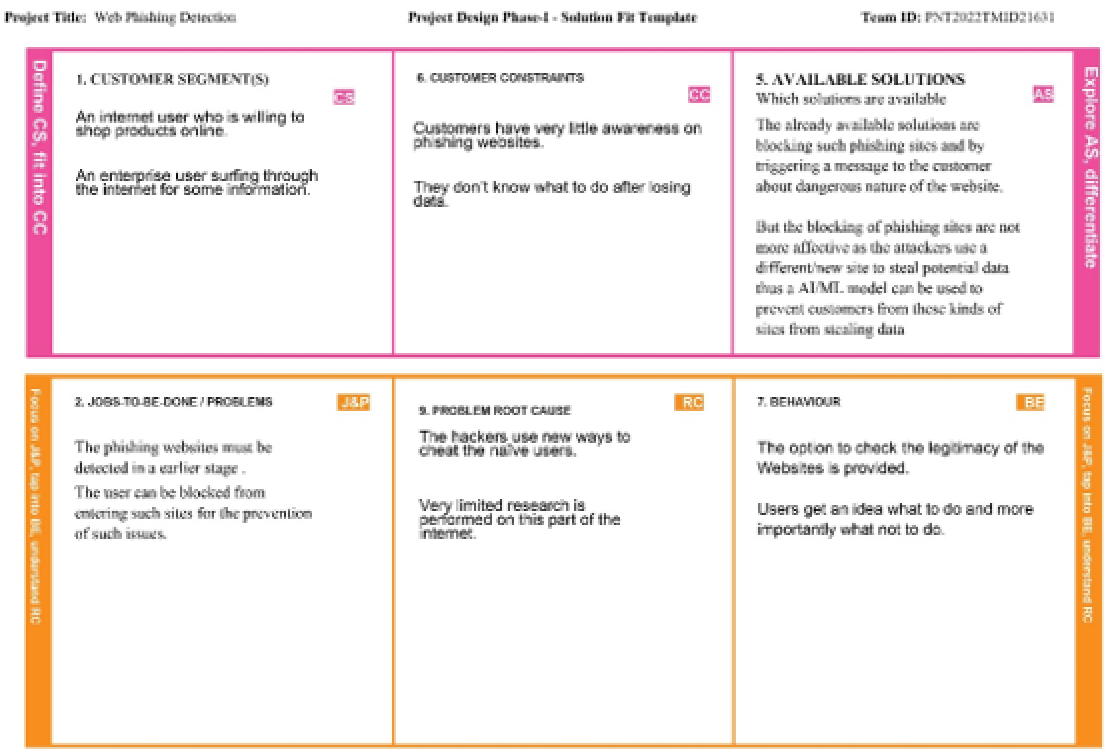
3.3 Proposed Solution

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 here every day, 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 their 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 efficieny to detect the malicious websites. thus scalability of this project will be high .

3.4 Problem Solution fit



Project Title: Web Phishing Detection

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID21631

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

An internet user who is willing to shop products online.

An enterprise user surfing through the internet for some information.

6. CUSTOMER CONSTRAINTS CC

Customers have very little awareness on phishing websites.

They don't know what to do after losing data.

5. AVAILABLE SOLUTIONS AS

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 website.

But the blocking of phishing sites are not more 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 from stealing data

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

The phishing websites must be detected in a earlier stage .

The user can be blocked from entering such sites for the prevention of such issues.

9. PROBLEM ROOT CAUSE RC

The hackers use new ways to cheat the naive users.

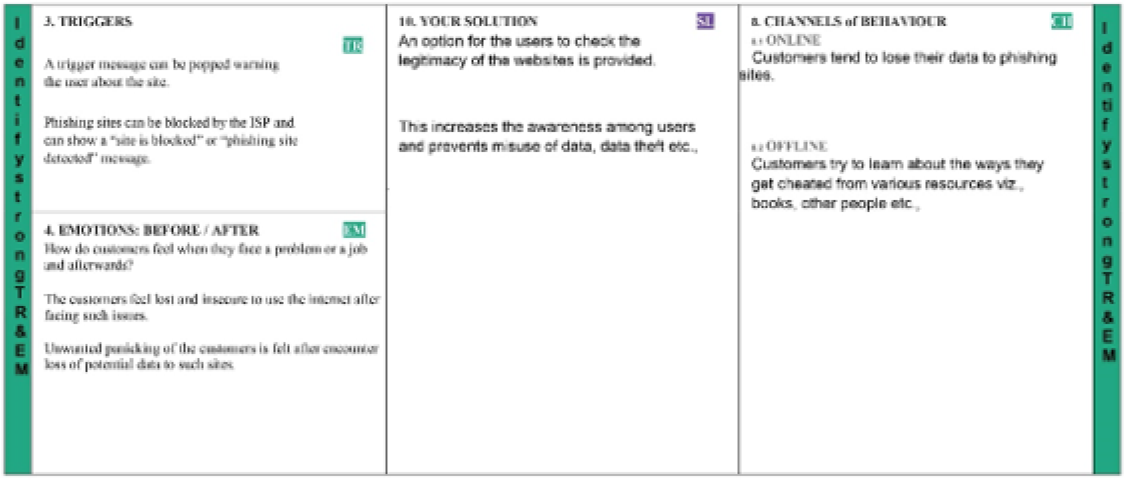
Very limited research is performed on this part of the internet.

7. BEHAVIOUR BE

The option to check the legitimacy of the Websites is provided.

Users get an idea what to do and more importantly what not to do.

Focus on J&P, tap into BE, understand RC



IdentifystrongTR&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.

This increases the awareness among users and prevents misuse of data, data theft etc.,

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Customers tend to lose their data to phishing sites.

8.2 OFFLINE

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

IdentifystrongTR&EM

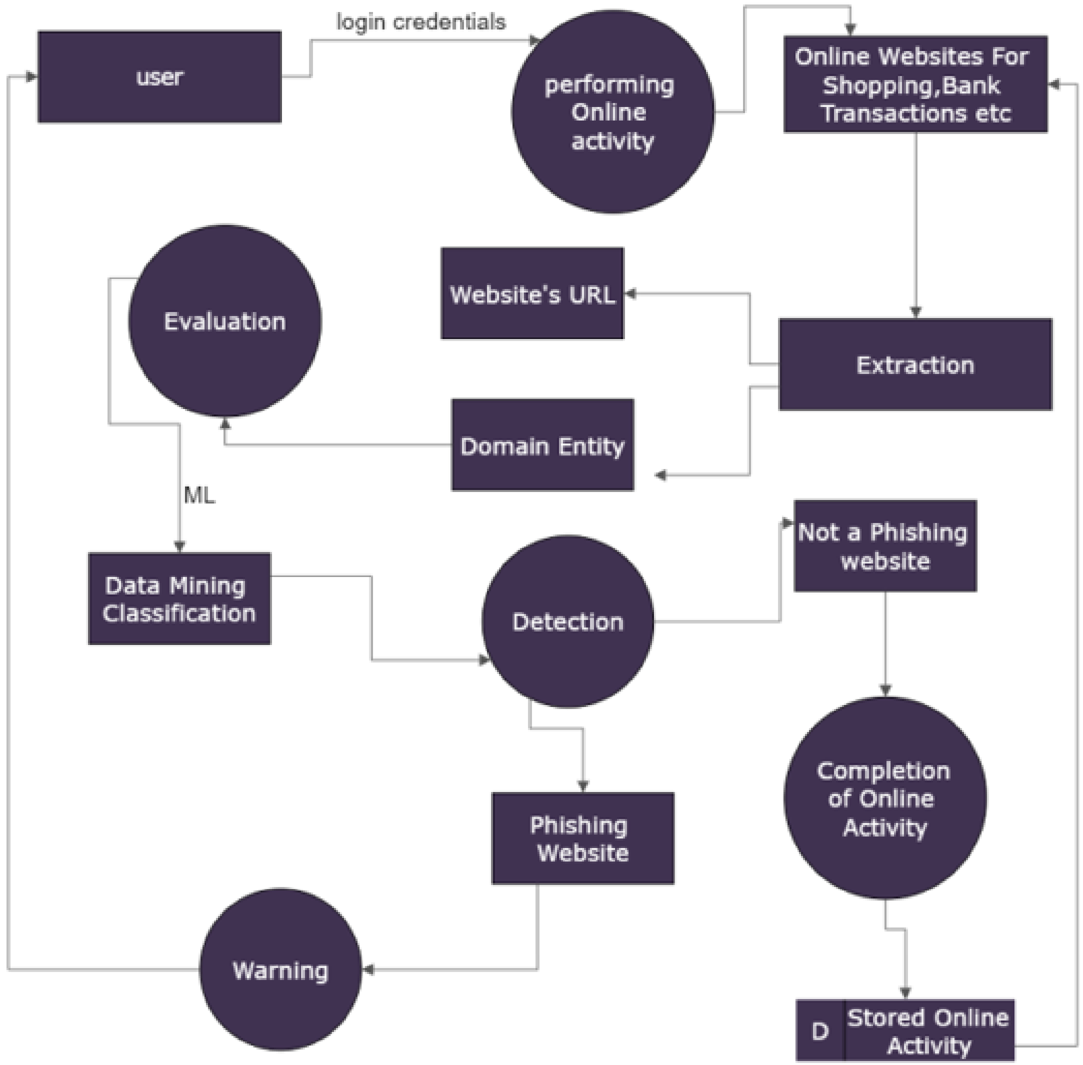
4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR NO.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Input	User inputs an URL in required field to check its validation.
FR-2	Website Comparison	Model compares the websites using Blacklist and Whitelist approach.
FR-3	Feature extraction	After comparing, if none found on comparison then it extracts feature using heuristic and visual similarity approach.
FR-4	Prediction	Model predicts the URL using Machine Learning algorithms such as Logistic Regression, KNN
FR-5	Classifier	Model sends all output to classifier and produces final result.
FR-6	Announcement	Model then displays whether website is a legal site or a phishing site.
FR-7	Events	This model needs the capability of retrieving and displaying accurate result for a website

5. PROJECT DESIGN

5.1 Data Flow Diagrams



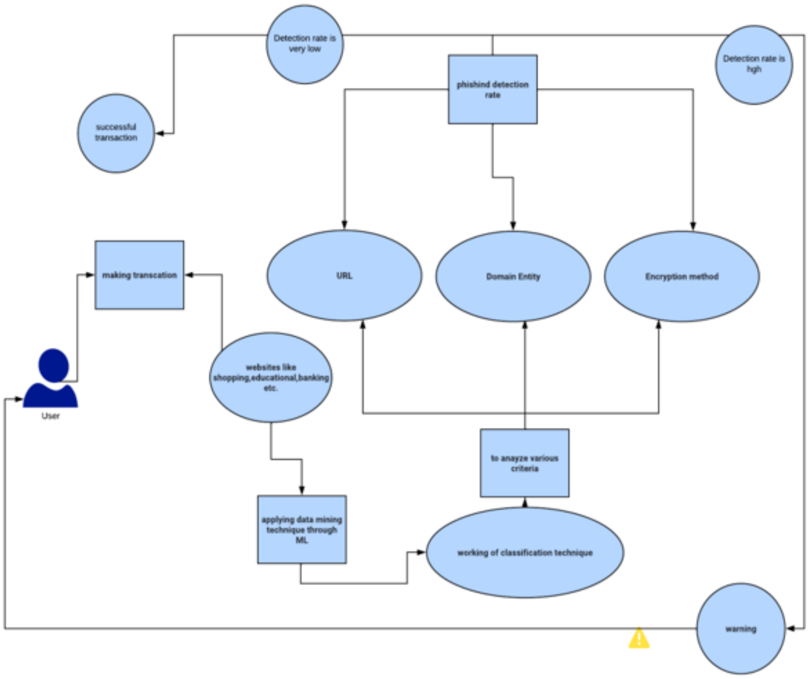
5.2 Solution & Technical Architecture

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.

Solution Architecture Diagram:


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

5.3 User Stories

User Stories

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

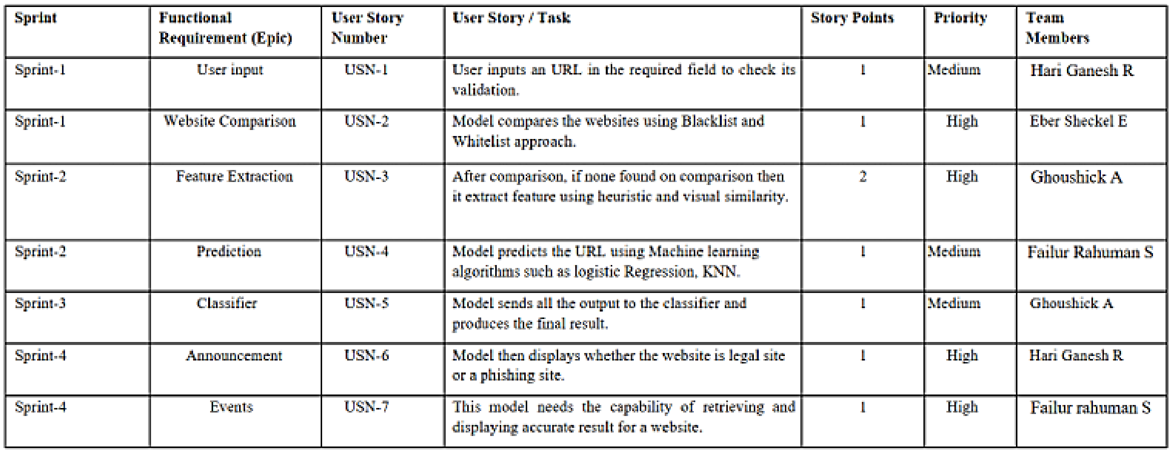
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Sign Up	USN-1	As a user I am able to sign up the application by providing my username, password and G-mail account and Facebook account.	I can access my account / dashboard	High	Sprint-1
USN-2	As a user,when I have completed my registration for the application I am capable of receiving a confirmation mail from that application	I can receive confirmation email & click confirm	High	Sprint-1		

USN-3	As a Facebook user I am capable of registering the application from this platform	I can register and access the dashboard with Facebook Login	Low	Sprint-2		
USN-4	As a G-mail user I am capable of registering the application from this platform as well	I can register and access the dashboard with G-Mail account	High	Sprint-1		
Sign in	USN-5	As a user, I can sign into the application by entering same username/email & password which I have been used for the sign in purpose	I can successfully able to login to the application	High	Sprint-1	
Dashboard						
Customer (Web user)	Input from User	USN-1	As a web user, I am capable of using the website URL, Domain entity for evaluation purpose to find out whether the currently using website is a phishing one or not	I can provide the URL and Domain entity for the evaluating the website	High	Sprint-1

Customer Care Executive	Extraction process	USN-1	When the website URL and domain entity has been provided, it will go under the process of extracting the information of that website for phishing detection	I can view the completion of the extraction process stage	High	Sprint-1
Administrator	Detection	USN-1	After the extraction purpose the model will be able to categorize it from other safe website through data mining classification technique through ML	In this scenario I can distinguish the phishing website from other secure websites	High	Sprint-1
Producing final result	USN-2	The model is able to produce a final result to the user after the completion of detection process	In this scenario I can view the final output given to me by the administrator	Medium	Sprint-2	

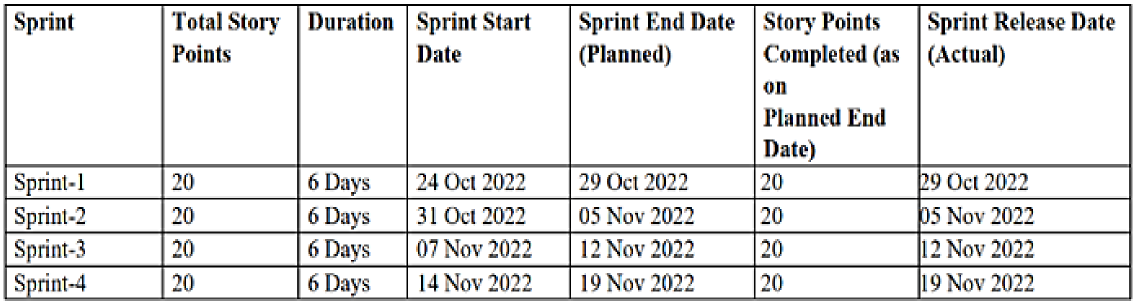
6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User input	USN-1	User inputs an URL in the required field to check its validation.	1	Medium	Hari Ganesh R
Sprint-1	Website Comparison	USN-2	Model compares the websites using Blacklist and Whitelist approach.	1	High	Eber Sheckel E
Sprint-2	Feature Extraction	USN-3	After comparison, if none found on comparison then it extract feature using heuristic and visual similarity.	2	High	Ghoushick A
Sprint-2	Prediction	USN-4	Model predicts the URL using Machine learning algorithms such as logistic Regression, KNN.	1	Medium	Failur Rahuman S
Sprint-3	Classifier	USN-5	Model sends all the output to the classifier and produces the final result.	1	Medium	Ghoushick A
Sprint-4	Announcement	USN-6	Model then displays whether the website is legal site or a phishing site.	1	High	Hari Ganesh R
Sprint-4	Events	USN-7	This model needs the capability of retrieving and displaying accurate result for a website.	1	High	Failur rahuman S

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

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the home page, when user enter the link in URL	1. Internet connection 2. Web browser such as Googl 3. User know the link http://127.0.0.1.5000 4.Mobile ,Laptop, or System...
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in home page	1. Internet connection 2. Web browser such as Googl 3. User know the link http://127.0.0.1.5000 4.Mobile ,Laptop, or System...
LoginPage_TC_OO3	Functional	Home page	Verify user is redirected to the about page, when the user click the "About "	1. Internet connection 2. Web browser such as Googl 3. User know the link http://127.0.0.1.5000

			button	4.Mobile ,Laptop, or System...
LoginPage_TC_OO4	Functional	Home page	Verify user is redirected to phishing website detection page when user click the "Get started" button in the home page.	1. Internet connection 2. Web browser such as Googl 3. User know the link http://127.0.0.1.5000 4.Mobile ,Laptop, or System...
LoginPage_TC_OO5	Functional	About page	Verify user is redirected to phishing website detection page when user click the "check your website" button in the about page.	1. Internet connection 2. Web browser such as Googl 3. User know the link http://127.0.0.1.5000 4.Mobile ,Laptop, or System...
	Functional	Phishing website detection page	Verify it shows whether the URL entered by the user is safe or unsafe.	https://portal.naanmudhalvan gin

LoginPage_TC_OO6				
LoginPage_TC_OO7	Functional	Phishing website detection page	Verify it shows whether the URL entered by the user is safe or unsafe.	https://www.searchonlineinfo

LoginPage_TC_OO8	Functional	Phishing website detection page	Verify it shows whether the URL entered by the user is safe or unsafe.	www.searchonlineinfo.com/
LoginPage_TC_OO9	Functional	Phishing website detection page	Verify it shows whether the URL entered by the user is safe or unsafe.	portal.naanmudhalvan.tn.gov.i

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

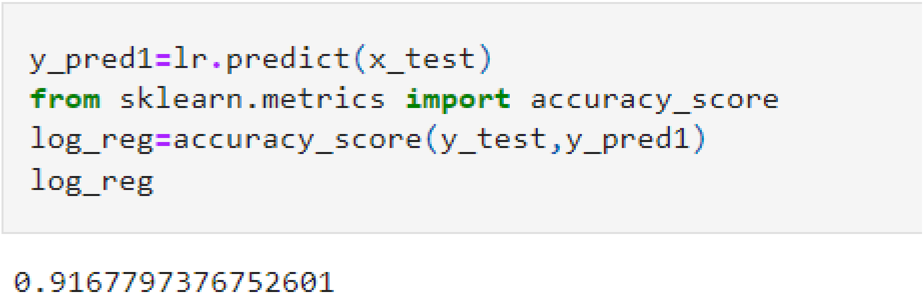
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	4	2	3	20
Duplicate	3	0	3	0	4
External	3	3	0	1	6
Fixed	12	2	4	18	39
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	3	4
Won't Fix	0	5	2	2	6
Totals	29	14	13	26	77

8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	48	0	0	48
Security	4	0	0	4
Outsource Shipping	1	0	0	1
Exception Reporting	3	0	0	3
Final Report Output	4	0	0	4
Version Control	5	0	0	5

9.Results

9.1 Performance Metrics:



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

```
0.9167797376752601
```

10. Advantages & Disadvanatages

Advantages

- This project tends to produce the result in a more structured, organized as well as in efficient manner.
- This project has high scalability and performance since it uses Ml technology for implementation.
- This project is able to find various kinds of malicious websites by using the concept of Classification algorithm in Data mining .
- This project tends to help the user to safeguard their privacy details like login credentials, account details etc.

Disadvantages

- This project finds difficult to detect any malicious activity in the website if the user opens many websites at the same time.
- This project may not able to find an malicious activity in other areas like spam emails,links other than websites.

11.Conclusion

This paper aims to enhance detection method to detect phishing website using machine learning technology. Also , classifiers generated by machine learning algorithms identify legitimate phishing websites. The proposed technique can detect new temporary phishing sites and reduce the damage caused by phishing attacks. The performance of the proposed technique based on machine learning is more effective that previous phishing detection technologies. In the future, it will be useful to investigate the impact of feature selection using various algorithms. This project is able to get implemented in lot of fields like medical, finance,education and military etc.

12.Future Scope

This project can be enhanced to detect in many areas other than websites like emails, various links etc. An enhancement in the technique of classification can be made in this project which helps to detect any malicious websites in a faster rate. The dataset which was generated for this project will gets improvement to store large amount of data as well as the train the model according to the pattern that has been given to acquire the expected output and generate the most preferable result as well. In future it may gets implemented together with Big data by using the tool Hadoop to process the informations that has been fed to it in very short period of time.

13.APPENDIX

app.py

```
import numpy as np

from flask import Flask,request,jsonify,render_template

import pickle

#importing the inputScript file used to analyze the URL

import inputScript
```

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


#fetches the URL given by the URL and passes to inputScript
@app.route('/_predict',methods=['POST'])
def y_predict():
    url = request.form['URL']
    checkprediction = inputScript.main(url)
    prediction =model.predict(checkprediction)
    print(prediction)
    output=prediction[0]
    if(output==1):
        pred="Your are safe!!  This is a Legitimate Website."
    else:
        pred="Your are on the wrong site. Be cautious!"
    return render_template('final.html', prediction_text='{}'.formate(pred),url=url)


#Takes the input parameters fetched from the URL by inputsScript and returns the
@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''

    data = request.get_json(force=True)
    prediction =model.y_predict([np.array(list(data.values()))])

    output =prediction[0]
    return jsonify(output)
```

```
if __name__ =='__main__':

    app.run( debug=True)
```

inputscript.py

```
import regex

from tldextract import extract

import ssl

import socket

from bs4 import BeautifulSoup

import urllib.request

import whois

import datetime




def url_having_ip(url):

#using regular function
```

```
#    symbol = regex.findall(r'(http((s)?)://)((((\d)+).)*)((\w)+)(/((\w)+))?',url)
 #   if(len(symbol)!=0):
  #      having_ip = 1 #phishing
   # else:
    #    having_ip = -1 #legitimate
    #return(having_ip)
    return 0
def url_length(url):
    length=len(url)
    if(length<54):
        return -1
    elif(54<=length<=75):
        return 0
    else:
        return 1
def url_short(url):
    #ongoing
    return 0
def having_at_symbol(url):
    symbol=regex.findall(r'@',url)
    if(len(symbol)==0):
        return -1
    else:
        return 1
def doubleSlash(url):
    #ongoing
    return 0
def prefix_suffix(url):
    subDomain, domain, suffix = extract(url)
    if(domain.count('-')):
        return 1
    else:
        return -1


def sub_domain(url):

    subDomain, domain, suffix = extract(url)

    if(subDomain.count('.')==0):

        return -1

    elif(subDomain.count('.')==1):

        return 0

    else:

        return 1


def SSLfinal_State(url):

    try:

#check wheather contains https

        if(regex.search('^https',url)):

            usehttps = 1
        else:

            usehttps = 0

#getting the certificate issuer to later compare with trusted issuer

        #getting host name

        subDomain, domain, suffix = extract(url)

        host_name = domain + "." + suffix

        context = ssl.create_default_context()

        sct = context.wrap_socket(socket.socket(), server_hostname = host_name)

        sct.connect((host_name, 443))

        certificate = sct.getpeercert()

        issuer = dict(x[0] for x in certificate['issuer'])

        certificate_Auth = str(issuer['commonName'])

        certificate_Auth = certificate_Auth.split()

        if(certificate_Auth[0] == "Network" or certificate_Auth == "Deutsche"):

            certificate_Auth = certificate_Auth[0] + " " + certificate_Auth[1]

        else:
            certificate_Auth = certificate_Auth[0]

        trusted_Auth = ['Comodo','Symantec','GoDaddy','GlobalSign','DigiCert','StartCom','Entrust','Verizon','Trustwave','Unizeto','Buypass','QuoVadis','Deutsche Telekom','Network Solutions','SwissSign','IdenTrust','Secom','TWCA','GeoTrust','Thawte','Doster','VeriSign']

#getting age of certificate

        startingDate = str(certificate['notBefore'])

        endingDate = str(certificate['notAfter'])

        startingYear = int(startingDate.split()[3])

        endingYear = int(endingDate.split()[3])

        Age_of_certificate = endingYear-startingYear



#checking final conditions

        if((usehttps==1) and (certificate_Auth in trusted_Auth) and (Age_of_certificate>=1) ):

            return -1 #legitimate

        elif((usehttps==1) and (certificate_Auth not in trusted_Auth)):

            return 0 #suspicious
        else:
            return 1 #phishing

    except Exception as e:
        return 1
def domain_registration(url):
    try:
        w = whois.whois(url)
        updated = w.updated_date
        exp = w.expiration_date
        length = (exp[0]-updated[0]).days
        if(length<=365):
            return 1
        else:
            return -1
    except:
        return 0

def favicon(url):

    #ongoing

    return 0

def port(url):

    #ongoing

    return 0

def https_token(url):

    subDomain, domain, suffix = extract(url)

    host =subDomain +'.' + domain + '.' + suffix

    if(host.count('https')): #attacker can trick by putting https in domain part

        return 1

    else:

        return -1

def request_url(url):

    try:
```

```
subDomain, domain, suffix = extract(url)
websiteDomain = domain
opener = urllib.request.urlopen(url).read()
soup = BeautifulSoup(opener, 'lxml')
imgs = soup.findAll('img', src=True)
total = len(imgs)

linked_to_same = 0
avg =0
for image in imgs:
    subDomain, domain, suffix = extract(image['src'])
    imageDomain = domain
    if(websiteDomain==imageDomain or imageDomain==''):
        linked_to_same = linked_to_same + 1
vids = soup.findAll('video', src=True)
total = total + len(vids)
```

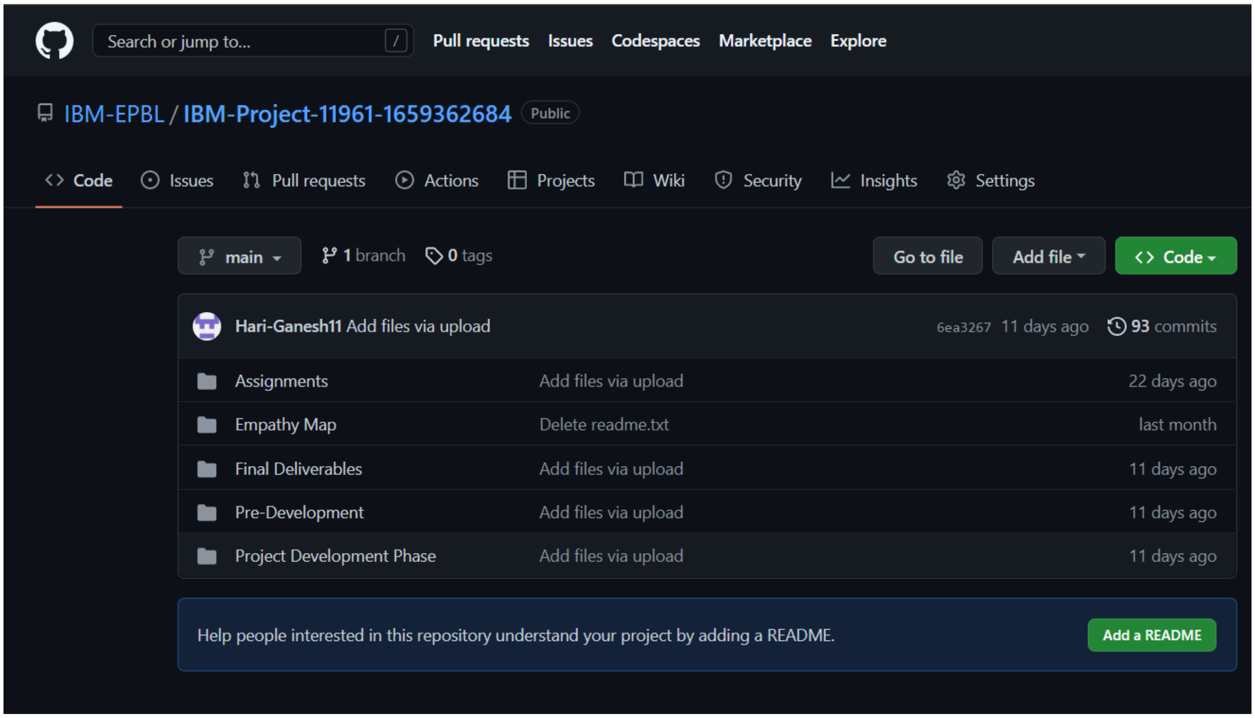
```
for video in vids:
    subDomain, domain, suffix = extract(video['src'])
    vidDomain = domain
    if(websiteDomain==vidDomain or vidDomain==''):
        linked_to_same = linked_to_same + 1
linked_outside = total-linked_to_same
if(total!=0):
    avg = linked_outside/total

if(avg<0.22):
    return -1
elif(0.22<=avg<=0.61):
    return 0
else:
    return 1
    except:

        return 0



def url_of_anchor(url):

    try:

        subDomain, domain, suffix = extract(url)

        websiteDomain = domain



        opener = urllib.request.urlopen(url).read()

        soup = BeautifulSoup(opener, 'lxml')

        anchors = soup.findAll('a', href=True)

        total = len(anchors)

        linked_to_same = 0

        avg = 0

        for anchor in anchors:

            subDomain, domain, suffix = extract(anchor['href'])
            anchorDomain = domain
            if(websiteDomain==anchorDomain or anchorDomain==''):
                linked_to_same = linked_to_same + 1
        linked_outside = total-linked_to_same
        if(total!=0):
            avg = linked_outside/total

        if(avg<0.31):
            return -1
        elif(0.31<=avg<=0.67):
            return 0
        else:
            return 1
    except:
        return 0
```

```
def Links_in_tags(url):

    try:

        opener = urllib.request.urlopen(url).read()

        soup = BeautifulSoup(opener, 'lxml')



        no_of_meta =0

        no_of_link =0

        no_of_script =0

        anchors=0

        avg =0

        for meta in soup.find_all('meta'):

            no_of_meta = no_of_meta+1

        for link in soup.find_all('link'):

            no_of_link = no_of_link +1

        for script in soup.find_all('script'):

            no_of_script = no_of_script+1
    for anchor in soup.find_all('a'):
        anchors = anchors+1
    total = no_of_meta + no_of_link + no_of_script+anchors
    tags = no_of_meta + no_of_link + no_of_script
    if(total!=0):
        avg = tags/total

    if(avg<0.25):
        return -1
    elif(0.25<=avg<=0.81):
        return 0
    else:
        return 1
except:
    return 0
```

```
def sfh(url):

    #ongoing

    return 0



def email_submit(url):

    try:

        opener = urllib.request.urlopen(url).read()

        soup = BeautifulSoup(opener, 'lxml')

        if(soup.find('mailto:')):

            return 1

        else:

            return -1

    except:

        return 0



def abnormal_url(url):
    #ongoing
    return 0

def redirect(url):
    #ongoing
    return 0

def on_mouseover(url):
    #ongoing
    return 0

def rightClick(url):
    #ongoing
    return 0

def popup(url):
    #ongoing

    return 0



def iframe(url):

    #ongoing

    return 0



def age_of_domain(url):

    try:

        w = whois.whois(url)

        start_date = w.creation_date

        current_date = datetime.datetime.now()

        age =(current_date-start_date[0]).days

        if(age>=180):

            return -1

        else:
            return 1

    except Exception as e:

        print(e)

        return 0



def dns(url):

    #ongoing

    return 0



def web_traffic(url):

    #ongoing

    return 0



def page_rank(url):

    #ongoing

    return 0
```

```
def google_index(url):

    #ongoing

    return 0




def links_pointing(url):

    #ongoing

    return 0



def statistical(url):

    #ongoing

    return 0



def main(url):

    check = [[url_having_ip(url),url_length(url),url_short(url),having_at_symbol(url),
            doubleSlash(url),prefix_suffix(url),sub_domain(url),SSLfinal_State(url),
            domain_registration(url),favicon(url),port(url),https_token(url),request_url(url),
            url_of_anchor(url),Links_in_tags(url),sfh(url),email_submit(url),abnormal_url(url),
            redirect(url),on_mouseover(url),rightClick(url),popup(url),iframe(url),
            age_of_domain(url),dns(url),web_traffic(url),page_rank(url),google_index(url),
            links_pointing(url),statistical(url)]]
    print(check)
    return check
```

Github



IBM-EPBL / IBM-Project-11961-1659362684 Public

Hari-Ganesh11 Add files via upload 6ea3267 11 days ago 93 commits

Assignments	Add files via upload	22 days ago
Empathy Map	Delete readme.txt	last month
Final Deliverables	Add files via upload	11 days ago
Pre-Development	Add files via upload	11 days ago
Project Development Phase	Add files via upload	11 days ago

Help people interested in this repository understand your project by adding a README.

github URL:https://github.com/IBM-EPBL/IBM-Project-11961-1659362684.git

Demo Video URL:https://github.com/IBM-EPBL/IBM-Project-11961-1659362684/blob/main/Final%20Deliverables/Demo_Video_PNT2022TMID21631.mp4