RMK ENGINEERING

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206, Gummidipoondi Taluk, Thiruvallur District.

PROJECT

WEB PHISHING DETECTION

DONE BY

TEAM ID: PNT2022TMID15890

VASANTHAZHAGAN R A (111719104170)

SHRINIWAAZ K G (111719104144)

SATHISH S (111719104138)

THARUN M V (111719104163)

INDEX

1. INTRODUCTION

For an average individual, it is very difficult to differentiate a phishing website from a legitimate website. It is not feasible to identify a fake website just by looking at them because they look similar to the original ones. This makes it hard to trust any websites. Users also share sensitive information like passwords, card details, etc. in websites in order to utilize the websites. Presence of these fake makes us feel less secure. These phishing sites also affects the reputation of the original website and its organization. Users feel frustrated, unsecured, confused, stressed, worried, annoyed, etc. which makes them give up. These web phishing sites are often used by the cypher harasser to web phish the user or normal people from their personal data and their online accounts. Around 8% of total email received by a person in his whole life is a spam or a phishing mail. These web phishing cypher harassers are increasing these days. So, it is not easy to catch all of them. Instead we can use some detection techniques to avoid being caught by these phishing sites.

1.1. Project Overview

These days most sites ask users for their personal data to process further. There are times where these data can be stolen. Those stolen data are often used to threaten the user mainly for ransom. One such way of stealing data is Web Phishing. The harassers create a fraudulent website that resembles the original websites and make the users to type in the data by scamming them. Thereby, stealing the user's data. This project creates a model that uses a dataset of 11,000 data samples. These data samples are used to train and test the model. The model that is trained is accessed for checking the fraudulence of the site using a web site that will be user friendly and easy to use.

1.2. Purpose

Web Phishing Detection is a technology where different machine learning algorithm can be used to differentiate a fraudulent website from an authentic one. This really helps users getting robbed of their personal data and prevent them from cyber harassers. The application that is to be created should detect the legitimate site from a suspicious one.

2. LITERATURE SURVEY

2.1. Existing problem

Detecting Phishing Websites Using Machine Learning [1] by Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh, Dr. Aram Alsedrani

The system is based on a machine learning method, particularly supervised learning. We have selected the Random Forest technique due to its good performance in classification. Accuracy of 98.8% and combination of 26 features. There are 36 features that can features that can be extracted from URL, page content and page rank. Using the combinations of these features irrelevant features are removed.

Result - Random combination of features and found it took the shape of normal distribution curve.

Review: Phishing Detection Approaches [2] by AlMaha Abu Zuraiq, Mouhammd Alkasassbeh

Uses different phishing detection approaches which include: Content-Based, Heuristic-Based, and Fuzzy rule-based approaches. Content-based approach does a deep analysis on pages' content. Heuristic Based Approach s discriminative features extracted by understanding and analyzing the structure of phishing web pages.

Result - Each approach has its advantages and disadvantages and improving these approaches is always required.

Real Time Detection of Phishing Websites [3] by Abdulghani Ali Ahmed, Nurul Amirah Abdullah

Proposes a detection technique of phishing websites based on checking Uniform Resources Locators (URLs) of web pages by checking the Uniform Resources Locators (URLs) of suspected web pages. There are few features that used to identify fake site from a legitimate one. Some are URLs, domain identity, security & encryption, source code, page style & contents, web address bar and social human factor. Features of URL and domain names are checked using several criteria such as IP Address, long URL address, adding a prefix or suffix, redirecting using the symbol “//”, and URLs having the symbol “@”.

Result - The paper checks the authenticity of the Universal Resource Locator (URLs) based only on few characteristics for detecting phishing attacks.

Detection of Phishing Websites by using Machine Learning-Based URL Analysis [4] by Mehmet Korkmaz, Ozgur Koray Sahingoz, Banu Diri

Aimed to implement a phishing detection system by analyzing the URL of the webpage. detected 58 different features on the web URL which included words, digits, "=", "?", IP address, etc. They implemented the system by using 8 different algorithms Logistic Regression (LR), K-Nearest Neighborhood (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), XGBoost, Random Forest (RF) and Artificial Neural Network (ANN).

Result - They used 3 datasets and obtained results in 8 different algorithms. Random Forest (RF) is the one seems to produce highest accuracy rate with 94.59%, 90.5%, 91.26% in the 3 datasets respectively.

Phishing Website Detection using Machine Learning Algorithms [5] Rishikesh Mahajan, Irfan Siddavatam

Deals with ML technologies for detection of phishing URLs by extracting and analyzing different features of legitimate and phishing URL. Python programming language is used to extract the features from URL. This paper talks about 3 machine learning algorithm Decision Tree, Random Forest and Support Vector Machine.

Result - Achieved 97.14% detection accuracy using Random Forest Algorithm with lowest false positive rate.

Phishing website detection using novel machine learning fusion approach [6] by A. Lakshmanarao,P.Surya Prabhakara Rao,M M Bala Krishna

Various machine learning algorithms like logistic regression, decision tree classifier, random forest classifier, AdaBoost, gradient boosting classifier for the phishing detection. A dataset from the UCI machine learning repository for the experiments. Two priority algorithms PA1, PA2. Based on the results of priority-based algorithms final fusion model was decided.

Result - A fusion classifier and achieved an accuracy of 97%. The proposed model was tested on one dataset only.

2.2. References

1. Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh, Dr. Aram Alsedrani ‘s ***“Detecting Phishing Websites Using Machine Learning”*** published during 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), 01-03 May 2019.

2. AlMaha Abu Zuraiq, Mouhammd Alkasassbeh ‘s ***“Review: Phishing Detection* Approaches”** published during 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), 09-11 October 2019.

3. Abdulghani Ali Ahmed, Nurul Amirah Abdullah ‘s ***“Real Time Detection of Phishing Websites”*** published during 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 13-15 October 2016.

4. Mehmet Korkmaz, Ozgur Koray Sahingoz, Banu Diri ‘s ***“Detection of Phishing Websites by using Machine Learning-Based URL Analysis”*** published during 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 01-03 July 2020.

5. Rishikesh Mahajan, Irfan Siddavatam ‘s ***“Phishing Website Detection using Machine Learning Algorithms”*** published during International Journal of Computer Applications, October 2018.

6. A. Lakshmanarao,P.Surya Prabhakara Rao,M M Bala Krishna ‘s ***“Phishing website detection using novel machine learning fusion approach”*** published during 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 25-27 March 2021.

2.3. Problem Statement Definition

Problem Statement defines the problem in hand. It helps the developers to understand the problem from the customer point of view and assist them in determining the ideal solution.

I am	Describe customer with 3-4 key characteristics - who are they?	Describe the customer and their attributes here
I'm trying to	List their outcome or "job" the care about - what are they trying to achieve?	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – what bothers them most?	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – what needs to be solved?	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – how does it impact them emotionally?	Describe the emotions the result from experiencing the problems or barriers

2.2.1 Problem Statement

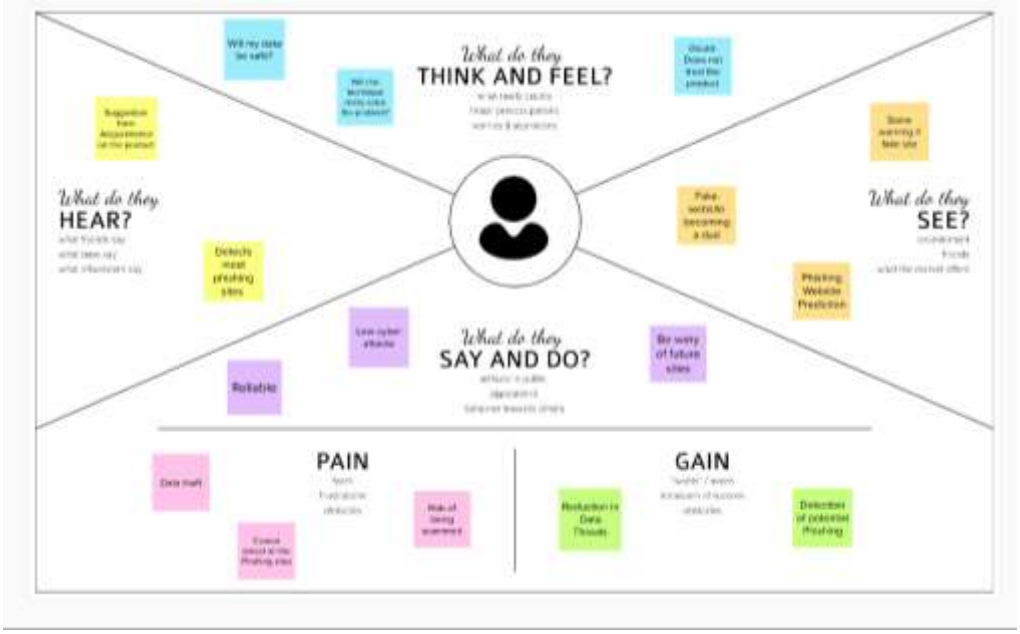
2.2.2 User Problem Statement

3. IDEATION & PROPOSED SOLUTION

3.1.Empathy Map Canvas

Empathy map is a way of determining the user's behavior and attitude. Web phishing concerns users and their data. So, it is required that the developers of the web phishing detection application understand user's perspective on web phishing and the application.

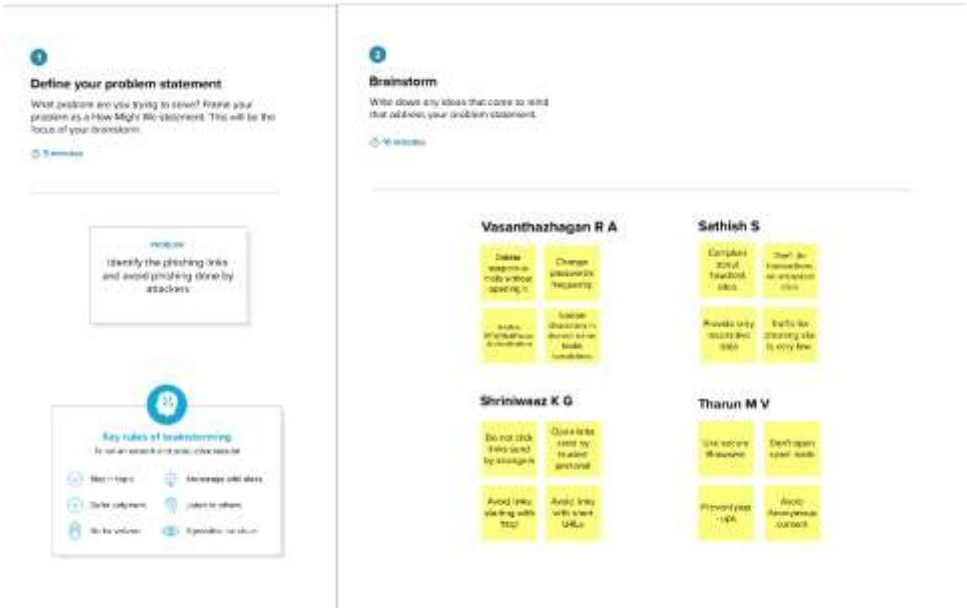
3.1.1 Empathy Map on Web Phishing



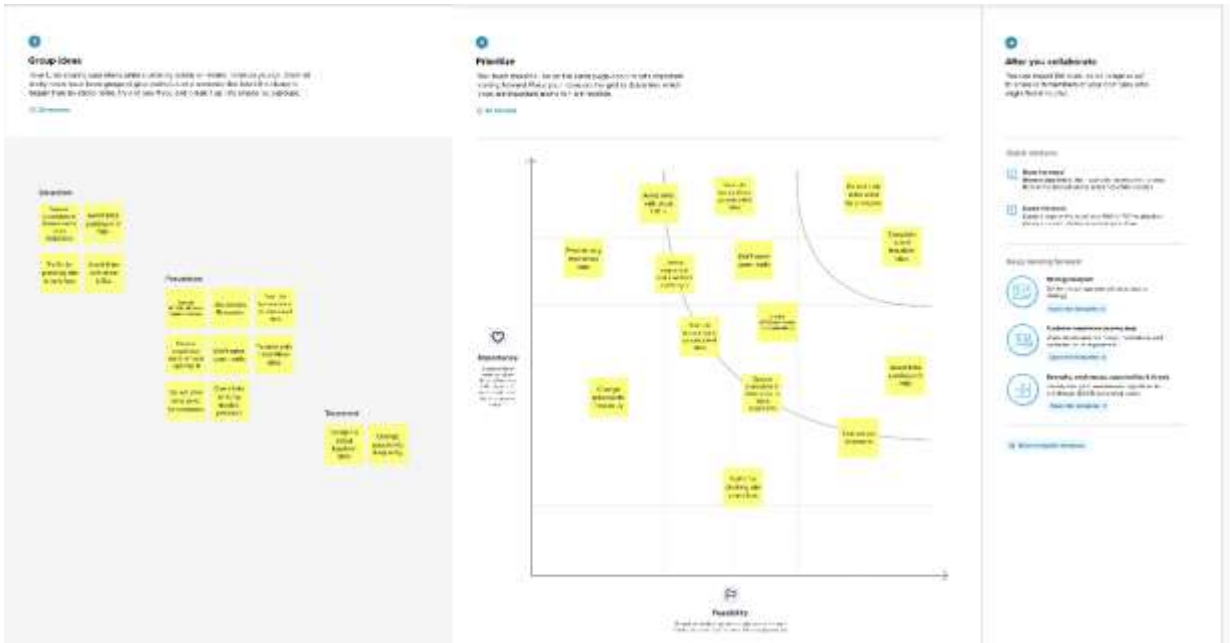
3.1.2 Empathy Map on Web Phishing Detection System

3.2. Ideation & Brainstorming

Finding the solution is not necessarily a single step process. It requires all the minds in the team to come together to find a solution. Ideation & Brainstorming is the phase where each member comes up with their idea and suggestion and the effective one is chosen.



3.2.1 Ideation a)



3.2.2 Ideation b)

3.3. Proposed Solution

Proposed solution is the solution that the developers comes up with from the ideas from the team members. It includes several areas like solution description, uniqueness, customer satisfaction and scalability.

The proposed solution describes an application that analyses the URL of the given site and produces a report showing if the website is a phishing website or an original one. The solution will detect 30 different features related to the given URL for the website like length, IP address, HTTPS token, page rank, google index etc. Then uses these features to detect if it is a phishing website or not. We have tested with three algorithms - Logistic Regression, Decision tree and Random Forest Tree. Among them, Random Forest Tree has given best result with the highest accuracy. It is an application to secure users from phishing websites.

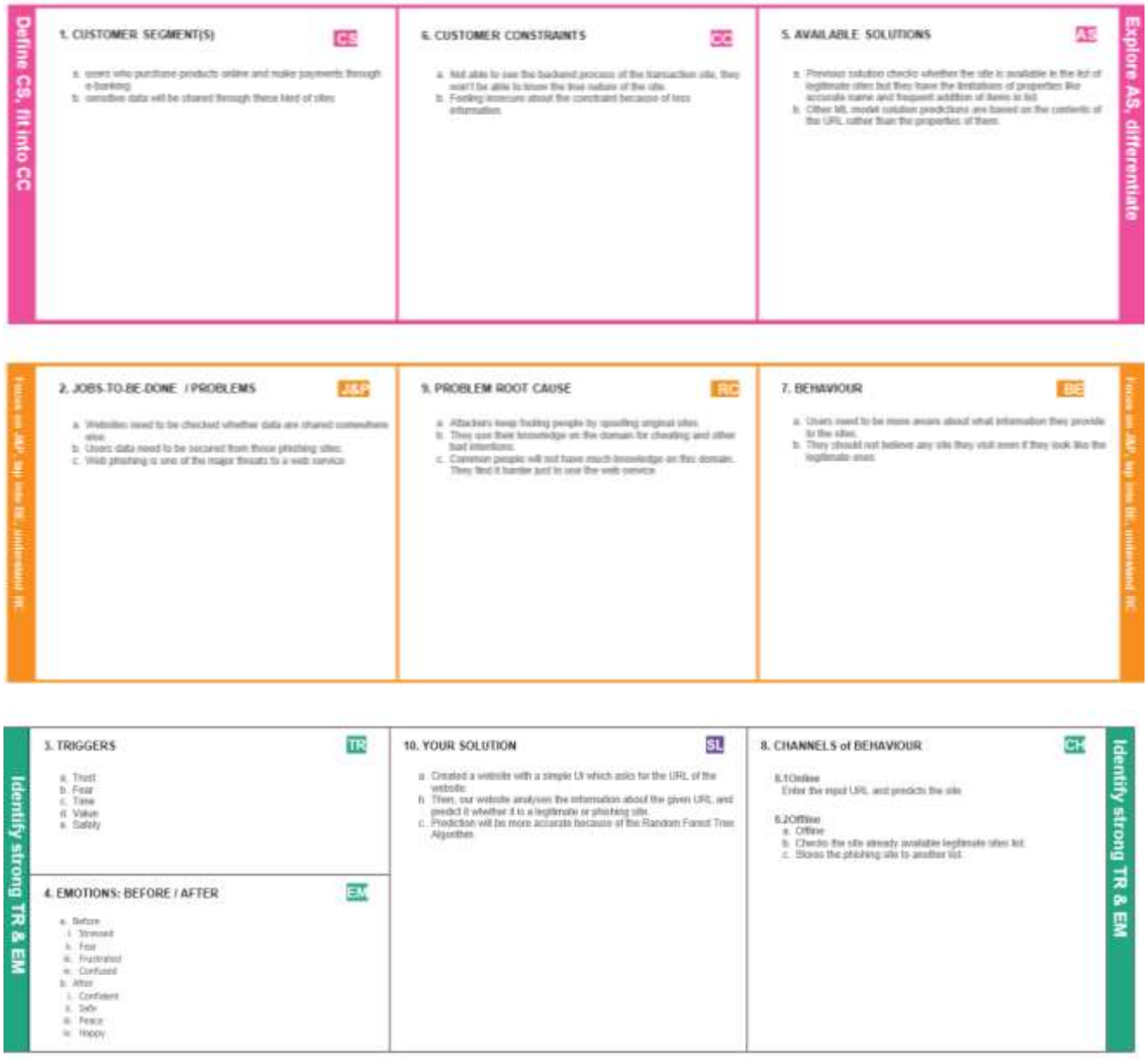
Sl. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	For an average individual, it is very difficult to differentiate a phishing website from a legitimate website. It is not feasible to identify a fake website just by looking at them because they look similar to the original ones. This makes it hard to trust any websites. Users also share sensitive information like passwords, card details, etc in websites in order to utilize the websites. Presence of these fake makes us feel less secure. These phishing sites also affects the reputation of the original website and its organisation. Users feel frustrated, unsecured, confused, stressed, worried, annoyed, etc. which makes them give up.
2.	Idea / Solution description	The solution to the problems described is a Phishing Website Detection application which analyses the URL of the site and give report about the site. It will detect 30 different features related to the URL of the website like length, IP address,

		HTTPS token etc. We had tested with three algorithms like Logistic Regression, Decision tree, Random Forest Tree. Among them, Random Forest Tree has given best result with more accuracy. It is an application to secure users from phishing websites.
3.	Novelty / Uniqueness	It uses Random Forest Tree algorithm, which provided higher accuracy rate among others with concept of binary classification on the website's properties like IP address, HTTPS token, etc to identify the originality of the websites.
4.	Social Impact / Customer Satisfaction	Phishing sites created fear among users' mind about losing their sensitive data. It has major impact on user's mental health and their privacy. By identifying the fake certificates, the user can access and provide information without any fear. It makes them feel that their data was not leaked. It provides immense customer satisfaction as it is very user-friendly and is easily accessible. It is a place where user can find security of the website.
5.	Business Model (Revenue Model)	The revenue is generated usage of the application. More the user using the application the more the revenue generated. If successful, it can also be developed as a browser extension.
6.	Scalability of the Solution	This application is very scalable as it can identify fake websites and original websites using binary classification ML models.

3.3.1 Table: Proposed Solution

3.4. Problem Solution fit

Problem solution fit is the one solution amongst many that actually solves the defined user problem



3.4.1 Problem Solution Fit

4. REQUIREMENT ANALYSIS

4.1.Functional requirement

Functional requirements include the functionality and the functional features that are required for the proposed solution.

FR No.	Functional Requirement	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	Registration through Gmail
FR-2	**Registered User Login**	Login through Gmail and Password
FR-3	**Data Fetch**	Get the URL of the website need to be verified
FR-4	**Result Prediction**	Predict the category of the website using the ML model trained under a certain dataset
FR-5	**Storing Result**	Store the outcome in the appropriate place
FR-6	**Displaying Result**	Notify the user about the nature of the given website

4.1.1 Table: Functional Requirement

4.2.Non-Functional requirements

Non-Functional requirements include the non-functionality features that are required for the proposed solution.

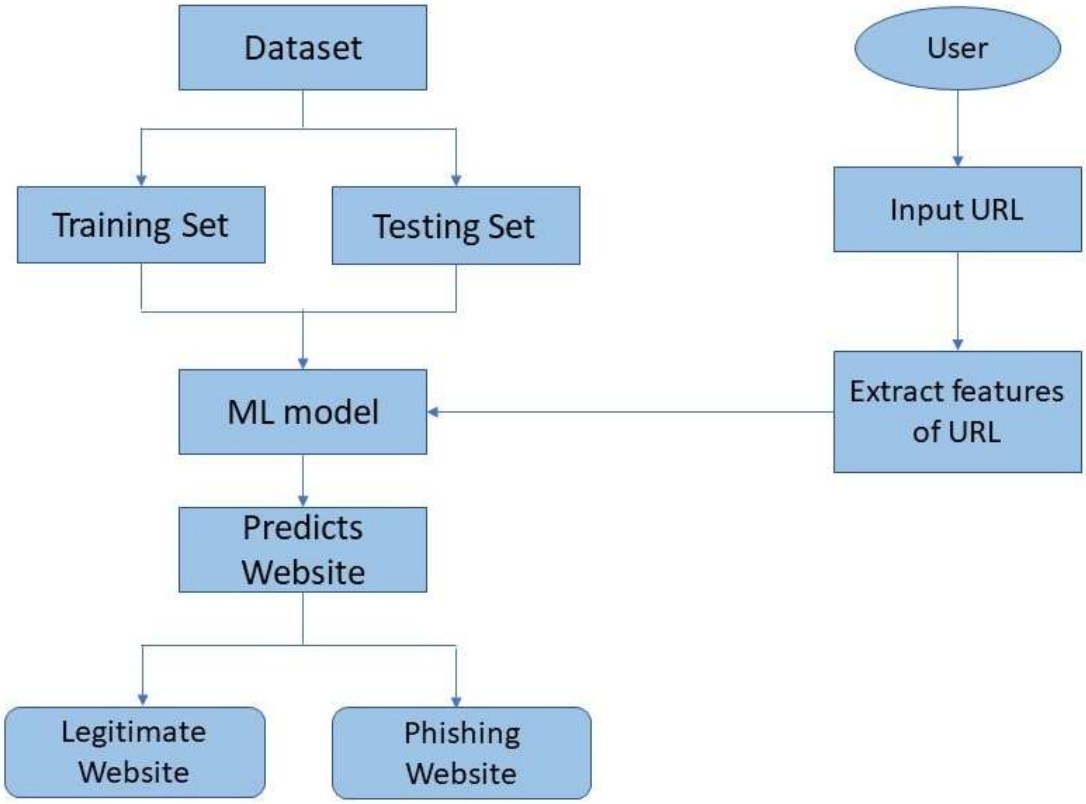
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Users will be able to register and access the website for detecting phishing sites easily. The User Interface should be simple and usable at ease.
NFR-2	**Security**	Data provided by the user should be secure and highest priority should be given to protect those data. Also, admin and authorized personnel can only be allowed to access those data.
NFR-3	**Reliability**	The data will be updated after every prediction. So, the database will be refreshed more often by adding more new websites.
NFR-4	**Performance**	The website shall be simple, dynamic and responsive. While the authentication of the website and other internal process will be done on background.
NFR-5	**Availability**	The website should be developed to identify phishing websites mainly among the payment and banking related domain. These are to be made available to user easily.
NFR-6	**Scalability**	The website shall be accessed by a greater number of users at the same time.

4.2.1 Table: Non - Functional Requirement

5. PROJECT DESIGN

5.1. Data Flow Diagrams

Data flow diagrams are the way of representing the flow of data and information in the system. It depicts the way in which the data travels through the system, how the data enters the system, what happens to the data in - between, where the data are stored.

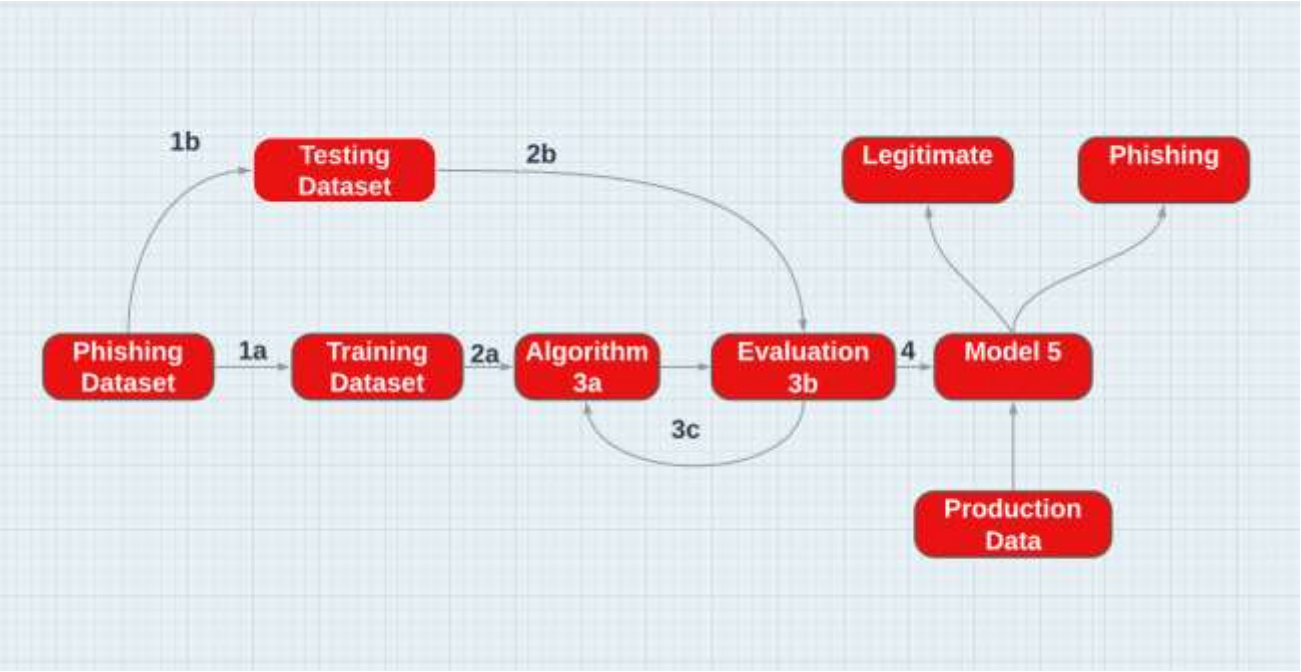


5.1.1 Data Flow Diagram

5.2. Solution & Technical Architecture

Solution Architecture:

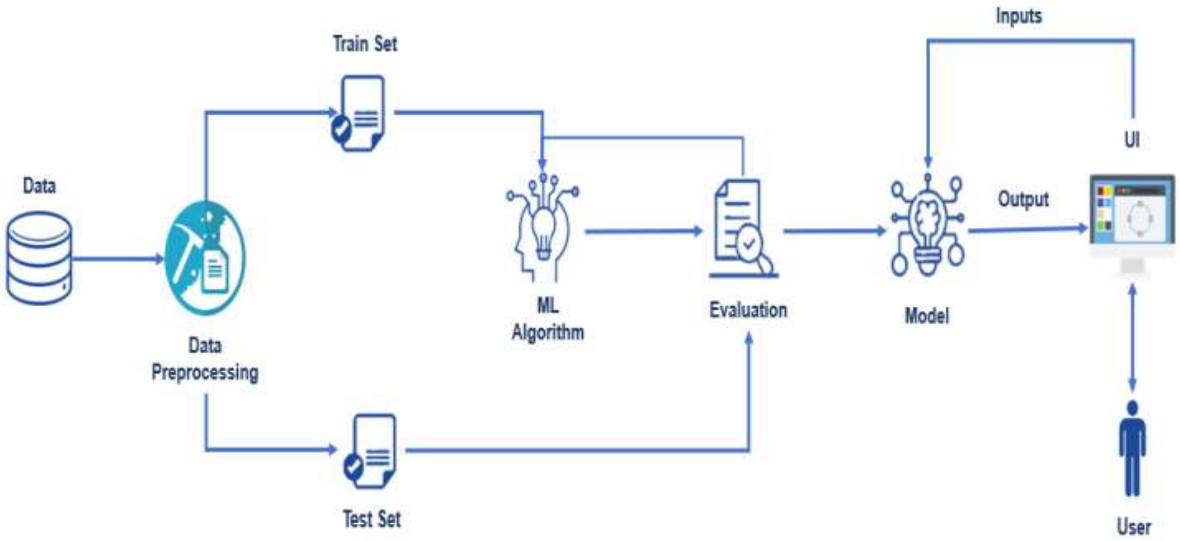
Solution architecture is a way of dividing the processes in the application into smaller sub processes and provides the specification for understanding and defining the solution.



5.2.1 Solution Architecture

Technology Architecture:

Technical Stack or Technical Architecture defines about the components and technologies used along with the characteristics of the application. Technical Architecture depicts the neat flow of the technologies being used.



5.2.2 Technical Architecture/Stack

5.3. User Stories

User stories represent the stories and actions that are actually performed by the user while using the application. It helps developers to gain knowledge on the user's perspective and determine how they use them.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	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	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-3	After login, I can access my dashboard		High	Sprint-1
	User Input	USN-4	In dashboard, I can provide input URL which needs to be predicted	Can Access the website dynamically	High	Sprint-1
	Predict Website	USN-5	The website predicts the category of website using ML model like Logistic Regression, Random Forest Tree Classification etc trained by a certain dataset		High	Sprint-2
	Result	USN-6	As a user, I will be shown the result of the prediction.		High	Sprint-2
Administrator	Update Dataset	USN-7	As a admin of the website, the dataset for the model needs to be updated often.		High	Sprint-3

5.3.1 Table: User Stories

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

Sprint planning is used to create a plan or schedule with an estimated time for completion that can be completed within the time period. The plan created shall segregate task for each team members.

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.	1	Medium	Sathish S, Tharun M V
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	1	Low	Sathish S, Tharun M V
Sprint-2	Prediction	USN-3	As a user, after given the input URL, I will wait until the website predicts the result.	1	High	Vasanthazhagan R A, Shriniwaaz K G
Sprint-3	Train Model on IBM Cloud	USN-4	Task - To make cloud access and prediction of website	2	High	Vasanthazhagan R A, Shriniwaaz K G
Sprint-3	Deploy model on IBM cloud	USN-5	Task - To make cloud access and prediction of website	2	High	Vasanthazhagan R A, Shriniwaaz K G
Sprint-4	Result	USN-6	Task - To make cloud access and prediction of website	1	High	Sathish S, Tharun M V

6.1.1 Table: Sprint Planning & Estimation

6.2. Sprint Delivery Schedule

Sprint Delivery Schedule gives the report for the Sprint Planning & Estimation along with a burn down chart.

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)

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

$AV_1 = 20/6 = 3.34$

$AV_2 = 20/6 = 3.34$

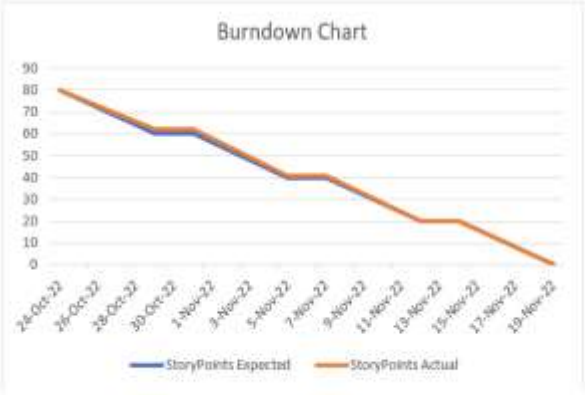
$AV_3 = 20/6 = 3.34$

$AV_4 = 20/6 = 3.34$

6.2.1 Sprint Delivery Schedule

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

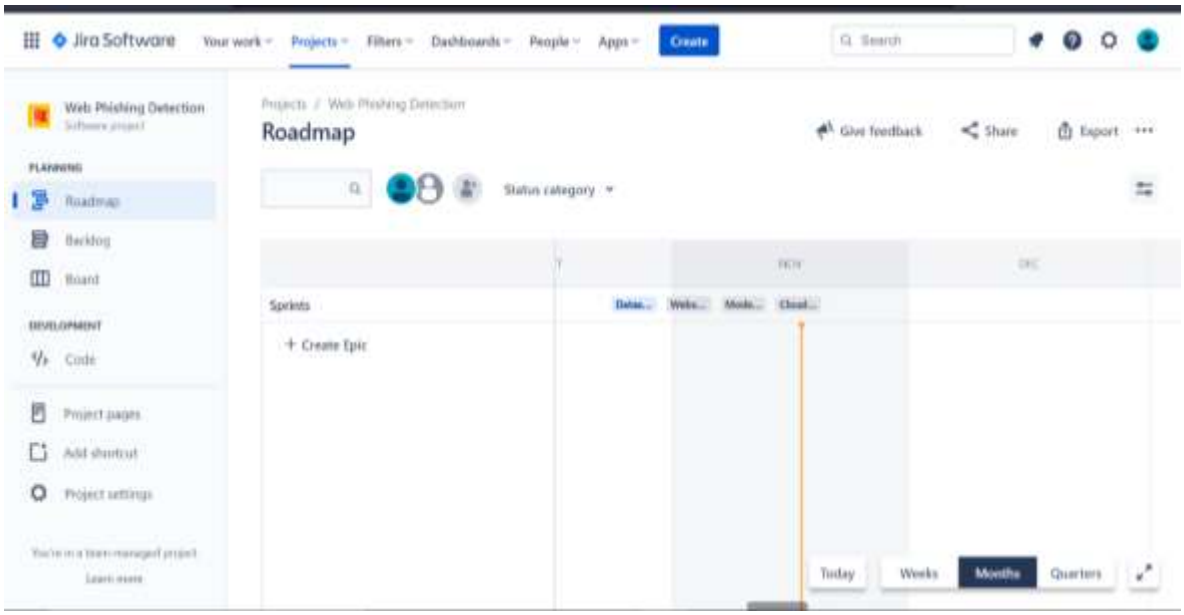


6.2.2 Burndown Chart

6.3.Reports from JIRA

Roadmap:

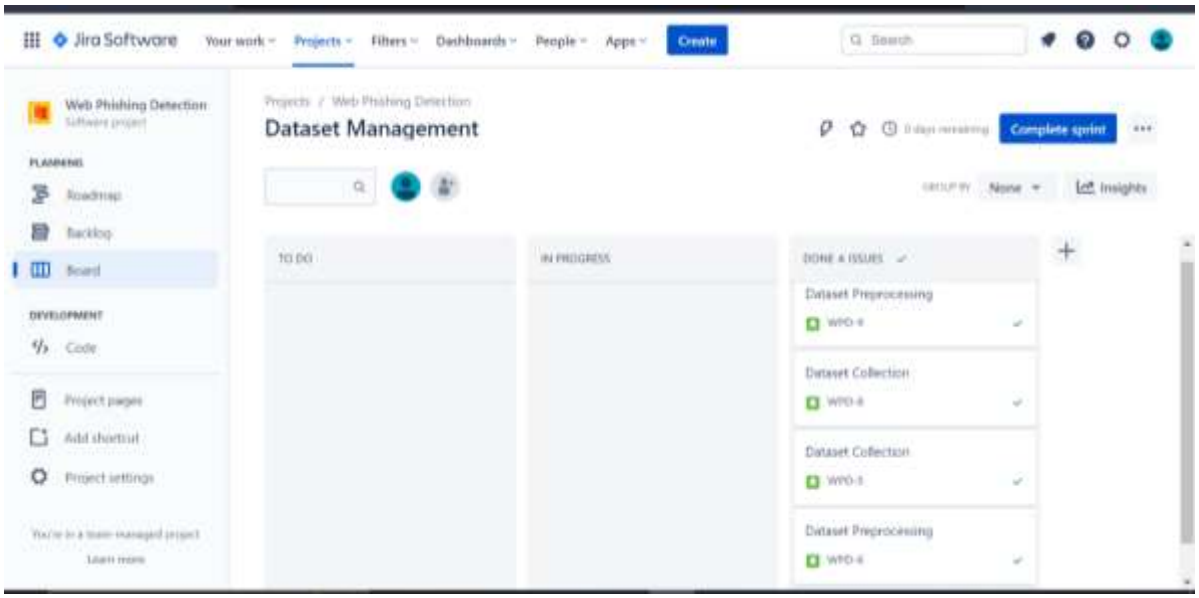
Roadmap depicts the flow of the project as per the duration scheduled.



6.3.1 Roadmap

Board:

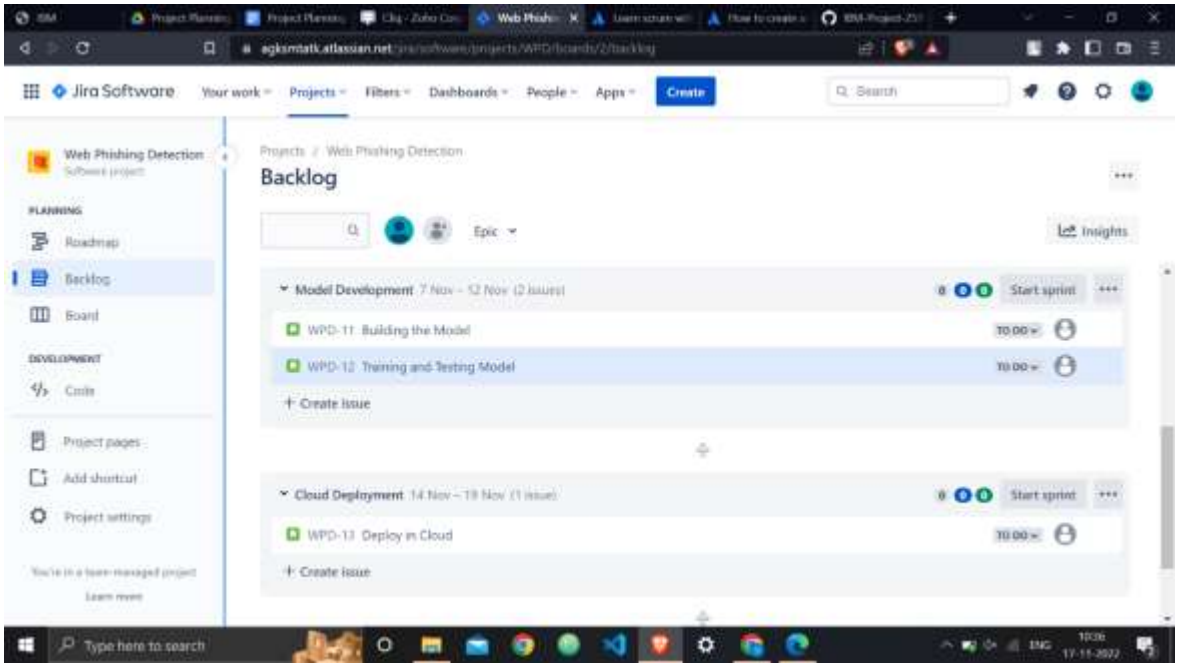
Board contains the detailed view on each sprint of the backlog.



6.3.2 Board

Backlog:

Backlog contains the sprints and issues for each sprint.

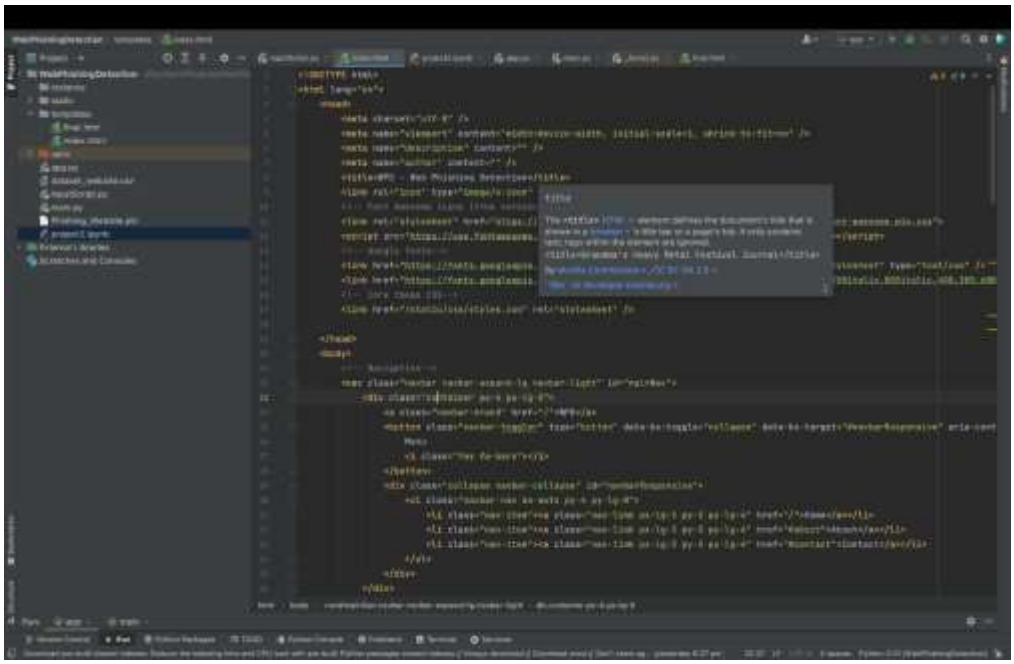


6.3.3 Backlog

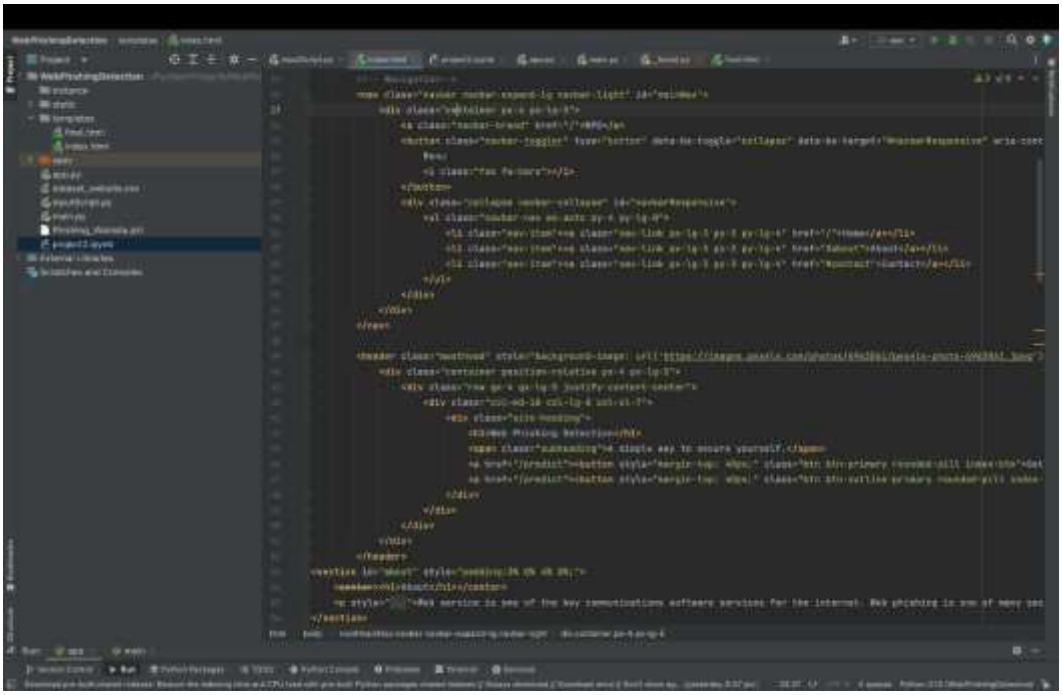
7. CODING & SOLUTIONING

7.1.Home Page

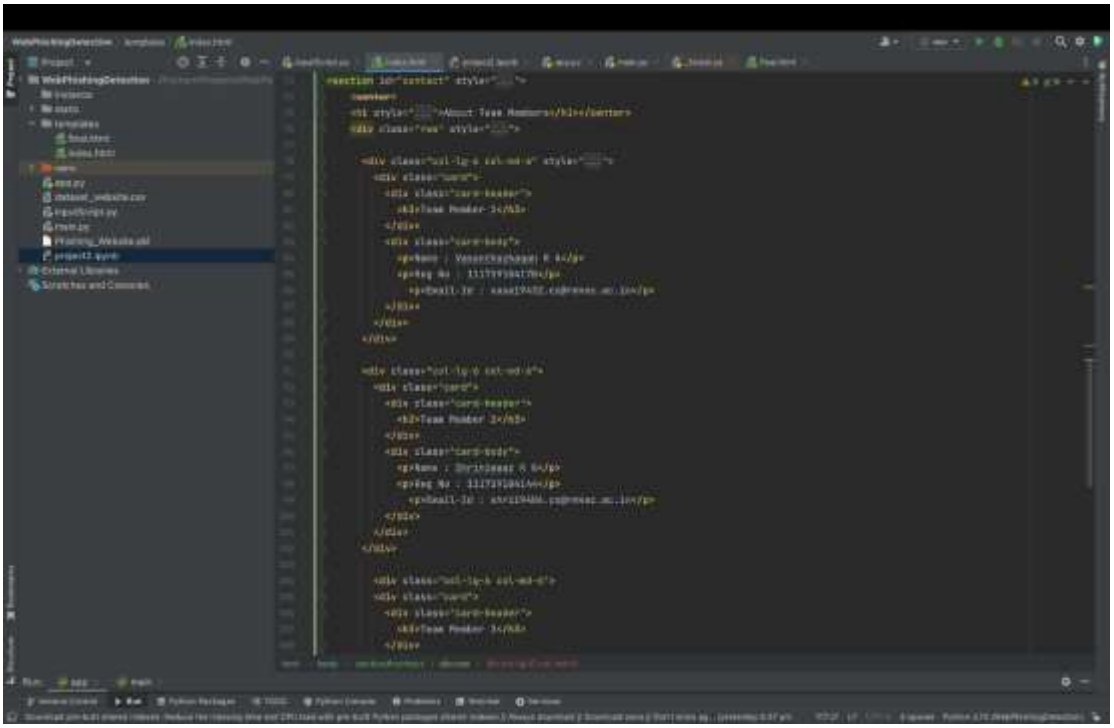
Home page is the starting page where the application takes the user to. The ‘Get Started’ Button takes the user to the next page, i.e. the ‘Prediction Page’.



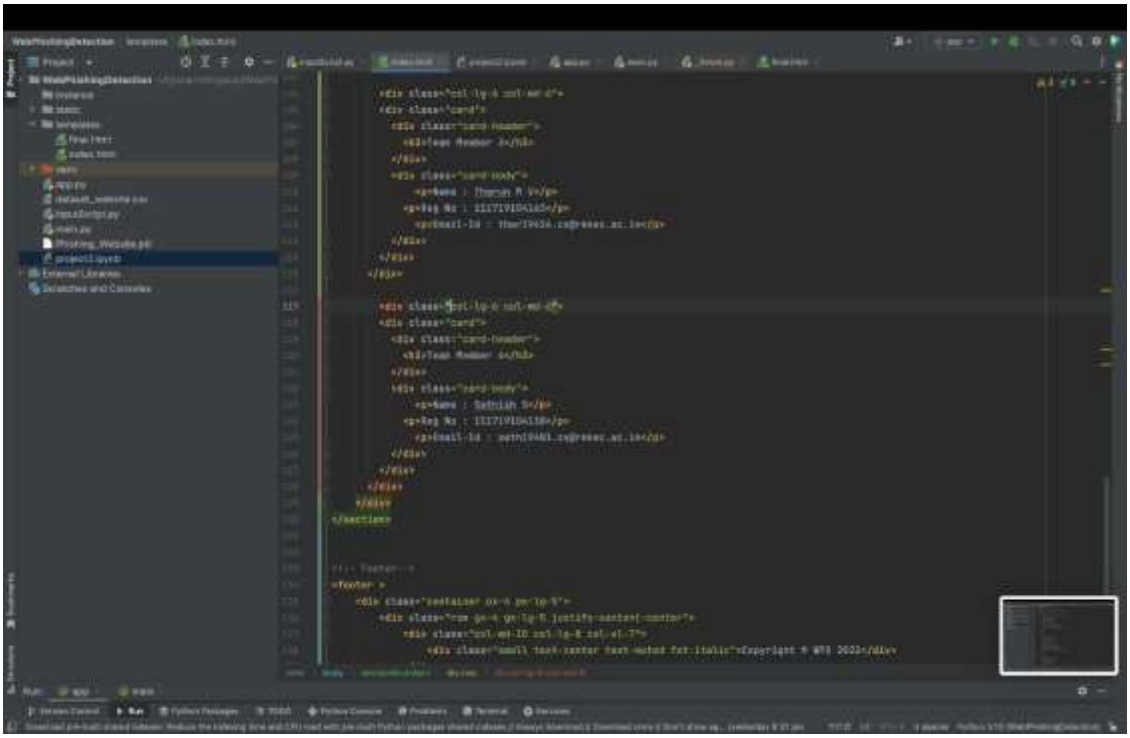
7.1.1 Home Page Code a)



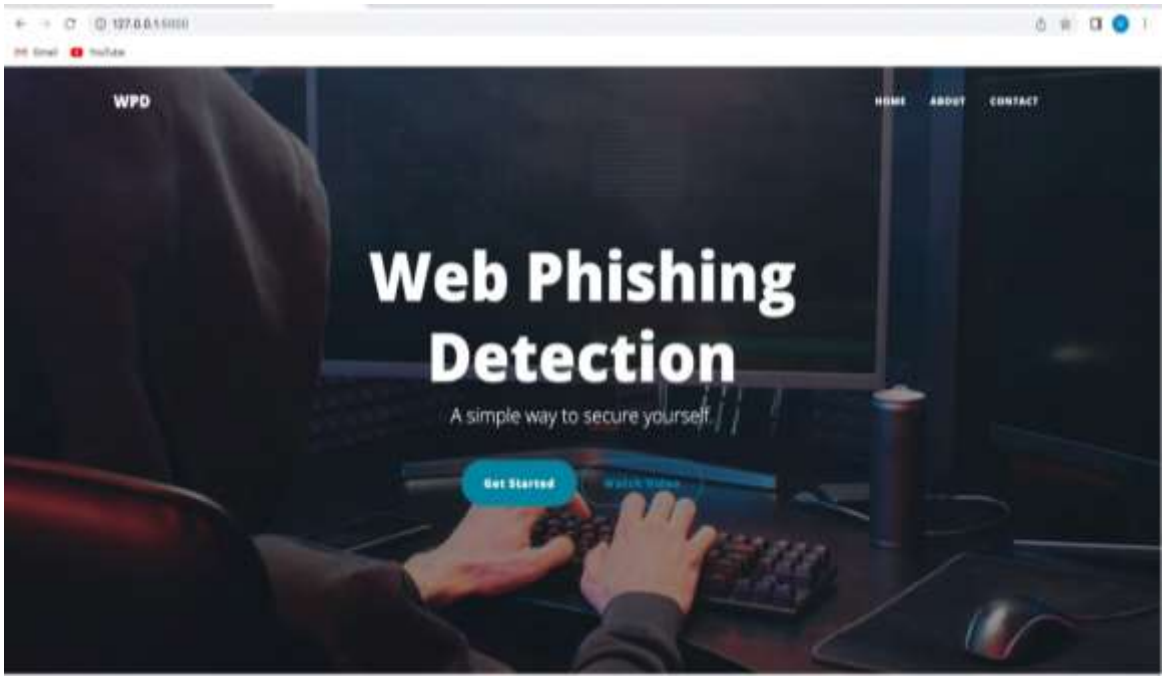
7.1.2 Home Page Code b)



7.1.3 Home Page Code c)



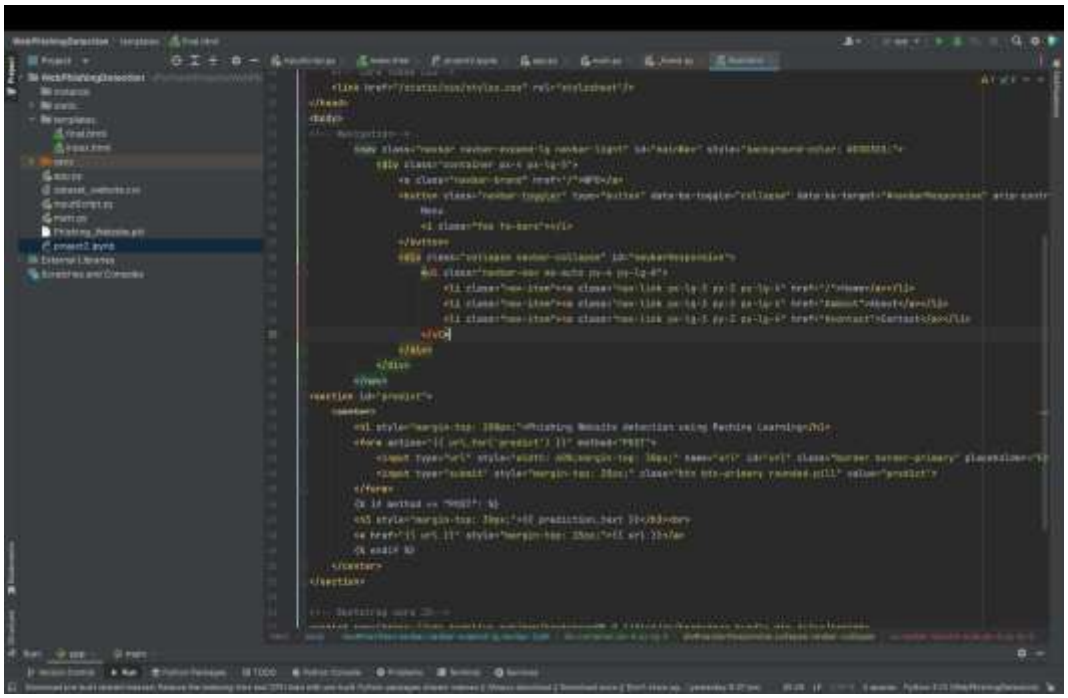
7.1.4 Home Page Code d)



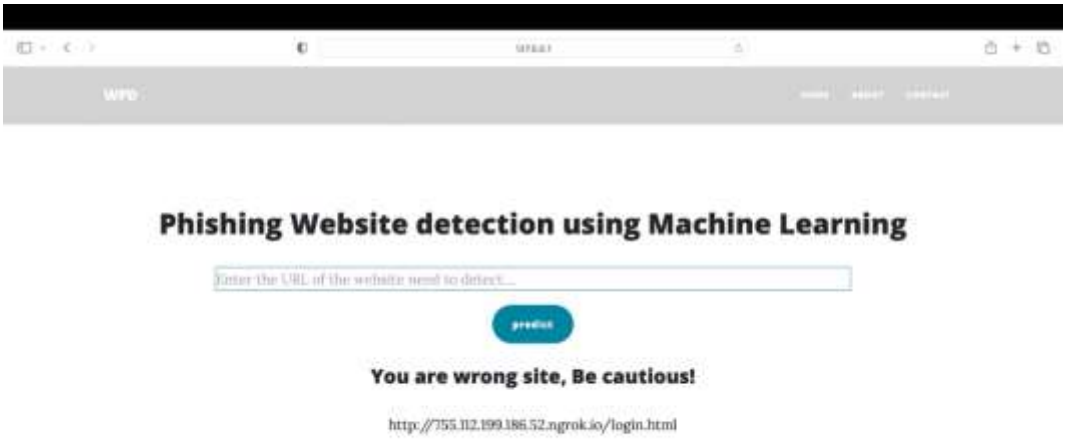
7.1.6 Home Page Output

7.2.Prediction Page

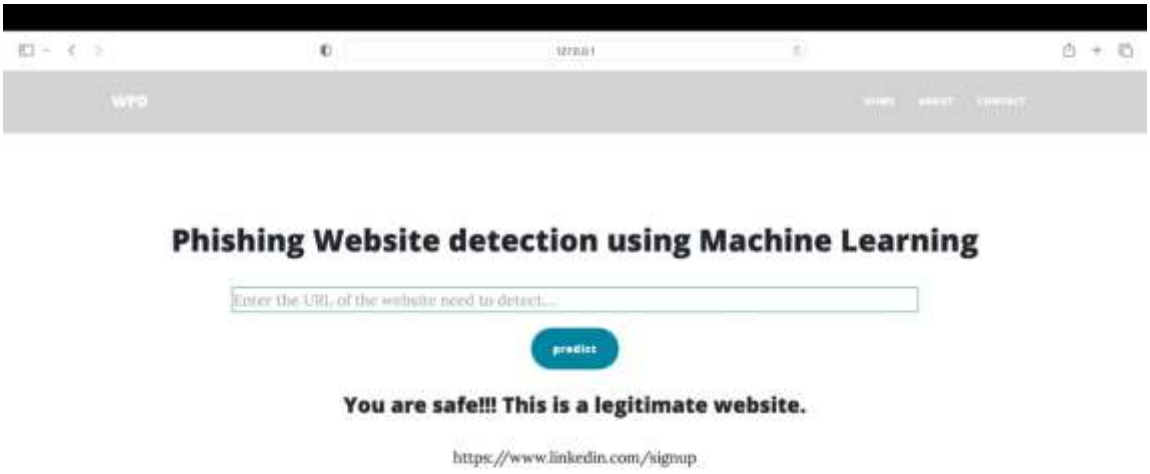
Prediction Page allows the user to input the website's URL and clicking the 'Predict' button tells the user if the page is Legitimate or suspicious one.



7.2.1 Prediction Page Code



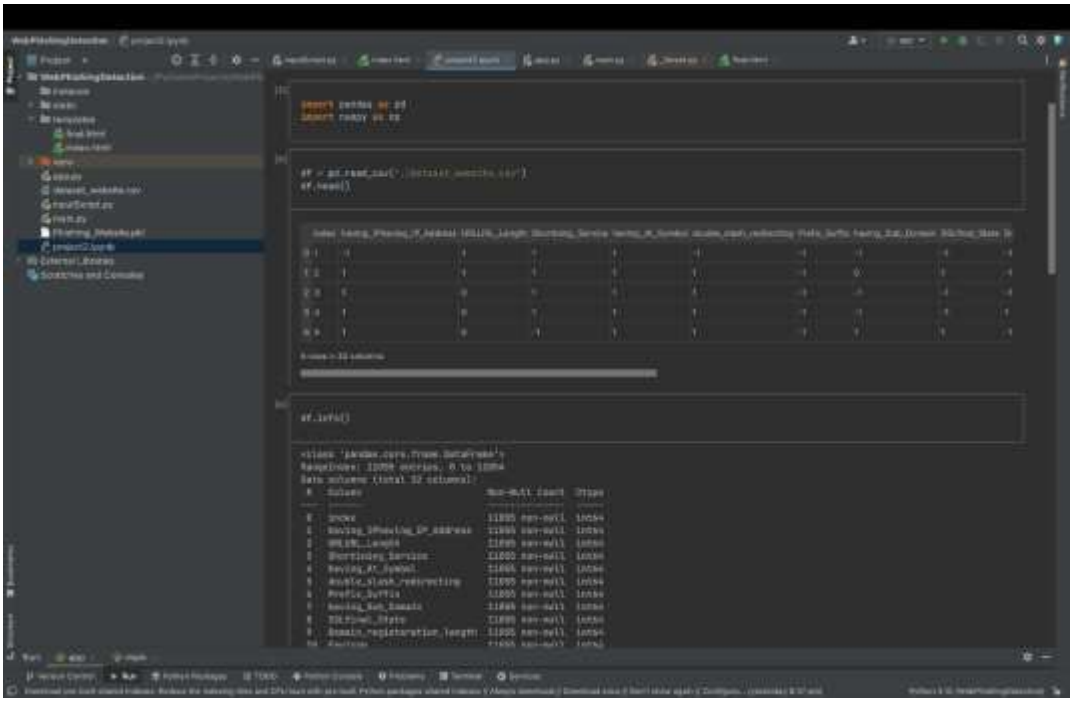
7.2.2 Prediction Page Output a)



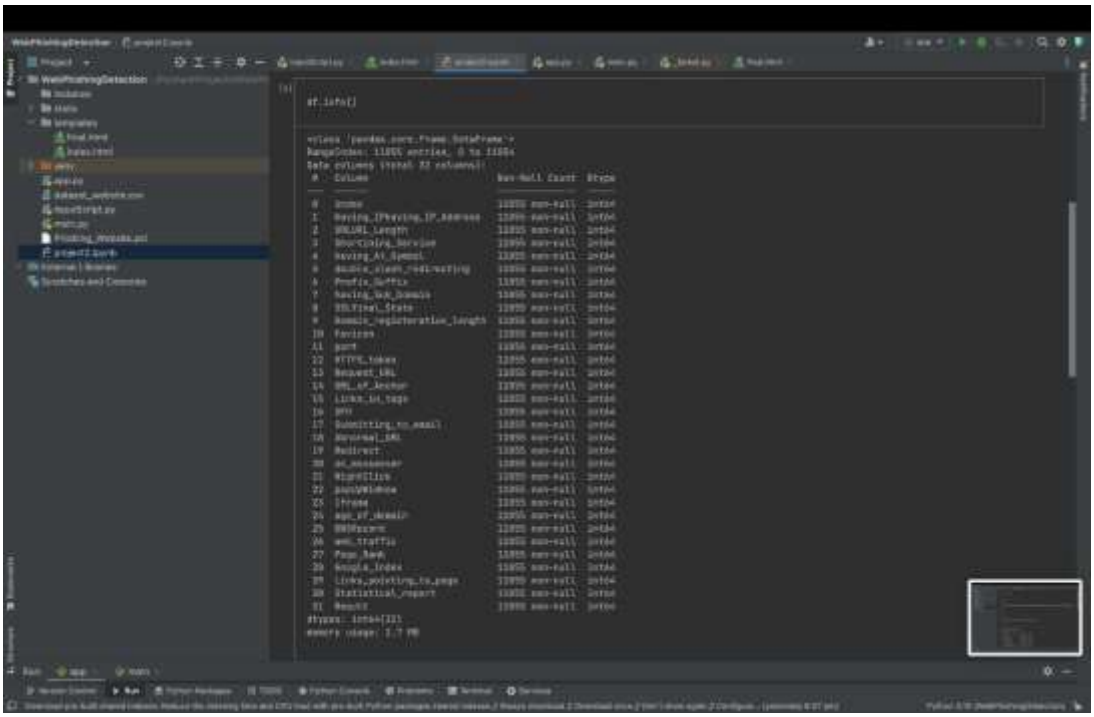
7.2.3 Prediction Page Output b)

7.3.Model Building using Random Forest Classifier

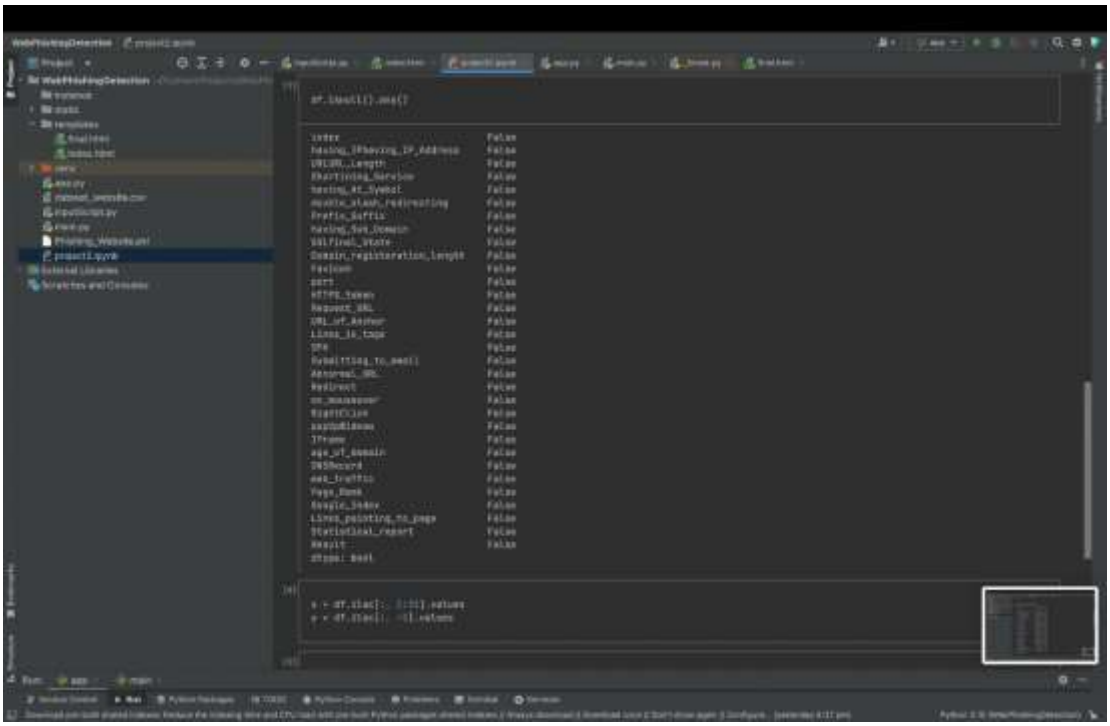
The model is build using Random Forest Classifier.



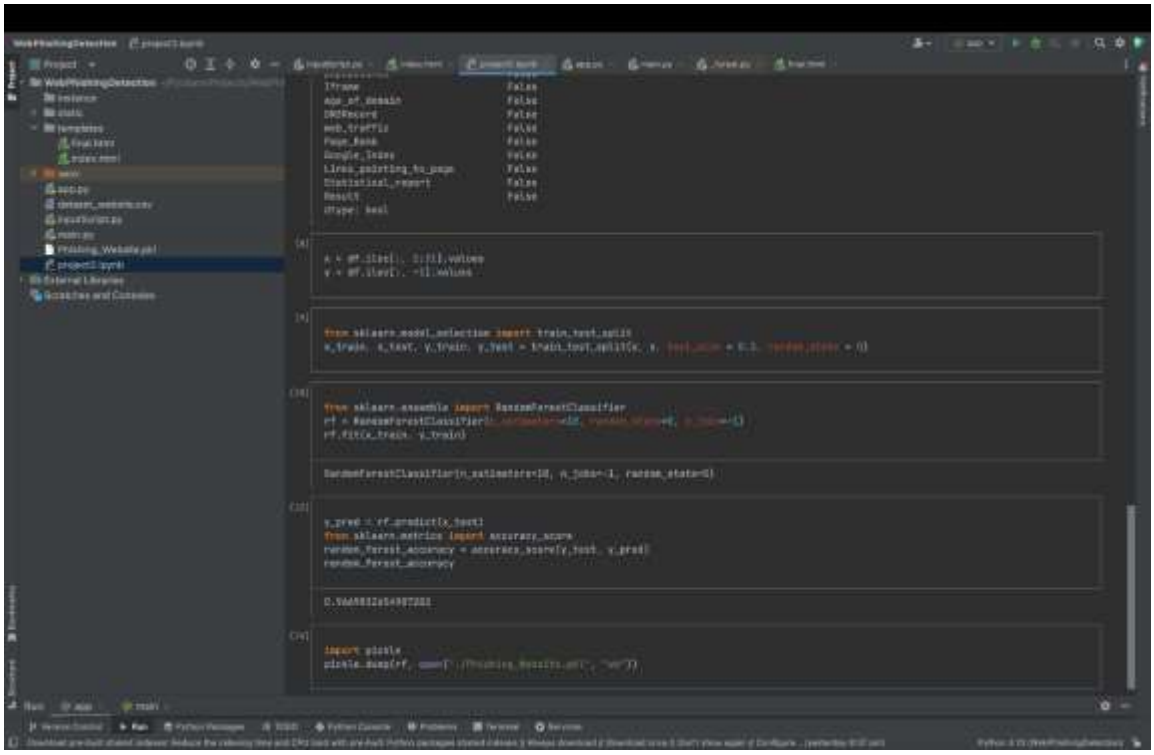
7.3.1 Model Building Code a)



7.3.2 Model Building Code b)



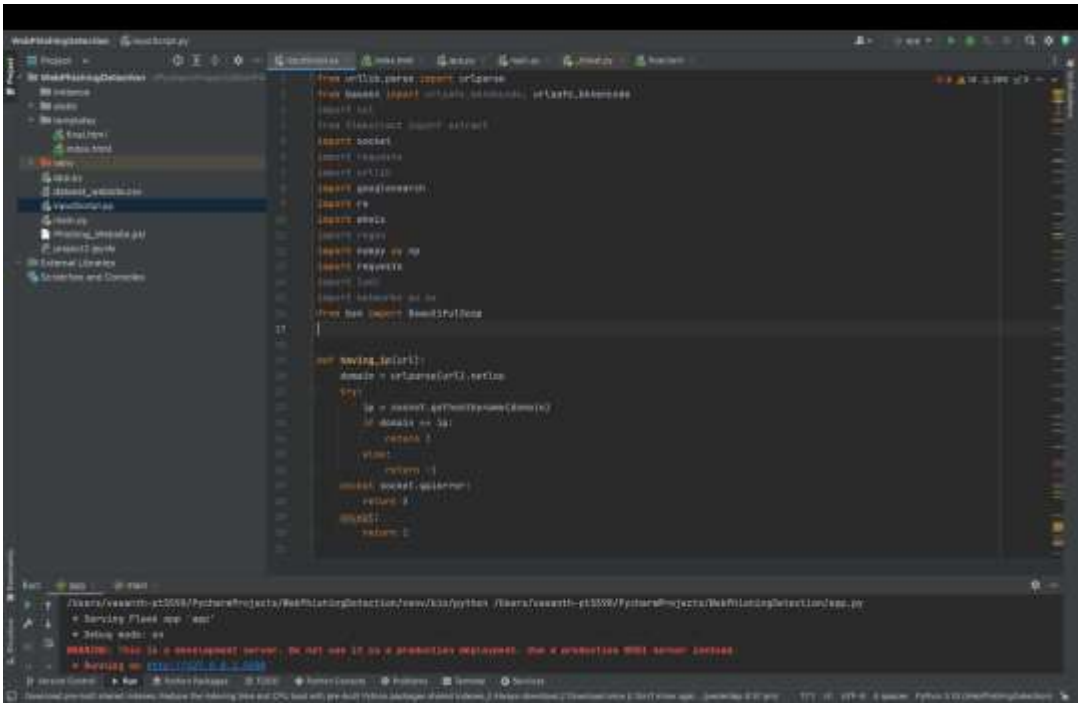
7.3.3 Model Building Code c)



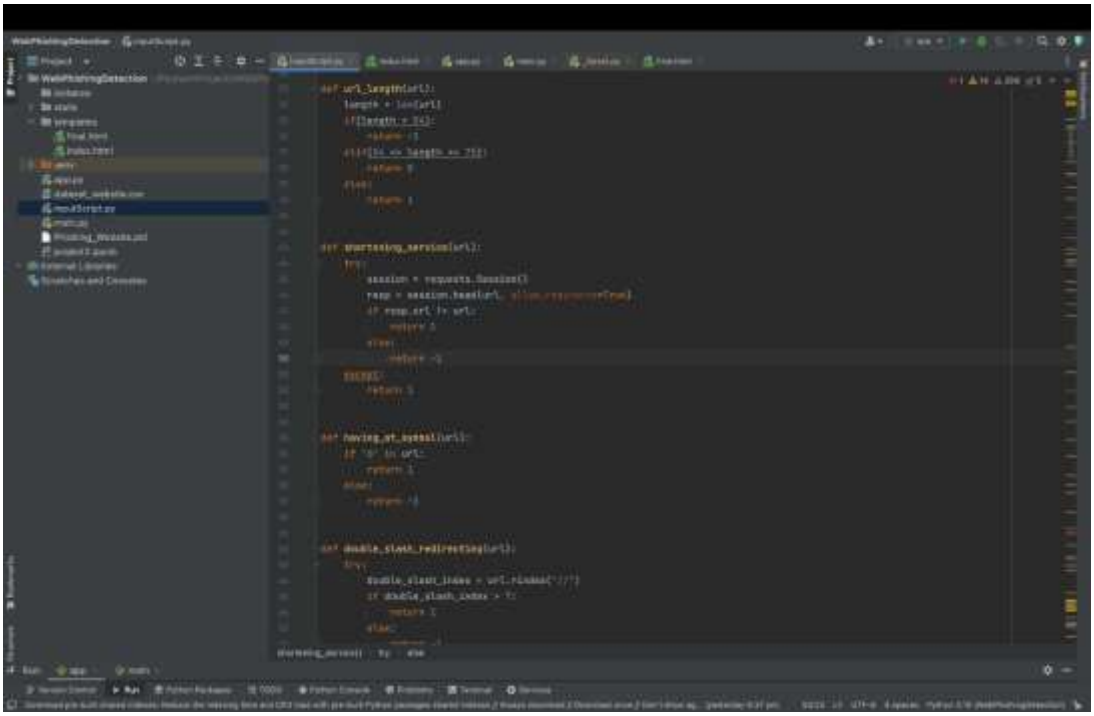
7.3.4 Model Building Code d)

7.4.URL Features checking

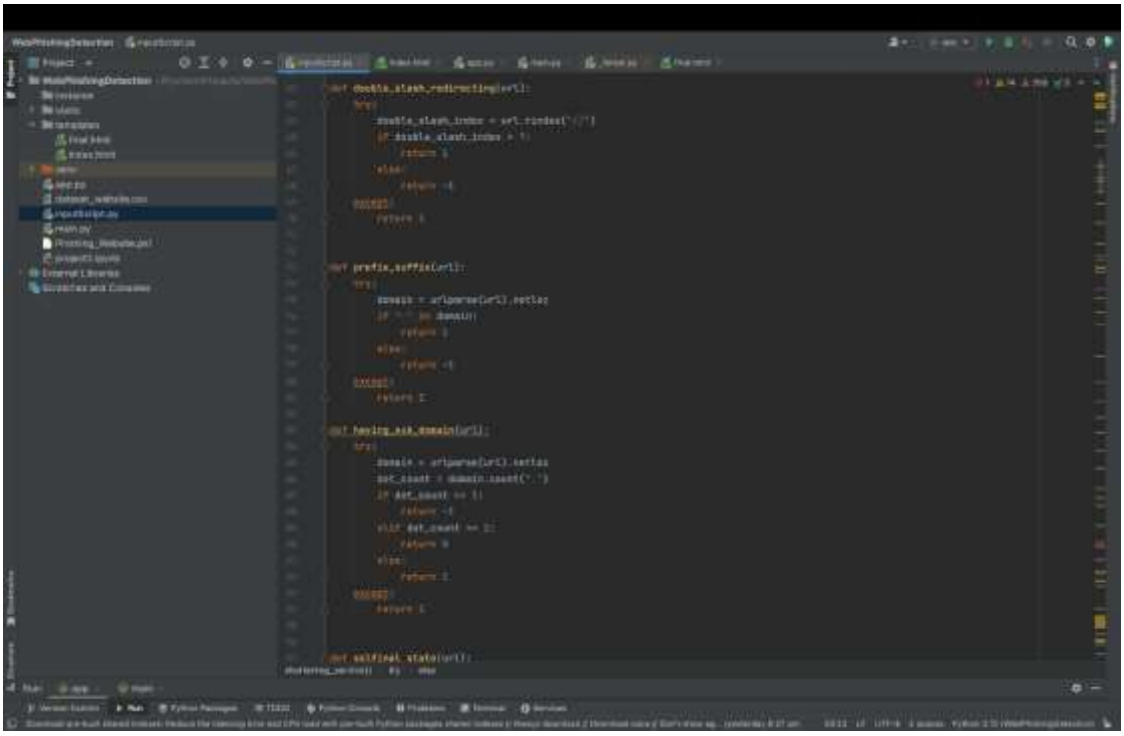
Takes 30 features from the URL and to predict the fraudulence.



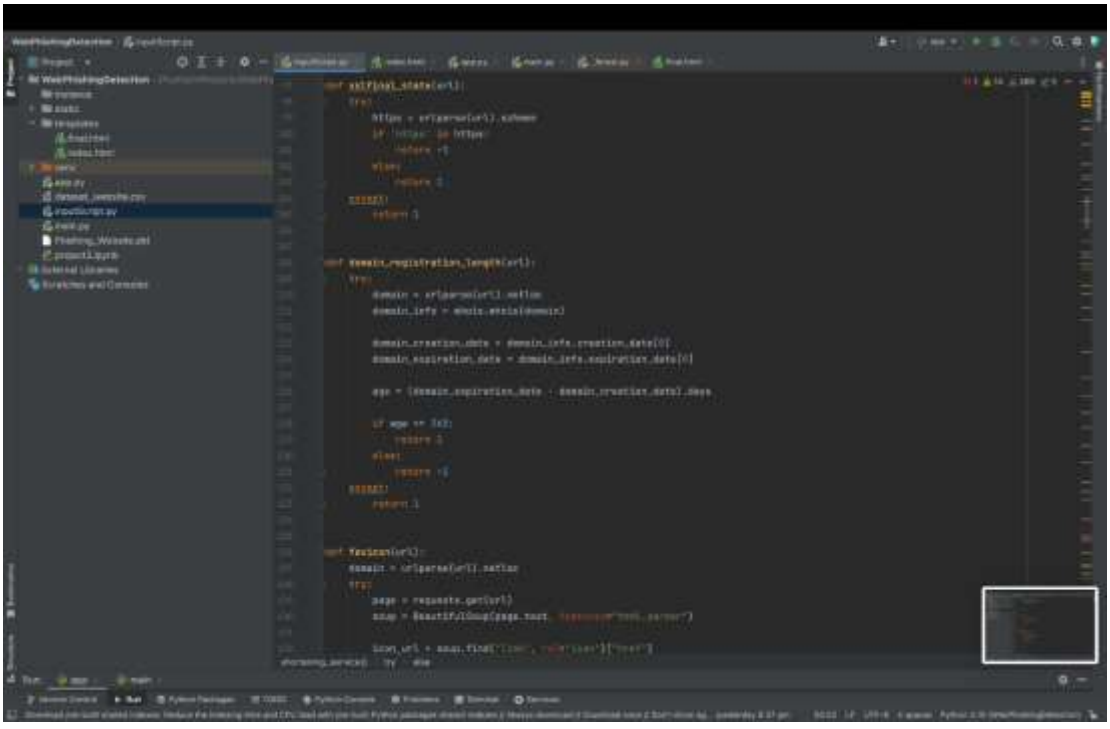
7.4.1 URL Feature Checking Code a)



7.4.2 URL Feature Checking Code b)



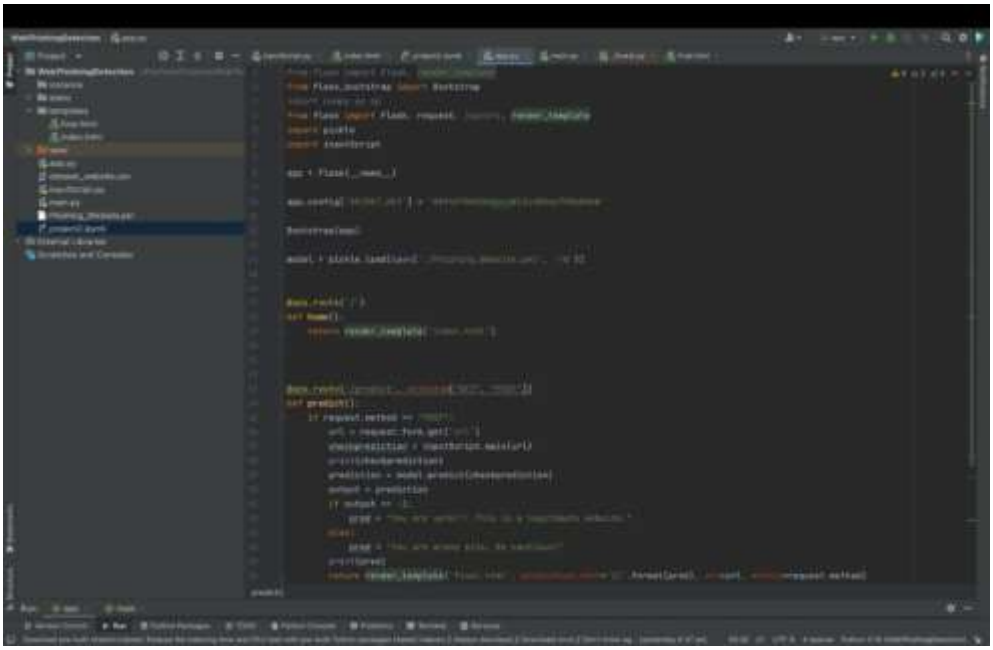
7.4.3 URL Feature Checking Code c)



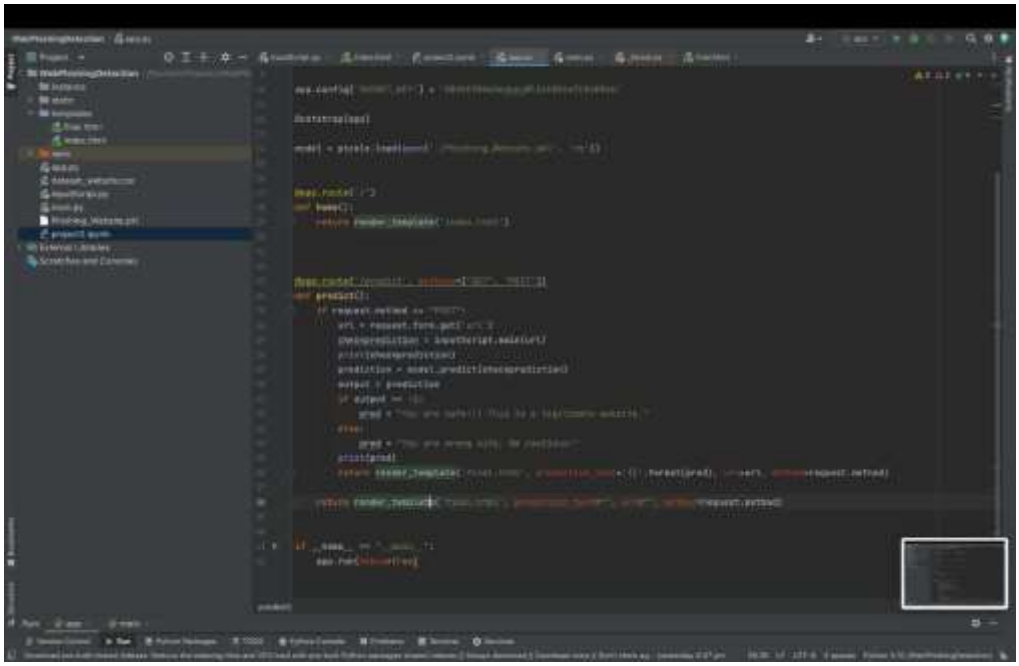
7.4.4 URL Feature Checking Code d)

7.5 Website Rendering

Acts as an interface between the HTML webpage and the model built.



7.5.1 Website Render Code a)



7.5.2 Website Render Code b)

8. TESTING

8.1. Test Cases

Test cases are sample data input that are used to obtain the efficiency of the application.

Date	3-Nov-22
Team ID	PNT2022TMID15890
Project Name	Project - Web Phishing Detection
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commets	TC for Automation(Y/N)	BUG ID	Executed By
Model_Building_TC_001	Functional	RFC Model	Verify if the model gives an accurate result	Model Build	1.Give the URL 2. Compile the RFC Model	Data from Dataset	Accuracy rate over 95%	Working as expected	Pass	nil	Y		Vasanthazhagan R A
Model_Building_TC_002	Functional	RFC Model	Verify if the model gives an accurate result	Model Build	1.Give the URL 2. Compile the RFC Model	Data from Dataset	Accuracy rate over 95%	Working as expected	Pass	nil	Y		Vasanthazhagan R A
Model_Building_TC_003	Functional	RFC Model	Verify if the model gives an accurate result	Model Build	1.Give the URL 2. Compile the RFC Model	Data from Dataset	Accuracy rate over 95%	Working as expected	Pass	nil	Y		Vasanthazhagan R A
Model_Building_TC_004	Functional	RFC Model	Verify if the model gives an accurate result	Model Build	1.Give the URL 2. Compile the RFC Model	Data from Dataset	Accuracy rate over 95%	Working as expected	Pass	nil	Y		Vasanthazhagan R A
Model_Building_TC_005	Functional	RFC Model	Verify if the model gives an accurate result	Model Build	1.Give the URL 2. Compile the RFC Model	Data from Dataset	Accuracy rate over 95%	Working as expected	Pass	nil	Y		Vasanthazhagan R A

8.1.1 Test cases for the build model

Date	3-Nov-22
Team ID	PNT2022TMID15890
Project Name	Project - Web Phishing Detection
Maximum Marks	4 marks

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
Application_Testing_TC_001	Functional	RFC Model, Webpage	Verify if the application gives an accurate result	Model Build, Website	1.Give the URL in the appropriate textblock in the website 2. Send the URL to the model 3. Compile the RFC Model	https://www.linkedin.com/login	Authentic Website	Working as expected	Pass	nil	Y		Shrinivaas K G
Application_Testing_TC_002	Functional	RFC Model, Webpage	Verify if the application gives an accurate result	Model Build, Website	1.Give the URL in the appropriate textblock in the website 2. Send the URL to the model 3. Compile the RFC Model	[illegible]	Suspicious Website	Working as expected	Pass	nil	Y		Shrinivaas K G
Application_Testing_TC_003	Functional	RFC Model, Webpage	Verify if the application gives an accurate result	Model Build, Website	1.Give the URL in the appropriate textblock in the website 2. Send the URL to the model 3. Compile the RFC Model	https://twitter.com/i/flow/login	Authentic Website	Working as expected	Pass	nil	Y		Shrinivaas K G
Application_Testing_TC_004	Functional	RFC Model, Webpage	Verify if the application gives an accurate result	Model Build, Website	1.Give the URL in the appropriate textblock in the website 2. Send the URL to the model 3. Compile the RFC Model	[illegible]	[illegible]	Working as expected	Pass	nil	Y		Shrinivaas K G
Application_Testing_TC_005	Functional	RFC Model, Webpage	Verify if the application gives an accurate result	Model Build, Website	1.Give the URL in the appropriate textblock in the website 2. Send the URL to the model 3. Compile the RFC Model	https://www.instagram.com/accounts/emailsignup/	Authentic Website	Working as expected	Pass	nil	Y		Shrinivaas K G

8.1.2 Test cases for the application

8.2. User Acceptance Testing

1. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	4	2	1	0	20
URL	3	0	3	4	6
External	2	4	0	1	6
Model	8	2	4	2	16
Skipped	0	5	1	0	1
Prediction	4	0	1	1	2
Won't Fix	1	2	2	1	8
Totals	24	14	13	26	77

8.2.1 Table: Defect Analysis

2. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
By Design	7	0	0	7
URL	51	0	0	51
External	2	0	0	2
Model	3	0	0	3
Skipped	9	0	0	9
Prediction	4	0	0	4
Won't Fix	2	0	0	2

8.2.2 Table: Test Case Analysis

9. RESULTS

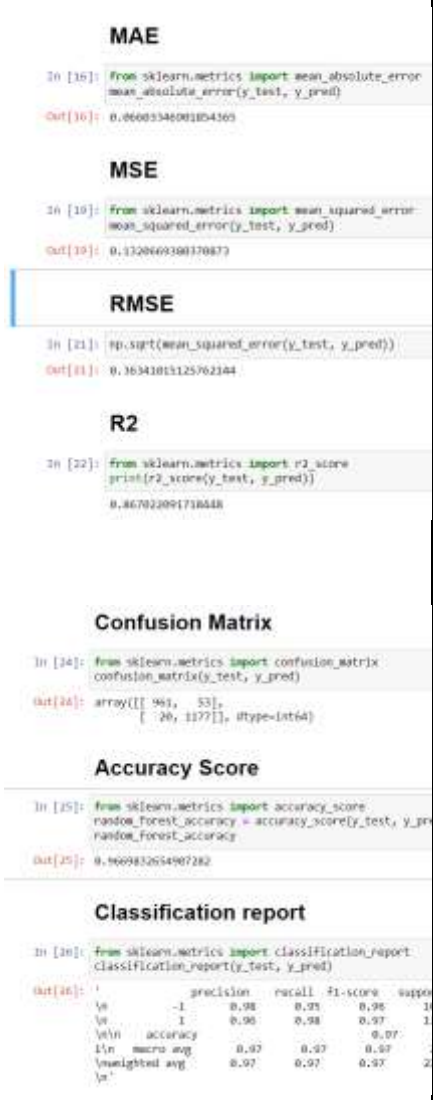
9.1. Performance Metrics

Performance metrics determines the performance of the application by checking the model for its accuracy, confusion matrix, etc.

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Metrics</td><td>
Regression Model:
MAE – 0.066
MSE – 0.132
RMSE – 0.363
R2 score – 0.867

Classification Model:
Confusion Matrix –
array([[961,53],
[20, 1177]],
dtype=int64)

Accuracy Score – 96.69%

Classification Report –
<table>
<tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr>
<tr><td>-1</td><td>0.98</td><td>0.95</td><td>0.96</td><td>1014</td></tr>
<tr><td>1</td><td>0.96</td><td>0.98</td><td>0.97</td><td>1197</td></tr>
<tr><td>Accuracy</td><td>0.97</td><td></td><td></td><td>2211</td></tr>
<tr><td>Macro avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>2211</td></tr>
<tr><td>Nweighted Avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>2211</td></tr>
</table>
</td><td>


</td></tr>
<tr><td>2.</td><td>Tune the Model</td><td>Hyperparameter Tuning - GridSearchCV
Validation Method</td><td></td></tr>
</table>

9.1.1 Table: Performance Testing

10. ADVANTAGES & DISADVANTAGES

Advantages:

- Easy and simple to use
- User-friendly website
- Error accuracy is $3.6x10^{-16}$.

Disadvantages:

- Prediction accuracy is not completely accurate (96.69%).
- Takes more time to process the URL.
- Requires really good internet connectivity.

11. CONCLUSION

The application thus developed uses a webpage as an interface to get the URL of a website as input from the user. It then predicts the fraudulence of the website using the model built and produces the result to the user as output. The model is developed using the Random Forest Classifier. It uses 80 percentage of the dataset as Training set and the remaining 20 percentage as Testing set. The model thus developed, produces the result with an accuracy of 96.69% and an error accuracy of $3.6x10^{-16}$. The application that has been developed has been tested with several websites for fraudulence and the result that is obtained is as the one that was expected.

12. FUTURE SCOPE

- The model currently used is fed with around 11,000 data samples to be built, which is not really sufficient in the current times where there is a lot of cases. So, the model must be trained with more data samples.
- The current system uses a website as an interface to interact with the user. In future, the system can be developed into a Browser Extension.

13. APPENDIX

13.1. Source Code

```
from flask import Flask, render_template
from flask_bootstrap import Bootstrap
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import inputScript

app = Flask(__name__)
Bootstrap(app)

model = pickle.load(open('./Phishing_Website.pkl', 'rb'))

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

@app.route('/predict', methods=["GET", "POST"])
def predict():
    if request.method == "POST":
        url = request.form.get('url')
        checkprediction = inputScript.main(url)
        print(checkprediction)
        prediction = model.predict(checkprediction)
        output = prediction
        if output == -1:
            pred = "You are safe!!! This is a legitimate website."
        else:
            pred = "You are wrong site, Be cautious!"
        print(pred)
        return render_template('final.html', prediction_text='{}'.format(pred), url=url, method=request.method)

    return render_template('final.html', prediction_text="", url="", method=request.method)

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

13.1.1 Source Code

13.2. GitHub Link

https://github.com/IBM-EPBL/IBM-Project-25999-1668681377

13.3. Project Demo Link

https://drive.google.com/file/d/1IpTDXoYWCE8qXtVvHYoCVfsjxrIw5B4A/view?usp=share_link