A MAJOR PROJECT REPORT

ON

# ONLINE PHISHING URL DETECTION USING MACHINE LEARNING TECHNIQUES AND FLASK FRAMEWORK

Submitted in partial fulfillment of the requirements for the award of the degree in

BACHELOR OF TECHNOLOGY

BY


| | |
|---|---|
| **POOJITHA KAVURI** | **(20NN1A1225)** |
| **DEEKSHITHA JONNAKUTI** | **(20NN1A1220)** |
| **SAKALA RAMYASRI** | **(20NN1A1255)** |
| **SHAIK APSANA** | **(20NN1A1256)** |


**Under the esteemed guidance of**

**Mrs. P. Sandhya Krishna**

**Assistant Professor**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VIGNAN'S NIRULA INSTITUTE OF TECHNOLOGY AND SCIENCE FOR WOMEN**

**PEDAPALAKALURU, GUNTUR-522005**

**(Approved by AICTE, NEW DELHI and Affiliated to JNTUK)**


**(2020-2024)**

**VIGNAN'S NIRULA INSTITUTE OF TECHNOLOGY AND SCIENCE FOR WOMEN**

**(Approved by AICTE, NEW DELHI and Affiliated to JNTUK)**

**PEDAPALAKALURU, GUNTUR-522005**

**(2020-2024)**

## DEPARTMENT OF INFORMATION TECHNOLOGY

# CERTIFICATE

This is to certify that the project work entitled **"ONLINE PHISHING URL DETECTION USING MACHINE LEARNING TECHNIQUES AND FLASK FRAMEWORK"** is a bonafide work submitted by **Poojitha Kavuri(20NN1A1225), Deekshitha Jonnakuti(20NN1A1220), Sakala Ramyasri(20NN1A1255), Shaik Apsana(20NN1A1256)** from the department of Information Technology in the partial fulfillment of the requirements forward of degree of Bachelor of Technology in Information Technology from Vignan's Nirula Institute of Technology & Science for Women, Guntur.

**Project Guide**
**Mrs. P. Sandhya Krishna**

**Head of the Department**
**Dr. K. V. S. S. Rama Krishna**

**External Examiner**

# DECLARATION

We hereby declare that the work described in this project work, entitled **"ONLINE PHISHING URL DETECTION USING MACHINE LEARNING TECHNIQUES AND FLASK FRAMEWORK"** which is submitted by us in partial fulfillment for the award of Bachelor of Technology in the department of Information Technology to Vignan's Nirula Institute of Technology & Science for women, affiliated to Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh, is the result of work done by us under the esteemed guidance of Mrs. P. Sandhya Krishna, Assistant Professor. The work is original and has not been submitted for any Degree of this or any other university.

**Poojitha Kavuri** **(20NN1A1225)**

**Deekshitha Jonnakuti** **(20NN1A1220)**

**Sakala Ramyasri** **(20NN1A1255)**

**Shaik Apsana** **(20NN1A1256)**

# ACKNOWLEDGEMENT

We profoundly grateful to express our deep sense of gratitude and respect towards our honorable chairman**, LAVU RATHAIAH** sir, Chairman of Vignan group for his precious support in the college.

We are much thankful to **Dr. P. RADHIKA**, Principal VNITSW, Guntur, for her support during and till the completion of the project.

We would like to thank, **Dr. K. V. S. S. RAMA KRISHNA**, Associate Professor and Head of the Department of Information Technology, for his extended and continuous support, valuable guidance and timely advices in the completion of this project thesis.

We wish to express our profound sense of sincere gratitude to our Project Guide, **Mrs. P. SANDHYA KRISHNA**, Associate Professor of Information Technology, without whose help, guidance and motivation this project thesis could not have been completed the project successfully.

We also thank all the faculty of the Department of Information Technology for their help and guidance of numerous occasions, which has given us the cogency to build-up adamant aspiration over the completion of our project thesis and finally, we thank one and all who directly or indirectly helped us to complete our project thesis successfully.

**POOJITHA KAVURI** **(20NN1A1225)**

**DEEKSHITHA JONNAKUTI** **(20NN1A1220)**

**SAKALA RAMYASRI** **(20NN1A1255)**

**SHAIK APSANA** **(20NN1A1256)**

## ABSTRACT

Phishing websites have proven to be a major security concern. The ways to recognize various network threats, specifically attacks not seen before, is a primary issue that needs to be looked into immediately. The aim of phishing site URLs is to collect the private information like user's identity, passwords and online money related exchanges. Phishers use the sites which are visibly and semantically like those of authentic websites. Since the majority of the clients go online to get to the administrations given by the government and money related organizations, there has been a vital increment in phishing threats and attacks since some years. The machine learning approaches to detect phishing websites have been proposed earlier and have been implemented. The central aim of this project is to implement the system with high efficiency, accuracy and cost effectively. That is been achieved. The project is implemented using 10 machine learning supervised classification models. Some of them are K-Nearest Neighbor, Support vector machine, decision tree and random forest classifier. It was established that the Gradient boost classifier provides best accuracy for the selected dataset and gives an accuracy score of 98%.

# TABLE OF CONTENTS

**CONTENTS** **PAGE NO**

# LIST OF FIGURES

| FIGURE NAME | PAGE NO |
|---|---|

| FIGURE NAME | PAGE NO |
| --- | --- |

# ONLINE PHISHING URL DETECTION USING MACHINE LEARNING TECHNIQUES AND FLASK FRAMEWORK

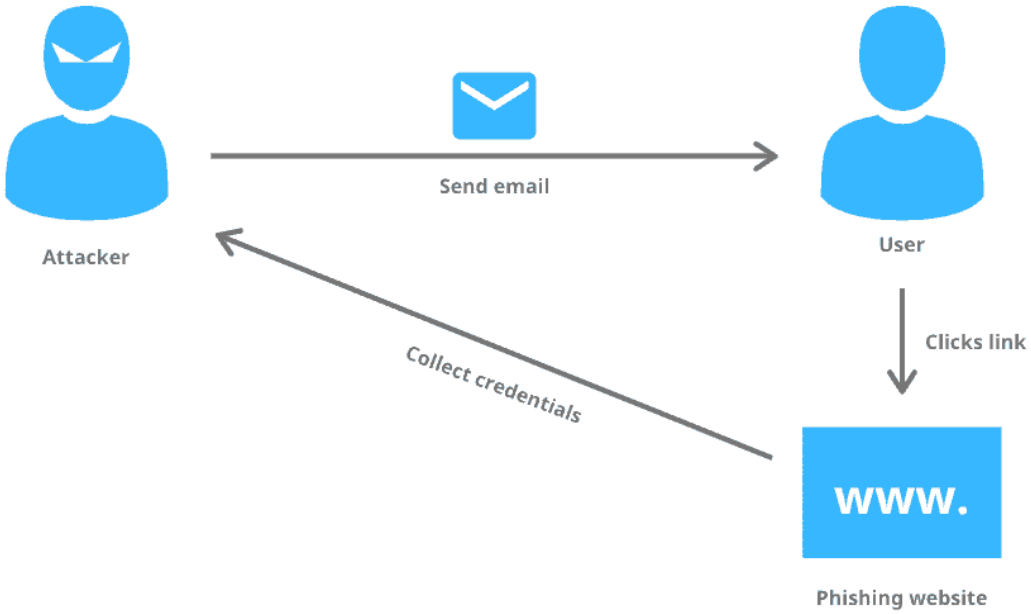
# CHAPTER – 1
# INTRODUCTION

## 1.1 Phishing Overview

Phishing is a fraudulent technique that uses social and technological tricks to steal customer identification and financial credentials. In our daily life, we carry out most of our work on digital platforms. Using a computer and the internet in many areas facilitates our business andprivate life. It allows us to complete our transaction and operations quickly in areas such as trade, health, education, communication, banking, aviation, research, engineering, entertainment, and public services. The users who need to access a local network have been able to easily connect to the Internet anywhere and anytime with the development of mobile and wireless technologies. Although this situation provides great convenience, it has revealed serious deficits in terms of information security. Thus, the need for users in cyberspace to takemeasures against possible cyber-attacks has emerged. These attacks are mainly targeted in thefollowing areas: fraud, forgery, force, shakedown, hacking, service blocking, malware applications, illegal digital contents and social engineering. According to Kaspersky's data, theaverage cost of an attack in 2019 (depending on the size of the attack) is between $ 108K and $ 1.4 billion. In addition, the money spent on global security products and services is around $124 billion. Among these attacks, the most widespread and also critical one is “phishing attacks”. It causes pecuniary loss and intangible damages.[1]

In United States businesses, there is a loss of US$2billion per year because their clients become victim to phishing. In 3rd Microsoft Computing Safer Index Report released in February 2014, it was estimated that the annual worldwide impact of phishing could be as high as $5 billion. Phishing attacks are becoming successful because lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them butit is very important to enhance phishing detection techniques. The general method to detect phishing websites by updating blacklisted URLs, Internet Protocol (IP) to the antivirus databasewhich is also known as “blacklist" method. To evade blacklists attackers uses creative techniques to fool users by modifying the URL to appear legitimate via obfuscation and manyother simple techniques including: fast-flux, in which

proxies are automatically generated to host the web-page; algorithmic generation of new URLs; etc.[2]




1.1 Phishing overview

Heuristic based detection which includes characteristics that are found to exist in phishing attacks in reality and can detect zero-hour phishing attack, but the characteristics are not guaranteed to always exist in such attacks and false positive rate in detection is very high. To overcome the drawbacks of blacklist and heuristic-based method, many security researchers now focused on machine learning techniques. Machine learning technology consists of many algorithms which requires past data to make a present model to detect future results.[3]

In order to receive confidential data, criminals develop unauthorized replicas of a real website and email, typically from a financial institution or other organization dealing with financial data. This e-mail is rendered using a legitimate company's logos and slogans. The design and structure of HTML allow copying of images or an entire website. Also, it is one ofthe factors for the rapid growth of Internet as a communication medium, and enables the misuseof brands, trademarks and other company identifiers that customers rely on as authentication mechanisms. To trap users, Phisher sends "spooled" mails to as

many people as possible. Whenthese e-mails are opened, the customers tend to be diverted from the legitimate entity to a spoofed website.[4]

Malicious Web sites largely promote the growth of Internet criminal activities and constrain the development of Web services. As a result, there has been strong motivation to develop systemic solution to stopping the user from visiting such Web sites.
URLs of the websites are separated into 3 classes:

**Benign:** Safe websites with normal services
**Spam:** Website performs the act of attempting to flood the user with advertising or sites suchas fake surveys and online dating etc.
**Malware:** Website created by attackers to disrupt computer operation, gather sensitive information, or gain access to private computer systems.

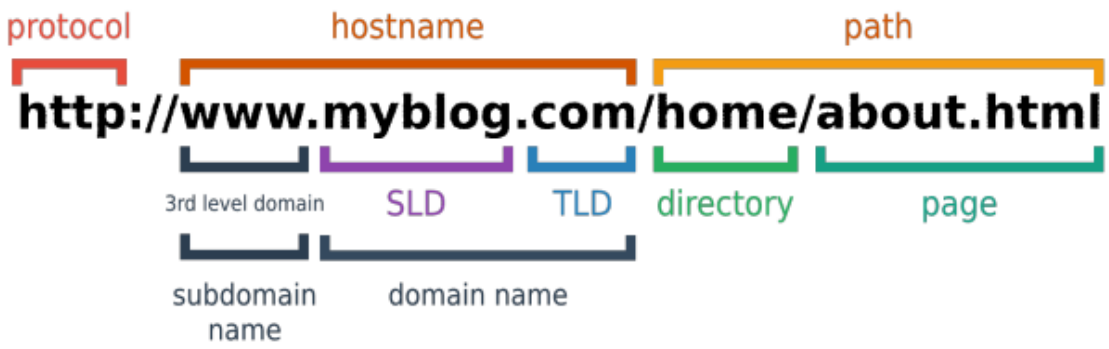
There is a significant chance of exploitation of user information. For these reasons, phishing in modern society is highly urgent, challenging, and overly critical. The method of reaching target users in phishing attacks has continuously increased since the last decade.

This method has been carried out in the 1990s as an algorithm-based, in the early 2000s based on e-mail, then as Domain Spoofing and in recent years via HTTPs. Due to the size of the mass attacked in recent years, the cost and effect of the attacks on the users have been high.

The average financial cost of the data breach as part of the phishing attacks in 2019 is $ 3.86 million,and the approximate cost of the BEC (Business Email Compromise) phrases is estimated to bearound $12 billion. Also, it is known that about 15% of people who are attacked are at least one more target.

With this result, it can be said that phishing attacks will continue to being carried out in the ongoing years.[5]

So, we proposed a system with the help of machine learning techniques and algorithms like Logistic Regression, KNN, SVC, Random Forest, Decision Tree, XGB Classifier and Naïve Bayes to predict Phishing Website based on different parameters like extracted by the website link entered by the user in the front end.

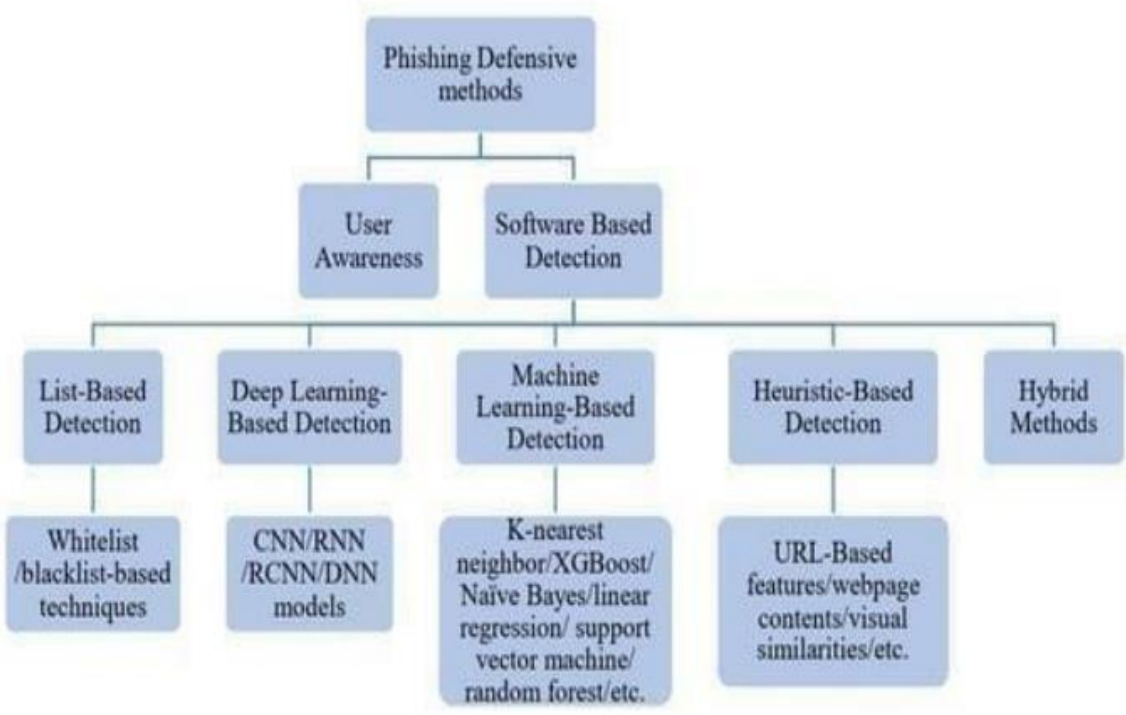



1.2 URL Structure

The objectives of the study are as follows:

1. To develop a novel approach to detect malicious URL.

2. To apply ML techniques in the proposed approach in order to analyze the real time URLs and produce effective results. 3. To implement the concept of Gradient Boosting, which is a familiar ML technique that has the capability to handle huge amount of data.[6]

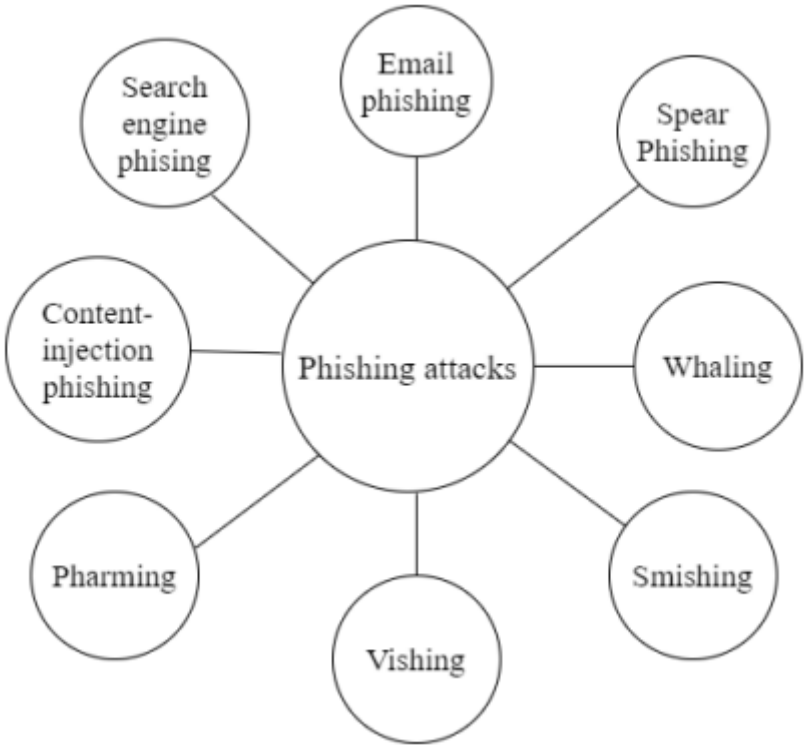



1.3 Phishing Defensive Methods

### 1.2 Types of Phishing Attacks:

There are different types of Phishing attacks. The main goal of these attacks is to obtain sensitive information from the end-users. Figure 1 shows various types of phishing attacks.[7]

- Email Phishing
- Spear Phishing
- Whaling
- Smishing
- Vishing
- Pharming
- Content-injection Phishing
- Search Engine Phishing




1.4 Phishing attacks

**Email Phishing:**

In this type of phishing an attacker sends an email regarding any problem, update or any sensitive matter that must be changed immediately once the user clicks the email and all the input the details entered by the end-users will be redirected to the attacker.

**Spear Phishing:**

In this attackers aims for specific individuals or enterprises, as opposed to random application users. It's a more in-depth version of phishing that requires special knowledge about an organization, including its power structure. In this attack emails are sent to specific persons unlike phishing.

**Whaling:**

It is known as whaling phishing or a whaling phishing attack. It is a form of spear phishing where, in this phishing attackers target high-profile employees, such as the CEO or CFO, in order to steal sensitive information from a company. As these people hold higher positions within the company, they will have complete access to sensitive data. It will be easy to obtain more information.[9]

**Smishing:**

It is also known as SMS phishing. It is a type of social engineering attack carried out in order to steal user data including personal information, financial information, and credentials. Smishing also aims at laundering money from victims. In Smishing, scammers send phishing messages via an SMS text that includes a malicious link. The phishing messages trick recipients into clicking the malicious link, which redirects them to a phishing page where personal information is harvested.[10]

Example : Lucky Draw campaign - In this attackers send SMS to end-users asking them to claim the account that they have won through lucky draw. Attackers ask them to click the link and provide their information so that amount will be transferred to their accounts.

**Vishing:**

It is also known as Voice Phishing. It is a type of phone fraud that uses voice messages to obtain personal information or money from victims. Vishing uses automated voice recordings to lure victims. In Vishing, an automated voice call stating that the recipients'

bank account has been compromised is sent. The voice message then asks the recipient to call a specified toll-free number. Once users call to that toll-free number, the user's bank account number and other personal details are harvested via the phone keypad

**Pharming:**

Pharming is sometimes known as "phishing without a lure". When a user attempts to navigate to a site, their computer can determine the IP address by either consulting a local file of defined mappings—a hosts file—or by consulting a DNS server on the Internet. Pharming is usually conducted either by changing the hosts file on a victim's computer (hosts file pharming) or by exploiting a vulnerability in DNS server software (DNS poisoning).[11]

**Content-injection Phishing:**

In this, the content of the legitimate website is replaced with some random content with different input fields similar to legitimate site so that end –users trust easily and give their data easily.

**Search Engine Phishing:**

It occurs when phishers create websites with attractive sounding offers and have them indexed legitimately with search engines. Users find these sites in the normal course of searching for products or services and are fooled into giving up their information.

### 1.3 Phishing Techniques:

There are different techniques used by attackers to execute different types of phishing attacks. By using these techniques, the attackers can bypass the security and are able to obtain confidential information from the end-users.[12]

1. Link Manipulation
2. Website Forgery
3. Pop ups

**Link Manipulation:**

Link manipulation is a widely used technique for phishing scams. It is done by directing a user through fraud to click a link to a fake website. Hackers are now using manipulative

ways to get the users to click such as:

1.Use of sub-domains

2.Hidden URLs

3.Misspelled URLs

4.IDN(internationalized domain name) Homograph attacks

**Website Forgery:**

Website forgery is another phishing technique that works by making a malicious website impersonate an authentic one, so as to make the visitors give up their sensitive information like account details, passwords, credit card numbers, etc. Web forgery is mainly carried out in two ways:

1. Cross-site scripting

2. Website spoofing.

**Pop ups:**

Pop-up messages are one of the easiest techniques to conduct successful phishing scams. They allow hackers to steal login details by sending users pop-up messages and eventually leading them to forged websites through these pop-ups. A variant of phishing attacks, also known as “in-session phishing,” works by displaying a pop-up window during an online banking session and appears to be a message from the bank.

## 1.4 Problem Statement

1. The Cyber-attacks are growing faster than usual rate, it became evident that necessary steps should be taken in-order to get them under control. Among various cyber-attacks, Phishing websites is one of the popular and commonly used attack to steal users personal information and financial information by manipulating the website URL and IP addresses.
2. The main focus in this project is to implement the better model for detecting these phishing websites using ML algorithms.

### 1.5 Advantages of Phishing URL detection

Phishing URL detection offers several advantages in combating online threats:

**1. Protects Users:** Phishing URL detection helps protect users from falling victim to phishing attacks. By identifying and blocking malicious URLs, it prevents users from accessing fraudulent websites designed to steal sensitive information such as login credentials, financial data, or personal details.[13]

**2. Prevents Data Breaches:** Effective phishing URL detection helps prevent data breaches by stopping users from inadvertently providing sensitive information to malicious actors. This reduces the risk of unauthorized access to personal or corporate data.

**3. Preserves Trust:** By detecting and blocking phishing URLs, organizations can preserve trust with their customers and users. When users feel confident that their information is secure, they are more likely to engage with online services and trust the organization handling their data.

**4. Reduces Financial Losses:** Phishing attacks can result in significant financial losses for individuals and organizations. By detecting phishing URLs and preventing successful attacks, businesses can avoid financial damages such as stolen funds, fraudulent transactions, or regulatory penalties.

**5. Enhances Security Awareness:** Phishing URL detection raises awareness about the prevalence of phishing attacks and educates users about the importance of scrutinizing URLs before clicking on them. This empowers users to recognize suspicious links and take appropriate precautions to protect themselves online.

**6. Maintains Reputation:** Organizations that effectively detect and mitigate phishing attacks safeguard their reputation. A reputation for strong cybersecurity practices can be a competitive advantage, reassuring customers, partners, and stakeholders that their data is safe from malicious threats.

**7. Compliance Requirements:** Many industries have regulatory requirements mandating the protection of sensitive data and the implementation of cybersecurity measures.

Effective phishing URL detection helps organizations meet these compliance standards, avoiding fines and legal consequences associated with data breaches.

**8. Adaptive Protection:** Advanced phishing URL detection systems often employ machine learning and artificial intelligence algorithms to continuously analyze and adapt to evolving phishing techniques. This adaptive protection ensures that detection mechanisms remain effective against new and sophisticated phishing attacks.

**9. Multi-Layered Defense:** Phishing URL detection is typically just one component of a multi-layered cybersecurity strategy. By integrating phishing detection with other security measures such as email filtering, endpoint protection, and employee training, organizations can create a robust defense against various cyber threats.[14]

**10. Global Impact:** Phishing is a global threat affecting users and organizations worldwide. Effective detection and mitigation of phishing URLs contribute to the overall cybersecurity landscape by reducing the success rate of phishing attacks and mitigating their global impact.

# CHAPTER-2
# LITERATURE SURVEY

## 1.6 Literature Survey

Waleed Ali et al.[1]- "Phishing Website Detection Based on Supervised Machine Learning with Wrapper Features Selection" is written by Waleed Ali. It is based on machine learning classifiers with a wrapper features selection method. In this some common supervised machine learning techniques are applied with effective and significant features selected using the wrapper features selection approach to accurately detect phishing websites.

Weiheng Bai et al.,[2]- "Phishing Website Detection Based on Machine Learning Algorithm," is written by Weiheng Bai. It extracts 12 kinds of features, and uses four machine learning algorithms for training. Then,use the best performing algorithm as their model to identify unknown URL's. After the recognition is completed, a snapshot of the web page is extracted and compared with the regularweb page snapshot to implement the recommendation of the original regular web page of the phishing web page.

A. D. Kulkarni, L. L. Brown et al.,[3]- "Phishing websites detection using machine learning," is written by A. D. Kulkarni, L. L. Brown. It has methods of defense utilizing various approaches to categorize websites. Specifically, they have developed a system that uses machine learning techniques to classify websites based on their URL. They used four classifiers: the decision tree, Naïve Bayesian classifier, support vector machine (SVM), and neural network. The classifiers were tested with a data set containing 1,353 real world URLs.

Ciza Thomas et al.,[4]- "Detection of Phishing URLs Using Machine Learning Techniques" is written by Joby James, Sandhya L, Ciza Thomas .This deals with methods for detecting phishing web sites by analyzingvarious features of benign and phishing URLs by Machine learning techniques. they discuss the methods used for detection of phishing websites based on lexical features, host properties and page importance properties. They consider various data mining algorithms for evaluation of the features in order to get a better understanding of the structure of URLs that spread phishing.

Atharva Deshpande et al.,[5]- "Detection of Phishing Websites using Machine Learning" is written by Atharva Deshpande, [2021] -In this study, machine learning-

based detection methods and their associated feature sets are surveyed. The characteristics of phishing domains (also known as fraudulent domains) are described here, along with the qualities that set them apart from valid domains, thesignificance of detecting these domains, and how ML and NLP methodologies are used to do so. For the purpose of identifying fraudulent web pages, Random Forest was chosen based on efficiency and incorporated into a Chrome extension. Using the Random Forest algorithm, phishing websites may be identified with great accuracy. Due to the fact that it does not offer online learning, this project cannot be scaled for web services. Receiving a phishing attempt ishighly likely.

Junaid Rashid et al.,[6]- Phishing is a sort of internet attack that illegally accesses data from the original website, including login credentials, passwords, and credit card information. An effective phishing detection method based on ML was suggested in this research. SVM model has an extremely low false-positive rate and offers great accuracy. The suggested method can stop the harm done by phishing assaults and find new temporary phishing sites. As this projectdoes not employ the feature selection technique, it delivers less accuracy when identifying realand phishing websites.

Mahajan Mayuri Vilas.,[7]- “Detection of Phishing Website Using machine Learning Approach” is written by Mahajan Mayuri Vilas [2019]- In this project, phishing websites are created using an extreme learning machine. When a person visits a website, its URL is used to extract its features. The outcome of the feature extraction will serve as the test data. This method's goal is to identify fraudulent or unlawful websites and alert users in advance, protecting consumers from having their personal information exploited. This method is designed to identify phoney or unlawfulwebsites and alert users in advance so that their personal information won't be exploited. This experiment is operating really accurately. This project's weakness is that, as the number of training sets is increased, bias may result.

Tsehay Admassu Assegie et al.,[8]- “K-Nearest Neighbor Based URL Identification Model for Phishing Attack Detection” is written by Tsehay Admassu Assegie [2021]- The proposed model detects phishing attack through URL classification. The performance of the proposed model is tested empirically and result is analyzed. Experimental result on test set reveals that the model is efficient on phishing attack detection. The advantage of the project is that the KNN algorithm is used in order to detect the phishing websites. And it gives an accuracy of 85%. The main disadvantage

is that KNN algorithm leads to overfitting as the training data is too small. And the accuracy is low in detecting phishing websites. “Phishing Websites detection using Machine Learning algorithms” is written by Rishikesh Mahajan, Irfan Siddavatam [2018]- This paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate.

Amani Alswailem et al.,[9]- An intelligent system was proposed for detecting phishing websites. The system acts as an additional functionality to an internet browser as an extension that automatically notifies the user when it detects a phishing website. The system is based on a machine learning method, particularly supervised learning. they have selected the Random Forest technique due to its good performance in classification. The advantage is that Random forest technique is used in order to develop the system which gives an high accuracy. In this paper they have used random features to process its combination because of that time computation become high and the performance gets decreased.

# CHAPTER – 3
# SYSTEM ANALYSIS AND PROPOSED MODEL

## 3.1 Existing System

Many Researches are done for phishing websites detection. Previously it was done using K-Nearest Neighbours (KNN) for different random states of train-test split function. KNN classifier is a non-parametric classification algorithm. Using KNN helps to classify the type of URL as phishing or legitimate that can help navie user into not revealing sensitive information like username, passwords and credit card numbers but the major disadvantage of using this classifier is that the accuracy falls with increase in the size of the training set and theremight be a chance for inaccurate predictions. And the KNN algorithm leads to overfitting if thetraining data is too small. As KNN depends more on training data, a small change in the data tends to cause a big difference, which causes instability and leads to high false-negative values.

### 3.1.1 KNN Algorithm

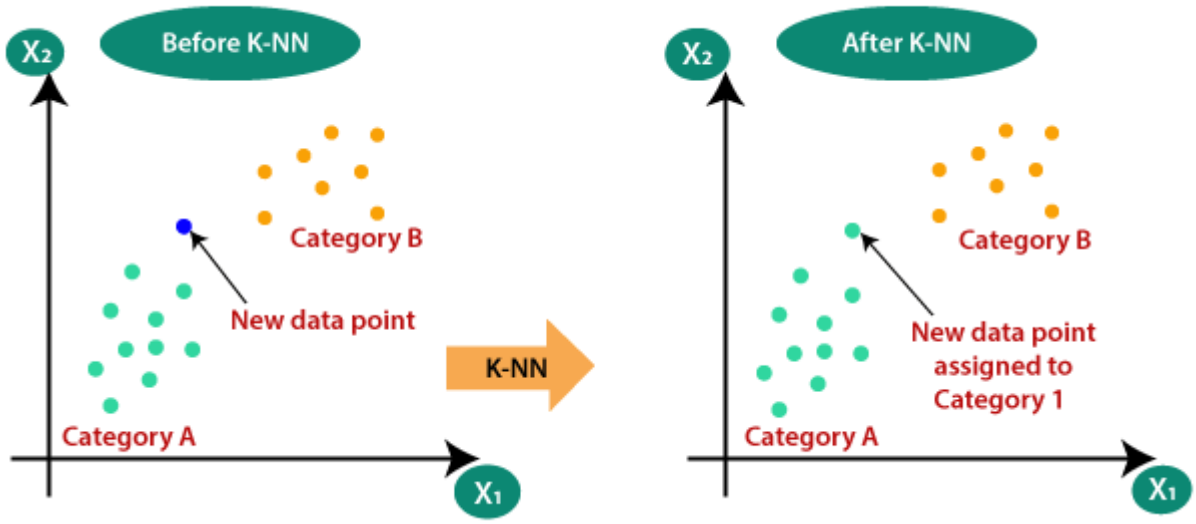
KNN is a supervised learning algorithm that can be used for both classification and regression problems. The main idea behind KNN is to find the k-nearest data points to a given test data point and use these nearest neighbours to make a prediction. The value of k is a hyperparameter that needs to be tuned, and it represents the number of neighbours to consider.

For classification problems, the KNN algorithm assigns the test data point to the class that appears most frequently among the k-nearest neighbours. In other words, the class with the highest number of neighbours is the predicted class.

For regression problems, the KNN algorithm assigns the test data point the average of the k-nearest neighbors' values.

The distance metric used to measure the similarity between two data points is an essential factor that affects the KNN algorithm's performance. The most commonly used distance metrics are Euclidean distance, Manhattan distance, and Minkowski distance.

K-Nearest Neighbors is one of the simplest supervised machine learning algorithms used for classification. It classifies a data point based on its neighbors' classifications. It stores all available cases and classifies new cases based on similar features.




3.1 KNN algorithm

### 3.1.2 Drawbacks of Existing System

- The accuracy falls with increase in the size of the training set when using KNNclassifier.
- KNN algorithm leads to overfitting if the training data is too small.
- The accuracy is also low in detecting the type of URL.
- As KNN depends more on training data, a small change in the data can leads toinstability and leads to high false-positive and false-negative values.

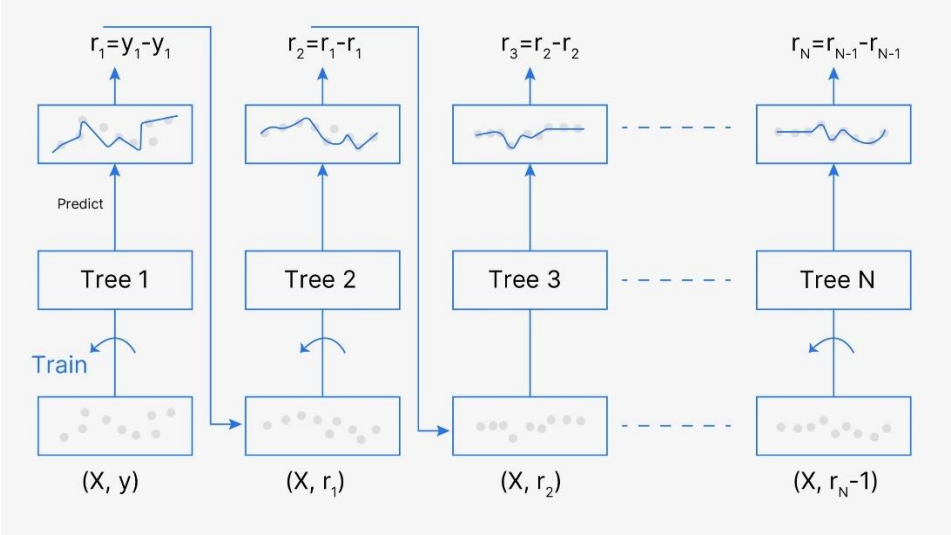
## 3.2 Proposed System

One of the most successful methods for detecting phishing websites is Machine Learning. This is because most phishing attacks have some common characteristics which can beidentified by machine learning. In this project, we are going to introduce a phishing detection system using Gradient Boosting algorithm which is a popular machine learning method, that belongs to supervised learning which can able to classify whether a website is legitimate or phishing withhigh accuracy and efficiency. In this work, we have collected a dataset from Kaggle which contains the URLsof both phishing

and legitimate websites along with 87 various extracted features. We have used correlation-based feature selection method in order to select the most appropriate featuresthat can be used to detect the type of website. Later we have developed Gradient Boosting model by training the required data. It is proven that our proposed model is very good at detection rate and with very low false-positive and false-negative values.

### 3.2.1 Gradient Boosting Algorithm

Gradient Boosting Classifier is a machine learning algorithm that belongs to the class of ensemble methods. It is used for classification tasks and works by combining multiple weak classifiers into a strong classifier. This algorithm works by iteratively adding new decision trees to the model, where each new tree tries to correct the errors of the previous tree. The algorithm starts by training an initial weak classifier on the data. Then, it calculates the residuals (difference between the predicted and actual values) of the weak classifier and fits a new weak classifier on the residuals. The process is repeated for a specified number of iterations or until the residuals are small enough. It aims to improve overall predictive performance by optimizing the model's weights based on the errors of previous iterations, gradually reducing prediction errors and enhancing the model's accuracy.

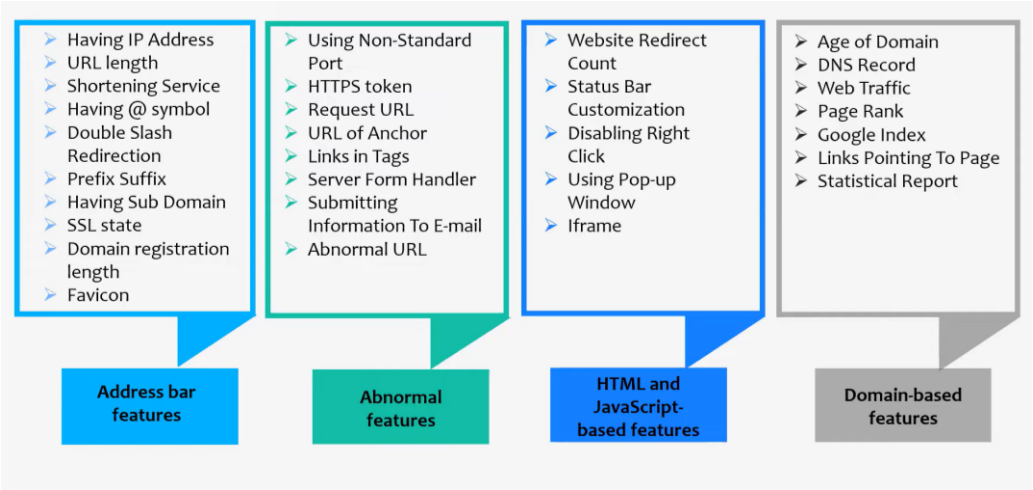



3.2 Gradient Boosting Algorithm

The main idea behind this algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model. But how do we do that? How do we reduce the error? This is done by building a new model on the errors or residuals of the previous model. When the target column is continuous, we use **Gradient Boosting Regressor** whereas when it is a classification problem, we

use **Gradient Boosting Classifier**. The only difference between the two is the *"Loss function"*. The objective here is to minimize this loss function by adding weak learners using gradient descent. Since it is based on loss function hence for regression problems, we'll have different loss functions like Mean squared error (**MSE**) and for classification, we will have different for e.g **log-likelihood**.

### 3.2.2 Dataset Explanation




3.3 Gradient Boosting Algorithm

A phishing URL detection dataset that consists of URL domains as 87 different features likely contains URLs where each URL is represented by its domain and additional features extracted from that domain. Here's a breakdown of what each part of such a dataset could entail:

1. **URL Domains:**

   Each URL in the dataset is parsed to extract its domain, which represents the main address of the website. For example, in the URL "https://www.example.com/page", the domain would be "example.com". These domains serve as the primary feature for analysis in this dataset.

2. **Additional Features:**

   Alongside each domain, there are 87 other features extracted to aid in phishing detection. These features could include various characteristics of the domain, such as:

- **Length of the domain:**

  Phishing domains may sometimes have longer or shorter names compared

to legitimate ones.

- **Presence of special characters:**
  Phishing domains might use special characters or unusual patterns to mimic legitimate domains.
- **Domain age:**
  Phishing domains may be newly registered, while legitimate domains tend to have a longer history.
- **Use of subdomains:**
  Phishing URLs may use additional subdomains to deceive users.
- **Lexical features:**
  Presence of specific keywords or patterns in the domain name that are associated with phishing.
- **Domain reputation scores:**
  Scores or ratings indicating the reputation of the domain based on factors such as previous phishing reports, blacklists, or security analyses.
- **SSL certificate status:**
  Presence or absence of SSL certificates, which can indicate whether the website uses encryption (a common feature of legitimate websites).
- **IP reputation:**
  Reputation of the IP address associated with the domain, as some phishing websites may share IPs with known malicious sites.
- **WHOIS information:**
  Details about the domain owner, registration date, and other WHOIS data, which can be used to assess the legitimacy of the domain.

**3. Label:**

Each domain in the dataset is labeled as either legitimate or phishing, indicating whether it belongs to a genuine website or a malicious one.

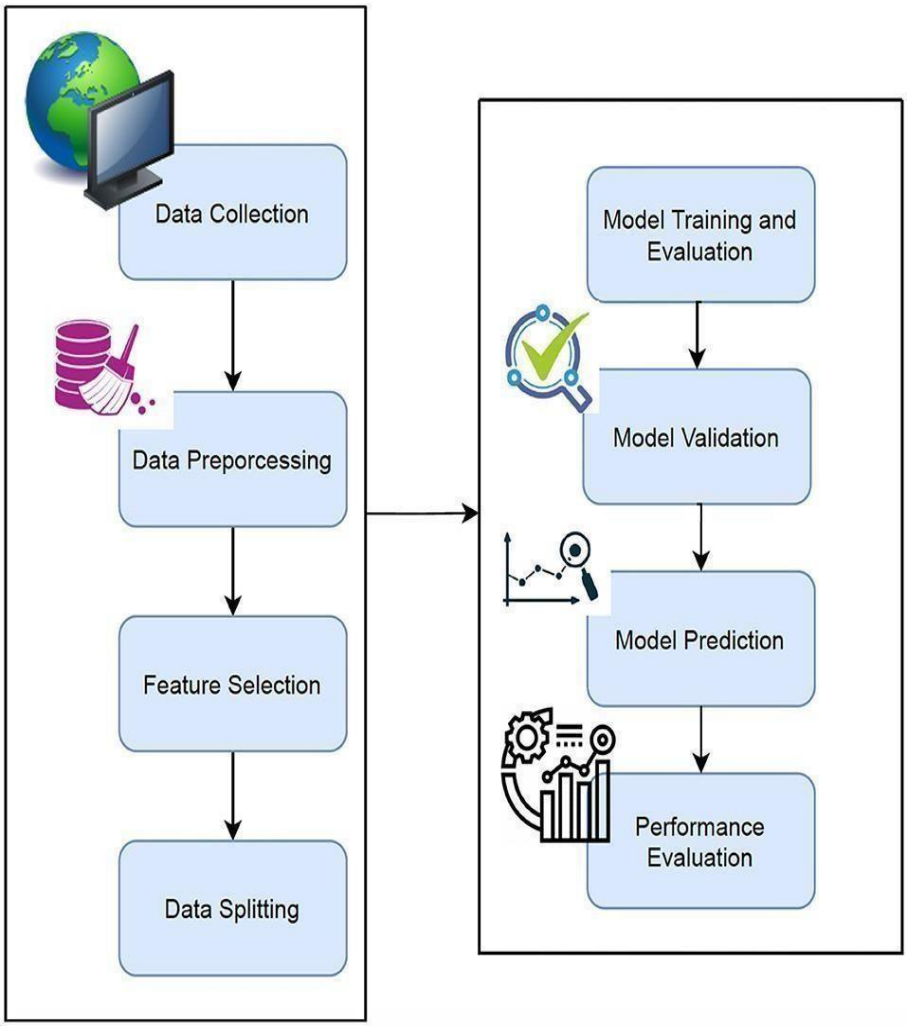
**Training and Testing Split:**

Similar to standard datasets, this dataset may be divided into training and testing sets for model development and evaluation. The training set is used to train machine learning models to distinguish between legitimate and phishing domains based on the provided features, while the testing set is used to assess the model's performance.

By using domains as features along with additional characteristics, this type of dataset enables the development of machine learning models capable of identifying phishing URLs based on various domain-related attributes.

### 3.2.3 Advantages of Proposed System

- The proposed system is fast and can be used to make real-time predictions.
- It has less overfit problem as compared to KNN algorithm.
- It is scalable and can be worked efficiently with the large datasets.
- It gives best accuracy as compared to KNN algorithm in order to detect whether awebsite is phishing or legitimate.
- As compared to other models random forest gives low false-positive and false-negativevalues.




3.4 Proposed model for phishing websites detection.

### 3.3 Purpose of the System

Now a days, everyone is highly dependent on the internet. Everyone performed online shopping and online activities such as online banking, online booking ,online recharge and more on internet. Consequently, the number of cyberattacks such as phishing has been increasing. Phishing is a cyberattack which targets naive online users tricking into revealing sensitive information such as username, password, social security number or credit card number etc. Attackers fool the internet users by masking webpage as a trustworthy or legitimate page to retrieve personal information. So, there is a need for a method which can prevent these phishing attacks. On research it is identified that machine learning plays a veryimportant role in detecting phishing websites. So, in this project we have used Gradient Boosting technique in order to detect the phishing websites accurately. Also, Gradient Boosting algorithm work so well as it provides high detection rate along with low false-positive and false- negative values.

### 3.4 Feasibility Study

Feasibility Study in Software Engineering is a study to evaluate feasibility of propose project or system. Feasibility study in one of stage among important four stages of Software Project Management Process. Feasibility study is carried out based on many proposes to analyze whether software product will be right in terms of development, implantation, contribution of project to the organization etc. During system analysis the feasibility study of the proposed system is to be carried out. Feasibility study is so important stage of Software Project Management Process as after completion of feasibility study it gives a conclusion of whether togo ahead with proposed project as itis practically feasible or to 13 stop proposed project here as it is not right/feasible to develop or to think/analyze about proposed project again. Three key considerations involved in the feasibility analysis are,

- Technical Feasibility
- Economical Feasibility
- Social Feasibility

**Technical Feasibility**

In Technical Feasibility current resources both hardware software along with required technology are analyzed/assessed to develop project .This technical feasibility study

gives report whether there exists correct required resources and technologies which will be used for project development. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modestrequirement, as only minimal or null changes are required for implementing this system.

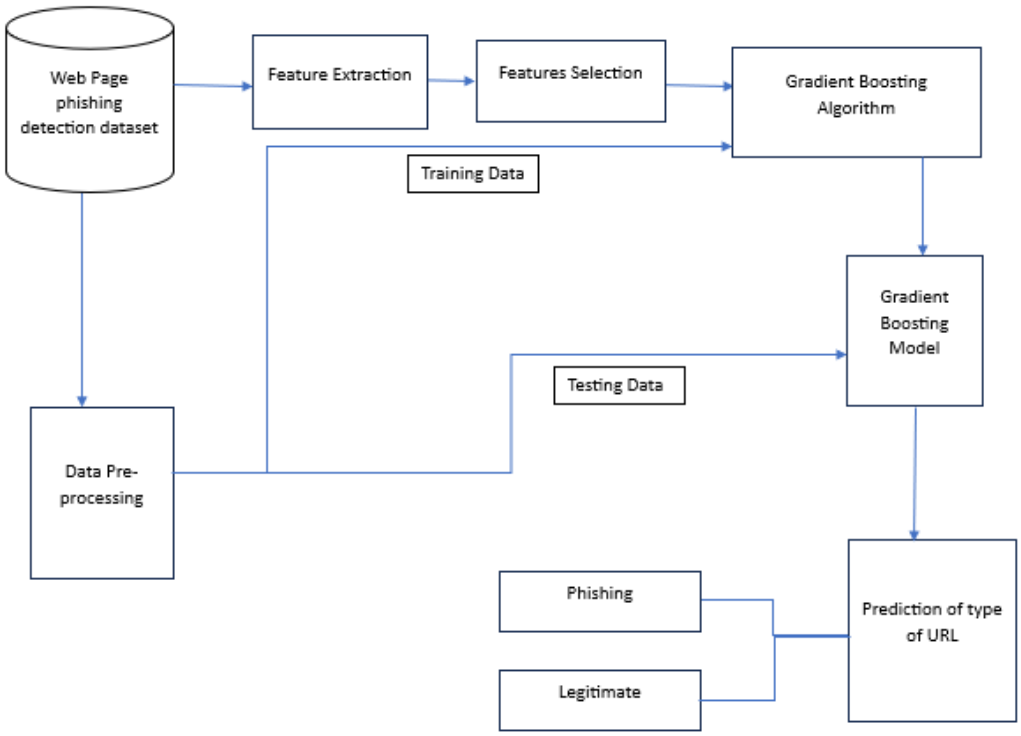
**Economic Feasibility**

In Economic Feasibility study cost and benefit of the project is analyzed. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well with in the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**Social Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includesthe process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solelydepends on the methods that are employed to educate the user about the system and to make him familiar within it.

### 3.5 System Architecture




3.5 System Architecture

The architecture of the system is as shown in figure 3.2. Initially we have acquired a dataset from Kaggle website which contains 11430 URLs of both phishing and legitimate and variousfeatures under the following categories: Hyper link-based features, content-based features andsome features are extracted from querying external services. This dataset is collected in its raw form, but it has been preprocessed to increase its dependability and usefulness for machinelearning algorithms. Data cleaning is done to get rid of unnecessary information, values that are missing and invalid, and data that the ML algorithms would find difficult to process. In the next step we have selected some important features by using Correlation based feature selection. Later ML model is created by utilizing extensible algorithm. The result of the modelwould be a forecast as to whether the attributes of the given data could indicate a legitimate website or a fraudulent website. Here Gradient Boosting algorithm is used in order to build the model.

## 3.6 Proposed Algorithm

**Input** : Training data in the form of URLs.

**Output** : Phishing or Legitimate.

**Step 1** : Collect the dataset which contains the information about both phishing andwebsites along with extracted features.

**Step 2** : The data is analyzed and preprocessed.

**Step 3** : Features are selected from the dataset based on correlation and are used fortraining and testing the model.

**Step 4** : Training the classifier using the training data and produces a forecast result from each decision of the sample.

**Step 5** : Make predictions using the testing data.

**Step 6** : Evaluated the accuracy and other metrics for the predicted model.

# CHAPTER - 4

# SOFTWARE REQUIREMENT SPECIFICATION

## 4.1 Software Requirement Specification

Software Engineering by James F Peters & Witold pedrycz Head First Java by Software Requirements Specification (SRS) is a description of particular software product, program or set of programs that performs a set of functions in a target environment (IEEE Sid. 830- 1993).

### 4.1.1 Purpose:

The purpose of software requirements specification specifies the intentions and intended audience of the SRS.

### 4.1.2 Scope:

The scope of the SRS identifies the software product to be produced, the capabilities, application relevant objects etc. We are proposed to implement Back Propagation Algorithm which takes the test and trained data set.

### 4.1.3 Definitions, Acronyms, Abbreviations Software Requirements:

It's a description of a particular software product, program or set of programs that performs aset of function in target environment.

### 4.1.4 References:

IEEE Std. 830-1993, IEEE Recommended Practice for Software Requirements Specificationsthy Sierra and Bert Bates.

### 4.1.5 Overview:

The SRS contains the details of process, DFD's, functions of the product, user characteristics.The non-functional requirements if any are also specified.

### 4.1.6 Overall Description:

The main functions associated with the product are described in this section of SRS. The characteristics of a user of this product are indicated. The assumptions in this section result from interaction with the project stakeholders.

## 4.2 Requirement Analysis

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The softwareproject is initiated bythe client needs. The SRS is the means of translating the ideas ofthe minds of clients (the input) into a formal document (the output

of the requirement phase.) Under requirement specification, the focus is on Under requirement specification, the focus is on specifying what has been found giving analysis such as representation, specification languagesand tools, and checking the specifications are addressed during this activity. The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the clientand user needs are accurately specified. A good SRS should satisfy all the parties involved in the sem. The software used are open source and easy to install. This is an independent application which can be easily run on to any system which has Python installed and Jupyter Notebook Product features the application is developed in such a way that phishing attack detection accuracy is done using Gradient Boosting algorithm. We can compare the accuracy for the implemented algorithms. User Requirements in which user can decide on the prediction accuracy to decide on which algorithm can be used in real-time predictions. Non-Functional Requirements why Dataset collected should be converted to image dataset from CSV format, why The column values should be numerical values why training set and test set are stored as image files, why Error rates can be calculated for prediction algorithms Product - Requirements Efficiency: Reliability: Maturity, fault tolerance and recoverability. Portability: can the software easily be transferred to another environment, including install ability. ability: How easy it is to understand, learn and operate the software system Organizational Requirements: Do not block some available ports through the windows firewall. The software may be safety- critical. If so, there are issues associated with its integrity of a lower Integrity level. Systems with different requirements for safety levels must be separated. Otherwise, the highest level ofintegrity required must be applied to all systems in the same environment.

**General**

These are the requirements for doing the project. Without using these tools and software's wecan't do the project. So, we have two requirements to do the project. They are

1. Hardware Requirements.
2. Software Requirements.

### 4.3 Hardware Requirements

The hardware requirements may serve as the basis for a contract for the implementation

of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should whatthe system does and not how it should be implemented.

- Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- RAM : 12.0 GB (11.8 GB usable)
- OS: Windows 11

### 4.4 Software Requirements

The software requirements document is the specification of the system. It should include botha definition and a specification of requirements. It is a set of what the system should do ratherthan how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performingtasks and tracking the teams and tracking the team's progress throughout the development activity.

- Python libraries such as pandas, NumPy, matplotlib and seaborn are used.
- Sklearn library using with many subordinate models such as model_selection,classifiers, and metrics etc.
- Visual Studio Code for implementation.

### 4.5 Software Description

#### 4.5.1 Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python works on different platforms (Windows, Mac, Linux, RaspberryPi, etc). Python has a simple syntax similar to theEnglish language. Python has syntax that allows developers to write programs with fewer linesthan some other programming languages. Python runs on an interpreter system. meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. python can be treated in a procedural way, an Object- orientated way or a functional way. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms. including object- oriented, imperative, functional and procedural, andhas a large and comprehensive standard library. Python interpreters are

available for many operating systems. C Python, the reference implementation of Python, is open-source softwareand has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

### 4.5.2 Pandas

Pandas are open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data — an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation.It had very little contribution towards data analysis. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance. Economics, Statistics, analytics, etc.

### 4.5.3 Numpy

NumPy is a general-purpose array-processing package. NumPy is numerical python. It provides a high-performance multidimensional array object. and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains featuresincluding these important ones:

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions.
- Tools for integrating C/C+ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities
- Besides its obvious scientific uses, NumPy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

### 4.5.4 Scikit-Learn

Sckit-learn is a in Python that provides many unsupervised and supervised learning

algorithms. It's built upon some of the technology.

- Simple and efficient tools for data mining and data analysis.
- Accessible to everybody, and reusable in various contexts.
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license.
- Regression, including Linear and logistic Regression.
- Classification
- Clustering
- Model Selection
- Pre-processing

### 4.5.5 Matplotlib

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely – histogram, bar charts, power spectra, error charts etc.

### 4.5.6 Functional Requirements

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior. In proposed system we have used Gradient Boosting Algorithm in order to build the model. Gradient Boosting works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. Gradient Boosting detect the type of website with high accuracy and produce very low false-positive and false-negative values. The mainaim of the project is to do it in a efficient , cost effectively and more scalable and having high accuracy and it is achieved using GB.

# CHAPTER – 5

# LANGUAGES OF IMPLEMENTATION

## 5.1 INTRODUCTION TO SCRIPT

### HTML:

- HTML (Hypertext Markup Language. is the standard markup language used for creating web pages and other web-based documents. It is a markup language that consists of a set of tags and attributes that define the structure and content of a web page.
- HTML documents are typically created using a text editor or an Integrated Development Environment (IDE., and are saved with a .html file extension. When a web browser loads an HTML document, it reads the document's tags and uses them to render the content of the page.
- HTML tags are used to create elements such as headings, paragraphs, images, links, lists, and tables, among others. Attributes can be added to these tags to provide additional information or functionality, such as the source URL for an image, or the destination URL for a link.
- HTML is an essential tool for web development, and is often used in combination with other web technologies such as CSS (Cascading Style Sheets. and JavaScript to create interactive and visually appealing web pages.

### CSS:

- CSS (Cascading Style Sheets. is a styling language used for describing the presentation of a document written in HTML or XML. CSS allows developers to separate the presentation of a document from its content, making it easier to create visually appealing and consistent web pages.
- CSS works by assigning styles to HTML elements using selectors. Selectors target specific HTML elements and define how they should be styled. Styles are defined using property- value pairs, such as color: red or font-size: 16px.
- CSS can be used to define a wide range of styles, including font styles, text colors, background colors, layout and positioning, and more. CSS can also be used to create responsive designs that adapt to different screen sizes and devices, and to create animations and transitions.

- CSS files are typically created using a text editor or an Integrated Development Environment (IDE., and are saved with a .CSS file extension. CSS can be included in an HTML document in a number of ways, including using the tag to link to an external CSS file.

## 5.2 WHY TO CHOOSE PYTHON

If you're going to write programs, there are literally dozens of commonly used languages to choose from Python .Here are some of the features that make Python an appealing choice.


5.1 Python

1. **Python is Popular**

- Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day.
- According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the Popularity of Programming Language Index.
- Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you'd like to dig deeper into Python salary statistics and job opportunities, you can do so here.

2. **Python is interpreted**

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an

interpreter that runs them directly.

- This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.
- One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

**3. Python is Free**

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes. A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

**4. Python is Portable**

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.

**5. Python is Simple**

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way. A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language. Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect. Python code has a Python code has simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code

structure that is easy to read. But It's Not That simple For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming. Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it. Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer. Python is an open-source programming language that was made to be easy-to-read and powerful. A Dutch programmer named Guidovan Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show. Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly. Python is a good programming language for beginners. It is a high- level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages. Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp. Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({}. are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language. Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks. Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that

make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

- Web development
- Scientific programming
- Desktop GUIs
- Network programming
- Game programming

### 5.3 Flask Framework

- Flask is a lightweight web framework for Python. It's designed to be simple and easy to use, allowing developers to quickly build web applications. Flask provides the basics for web development, such as routing URLs to Python functions, rendering templates, managing sessions, and handling requests and responses.

- One of the key features of Flask is its flexibility. It doesn't enforce any particular way of structuring your application, allowing developers to organize their code as they see fit. This makes Flask a popular choice for building a wide range of web applications, from simple prototypes to complex, production-ready systems.

- Flask is also known for its extensive ecosystem of extensions, which provide additional functionality for tasks such as database integration, form validation, authentication, and more. This allows developers to easily add features to their Flask applications without having to reinvent the wheel.

- Overall, Flask is a powerful tool for building web applications in Python, offering simplicity, flexibility, and a vibrant community of users and contributors.

**What is Flask Python Used For?**

Python’s Flask micro web framework is well-liked and frequently used to create online apps. It offers a straightforward and adaptable method for developing Python-based web applications and APIs (Application Programming Interfaces).

Flask is renowned for its straightforward design, which gives developers the freedom to select the elements they desire and customise their apps to meet their needs.

**Common Uses of Flask:**

- **Web Applications:** Flask builds various types of web applications, including blogs, e-commerce sites, social media platforms, and more. Its simplicity and flexibility make it suitable for both small projects and larger, more complex applications.
- **API Development:** Create RESTful APIs that enable communication between different software systems. Use these APIs to share data, perform actions, and integrate various services.
- **Prototyping:** Flask's lightweight nature makes it a great choice for quickly prototyping web-based ideas and concepts. It allows developers to rapidly create and test their ideas without the overhead of more complex frameworks.
- **Microservices:** Flask lends itself well to constructing microservices, which constitute small, autonomously deployable constituents within more extensive applications. Developers can construct each microservice utilizing Flask to offer distinct functionalities.
- **Webhooks:** Webhooks function by obtaining data from external services upon the occurrence of particular events. Flask enables the creation of endpoints capable of listening for these events and initiating actions within your application.
- **Interactive Dashboards:** Flask can be used to build interactive dashboards that visualize data in real-time. This is useful for data analysis, reporting, and monitoring purposes.
- **Educational Projects:** Flask is often used in educational settings to teach web development concepts due to its simplicity and clear structure. Students can quickly learn about routes, templates, and interactions between the front end and back end.

- **Small to Medium-sized Websites:** For websites that don't require the complexity of larger frameworks, Flask provides a lightweight alternative that still supports various features.
- **Integration with Data Science and Machine Learning:** Flask empowers the creation of web interfaces for data science and machine learning models, facilitating user interaction and utilization of these models without the necessity of comprehending the underlying code.

**Flask Framework**

Flask is used for developing web applications using **python**, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are:

- There is a built-in development server and a fast debugger provided.
- Lightweight
- Support Secure cookies
- Templating using Jinja2
- Request dispatching using REST
- Support for unit testing is built-in

# CHAPTER – 6

# SYSTEM DESIGN

## 6.1 UML INTRODUCTION

### Unified Modeling Language (UML)

UML is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

### Do we really need UML?

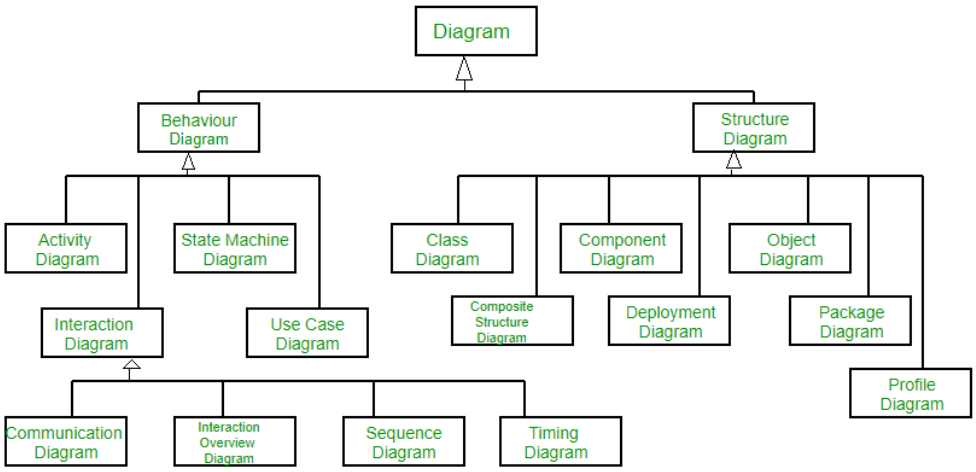
- Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.
- Businessmen do not understand code. So UML becomes essential to communicate with non-programmers essential requirements, functionalities and processes of the system.
- A lot of time is saved down the line when teams are able to visualize processes, user interactions and static structure of the system.

UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

- **Structural Diagrams** – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

- **Behavior Diagrams –** Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

The image below shows the hierarchy of diagrams according to UML 2.2




6.1 Types of UML diagrams

**Object Oriented Concepts Used in UML :**

- **Class –** A class defines the blue print i.e., structure and functions of an object.
- **Objects** – Objects help us to decompose large systems and help us to modularize our system. Modularity helps to divide our system into understandable components so that we can build our system piece by piece. An object is the fundamental unit (building block) of a system which is used to depict an entity.
- **Inheritance** – Inheritance is a mechanism by which child classes inherit the properties of their parent classes.
- **Abstraction** – Abstraction in UML refers to the process of emphasizing the essential aspects of a system or object while disregarding irrelevant details. By abstracting away unnecessary complexities, abstraction facilitates a clearer understanding and communication among stakeholders.
- **Encapsulation** – Binding data together and protecting it from the outer world is referred to as encapsulation.
- **Polymorphism** – Mechanism by which functions or entities are able to exist in different forms.

Originally UML specified 9 diagrams. UML 2.x has increased the number of diagrams from 9 to 13. The four diagrams that were added are : timing diagram, communication diagram, interaction overview diagram and composite structure diagram. UML 2.x renamed state chart diagrams to state machine diagrams.

UML 2.x added the ability to decompose software system into components and sub-components.

**STRUCTURAL UML DIAGRAMS:**

- **Class Diagram** – The most widely use UML diagram is the class diagram. It is the building block of all object-oriented software systems. We use class diagrams to depict the static structure of a system by showing system's classes, their methods and attributes. Class diagrams also help us identify relationship between different classes or objects.
- **Composite Structure Diagram** – We use composite structure diagrams to represent the internal structure of a class and its interaction points with other parts of the system. A composite structure diagram represents relationship between parts and their configuration which determine how the classifier (class, a component, or a deployment node) behaves. They represent internal structure of a structured classifier making the use of parts, ports, and connectors. We can also model collaborations using composite structure diagrams. They are similar to class diagrams except they represent individual parts in detail as compared to the entire class.
- **Object Diagram** – An Object Diagram can be referred to as a screenshot of the instances in a system and the relationship that exists between them. Since object diagrams depict behavior when objects have been instantiated, we are able to study the behavior of the system at a particular instant. An object diagram is similar to a class diagram except it shows the instances of classes in the system. We depict actual classifiers and their relationships making the use of class diagrams. On the other hand, an Object Diagram represents specific instances of classes and relationships between them at a point of time.
- **Component Diagram** – Component diagrams are used to represent how the physical components in a system have been organized. We use them for modelling implementation details. Component Diagrams depict the structural relationship between software system elements and help us in understanding if functional

requirements have been covered by planned development. Component Diagrams become essential to use when we design and build complex systems. Interfaces are used by components of the system to communicate with each other.

- **Deployment Diagram** – Deployment Diagrams are used to represent system hardware and its software. It tells us what hardware components exist and what software components run on them. We illustrate system architecture as distribution of software artifacts over distributed targets. An artifact is the information that is generated by system software. They are primarily used when a software is being used, distributed or deployed over multiple machines with different configurations.

- **Package Diagram** – We use Package Diagrams to depict how packages and their elements have been organized. A package diagram simply shows us the dependencies between different packages and internal composition of packages. Packages help us to organize UML diagrams into meaningful groups and make the diagram easy to understand. They are primarily used to organize class and use case diagrams.

**BEHAVIOUR DIAGRAMS**:

- **State Machine Diagrams** – A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams . These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli.
- **Activity Diagrams** – We use Activity Diagrams to illustrate the flow of control in a system. We can also use an activity diagram to refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.
- **Use Case Diagrams** – Use Case Diagrams are used to depict the functionality of a system or a part of a system. They are widely used to illustrate the functional requirements of the system and its interaction with external agents(actors). A use case is basically a diagram representing different scenarios where the system can be used.

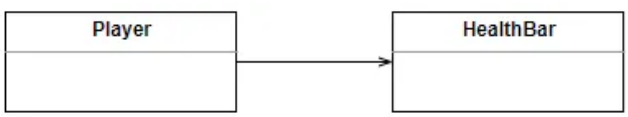
A use case diagram gives us a high-level view of what the system or a part of the system does without going into implementation details.

- **Sequence Diagram** – A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.
- **Communication Diagram** – A Communication Diagram(known as Collaboration Diagram in UML 1.x) is used to show sequenced messages exchanged between objects. A communication diagram focuses primarily on objects and their relationships. We can represent similar information using Sequence diagrams; however, communication diagrams represent objects and links in a free form.
- **Timing Diagram** – Timing Diagram are a special form of Sequence diagrams which are used to depict the behavior of objects over a time frame. We use them to show time and duration constraints which govern changes in states and behavior of objects.
- **Interaction Overview Diagram** – An Interaction Overview Diagram models a sequence of actions and helps us simplify complex interactions into simpler occurrences. It is a mixture of activity and sequence diagrams.

**Association :**

An Association reflects relation between two classes. Use Association arrow when two classes need to communicate and either (or both) class(es) hold reference to the second one. Association relationship is “stronger” than Dependency relationship, it implies a closer connection between entities.

Association

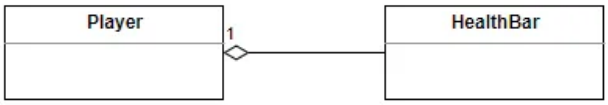



6.2 Association

**Aggregation :**

Aggregation implies that two classes are associated, it brings in more details regarding the nature of the relationship: the child can exist independently of the parent.

Aggregation

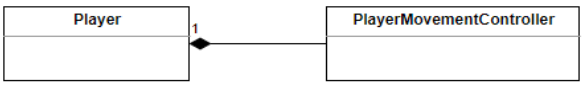



6.3 Aggregation

**Composition :**

Composition implies that two classes are associated and it adds the following details: within a Composition sub-objects are strongly dependent on a whole. Objects of the types are instantiated together and have common lifecycle.

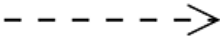
Composition




6.4 Composition

**Dependency :**

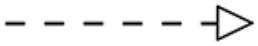
Dependency relationship implies that two elements are dependent on each other. It is used to reflect that one class interacts with another one, receives an instance this class as a method parameter. Compared to Association, Dependency relationship is weaker.



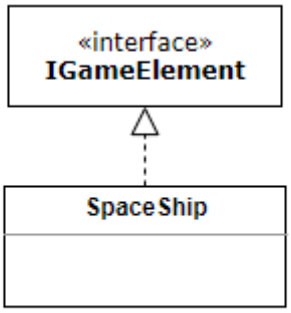
6.5 Dependency

**Implementation / Interfaces :**

Implementation (Implementation) is mainly used to specify the relationship between interfaces and implementation classes . The name of an interface is italicized and «interface» is placed above the interface name. A dashed line going from a class to an interface, terminating with an open arrow signifies that the class implements the interface.



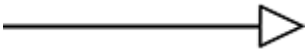
6.6 Implementation



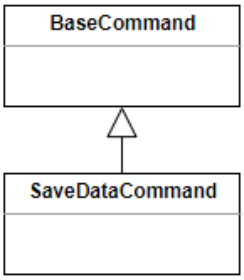

6.7 Implementation

**Inheritance :**

Use Inheritance when you want to show that one class inherits from another one. The inheritance relationship is shown in UML class diagrams using an open arrow from the subclass to the superclass. The open arrow signifies that the superclass is a generalization of the subclass.



Inheritance

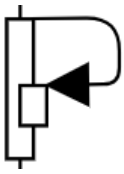



6.8 Inheritance

**Self-call:**
Method calls another method in same object – An object sends a message to itself.

6.9 Self-call

**Call-back:** A calls B, then B calls A. – Not the same as "return"!



6.10 Call-back

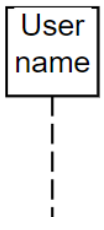
**Delete message:**

Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.

6.11 Delete message

**Lifeline:**

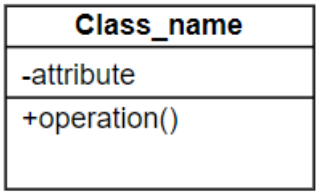
Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol.




6.12 Lifeline

**Class:**

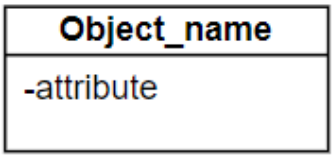
A class is used to represent various objects. It is used to define the properties and operations of an object. In UML, we can also represent an abstract class. A class whose functionalities are not defined is called an abstract class. Any UML class diagram notations are generally expressed as below UML class diagrams example.




6.13 Class symbol

**Object:**

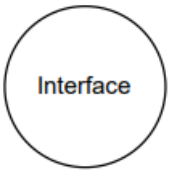
An object is an entity which is used to describe the behavior and functions of a system. The class and object have the same notations. The only difference is that an object name is always underlined in UML. The UML notation of any object is given below.




6.14 Object

**Interface:**

An interface is similar to a template without implementation details. A circle notation represents it. When a class implements an interface, its functionality is also implemented.




6.15 UML Interface Symbol

**Use-case:**

Use-cases are one of the core concepts of object-oriented modeling. They are used to

represent high-level functionalities and how the user will handle the system.


6.16 UML Use Case

**Actor:**

It is used inside use case diagrams. The Actor notation is used to denote an entity that interacts with the system. A user is the best example of an actor. The actor notation in UML is given below.


6.17 UML Actor

**Initial state:**

Each state diagram typically begins with a dark circle that indicates the initial state and ends with a bordered circle that denotes the final state. However, despite having clear start and end points, state diagrams are not necessarily the best tool for capturing an overall progression of events.

6.18 UML Initial state

**Final state:**

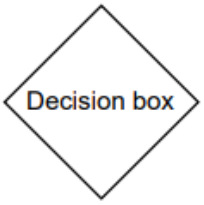
The final state of a state machine diagram is shown as concentric circles. An open loop state machine represents an object that may terminate before the system terminates, while a closed loop state machine diagram does not have a final state; if it is the case, then the object lives until the entire system terminates.

6.19 UML Final state

**Decision box:**

It makes sure that the control flow or object flow will follow only one path. A Decision is an element of an Activity diagram or Interaction Overview diagram that indicates a

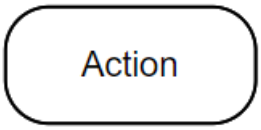
point of conditional progression: if a condition is True, then processing continues one way; if not, then another.




6.20 Decision Box

**Action Box:**

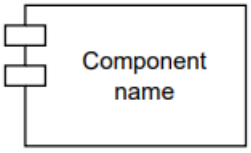
It represents the set of actions that are to be performed. In UML, an action represents a discrete unit of functionality in an activity. Actions have incoming and outgoing activity edges that specify the flow of control and data to and from other activity nodes. The actions in an activity start when all of the input conditions are met.




6.21 UML **Action Box**

**Component:**

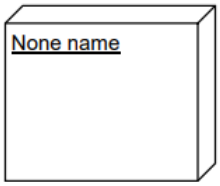
A component notation is used to represent a part of the system. It is denoted in UML like given below.




6.22 UML Component

**Node:**

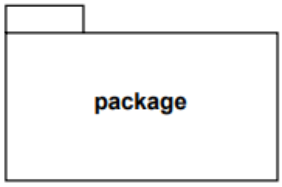
A node is used to describe the physical part of a system. A node can be used to represent a network, server, routers, etc. Its notation is given below.




6.23 UML Node

**Grouping things:**

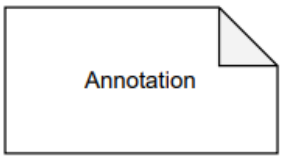
It is the package which is used to group semantically related modeling elements into a single cohesive unit. The package is the only grouping thing available in the UML.




6.24 UML Package

**Annotational things:**

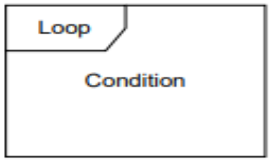
It is like a note, which may be written to the model to capture some vital information. It is similar to the yellow sticky note. Here is an example for annotation things in UML.




6.25 UML Annotation

**Option loop:**

In UML this symbol id used to model if/then scenarios, i.e., a circumstance that will only occur under certain conditions.

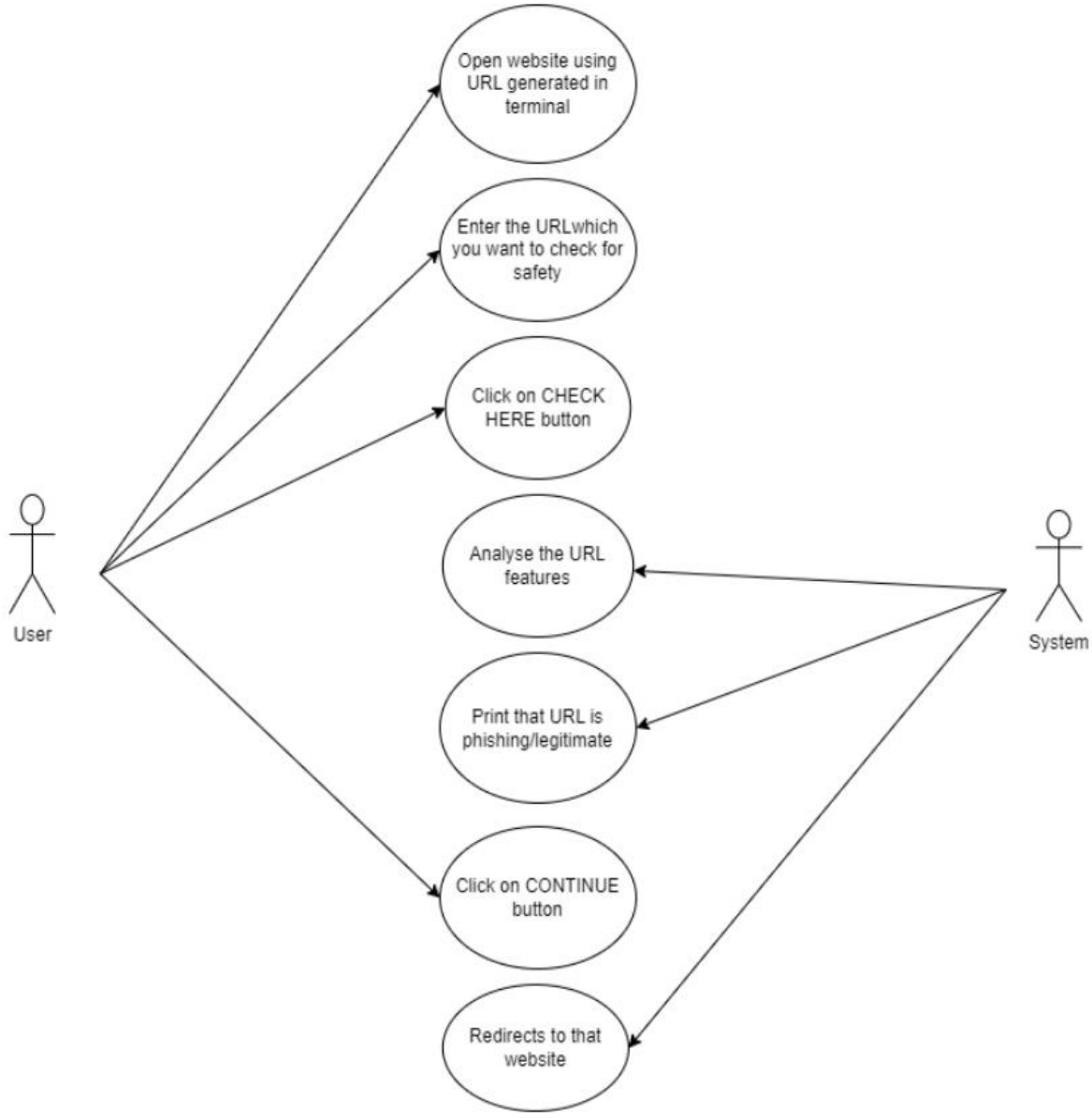



6.26 UML Option loop

## 5.2 UML Diagrams

### 5.2.1 Use Case Diagram

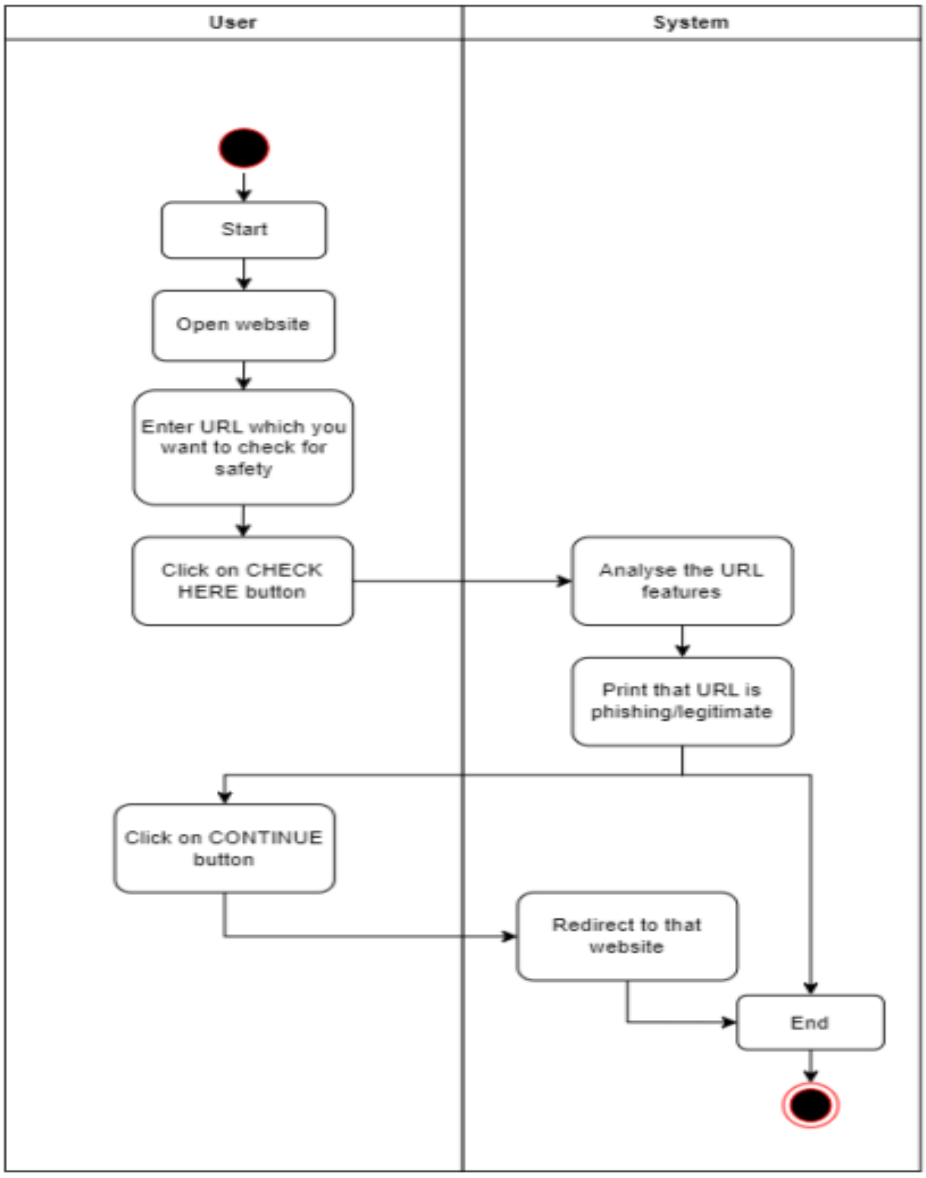
The primary motivation behind a utilization case chart is to show what framework capacities are performed for which entertainer. Parts of the entertainers in the framework can be portrayed.The above chart comprises of client as entertainer. Each will assume a specific part idea.




6.27 Use Case Diagram

### 5.2.2 Activity Diagram

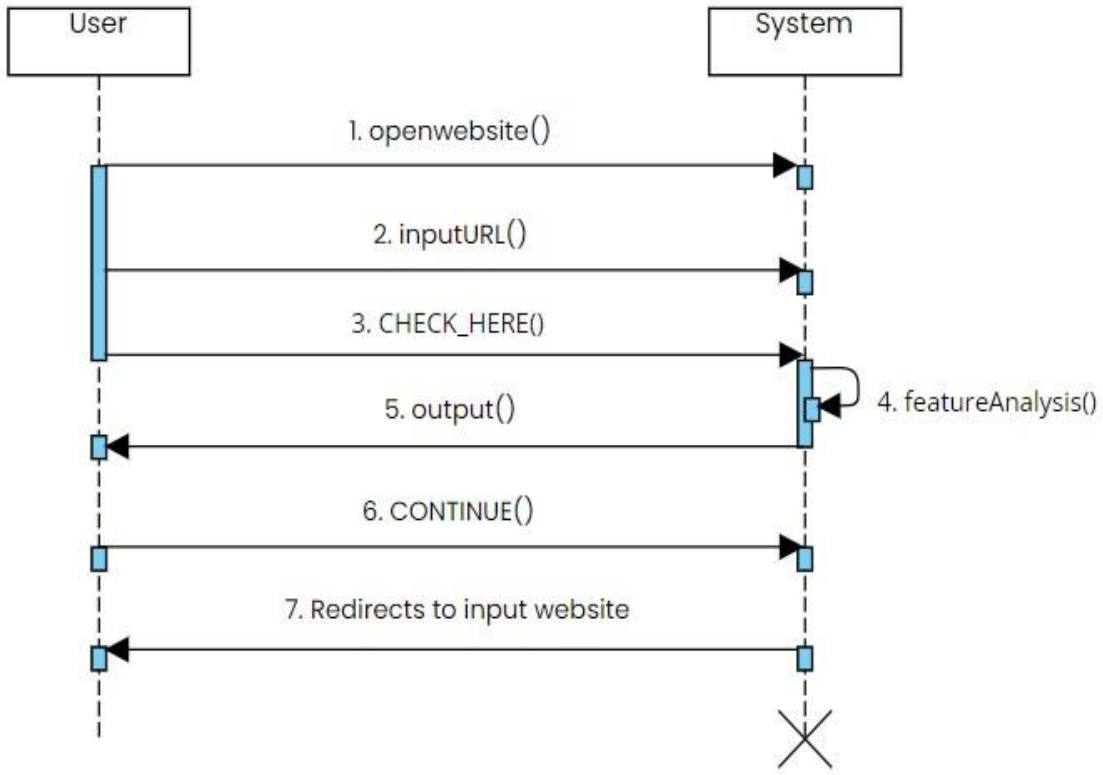
Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chartthat modeling the flow from one activity to another activity. Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the eventsin a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.




6.28 Activity Diagram

### 5.2.3 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

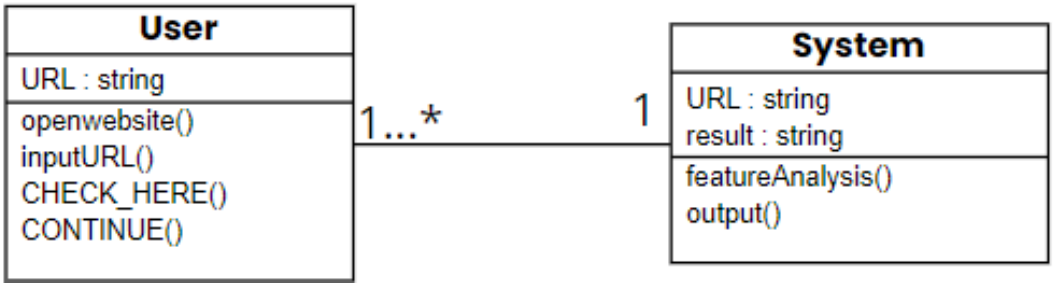



6.29 Sequence Diagram

The figure 5.2.3 explains the sequence of actions for performing an action in phishing detection and prevention. The appropriate processing will be done between the user, phishing page and attacker.

### 5.2.4 Class Diagram

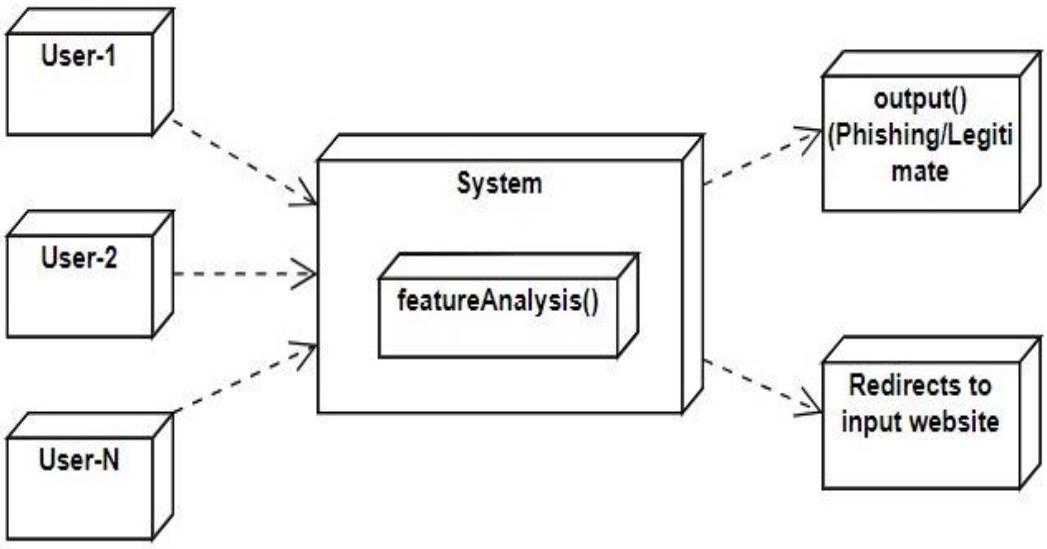
A class is used to represent various objects. It is used to define the properties and operations of an object. In UML, we can also represent an abstract class. A class whose functionalities are not defined is called an abstract class. Any UML class diagram notations are generally expressed as below UML class diagrams example.




6.30 Class Diagram

### 5.2.5 Deployment Diagram

Deployment diagrams model the physical architecture of a system. Deployment diagrams show the relationships between the software and hardware components in the system and the physical distribution of the processing. Deployment diagrams, which you typically prepare during the implementation phase of development, show the physical arrangement of the nodes in a distributed system. Nodes represent hardware devices such as computers, sensors, and printers, as well as other devices that support the runtime environment of a system. Communication paths and deploy relationships model the connections in the system.




6.31 Deployment Diagram

# CHAPTER – 7

# IMPLEMENTATION

## 7.1 Implementation Procedure

### Implementation on Python

What is a Script?

A script or scripting language is a computer language with a series of commands within a file that is capable of being executed without being compiled. This is a very useful capability that allows us to type in a program and to have it executed immediately in an interactive mode .

- Scripts are reusable
- Scripts are editable

### Difference between a script and programScript:

Scripts are distinct from the core code of the application, which is usually written in a differentlanguage, and are often created or at least modified by the end-user. Scripts are often interpretedfrom source code or byte code, whereas the applications they control are traditionally compiledto naïve machine code.

### Program:

The program has an executable from that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executableprograms are derived (e.g., compiled).

### Python:

What is Python? Python is an interpreter, high-level, general-purpose programming language. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

### Python concepts:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, than other languages.

- **Python is interpreted –** Python is processed at runtime by the interpreter. You do not needto compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive -** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented –** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language –** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at theNational Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68,Smalltalk, and Unix shell and Other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU GeneralPublic License (GPL).

Python is now maintained by a core development team at the institute, although Guido vanRossum still holds a vital role in directing its progress.

**Python Features**

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax.This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python 's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, windows, and Macintosh.
- **Interactive Mode:** Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has the sameinterface on all platforms.
- **Extensible:** You can add low-level modules to the Python interpreter. These

modules enable programmers to add to or customize their tools to be more efficient.

- **Databases:** Python provides interfaces to all major commercial databases.
- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

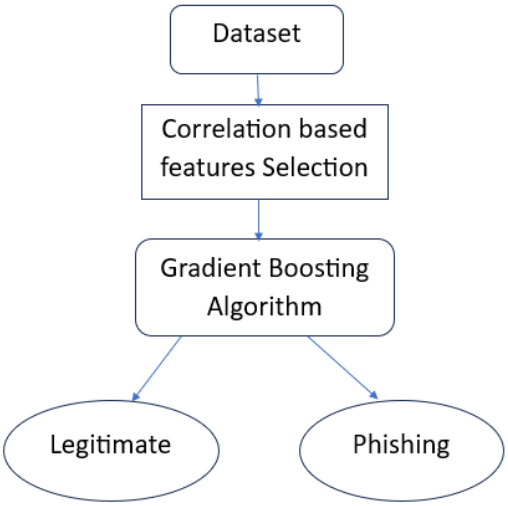
**Python modules:**

Python allows us to store our code in files(also called modules). To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a module; definitions from a module can be imported into othermodule or into the main module.

**Testing code:**

- Code is usually developed in a file using an editor.
- To test the code, import it into a python session and try to run it.
- Usually there is an error, so you can check by go to file, make a correction, and test again. This process is repeated until you are satisfied that the code works. The entire process is known as the development cycle.

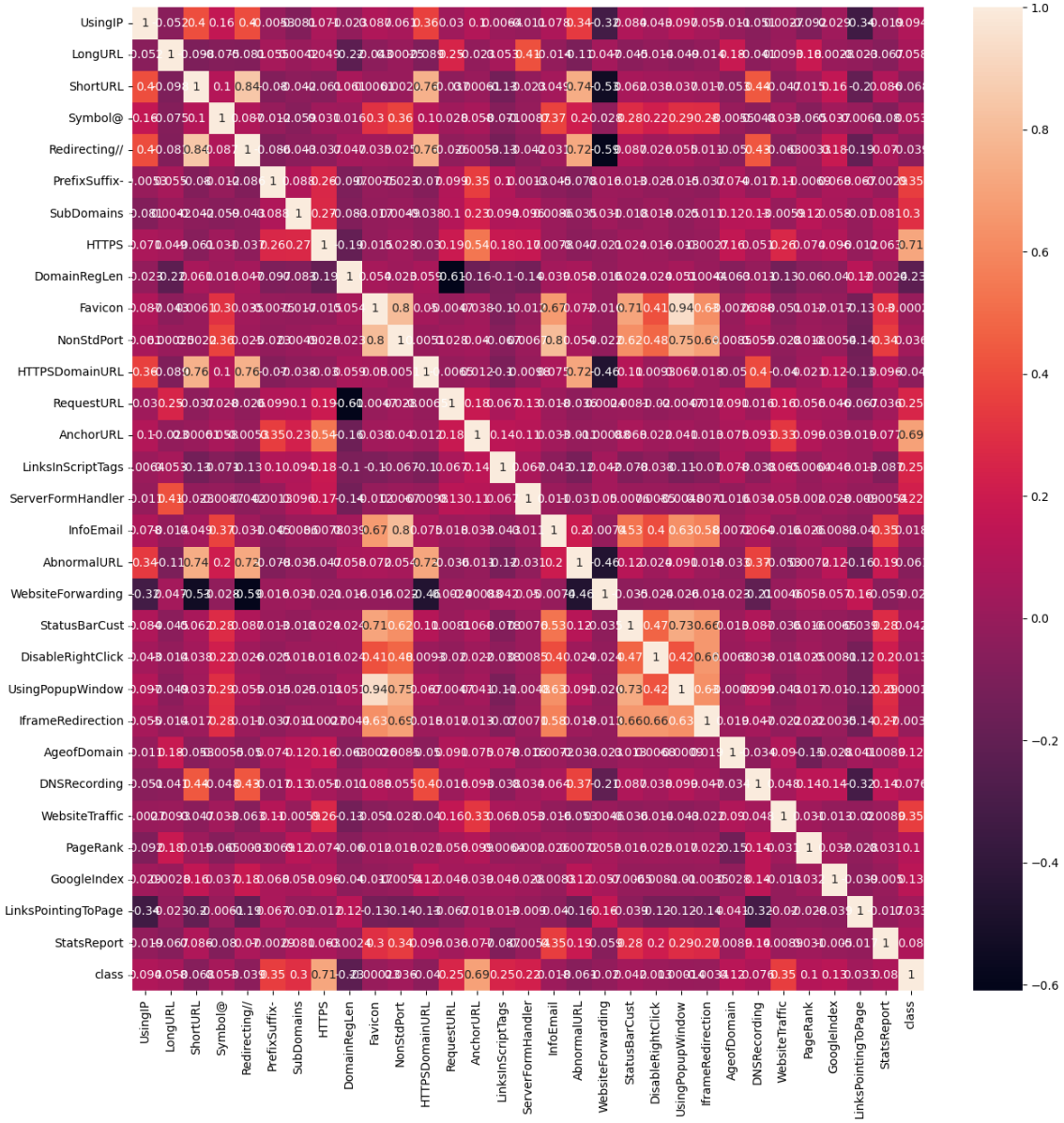
### 7.1.1 Architecture of predicting the type of website using Features of URLs




7.1 Architecture of predicting the type of website

### 7.1.2 Feature Selection

Performance is greatly influenced by the characteristics of the data that are trained into ML models. Model performance may be negatively impacted by irrelevant features or partially relevant. For feature selection, we employ a correlation-based feature selection approach. Correlation-based feature selection selects features that have low linear relations with other features, but are correlated with the label. The following is the correlation matrix that depicts the correlation between various features that are present in the dataset.




7.2 Features Overview

The relationship between various characteristics of the URLs that are present in the dataset is shown in by the correlation matrix, which shows the correlation. The correlation matrix correlates every feature in the dataset with every other feature, returning the findings as a graph.It is a potent tool for finding and displaying trends in the provided data as well as for summarizing a sizable dataset. In dataset we have a total of 87 Extracted features from the URL.From them we have selected the most important 23 features using correlation-based feature selection method. The following are the features that are selected from CBFS method.

length_url
longest_words_raw
phish_hints
ratio_intHyperlinks
tld_in_subdomain
prefix_suffix
ratio_intHyperlinks
empty_title
ip
nb_qm
nb_www
ratio_digits_url
shortest_word_host
longest_word_path
nb_hyperlinks
length_hostname,
domain_age
page_rank
domain_in_title
google_index
nb_dots
nb_eq
nb_slash
ratio_digits_host

**7.2 Coding**

#importing required libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import seaborn as sns
from sklearn import metrics
import warnings
warnings.filterwarnings('ignore')

#Loading data into dataframe

data = pd.read_csv("phishing.csv")
data.head()

#Shape of dataframe

data.shape

#Listing the features of the dataset

data.columns

#Information about the dataset

data.info()

# nunique value in columns

data.nunique()

#droping index column

data = data.drop(['Index'],axis = 1)

#description of dataset

data.describe().T
```

```
#Correlation heatmap

plt.figure(figsize=(15,15))
sns.heatmap(data.corr(), annot=True)
plt.show()

# Splitting the dataset into dependant and independant fetature

X = data.drop(["class"],axis =1)
y = data["class"]

# Splitting the dataset into train and test sets: 80-20 split

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 =
42)
X_train.shape, y_train.shape, X_test.shape, y_test.shape

# Creating holders to store the model performance results
ML_Model = []
accuracy = []
f1_score = []
recall = []
precision = []

#function to call for storing the results
def storeResults(model, a,b,c,d):
  ML_Model.append(model)
  accuracy.append(round(a, 3))
  f1_score.append(round(b, 3))
  recall.append(round(c, 3))
  precision.append(round(d, 3))
# Gradient Boosting Classifier Model
```

```
from sklearn.ensemble import GradientBoostingClassifier

# instantiate the model
gbc = GradientBoostingClassifier(max_depth=4,learning_rate=0.7)
# fit the model
gbc.fit(X_train,y_train)

#predicting the target value from the model for the samples
y_train_gbc = gbc.predict(X_train)
y_test_gbc = gbc.predict(X_test)

#computing the accuracy, f1_score, Recall, precision of the model performance

acc_train_gbc = metrics.accuracy_score(y_train,y_train_gbc)
acc_test_gbc = metrics.accuracy_score(y_test,y_test_gbc)
print("Gradient Boosting Classifier : Accuracy on training Data:
{:.3f}".format(acc_train_gbc))
print("Gradient Boosting Classifier : Accuracy on test Data:
{:.3f}".format(acc_test_gbc))
print()

f1_score_train_gbc = metrics.f1_score(y_train,y_train_gbc)
f1_score_test_gbc = metrics.f1_score(y_test,y_test_gbc)
print("Gradient Boosting Classifier : f1_score on training Data:
{:.3f}".format(f1_score_train_gbc))
print("Gradient Boosting Classifier : f1_score on test Data:
{:.3f}".format(f1_score_test_gbc))
print()

recall_score_train_gbc = metrics.recall_score(y_train,y_train_gbc)
recall_score_test_gbc =  metrics.recall_score(y_test,y_test_gbc)
print("Gradient Boosting Classifier : Recall on training Data:
{:.3f}".format(recall_score_train_gbc))
print("Gradient Boosting Classifier : Recall on test Data:
```

```
{:.3f}".format(recall_score_test_gbc))
print()
precision_score_train_gbc = metrics.precision_score(y_train,y_train_gbc)
precision_score_test_gbc = metrics.precision_score(y_test,y_test_gbc)
print("Gradient Boosting Classifier : precision on training Data:
{:.3f}".format(precision_score_train_gbc))
print("Gradient Boosting Classifier : precision on test Data:
{:.3f}".format(precision_score_test_gbc))

#computing the classification report of the model

print(metrics.classification_report(y_test, y_test_gbc))

training_accuracy = []
test_accuracy = []
# try learning_rate from 0.1 to 0.9
depth = range(1,10)
for n in depth:
    forest_test =  GradientBoostingClassifier(learning_rate = n*0.1)

    forest_test.fit(X_train, y_train)
    # record training set accuracy
    training_accuracy.append(forest_test.score(X_train, y_train))
    # record generalization accuracy
    test_accuracy.append(forest_test.score(X_test, y_test))


#plotting the training & testing accuracy for n_estimators from 1 to 50
plt.figure(figsize=None)
plt.plot(depth, training_accuracy, label="training accuracy")
plt.plot(depth, test_accuracy, label="test accuracy")
plt.ylabel("Accuracy")
plt.xlabel("learning_rate")
plt.legend()
```

```
training_accuracy = []
test_accuracy = []
# try learning_rate from 0.1 to 0.9
depth = range(1,10,1)
for n in depth:
    forest_test =  GradientBoostingClassifier(max_depth=n,learning_rate = 0.7)

    forest_test.fit(X_train, y_train)
    # record training set accuracy
    training_accuracy.append(forest_test.score(X_train, y_train))
    # record generalization accuracy
    test_accuracy.append(forest_test.score(X_test, y_test))
#plotting the training & testing accuracy for n_estimators from 1 to 50
plt.figure(figsize=None)
plt.plot(depth, training_accuracy, label="training accuracy")
plt.plot(depth, test_accuracy, label="test accuracy")
plt.ylabel("Accuracy")
plt.xlabel("max_depth")
plt.legend()

#storing the results. The below mentioned order of parameter passing is important.

storeResults('Gradient Boosting Classifier',acc_test_gbc,f1_score_test_gbc,
          recall_score_train_gbc,precision_score_train_gbc)
```

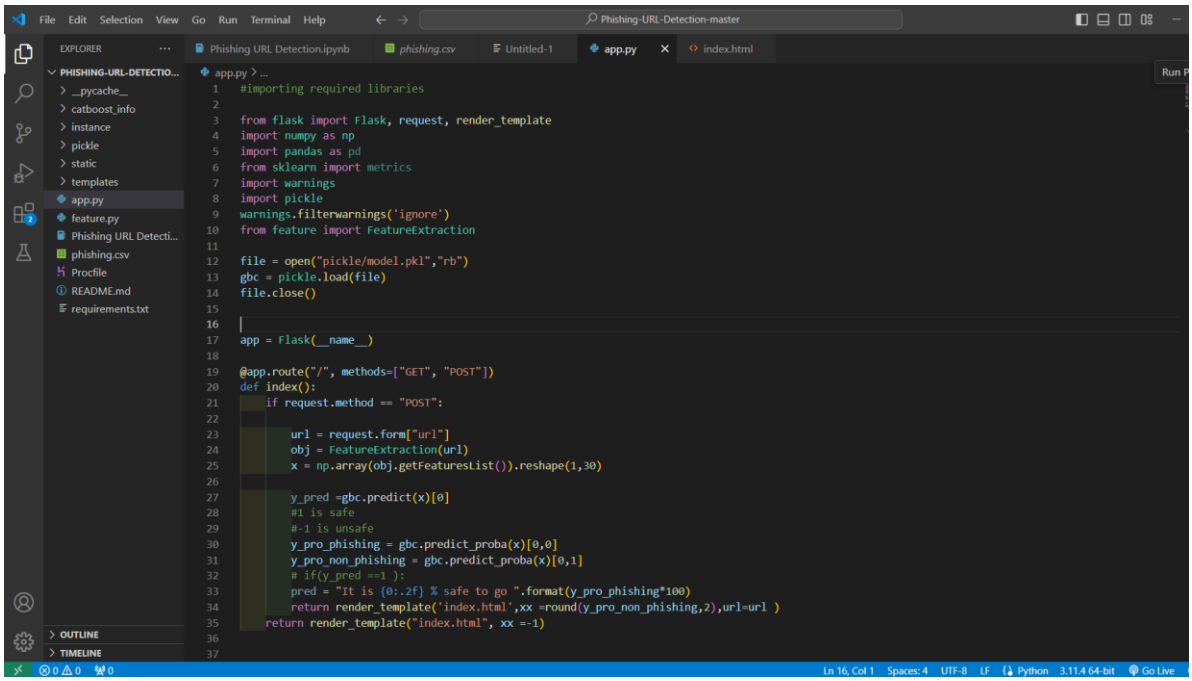
**app.py:**

```
#importing required libraries

from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
```

```
import pickle
warnings.filterwarnings('ignore')
from feature import FeatureExtraction

file = open("pickle/model.pkl","rb")
gbc = pickle.load(file)
file.close()
app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":

        url = request.form["url"]
        obj = FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)

        y_pred =gbc.predict(x)[0]
        #1 is safe
        #-1 is unsafe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
        return render_template('index.html',xx =round(y_pro_non_phishing,2),url=url )
    return render_template("index.html", xx =-1)
if __name__ == "__main__":
    app.run(debug=True)
```

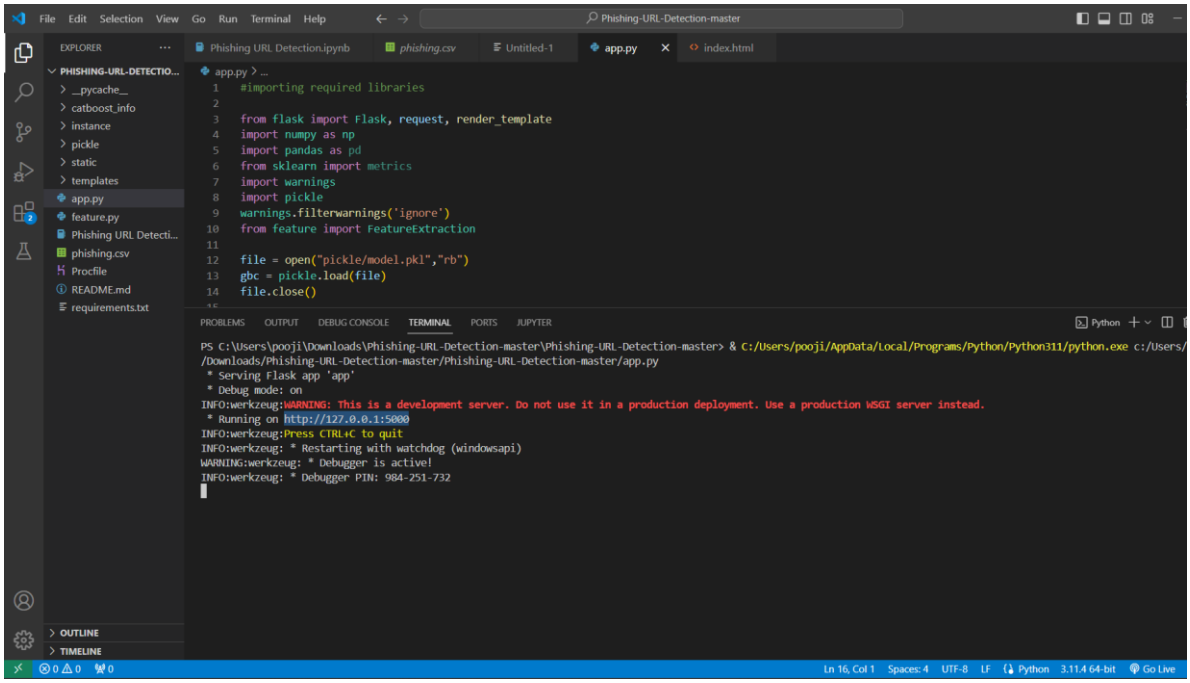
## 6.3 Execution

**Step – 1:** Open VSCode and go to “app.py” file. Run the file.



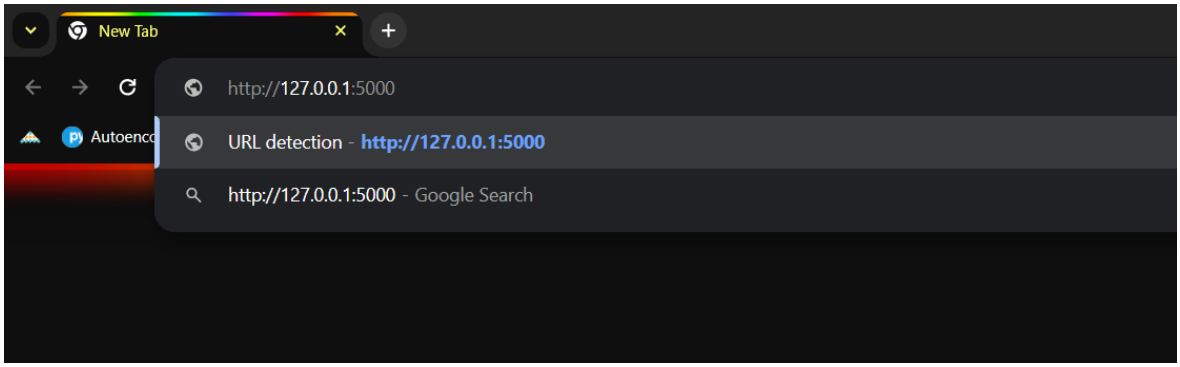

7.3 Step 1

**Step – 2:** After running the file a link will be generated. Copy the link.



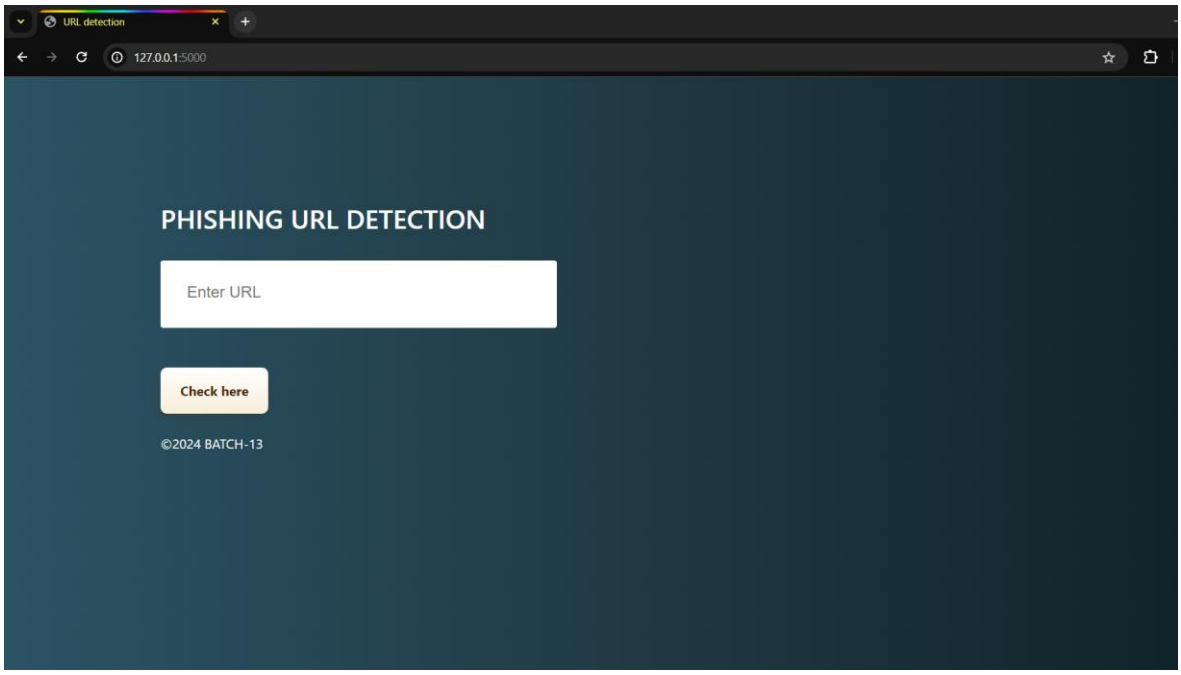

7.4 Step 2

**Step – 3:** Paste that particular link in a website and run it.



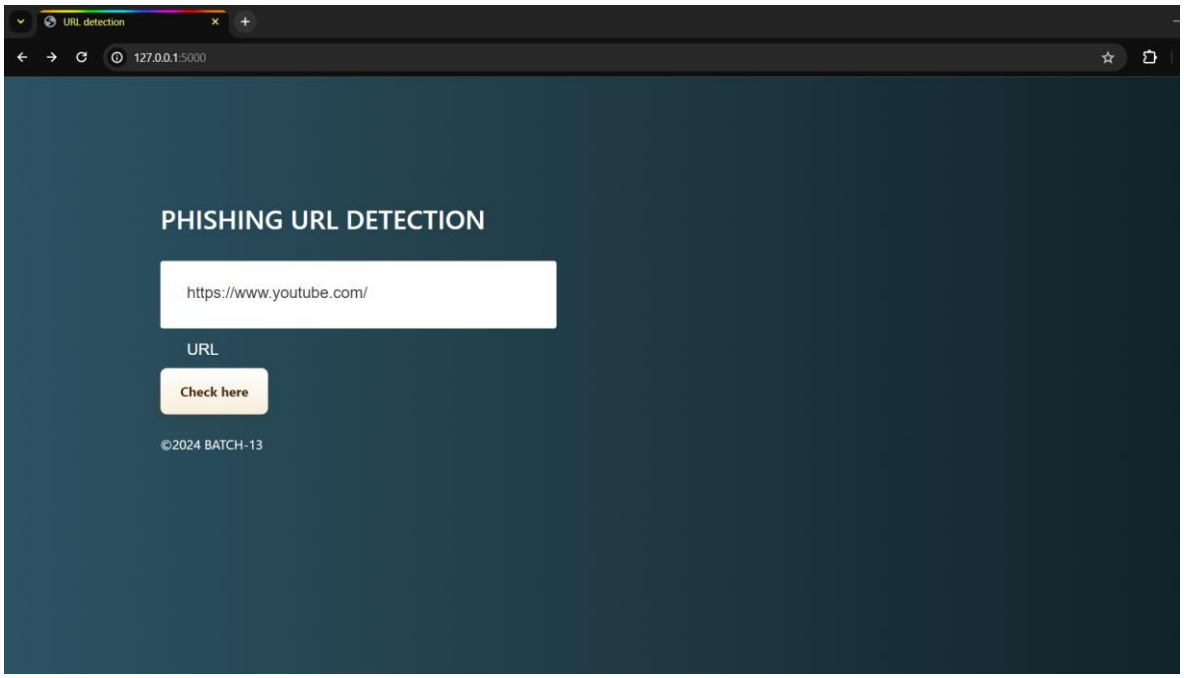

7.5 Step 3

**Step – 4:** After that a webpage will be generated. This is the phishing URL detection website.



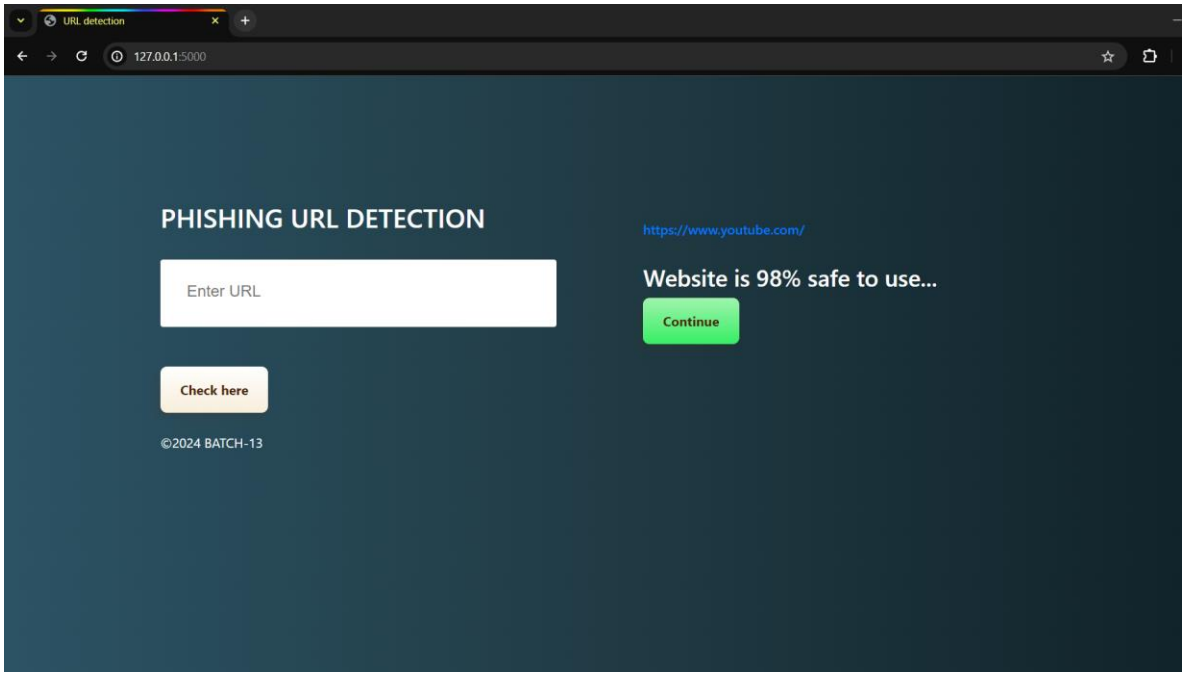

7.6 Step 4

**Step – 5:** Paste any link and click on “Check Here”.



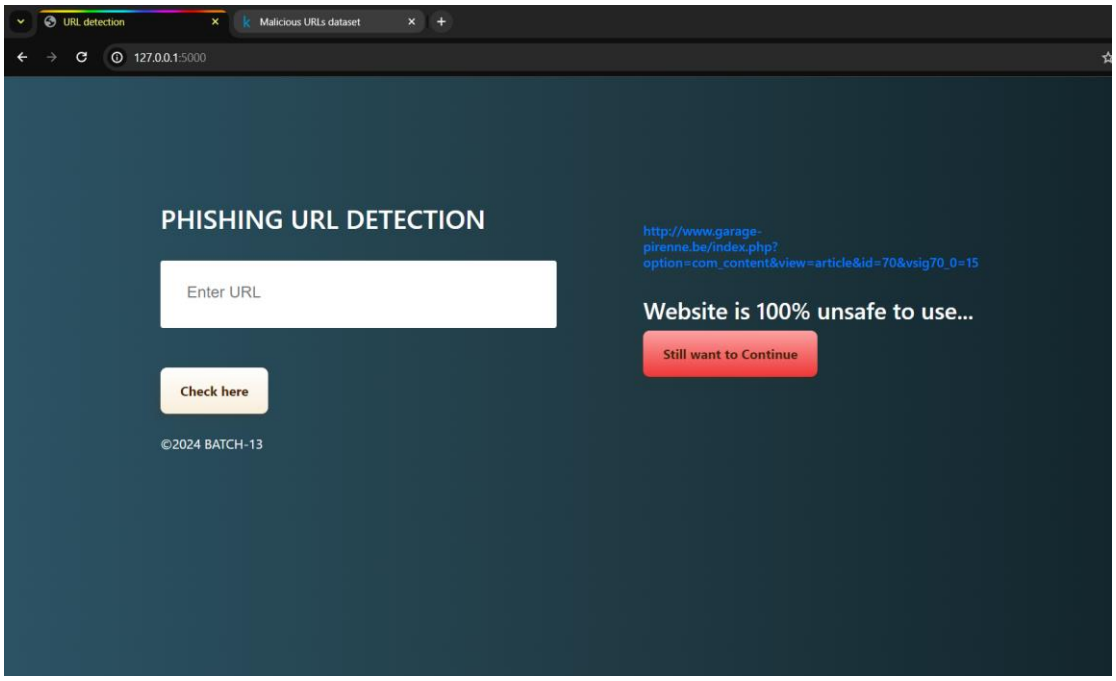

7.7 Step 5

**Safe Link:**




7.8 Safe Link

**Phishing:**




7.9 Phishing Link

# CHAPTER – 8
# SYSTEM TESTING

## 8.1 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality ofcomponents, sub-assemblies, assemblies and/or a finished product It is the process of exercisingsoftware with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Eachtest type addresses a specific testing requirement.

**Manual Testing**

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such asunit testing, integration testing, system testing, and user acceptance testing.

**Automation Testing**

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

**What to Automate?**

It is not possible to automate everything in a software. The areas at which a user can make transactions such as the login form or registration forms, any area where large number of userscan access the software simultaneously should be automated.

**When to Automate?**

Test Automation should be used by considering the following aspects of a software

- Large and critical projects
- Projects that require testing the same areas frequently
- Requirements not changing frequently
- Accessing the application for load and performance with many virtual users
- Stable software with respect to manual testing.
- Availability of time.

**How to Automate?**

Automation is done by using a supportive computer language like VB scripting and an automated software application. There are many tools available that can be used to write automation scripts. Before mentioning the tools, let us identify the process that can be used to automate the testing process.

- Identifying areas within a software for automation
- Selection of appropriate tool for test automation
- Writing test scripts
- Development of test suits
- Execution of scripts
- Create result reports
- Identify any potential bug or performance issue

## 8.2 TYPES OF TESTS

### 8.2.1 Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests performbasic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accuratelyto the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**:

- Field testing will be performed manually and functional tests will be written in detail.
- Test objectives
- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested:

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

### 8.2.2 Integration Testing:

Integration tests are designed to test integrated software components to determine if they actuallyrun as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct andconsistent. Integration testing is specifically aimed at exposing the problems that arise from thecombination of components. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

### 8.2.3 Functional Testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.Functional testing is centered on the following items:

- **Valid Input:** identified classes of valid input must be accepted.
- **Invalid Input:** identified classes of invalid input must be rejected.
- **Functions:** identified functions must be exercised.
- **Output:** identified classes of application outputs must be exercised.
- **Systems/Procedures:** interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify

Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### 8.2.4 System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### 8.2.5 White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is usedto test areas that cannot be reached from a black box level.

### 8.2.6 Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structureor language of the module being tested. Black box tests must be written from a definitive sourcedocument, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box.you cannot ―see‖ into it. The test provides inputs and responds to outputs without considering how the software works.

### 8.3 Testcases

| Testcase ID | Testcase Description | Testcase Input | Expected Outcome | Actual Outcome | Testcase Status |
|---|---|---|---|---|---|
| **T1** | Open website | User tries to open the website through live hosting link | Website should be displayed to the user | Website will be displayed to the user | Pass |
| **T2** | Identifying URL is legitimate or Phishing | User inputs an URL to check whether it is legitimate or not. | The URL must be legitimate. | The URL is legitimate | Pass |
| **T3** | Identifying URL is legitimate or Phishing | User inputs an URL to check whether it is legitimate or not. | The URL must be phishing. | The URL is phishing | Pass |

# CHAPTER – 9

# CONCLUSION

Phishing websites are growing quickly today and harming individuals and businesses more and more. The largest threat to people's everyday lives and networking environments is growing. In such attacks,the attacker poses as a reputable entity with the purpose to steal important and legally admissible data. A phishing website is a fake website with a similar look but a different location. Unknown people post their information believing that these websites are affiliated with reputable financial organizations. Therefore, a reliable method of phishing website identification is required. In our project, we created a model that can primarily be used to identify a website as either genuine or fraud by employing featuresselection methodologies. This Feature selection technique is very helpful In finding the important features of URLs that can be very much useful in URLs type classification. In this case, we used the established model in our project to apply approaches like LR, KNN, Decision Tree, and Random Forest.It has been stated that the system has functioned effectively and in line with expectations throughout testing. This study uses machine learning technologies to improve the diagnosis of fake websites. We used the Random Forest classifier, which had the lowest false positive rate, to reach a detection accuracyof 97.2%. As classifiers perform better when more data is utilized as training data.

# CHAPTER – 10
# FUTURE ENHANCEMENT

## 9.1 Future Enhancement

In the future, hybrid technology that combines the blacklist approach with the random forest algorithm of machine learning technology will be utilised to more reliably detect phishing websites. And we are planning to build a chrome extension for detecting phishing webpages.

# REFERENCES

[1] Deshpande, A., Pedamkar, O., Chaudhary, N., & Borde, S. (2021). Detection of Phishing Websites using Machine Learning. INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)
Volume, 10.

[2] Rashid, J., Mahmood, T., Nisar, M. W., & Nazir, T. (2020, November). Phishing detectionusing machine learning technique. In 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH) (pp. 43-46). IEEE.

[3] Bai, W. (2020, August). Phishing website detection based on machine learning algorithm. In 2020 International Conference on Computing and Data Science (CDS) (pp. 293-298). IEEE.

[4] Vilas, M. M., Ghansham, K. P., Jaypralash, S. P., & Shila, P. (2019, December). Detectionof phishing website using machine learning approach. In 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques(ICEECCOT) (pp.384-389). IEEE.

[5] Janet, B., & Kumar, R. J. A. (2021, March). Malicious URL detection: a comparative study. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) (pp. 1147-1151). IEEE.

[6] Patil, V., Thakkar, P., Shah, C., Bhat, T., & Godse, S. P. (2018, August). Detection and prevention of phishing websites using machine learning approach. In 2018 Fourth international conference on computing communication control and automation (ICCUBEA) (pp. 1-5). Ieee.

[7] Shahrivari, V., Darabi, M. M., & Izadi, M. (2020). Phishing detection using machine learning techniques. arXiv preprint arXiv:2009.11116.

[8] Mahajan, R., & Siddavatam, I. (2018). Phishing website detection using machine learning algorithms. International Journal of Computer Applications, 181(23), 45-47.

Ali, W. (2017). Phishing website detection based on supervised machine learning with wrapper features selection. International Journal of Advanced Computer Science and Applications, 8(9).

[9] Maddumala, V.R., Arunkumar, R. (2020). Big data-driven feature extraction and

clustering based on statistical methods.Treatment du Signal, Vol. 37, No. 3, pp. 387-394.

[10] Maddumala, V.R., R, A. (2021). Body mass index prediction and classification based on facial morphological cues using multinomial logistic regression. Revued 'IntelligenceArtificial, Vol. 35, No.2, pp. 105-113.

[11] Maddumala, V.R., R, A. (2020). A weight based feature extraction model on multifaceted multimedia bigdata using convolutional neural network. Ingénierie des Systems d'Information, Vol. 25,No. 6, pp. 729-735.

[12]A Treatment to Cure Diabetes Using Plant-Based Drug Discovery Mahankali, S., Kalava, J., Garapati, Y., Maddumala, V.R., Sundramurty, V.P.Evidence based complementary and alternative medicine, 2022, 2022, 8621665.

[13] Enhanced morphological operations for improving thepixel intensity level Maddumala, V.R., Maha Lakshmi, K., Anusha, P., Lakshman Narayana, V. International Journal of Advanced Science and Technology, 2020, 29(3), pp. 9191–920.

[14] Medical data clustering using particle swarm optimization method Lakshmi Patibandla, R.S.M., Tarakeswara Rao, B., Sandhya Krishna, P., Maddumala, V.R. Journal of Critical Reviews, 2020, 7(6), pp. 363–367.

[15] J. Rashid, T. Mahmood, M. W. Nisar and T. Nazir, "Phishing Detection Using Machine Learning Technique," 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), 2020, pp. 43-46.

[16] M. H. Alkawaz, S. J. Steven and A. I. Hajamydeen, "Detecting Phishing Website Using Machine Learning," 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), 2020, pp. 111-114.

[17] V. Patil, P. Thakkar, C. Shah, T. Bhat and S. P. Godse, "Detection and Prevention of Phishing Websites Using Machine Learning Approach," 2018 Fourth International Conference onComputing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5.

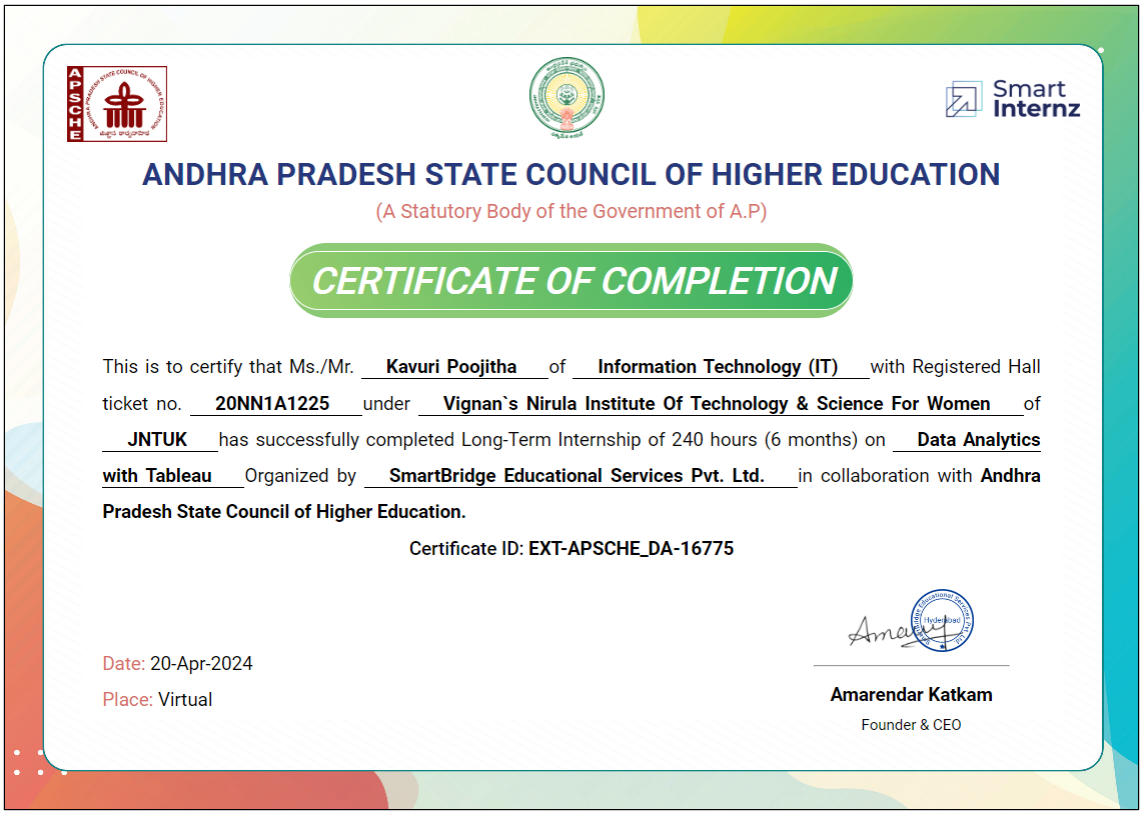
[18] W. Bai, "Phishing Website Detection Based on Machine Learning Algorithm," 2020 International Conference on Computing and Data Science (CDS), 2020, pp. 293-298.

[19] A. Razaque, M. B. H. Frej, D. Sabyrov, A. Shaikhyn, F. Amsaad and A. Oun, "Detection of Phishing Websites using Machine Learning," 2020 IEEE Cloud Summit, 2020, pp. 103-107.

[20] M. M. Vilas, K. P. Ghansham, S. P. Jaypralash and P. Shila, "Detection of Phishing

Website Using Machine Learning Approach," 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), 2019, pp. 384-389.

[21] A. Alswailem, B. Alabdullah, N. Alrumayh and A. Alsedrani, "Detecting Phishing Websites Using Machine Learning," 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), 2019, pp. 1-6.



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Kavuri Poojitha** of **Information Technology (IT)** with Registered Hall ticket no. **20NN1A1225** under **Vignan`s Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Data Analytics with Tableau** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

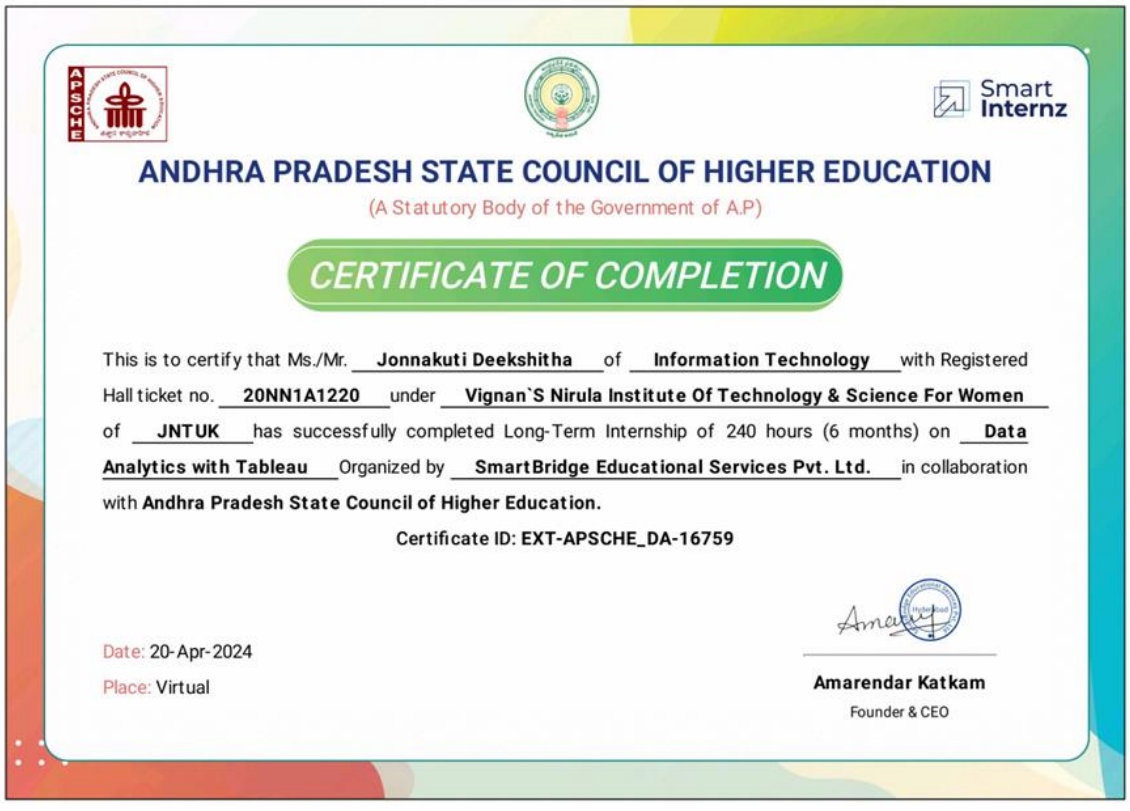
Certificate ID: **EXT-APSCHE_DA-16775**

Date: 20-Apr-2024

Place: Virtual

**Amarendar Katkam**

Founder & CEO



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Jonnakuti Deekshitha** of **Information Technology** with Registered Hall ticket no. **20NN1A1220** under **Vignan`S Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Data Analytics with Tableau** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

Certificate ID: **EXT-APSCHE_DA-16759**

Date: 20-Apr-2024

Place: Virtual

**Amarendar Katkam**

Founder & CEO

# ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

## CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Sakala Ramya Sri** of **Information Technology** with Registered Hall ticket no. **20NN1A1255** under **Vignan`S Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Data Analytics with Tableau** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

**Certificate ID: EXT-APSCHE_DA-16904**

Date: 21-Apr-2024

Place: Virtual

**Amarendar Katkam**

Founder & CEO

# ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

## CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Shaik Apsana** of **Information Technology** with Registered Hall ticket no. **20NN1A1256** under **Vignan`S Nirula Institute Of Technology & Science For Women** of **JNTUK** has successfully completed Long-Term Internship of 240 hours (6 months) on **Data Analytics with Tableau** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

**Certificate ID: EXT-APSCHE_DA-16910**

Date: 22-Apr-2024

Place: Virtual

**Amarendar Katkam**

Founder & CEO