A Project Report on

PHISHING SITES PREDICTOR

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

POLUPARTHI KAMALATHA	**(19P31A05A6)**
NALLI PRATHYUSHA	**(19P31A0598)**
GANAPAVARAPU POOJITHA	**(19P31A0584)**
VANAMADI RAVI SAI KUMAR	**(19P31A05B8)**

Under the esteemed supervision of

Ms.RAYAVARAPU SRI DIVYA, M.Tech

Assistant Professor

ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada

Accredited by NAAC (A+) and NBA

Surampalem, Kakinada District, Andhra Pradesh - 533 437

2019-2023

VISION

To induce higher planes of learning by imparting technical education with

- ✓ International standards
- ✓ Applied research
- ✓ Creative Ability
- ✓ Value based instruction and to emerge as a premiere institute

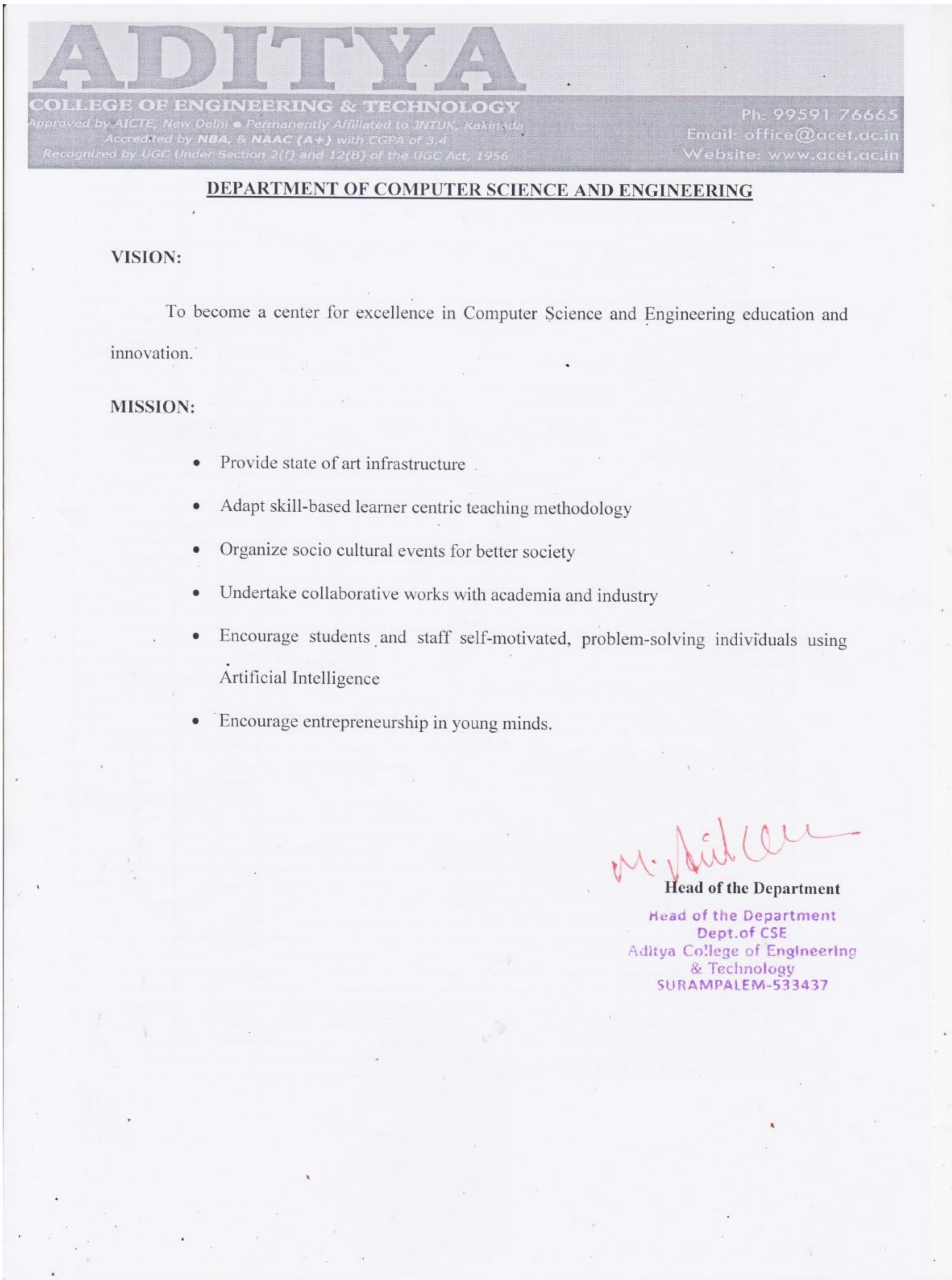
MISSION

Achieving academic excellence by providing globally acceptable technicaleducation by forecasting technology through

- ✓ Innovative Research And development
- ✓ Industry Institute Interaction
- ✓ Empowered Manpower

PRINCIPAL

PRINCIPAL
Aditya College of
Engineering & Technology
SURAMPALEM- 533 437



ADITYA

COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, New Delhi • Permanently Affiliated to JNTUK, Kakinada
Accredited by **NBA, & NAAC (A+)** with CGPA of 3.4
Recognized by UGC Under Section 2(f) and 12(B) of the UGC Act, 1956

Ph: 99591 76665
Email: office@acet.ac.in
Website: www.acet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION:

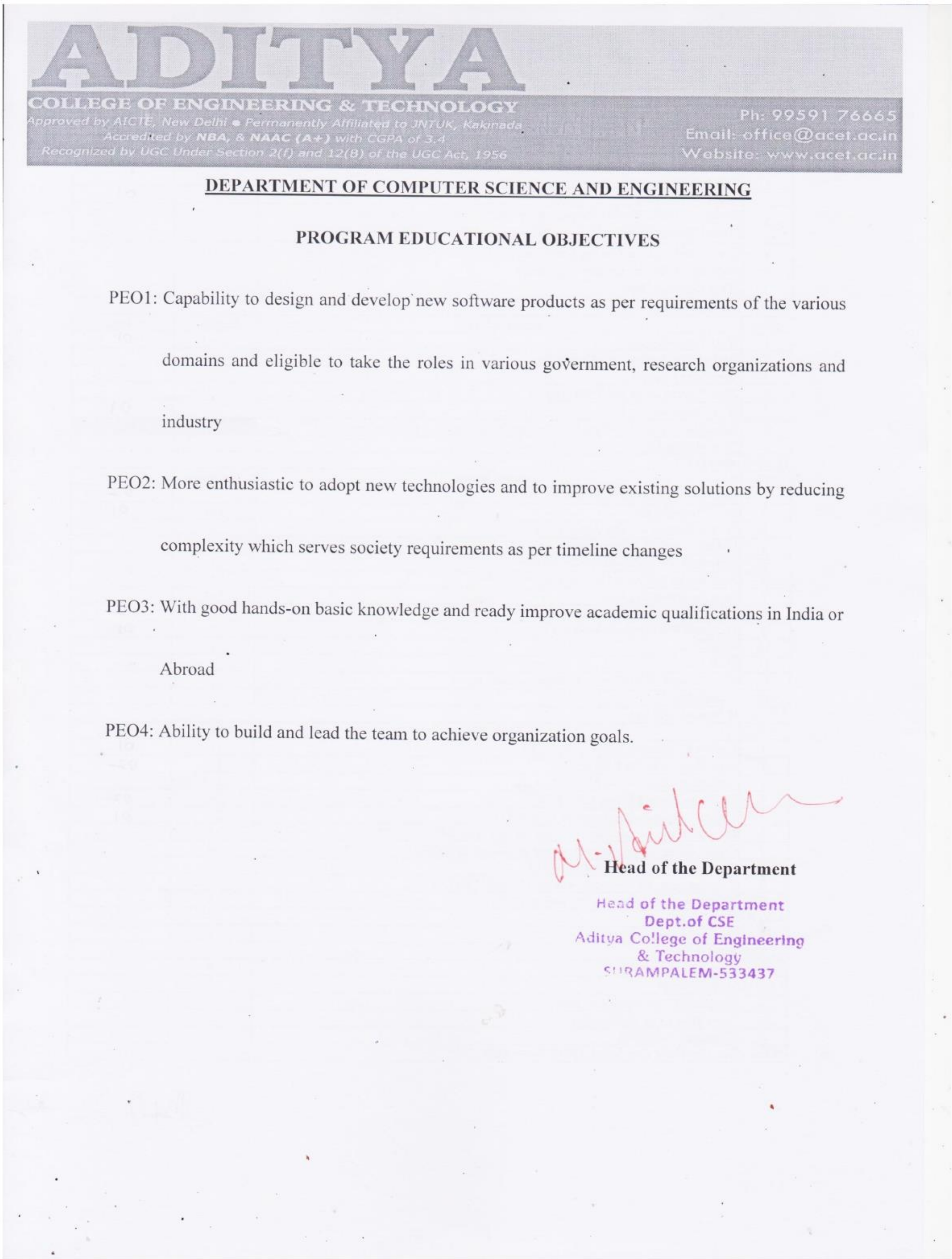
To become a center for excellence in Computer Science and Engineering education and innovation.

MISSION:

- Provide state of art infrastructure
- Adapt skill-based learner centric teaching methodology
- Organize socio cultural events for better society
- Undertake collaborative works with academia and industry
- Encourage students and staff self-motivated, problem-solving individuals using Artificial Intelligence
- Encourage entrepreneurship in young minds.

Head of the Department

Head of the Department
Dept.of CSE
Aditya College of Engineering
& Technology
SURAMPALEM-533437



ADITYA
COLLEGE OF ENGINEERING & TECHNOLOGY
Approved by AICTE, New Delhi • Permanently Affiliated to JNTUK, Kakinada
Accredited by NBA, & NAAC (A+) with CGPA of 3.4
Recognized by UGC Under Section 2(f) and 12(B) of the UGC Act, 1956

Ph: 99591 76665
Email: office@acet.ac.in
Website: www.acet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: Capability to design and develop new software products as per requirements of the various domains and eligible to take the roles in various government, research organizations and industry

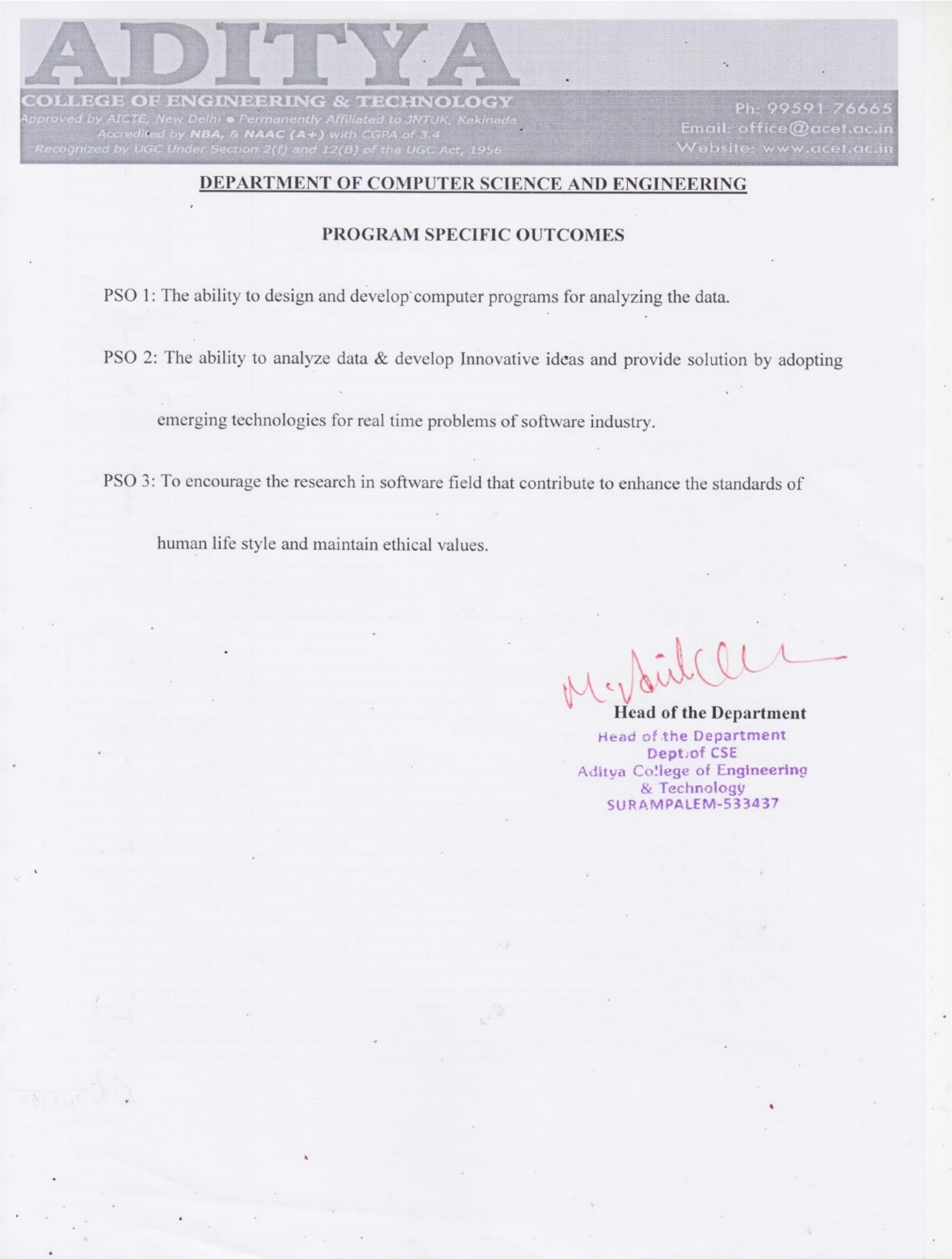
PEO2: More enthusiastic to adopt new technologies and to improve existing solutions by reducing complexity which serves society requirements as per timeline changes

PEO3: With good hands-on basic knowledge and ready improve academic qualifications in India or Abroad

PEO4: Ability to build and lead the team to achieve organization goals.

Head of the Department

Head of the Department
Dept.of CSE
Aditya College of Engineering
& Technology
SURAMPALEM-533437



ADITYA
COLLEGE OF ENGINEERING & TECHNOLOGY
Approved by AICTE, New Delhi • Permanently Affiliated to JNTUK, Kakinada
Accredited by **NBA**, & **NAAC (A+)** with CGPA of 3.4
Recognized by UGC Under Section 2(f) and 12(B) of the UGC Act, 1956

Ph: 99591 76665
Email: office@acet.ac.in
Website: www.acet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM SPECIFIC OUTCOMES

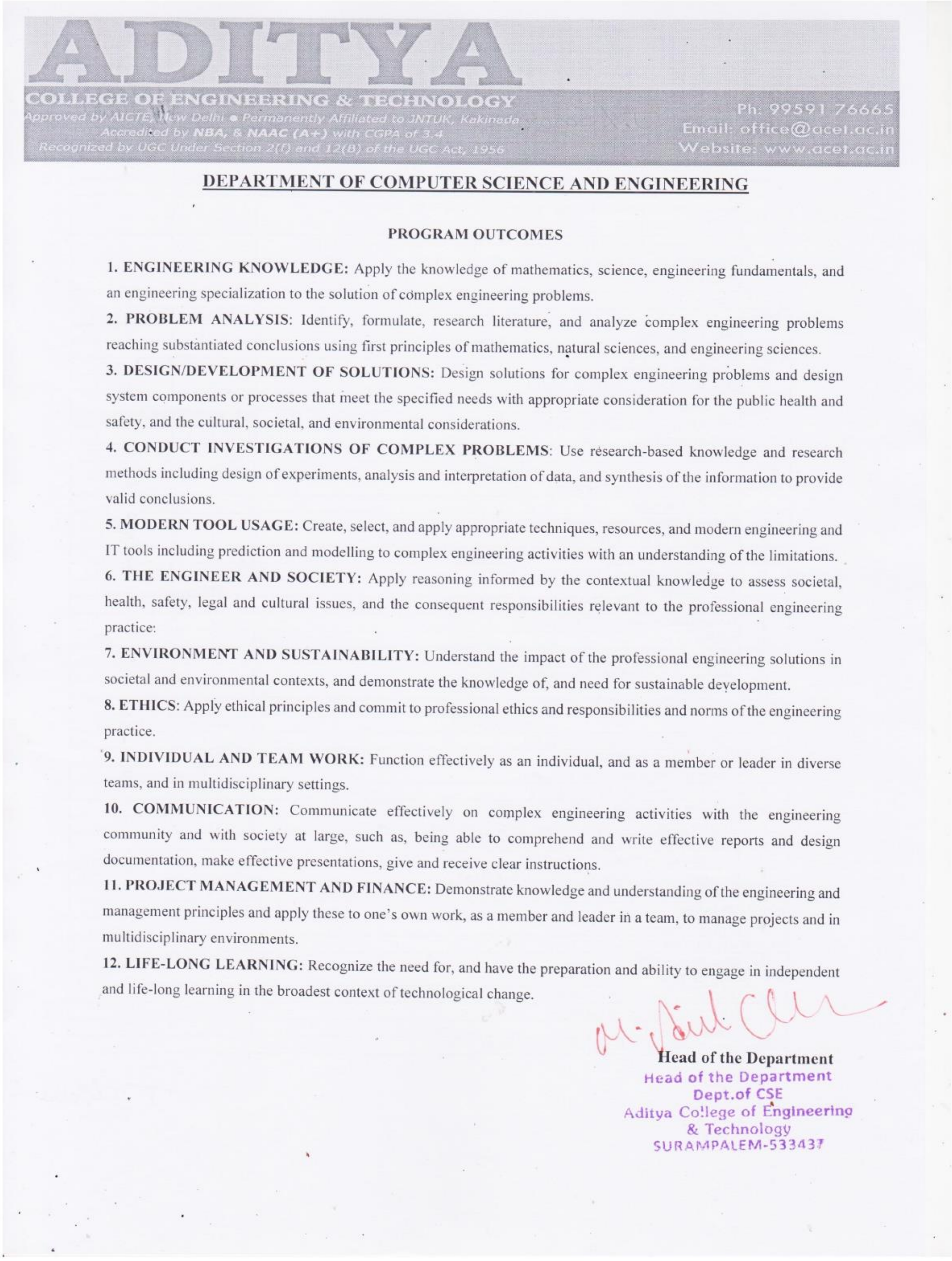
PSO 1: The ability to design and develop computer programs for analyzing the data.

PSO 2: The ability to analyze data & develop Innovative ideas and provide solution by adopting emerging technologies for real time problems of software industry.

PSO 3: To encourage the research in software field that contribute to enhance the standards of human life style and maintain ethical values.

Head of the Department

Head of the Department
Dept. of CSE
Aditya College of Engineering
& Technology
SURAMPALEM-533437



ADITYA
COLLEGE OF ENGINEERING & TECHNOLOGY
Approved by AICTE, New Delhi • Permanently Affiliated to JNTUK, Kakinada
Accredited by NBA, & NAAC (A+) with CGPA of 3.4
Recognized by UGC Under Section 2(f) and 12(B) of the UGC Act, 1956

Ph: 99591 76665
Email: office@acet.ac.in
Website: www.acet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM OUTCOMES

1. ENGINEERING KNOWLEDGE: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. PROBLEM ANALYSIS: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. DESIGN/DEVELOPMENT OF SOLUTIONS: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. CONDUCT INVESTIGATIONS OF COMPLEX PROBLEMS: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. MODERN TOOL USAGE: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. THE ENGINEER AND SOCIETY: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues, and the consequent responsibilities relevant to the professional engineering practice:

7. ENVIRONMENT AND SUSTAINABILITY: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. ETHICS: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. INDIVIDUAL AND TEAM WORK: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. COMMUNICATION: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.

11. PROJECT MANAGEMENT AND FINANCE: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. LIFE-LONG LEARNING: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Head of the Department

Head of the Department
Dept.of CSE
Aditya College of Engineering
& Technology
SURAMPALEM-533437

ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada

Accredited by NAAC (A+) and NBA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled, "**PHISHING SITES PREDICTOR**", is a bonafide work carried out by **POLUPARTHI KAMALATHA (19P31A05A6), NALLI PRATHYUSHA (19P31A0598),GANAPAVARAPU POOJITHA (19P31A0584), VANAMADI RAVI SAI KUMAR (19P31A05B8)**, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from **ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY**, Surampalem, during the academic year 2022-2023.

This project work has not been submitted in full or part to any other University or educational institute for the award of any degree or diploma.

PROJECT SUPERVISOR
Ms.Rayavarapu Sri Divya,
M.Tech
Assistant Professor

HEAD OF THE DEPARTMENT
Dr.M.Anil Kumar, M.Tech., Ph.D.
Professor

EXTERNAL EXAMINER

DECLARATION

We hereby declare that this project entitled "**PHISHING SITES PREDICTOR**" has been undertaken by us and this work has been submitted to **ADITYA COLLEGE OF ENGINEERING & TECHONOLOGY,** Surampalem affiliated to JNTUK, Kakinada, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING.**

We further declare that this project work has not been submitted in full or part to any other University or educational institute for the award of any degree or diploma.

PROJECT ASSOCIATES

POLUPARTHI KAMALATHA	**(19P31A05A6)**
NALLI PRATHYUSHA	**(19P31A0598)**
GANAPAVARAPU POOJITHA	**(19P31A0584)**
VANAMADI RAVI SAI KUMAR	**(19P31A05B8)**

ACKNOWLEDGEMENT

It is with immense pleasure that we would like to express our indebted gratitude to my **project supervisor**, **Ms.RAYAVARAPU SRI DIVYA, M.Tech** who has guided us a lot and encouraged us in every step of project work, his valuable moral support and guidance has been helpful in successful completion of this Project.

We wish to express our sincere thanks to **Dr.M.ANIL KUMAR M.Tech.,Ph.D., Head of the Department of CSE**, for his valuable guidance given to us throughout the period of the project work.

We feel elated to thank **Principal**, **Dr.DOLA SANJAY S M.Tech.,Ph.D.**, of Aditya College of Engineering and Technology for his cooperation in completion of our project and throughout our course.

We feel elated to thank **Dr.A.RAMA KRISHNA M.Tech.,Ph.D.**, **Dean (Academics & Administration)** of Aditya College of Engineering and Technology for his cooperation in completion of our project work.

We wish to express our sincere thanks to all faculty members, and lab programmers for their valuable guidance given to us throughout the period of the project.

We avail this opportunity to express our deep sense and heart full thanks to the **Management** of **Aditya College of Engineering & Technology** for providing a great support for us by arranging the trainees, and facilities needed to complete our project and for giving us the opportunity for doing this work.

PROJECT ASSOCIATES

POLUPARTHI KAMALATHA	**(19P31A05A6)**
NALLI PRATHYUSHA	**(19P31A0598)**
GANAPAVARAPU POOJITHA	**(19P31A0584)**
VANAMADI RAVI SAI KUMAR	**(19P31A05B8)**

ABSTRACT

With the rapid growth of technology, many people are currently working remotely and consume social media more often. Over the last few years, the Web has seen a massive growth in the number and kinds of web services. Web facilities such as online banking, gaming, and social networking have promptly evolved as people perform routine tasks. As a result, a large amount of information is uploaded on a daily basis to the Web. As these web services drive increased opportunities for people to interact, they equally offer new opportunities for criminals. URLs are launch pads for any web attacks such that any malicious intention user can steal the identity of the authorized person by sending the malicious URL. Malicious URLs are a keystone of Internet illegitimate activities. It is a form of cyber-attack, which has an adverse effect on people where the user is directed to fake websites and duped to reveal their sensitive and personal information which includes passwords of accounts, bank details, atm pin-card details etc. Hence protecting sensitive information from malwares or web phishing is difficult Aim of the phishers is to acquire critical information like username, password and bank account details. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. Hence protecting sensitive information from malwares or web phishing is difficult. By using Machine learning algorithms we will be identifying phishing attacks and report their positives and negatives. The proposed approach is that classifies URLs automatically by using Machine learning algorithm called logistic regression that uses binary classification.

INDEX

LIST OF FIGURES

LIST OF TABLES

1.INTRODUCTION

Phishing is one of the most challenging security problems faced by the world today, in part due to the large number of online transactions that take place daily. It refers to the practice of trying to obtain sensitive information, like user names, passwords and credit card details for malicious reasons by mimicking a trustworthy entity, like a well-knownand trusted website. It can be carried out by email spoofing, messaging, and generally appears to be from socialnetworking websites, auction sites as well as online pay-ment processing websites. Phishing websites deceive users,and exploit weaknesses of web security technologies.

Phishing site prediction is the process of identifying websites or online platforms that attempt to steal sensitive information such as usernames, passwords, and credit card details from unsuspecting users. These websites often mimic legitimate sites, such as banking or social media sites, and use various methods to trick users into providing their confidential information. Phishing attacks have become increasingly sophisticated over the years, with scammers using a variety of tactics to deceive users. As a result, detecting and preventing phishing attacks has become a critical task for individuals, businesses, and organizations alike.

Phishing site prediction involves using various tools and techniques to identify potential phishing websites before they can cause any harm. These tools often rely on machine learning algorithms to analyze the characteristics of known phishing sites and identify patterns that may indicate the presence of a phishing site. Some common features that can be used to identify phishing sites include suspicious domain names, incorrect or inconsistent SSL certificates, and unusual URLs.

By predicting potential phishing sites, organizations can take proactive measures to prevent their users from falling victim to these attacks. This can include implementing stronger authentication measures, providing user education and awareness training, and using advanced security solutions to detect and block phishing attempts. A Machine Learning Approach to Phishing Detection and Defence financial losses or valuables.

In this project, logistic regression which is a supervised machine learning model will betrained and tested with URL dataset to show that it has a hight detection accuracy rate and alow false negative rate. At the end of the study, the model will be used to deploy an app thatwill be used to predict benign URLs from good ones using FastAPI.

1.1 EXISTING SYSTEM

Phishing attacks have misled a lot of users by impersonating legitimate websites and stealing private information and/or financial data (Afroz and Greenstadt, 2011). To protect users against phishing, various anti-phishing techniques have been proposed that following different strategies.

- Content filtering

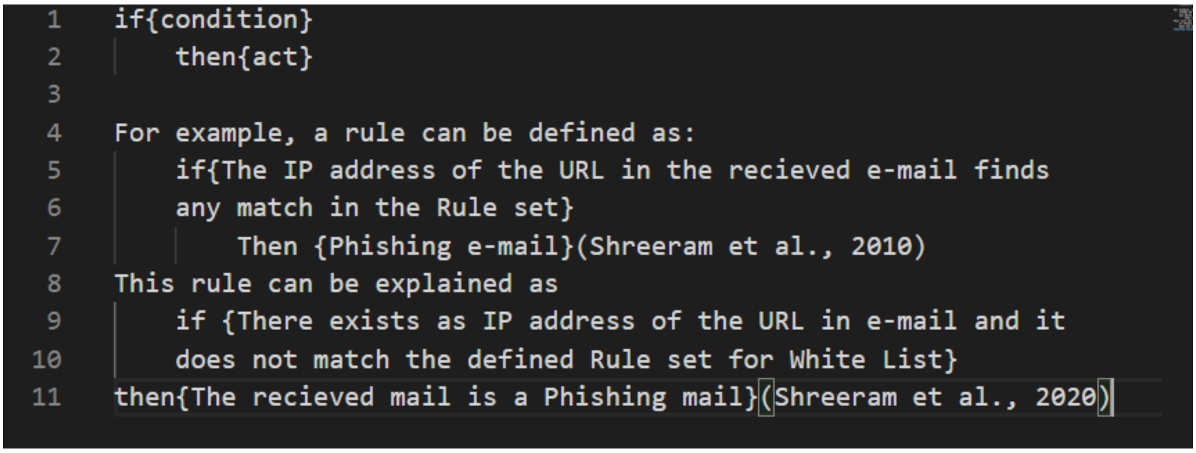
In this methodology, content/email is filtered as it enters in the victim's mail box by means of machine learning methods, such as Support Vector Machines (SVM) or Bayesian Additive Regression Trees (BART) (Tout and Hafner, 2009).

- Blacklisting

Blacklist is collection of recognized phishing Websites/addresses published by dependable entities like Google's and Microsoft's blacklist. It involves both a client and a server component. The client component is employed as either an email or browser plug-in that relates with a server component, which in this case is a public website that make available a list of identified phishing sites (Tout and Hafner, 2009).

- Genetic Algorithm-Based Anti-Phishing Technique

It is an approach that uses genetic algorithm for phishing web pages' detection. Genetic algorithms can be used to develop simple rules for preventing phishing attacks. These rules are used to discern normal web-site from anomalous website. These anomalous websites denote events with probability of phishing attacks. The rules stored in the rule base are typically in the following form.



```
if{condition}
    then{act}

For example, a rule can be defined as:
    if{The IP address of the URL in the recieved e-mail finds
    any match in the Rule set}
        Then {Phishing e-mail}(Shreeram et al., 2010)
This rule can be explained as
    if {There exists as IP address of the URL in e-mail and it
    does not match the defined Rule set for White List}
then{The recieved mail is a Phishing mail}(Shreeram et al., 2020)
```

Figure 1.1: Rules stored in rules based.

The main advantage is that it provides the feature of malicious status notification before the user reads the mail. It also provides malicious web-link detection in addition of phishing detection.

The disadvantage of this technique is more to its complex algorithms; single rule for phishing detection like in case of URL is far from enough, so we need multiple rule set for only one type of URL based phishing detection. Likewise, for another parameter we need to write other rule which may lead to more complex algorithm.

1.2 SCOPE OF THE EXISTING SYSTEM

Phishing detection techniques do suffer low detection accuracy and high false alarmespecially when novel phishing approaches are introduced. Besides, the most commontechnique used, blacklist-based method is inefficient in responding to emanating phishingattacks since registering new domain has become easier, no comprehensive blacklist canensure a perfect up-to-date database. Furthermore, page content inspection has been used bysome strategies to overcome the false negative problems and complement the vulnerabilities of the stale lists. Therefore, ensemble can be seen to be a better solution as it can combine the similarity in accuracy and different error-detection rate properties in selected algorithms

1.3 PROPOSED SYSTEM

Here we proposed a new method of anti-phishing technology. The Anti-Phishing Technology using Machine Learning Approach is a mechanism that is proposed in order to ensure high security. In this mechanism we deal with the URLs (Uniform Resource Locaters) and the URI (Uniform Resource Identifies) check with machine learning technique and predict whether it is phishing website or not. Here we create a web app for browsing imputed URLs. Each time we browse a site the corresponding URL (Uniform Resource Locater) of site will be checked with machine learning technique. Phishing site detection is truly an unpredictable and element issue including numerous components and criteria that are not stable. On account of the last and in addition ambiguities in arranging sites because of the intelligent procedures programmers are utilizing, some keen proactive strategies can be helpful and powerful tools can be utilized, for example, fuzzy, neural system and data mining methods can be a successful mechanism in distinguishing phishing sites. We applied logistic regression algorithms to model our train out model and at the end logistic regression which gave a more accurate prediction was used in our system.

1.4 NOVELTY OF PROPOSED SYSTEM

Phishing site prediction using logistic regression is a novel approach to identifying and preventing phishing attacks. While there are various methods for detecting phishing sites, such as blacklisting or heuristic analysis, logistic regression offers several advantages in terms of accuracy, speed, and flexibility.

One of the key novelties of using logistic regression for phishing site prediction is that it can effectively handle large datasets with numerous variables. Logistic regression can analyze a large number of variables, such as website domain names, IP addresses,, to accurately predict the likelihood of a phishing attack.

Another novelty of logistic regression is its ability to adapt to changing patterns and trends in phishing attacks. By continually analyzing new data and adjusting the model accordingly, logistic regression can stay up-to-date and provide accurate predictions even as phishing techniques evolve and become more sophisticated. Additionally, logistic regression is a transparent and interpretable method, which allows security experts to understand and explain how the model is making its predictions. This transparency can help build trust in the system and allow for more effective collaboration between security teams and other stakeholders. Overall, the novelty of using logistic regression for phishing site prediction lies in its ability to accurately analyze large and complex datasets, adapt to changing trends, and provide transparent and interpretable results. By leveraging these advantages, logistic regression can help prevent phishing attacks and improve the overall security of computer systems.

Here we proposed a new method of anti-phishing technology. The Anti-Phishing Technology using Machine Learning Approach i.e., Logistic regression and MultinomialNB to ensure best prediction. In this mechanism we deal with the URLs (Uniform Resource Locaters) technique and predict whether it is phishing website or not. Here we create a web API. Each time we browse a site the corresponding URL (Uniform Resource Locater) of site will be checked with machine learning technique. Implementation of this project is done by creating an app using python programming language and FASTAPI.

- The phishing dataset is collected from kaggle .
- First, the dataset is divided into two set which are then used to train and test our logistic algorithm model.
- The performance metrics of the reference algorithms based on precision, recall, f1-score and accuracy of our algorithm is analysed.
- Implementation of this project is done by creating an app using python programming language and FASTAPI.

2. REQUIREMENT ANALYSIS

2.1 FUNCTIONAL REQUIREMENTS

The workflow of a machine project which includes all the steps required to build the proper machine learning project from scratch. We will also go over data pre-processing, data cleaning, feature exploration and various methods show the impact that it has on Machine Learning Model Performance. It also covers a couple of the pre-modelling steps that can help to improve the model.

1 . Data Collection:

The system must be capable of collecting data related to the website's URL, domain name, information that can help in identifying a phishing site.

2 . Data Preprocessing:

The collected data must be preprocessed to extract the relevant features and eliminate any redundant or irrelevant information.

3 . Feature Selection:

The system must select the most relevant features that can be used for predicting whether a website is a phishing site or not.

4 . Model Building:

The logistic regression model must be built using the preprocessed data and selected features.

Logistic Regression:

Logistic Regression is a classification model that is used when the dependent variable(output) is in the binary format such as 0 (False) or 1 (True). This makes logistic regression agood algorithm fit for the purpose of our work in predicting if a URL is a phishing URL (1)or not (0).

Logistic Regression has an S-shaped curve and can take values between 0 and 1 but never exactly at those limits.

And it looks like this;

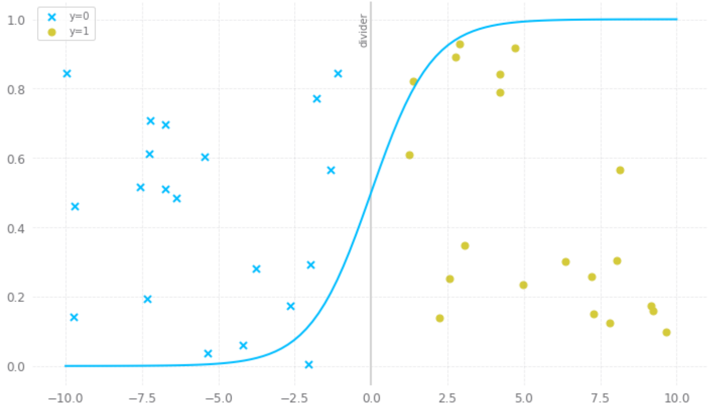


Figure 2.1: logistic Regression curve

5 . Model Training:

The logistic regression model must be trained on a labeled dataset of phishing and legitimate websites.

6 . Model Testing:

The system must test the performance of the logistic regression model on a separate dataset of phishing and legitimate websites to evaluate its accuracy, precision, recall, and F1 score

7 . Model Deployment:

The trained logistic regression model is deployed using FastAPI to predict phishing websites in real-time.

Prediction_app.py: This file will contain the code which will start the uvicorn server. It will also contain the code to serve a request and return a response asynchronously. This file will also create an instance of FastAPI () which is the main point of interaction to create the API.

Phishing.pkl: This file contains the dump of the machine learning model with function whichwill be called when our program is running to predict malicious URL.

let’s create the following folder structure to understand Deployment.

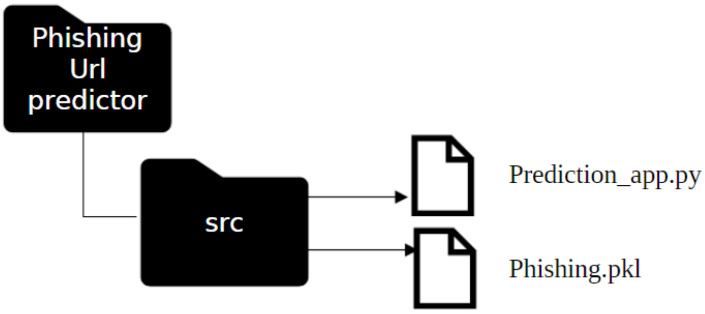


Figure 2.2: Deployment model

2.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements refer to the aspects of a system or software application that do not directly relate to its functional capabilities or features. These requirements are often related to the quality attributes or characteristics of the system, such as performance, scalability, reliability, usability, security, and maintainability. Non-functional requirements are just as important as functional requirements because they define the overall quality and usability of the system.

2.2.1 USER INTERFACE AND HUMAN FACTORS

When using logistic regression to predict phishing attacks, it is important to consider the human factors involved in the attack. This includes factors such as the user's level of awareness and understanding of phishing attacks, their susceptibility to social engineering tactics, and their ability to identify suspicious emails or messages. Below diagram shows how the user interface looks like

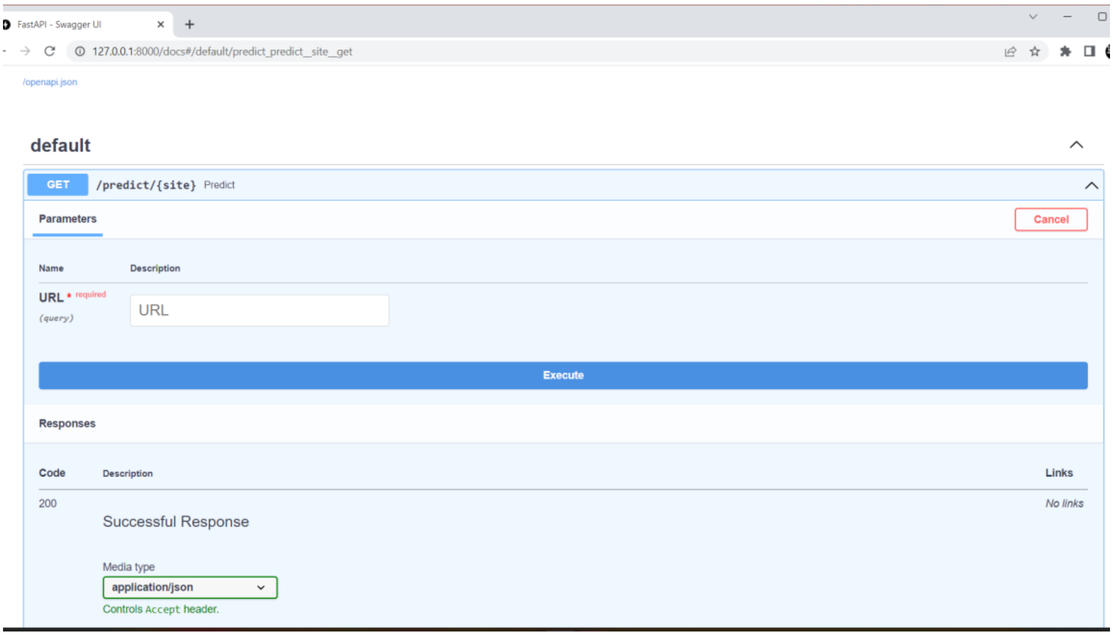


Figure 2.3: User Interface

2.2.2 SOFTWARE REQUIREMENTS

- Python (Jupiter notebook)

  Libraries used

- RegexpTokenizer:

  A tokenizer that splits a string using a regular expression, which matches either the tokens or the separators between tokens.

- Pandas:

  Pandas library is used for data manuplation and analysis

- Logistic Regression:

  Algorithm used to predict good or bad.

- SnowballStemmer:

  It is the process of reducing the word to its word stem that affixes to suffixes and prefixes or to roots of words known as a lemma.

- Count vectorizer:

  Create sparse matrix of words using regexptokenizes

- Pickle:

  Pickle library is used to dump models.

- FastAPI

Fast API is a high performance web framework. It is based on standard Python type hints. Not only is FastAPI intuitive and straight-forward to use in projects, but the FastAPI code alsohas 100% test coverage .

2.2.3 HARDWARE REQUIREMENTS

- RAM (8GB)
- Windows 10 operating system
- Processor(Intel core i5)
- 256 GB HDD (Minimum)

2.2.4 USABILITY

Usability in the context of phishing site prediction refers to how well the prediction system performs its task and how easy it is to use. To ensure the usability of a phishing site prediction system, the following factors should be considered:

1. Accuracy: The accuracy of the prediction system is a crucial factor in its usability. The system should be able to accurately identify and classify phishing sites. The accuracy can be measured using metrics such as precision, recall, and F1-score.

2. Speed: The prediction system should be able to classify phishing sites quickly to ensure that users are protected in real-time. The speed of the system can be improved by optimizing the algorithms used to classify sites.

3. User-friendly interface: The prediction system should have a user-friendly interface that is easy to use for non-technical users. The interface should provide clear instructions on how to use the system, and the results of the classification should be presented in an easily understandable format.

4. Continuously updated: The prediction system should be continuously updated to ensure that it can detect new and emerging phishing sites. The system should also be regularly maintained to ensure that it is functioning properly.

5. Integration: The prediction system should be easily integrated with existing security solutions to provide a comprehensive defense against phishing attacks. This integration can be achieved through APIs or other integration methods.

Overall, the usability of a phishing site prediction system can be improved by focusing on accuracy, speed, a user-friendly interface, continuous updating, and integration with other security solutions. By considering these factors, the prediction system can be designed to be effective and easy to use, helping to protect users from phishing attacks.

2.2.5 RELIABILITY

Reliability is an important factor when it comes to phishing site prediction. A reliable phishing site prediction system is one that can accurately identify and classify phishing sites with a high degree of certainty. By ensuring that the system's dataset, features, algorithms, testing, and monitoring are all reliable, the prediction system can accurately identify and classify phishing sites with a high degree of certainty.Logistic Regression model gives 96 % Accuracy.

2.2.6 PERFORMANCE

The performance of a phishing site prediction system based on logistic regression can be evaluated using a combination of the metrics i.e, Accuracy, Precision, Recall and F1-Score. The performance can be improved by optimizing the feature selection, dataset quality, and model hyperparameters. The system can also be continuously updated to adapt to new and emerging types of phishing attacks**.**

2.2.7 SUPPORTABILITY

By using th that we are using in this project will predict the URL and determine it whether it is a phishing website or not. Supportability in the context of phishing site prediction using logistic regression refers to the system's ability to be maintained and updated over time. Supportability is important because the threat landscape for phishing attacks is constantly evolving, and the prediction system needs to be updated to stay effective.

2.2.8 PHYSICAL ENVIRONMENT

The physical environment can have an impact on the performance of a phishing prediction system using logistic regression. Phishing attack relies heavily on human interaction and often involves using psychological tricks aimed at making victims agree to things they would not have done normally. Phishing traditionally functions by mimicking an online website, which is carefully design to look like the genuine site.

2.2.9 SECURITY REQUIREMENTS

The security requirements for a phishing site prediction system using logistic regression are crucial to ensure the system's protection against unauthorized access, data breaches, and other security threats.Anti-spyware and firewall settings should be used to prevent phishing attacks and users should update the programs regularly. Firewall protection prevents access to malicious files by blocking the attacks. Antivirus software scans every file which comes through the internet to user computer.

2.2.10 RESOURCE REQUIREMENTS

The resource requirements for a phishing site prediction system using logistic regression depend on several factors, such as the size of the dataset, the complexity of the model, and the number of requests the system needs to handle. Here are some of the resource requirements to consider when designing and deploying a phishing site prediction system using logistic regression.They are like Hardware and Software Requirements.

3. SYSTEM ANALYSIS

3.1 INTRODUCTION

System analysis is a process of studying and understanding complex systems, such as software applications or technological systems, in order to identify and solve problems or improve their performance. Phishing is a type of cyber attack where criminals try to trick individuals into providing sensitive information, such as passwords or credit card numbers, by disguising themselves as a trustworthy entity through emails, websites or other means.

In order to predict and prevent phishing attacks, logistic regression can be used as a machine learning technique. Logistic regression is a statistical method that is used to predict the probability of a binary outcome, such as the occurrence of a phishing attack.

System analysis can be used to study the different variables that affect the likelihood of a phishing attack, such as the type of email, the language used, the source of the email, and the behavior of the recipient. By analyzing these variables and identifying patterns, it is possible to develop a logistic regression model that can accurately predict the likelihood of a phishing attack.

The model can be trained using historical data on phishing attacks and non-phishing emails, and then tested on new data to evaluate its accuracy. By integrating the logistic regression model into a larger system, such as an email filtering system or a web browser extension, it is possible to provide real-time protection against phishing attacks.

Overall, system analysis using logistic regression is a powerful tool for predicting and preventing phishing attacks, and can be used to improve the security of computer systems and protect against cybercrime

3.2 USE CASES

A use case represents the functionality provided by the system to the user. A use case is defined as “a set of actions performed by the system, which produces an observable result that is, typically, of some value to one or more actors or other stakeholders of the system”. The actions can include communicating with other actors or systems as well as performing calculations inside the system.

The characteristics of a use case are:

1. A use case is always initiated by an actor.
2. A use case provides value to an actor.
3. A use case is complete.

Use cases are connected to actors through associations, which are sometimes referred to as communication associations. Associations represent which actors the use case is communicating with. The association should always be binary, implying a dialog between the actor and system.

Graphical Representation of Use case be:

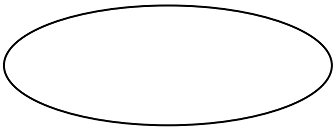


Figure 3.1: Use Case

3.2.1 ACTORS

In UML, an actor is a model element that interacts with a system. As a model element, it can be an abstract person or another external system. In this proposed Approach, Actors are user and system.

Actor can be represented as the following:

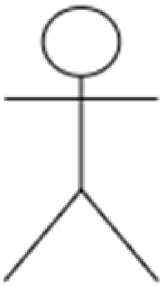


Figure 3.2 Actor

3.2.2 LIST OF USE CASES

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use case) that some system or systems(subject)should or can perform in collaboration with one or more external users of the system(actors). Each use case should provide some observable and valuable result to the actors of the system.

The following are the list of Use Cases:

1.Application:

Firstly, user opens Application to check a URL is phishing site or Legitmate site.

2.Input URL:

Next, user enters the URL to know the site is phishing or not.

3.Preprocessing:

Then System preprocess the URL entered by the user and checks missing values.

4.Feature Extraction:

Then System will Extract features by using tokenizer, snowball stemmer and uses count vectorizer to convert text into binary format.

5.Binary classifier:

The system uses Logistic Regression, i.e, Binary classification and classifies phishing sites or legitmate sites.

6.Predict output:

Finally,the System predicts whether a site is phishing or not to the user.

Graphical Representation are shown below:

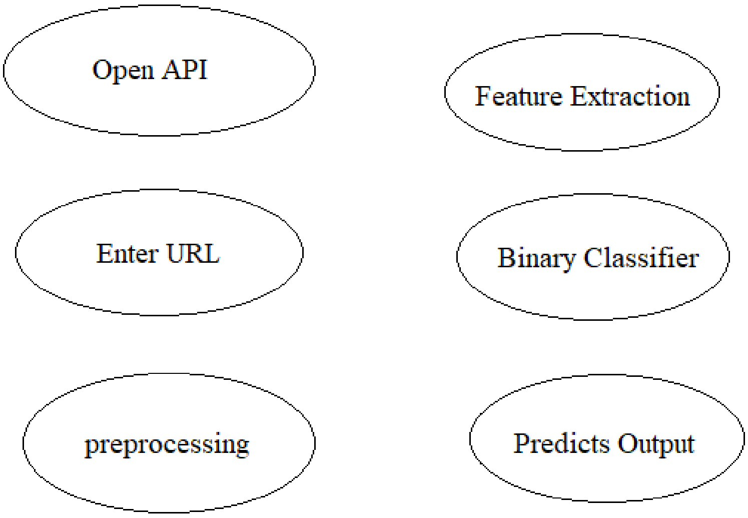


Figure 3.3: Use Cases

3.3.3 USE CASE DIAGRAMS

Use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process. Use-case diagrams are helpful in the following situations: Before starting a project, you can create use-case diagrams to model a business so that all participants in the project share an understanding of the workers, customers, and activities of the business. While gathering requirements, you can create use-case diagrams to capture the system requirements and to present to others what the system should do. During the analysis and design phases, you can use the use cases and actors from your use-case diagrams to identify the classes that the system requires. During the testing phase, you can use use-case diagrams to identify tests for the system.

Below Diagram explains Use Case Diagram for this project like,User opens API and enters URL then System preprocess the URL,nextly feature Extraction is done and then Logistic Regression classifies URL using Binary classification and finally predicts output.

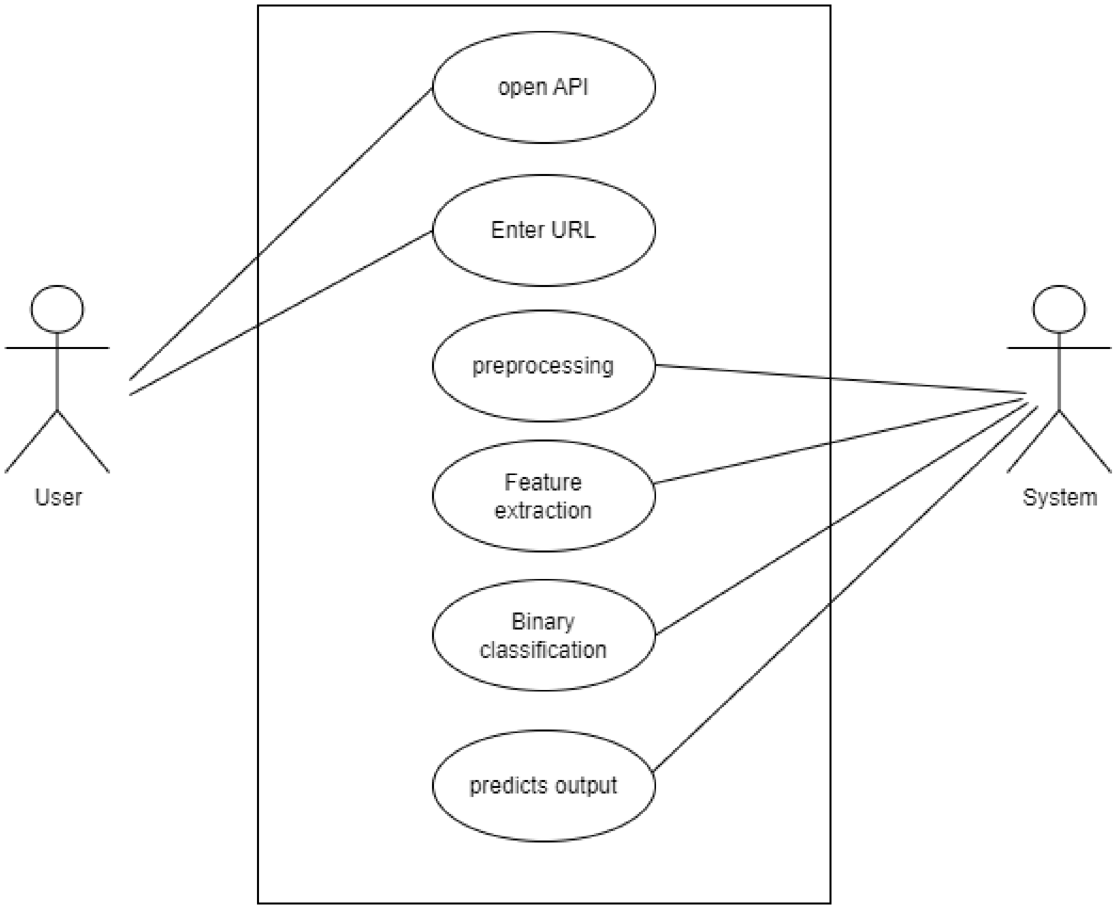


Figure 3.1 :Use Case Diagram

04. SYSTEM DESIGN

4.1 INTRODUCTION

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business organization.

4.2 SYSTEM ARCHITECTURE

A system architecture is a conceptual model that defines the structure, behavior and more views of the system. An architecture is a formal description and representation of a system organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture.

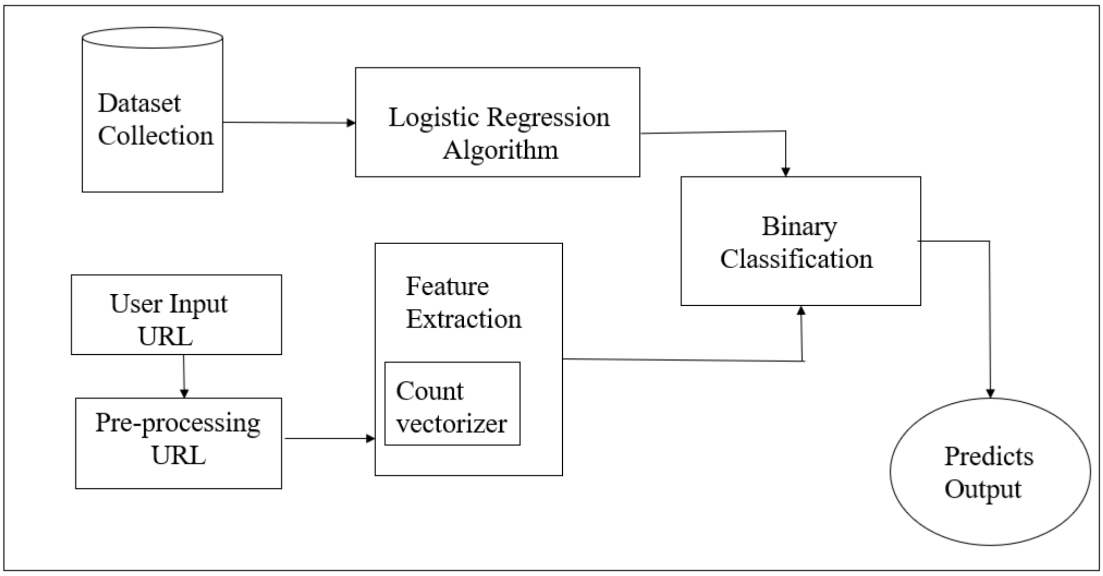


Figure 4.1:System Architecture

4.3 SYSTEM OBJECT MODEL

4.3.1 INTRODUCTION

SOM (System Object Model) defines an interface between programs, or between libraries and programs, so that an object's interface is separated from its implementation. SOM allows classes of objects to be defined in one programming language and used in another, and it allows libraries of such classes to be updated without requiring client code to be recompiled.

A SOM library consists of a set of classes, methods, static functions, and data members. Programs that use a SOM library can create objects of the types defined in the library, use the methods defined for an object type, and derive subclasses from SOM classes, even if the language of the program accessing the SOM library does not support class typing. A SOM library and the programs that use objects and methods of that library need not be written in the same programming language.

4.3.2 SUBSYSTEMS

In UML models, subsystems are a type of stereotyped component that represent independent, behavioral units in a system. Subsystems are used in class, component, and use-case diagrams to represent large-scale components in the system that you are modeling.

You can model an entire system as a hierarchy of subsystems. You can also define the behavior that each subsystem represents by specifying interfaces to the subsystems and the operations that support the interfaces.

4.4 OBJECT DESCRIPTION

4.4.1 OBJECTS

In UML models, objects are model elements that represent instances of a class or of classes. You can add objects to your model to represent concrete and prototypical instances. A concrete instance represents an actual person or thing in the real world. For example, a concrete instance of Customer class represents an actual customer. A prototypical instance of Customer class contains data that represents a typical customer.

A class represents an abstraction of a concept or of a physical thing, whereas an object represents a concrete entity.

4.4.2 CLASS DIAGRAMS

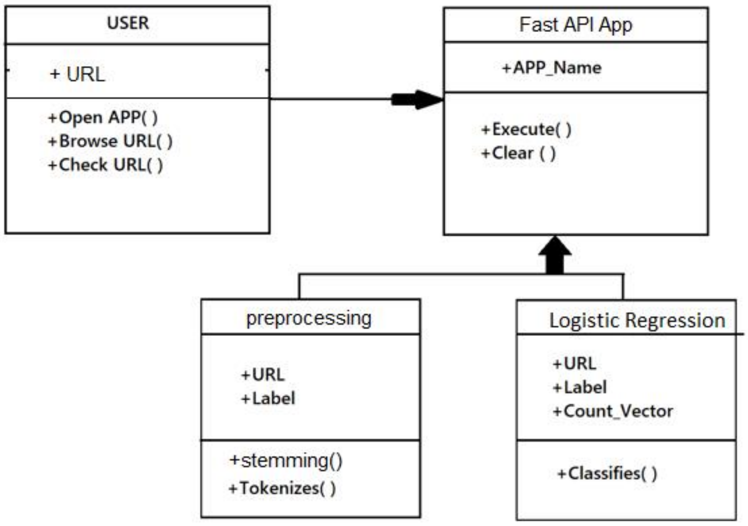
In software Engineering, A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's methods), and the relationships among objects translating the models into application.

The class diagram is made up of three sections:

Upper Section: The upper section encompasses the name of the class. A class is a representation of similar objects that shares the same relationships, attributes, operations, and semantics.

Middle Section: The middle section constitutes the attributes, which describe the quality of the class.

Lower Section: The lower section contain methods or operations. The methods are represented in the form of a list, where each method is written in a single line. It demonstrates how a class interacts with data.



.

Figure 4.2 :Class Diagram

4.5 DYNAMIC MODEL

The dynamic model is used to express and model the behaviour of the system over time. It includes support for activity diagrams,sequence diagrams and extensions including business process modelling.

4.5.1 SEQUENCE DIAGRAMS

A sequence diagram is a type of interaction diagram because it describes howand in what order a group of objects works together. The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

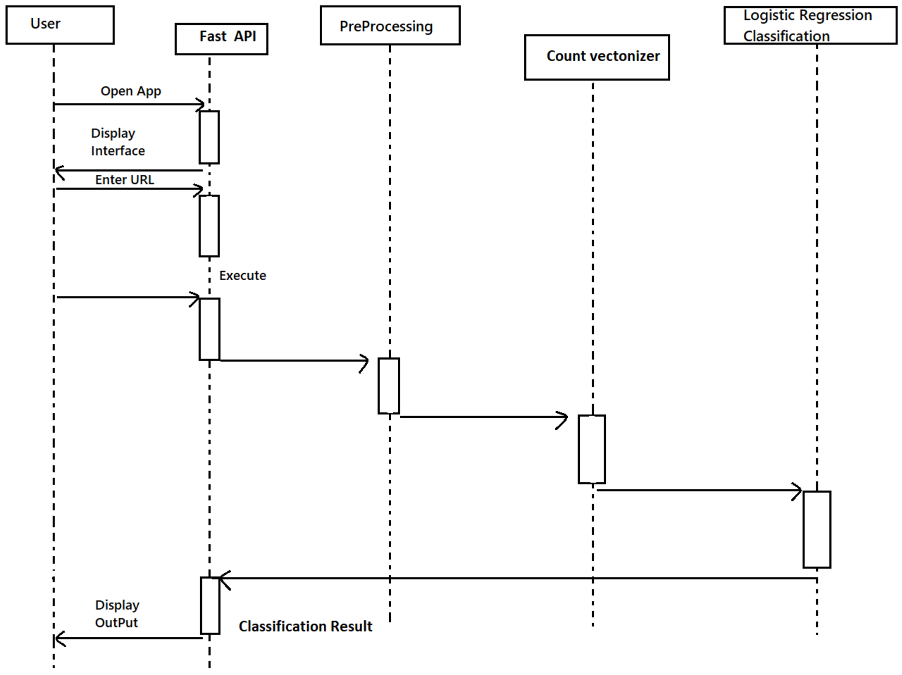


Figure 4.3 : Sequence Diagram

4.5.2 ACTIVITY DIAGRAM

An activity diagram is a behavioural diagram i.e., it depicts the flow of control during a system and ask the steps involved within the execution of a use case.They are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the uml, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

Activity diagram constitutes following notations:

1.Initial State: It depicts the initial stage or beginning of the set of actions.

2.Final State: It is the stage where all the control flows and object flows end.

3.Decision Box: It makes sure that the control flow or object flow will follow only one path. It is shown as Diamond shape.

4.Action Box: It represents the set of actions that are to be performed.It is represented as Rectangle shape.

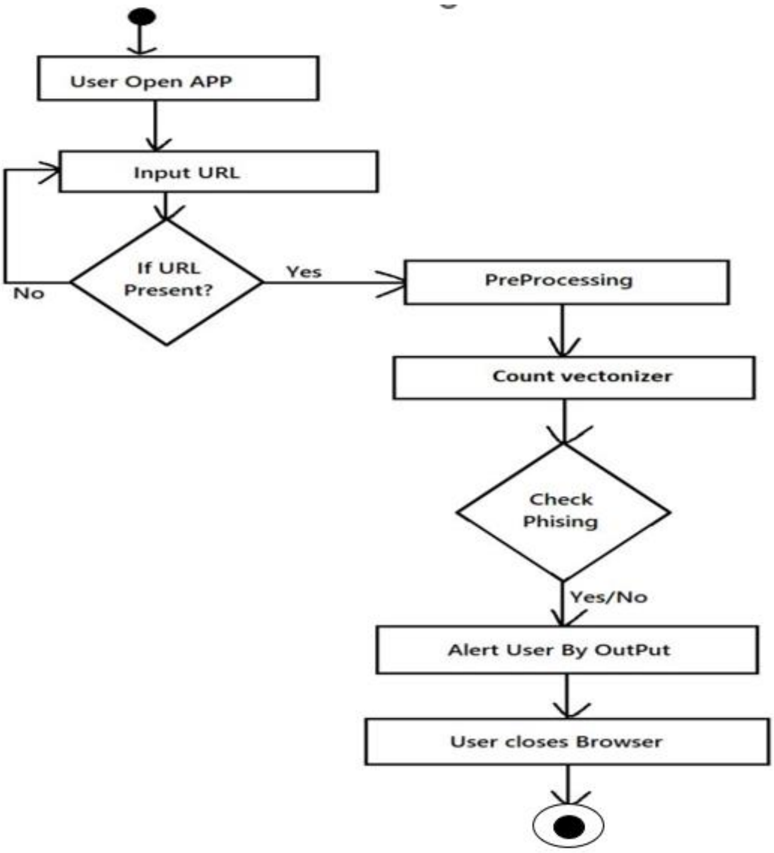


Figure 4.4 : Activity Diagram

4.6 STATIC MODEL

In particular, a static model defines the classes in the system, the attributes of the classes, the relationships between classes, and the operations of each class.

4.6.1 DEPLOYMENT DIAGRAM

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

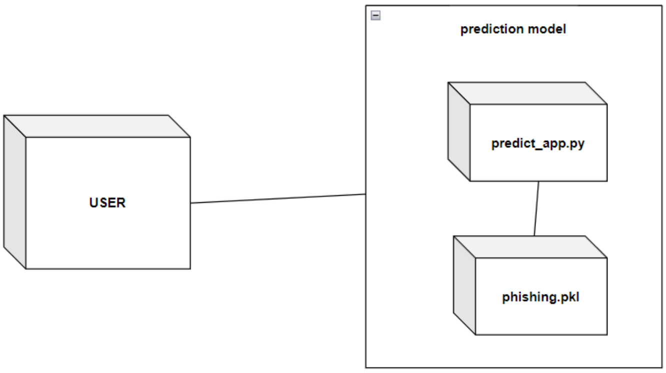


Figure 4.5:Deployment Diagram

From above diagram,

Prediction_app.py: This file will contain the code which will start the uvicorn server. It will also contain the code to serve a request and return a response asynchronously. This file will also create an instance of FastAPI () which is the main point of interaction to create the API.

Phishing.pkl: This file contains the dump of the machine learning model with function whichwill be called when our program is running to predict malicious URL.

let’s create the following folder structure to understand Deployment

5. IMPLEMENTATION

5.1 SOFTWARE USED

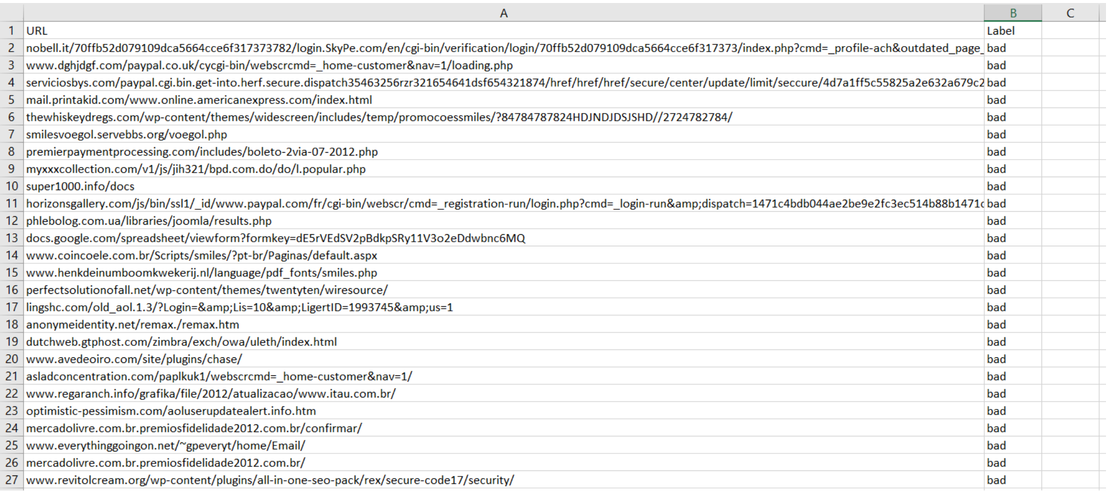
PyCharm is used in this project.Pycharm is a hybrid platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE.

We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

5.2 SOURCE CODE

5.2.1 DATA SET STRUCTURE:

The dataset used in this implementation is gotten from already processed data stored at https://www.kaggle.com/taruntiwarihp/phishing-site-urls. the output of the features extracted will be used as input in evaluating our models. Also, the dataset format was saved as a .CSVfile for better processing with python. This dataset if made of entries divided into two columns. Label column is prediction col which has 2 categories; Good - which means theURLs is not containing malicious stuff and this site is not a Phishing Site, Bad - which meansthe URLs contains malicious stuff and this site is a Phishing Site.



	A	B	C
1	URL	Label	
2	nobell.it/70ffb52d079109dca5664cce6f317373782/login.SkyPe.com/en/cgi-bin/verification/login/70ffb52d079109dca5664cce6f317373/index.php?cmd=_profile-ach&outdated_page_	bad	
3	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrcmd=_home-customer&nav=1/loading.php	bad	
4	serviciosbys.com/paypal.cgi.bin.get-into.herf.secure.dispatch35463256rzr321654641dsf654321874/href/href/href/secure/center/update/limit/seccure/4d7a1ff5c55825a2e632a679c2	bad	
5	mail.printakid.com/www.online.americanexpress.com/index.html	bad	
6	thewhiskeydregs.com/wp-content/themes/widescreen/includes/temp/promocoessmiles/?84784787824HDJNDJDSJSHD//2724782784/	bad	
7	smilesvoegol.servebbs.org/voegol.php	bad	
8	premierpaymentprocessing.com/includes/boleto-2via-07-2012.php	bad	
9	myxxxcollection.com/v1/js/jih321/bpd.com.do/do/l.popular.php	bad	
10	super1000.info/docs	bad	
11	horizonsgallery.com/js/bin/ssl1/_id/www.paypal.com/fr/cgi-bin/webscr/cmd=_registration-run/login.php?cmd=_login-run&amp;dispatch=1471c4bdb044ae2be9e2fc3ec514b88b1471c	bad	
12	phlebolog.com.ua/libraries/joomla/results.php	bad	
13	docs.google.com/spreadsheet/viewform?formkey=dE5rVEdSV2pBdkpSRy11V3o2eDdwbnc6MQ	bad	
14	www.coincoele.com.br/Scripts/smiles/?pt-br/Paginas/default.aspx	bad	
15	www.henkdeinumboomkwekerij.nl/language/pdf_fonts/smiles.php	bad	
16	perfectsolutionofall.net/wp-content/themes/twentyten/wiresource/	bad	
17	lingshc.com/old_aol.1.3/?Login=&amp;Lis=10&amp;LigertID=1993745&amp;us=1	bad	
18	anonymeidentity.net/remax./remax.htm	bad	
19	dutchweb.gtphost.com/zimbra/exch/owa/uleth/index.html	bad	
20	www.avedeoiro.com/site/plugins/chase/	bad	
21	asladconcentration.com/paplkuk1/webscrcmd=_home-customer&nav=1/	bad	
22	www.regaranch.info/grafika/file/2012/atualizacao/www.itau.com.br/	bad	
23	optimistic-pessimism.com/aoluserupdatealert.info.htm	bad	
24	mercadolivre.com.br.premiosfidelidade2012.com.br/confirmar/	bad	
25	www.everythinggoingon.net/~gpeveryt/home/Email/	bad	
26	mercadolivre.com.br.premiosfidelidade2012.com.br/	bad	
27	www.revitolcream.org/wp-content/plugins/all-in-one-seo-pack/rex/secure-code17/security/	bad	

Figure 5.1:Dataset Structure

5.2.2 SOURCE CODE FOR PREDICTION:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from nltk.tokenize import RegexpTokenizer
from nltk.stem.snowball import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
import warnings # ignores pink warnings
warnings.filterwarnings('ignore')
phish_data = pd.read_csv('phishing_site_urls.csv')
print("data loaded")
#phish_data.isnull().sum()
tokenizer = RegexpTokenizer(r'[A-Za-z]+')
print('Getting words tokenized ...')
phish_data['text_tokenized'] = phish_data.URL.map(lambda t: tokenizer.tokenize(t))
stemmer = SnowballStemmer("english") # choose a language
print('Getting words stemmed ...')
phish_data['text_stemmed'] = phish_data['text_tokenized'].map(lambda l:
[stemmer.stem(word) for word in l])
print('Getting joining words ...')
phish_data['text_sent'] = phish_data['text_stemmed'].map(lambda l: ' '.join(l))
cv = CountVectorizer()
feature = cv.fit_transform(phish_data.text_sent)
trainX, testX, trainY, testY = train_test_split(feature, phish_data.Label,
test_size=0.20)
lr = LogisticRegression()
lr.fit(trainX,trainY)
threshold = 0.5
print(lr.predict(testX))
Scores_ml = {}
Scores_ml['Logistic Regression'] = np.round(lr.score(testX,testY),2)
print('Training Accuracy :',lr.score(trainX,trainY))
print('Testing Accuracy :',lr.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(lr.predict(testX), testY), columns =
['Predicted:Bad', 'Predicted:Good'],index = ['Actual:Bad', 'Actual:Good'])
print('\nLogistic Regression -CLASSIFICATION REPORT\n')
print(classification_report(lr.predict(testX), testY, target_names =['Bad','Good']))
print("multi")
mnb = MultinomialNB()
mnb.fit(trainX,trainY)
Scores_ml['MultinomialNB'] = np.round(mnb.score(testX,testY),2)
print('Training Accuracy :',mnb.score(trainX,trainY))
print('Testing Accuracy :',mnb.score(testX,testY))
con_mat = pd.DataFrame(confusion_matrix(mnb.predict(testX), testY),columns =
['Predicted:Bad', 'Predicted:Good'],index = ['Actual:Bad', 'Actual:Good'])
print('\nMultinomialNB-CLASSIFICATION REPORT\n')
print(classification_report(mnb.predict(testX), testY,target_names =['Bad','Good']))
```

5.2.3 EVALUATION METRICS:

```
Logistic Regression -CLASSIFICATION REPORT

              precision   recall    f1-score    support

         Bad      0.91      0.97       0.94        29589
        Good      0.99      0.97       0.98        80281

    accuracy                           0.97       109870
   macro avg      0.95      0.97       0.96       109870
weighted avg      0.97      0.97       0.97       109870


MultinomialNB-CLASSIFICATION REPORT

              precision   recall    f1-score   support

         Bad      0.92      0.94       0.93       30892
        Good      0.97      0.97       0.97       78978

    accuracy                           0.96       109870
   macro avg      0.95      0.95       0.95       109870
weighted avg      0.96      0.96       0.96       109870
```

By comparing both algorithms Logistic Regression has better Accuracy when compared to Multinomial Naïve Bayes. From above confusion matrix clearly precision, recall, f1-score are better for logistic regression when compared to MultinomialNB. So, Logistic Regression is used for phishing sites prediction in this project.

Precision:

precession is about being precise, i.e., how accurate your mode is. In other words,you can say, when a model makes a prediction, how often it is correct. In our case logisticregression predicts a URL is going to be a phishing URL have 91% of the times and it’sgoing to be a good site have 99%.

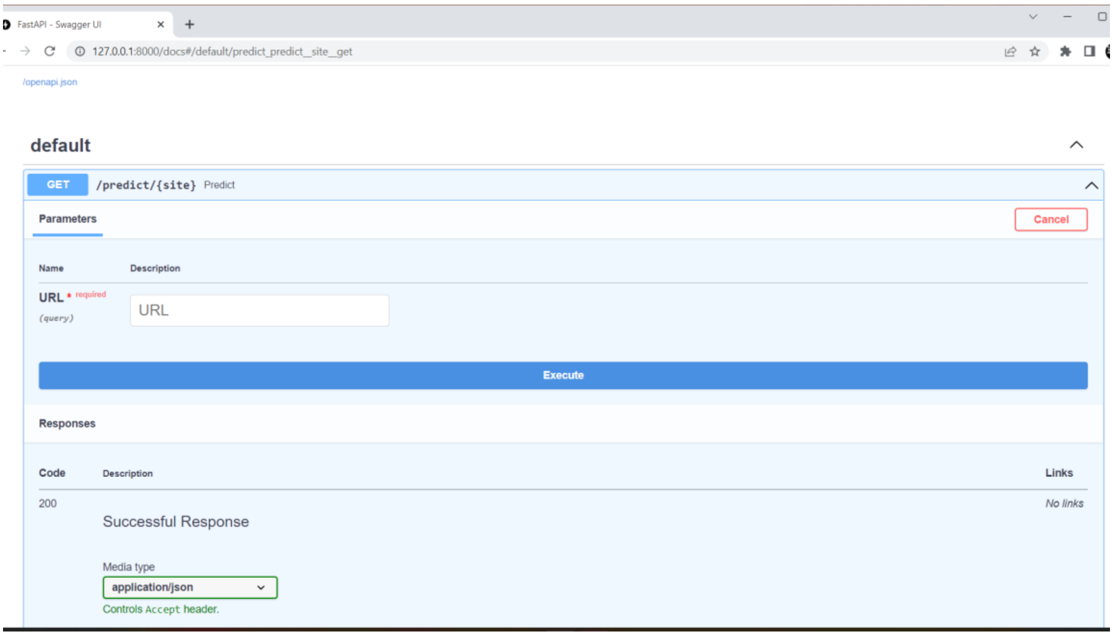
Recall:

It there are phishing URLs in the test set and logistic regression model can identify it 97% and good URLs will be identifying 96% of the time.

5.2.4 SOURCE CODE FOR API:

```
import uvicorn
from fastapi import FastAPI
import joblib,os

app = FastAPI()

#pkl
phish_model = open('phishing_k.pkl','rb')
phish_model_ls = joblib.load(phish_model)

# ML Aspect
@app.get('/predict/{site}')
async def predict(URL):
   X_predict = []
   X_predict.append(str(URL))
   y_Predict = phish_model_ls.predict(X_predict)
   if y_Predict == 'bad':
      result = "This is a Phishing Site"
   else:
      result = "This is not a Phishing Site"

   return (URL, result)
if __name__ == '__main__':
   uvicorn.run(app,host="127.0.0.1",port=8000)
```

5.2.5 USER INTERFACE SCREEN:

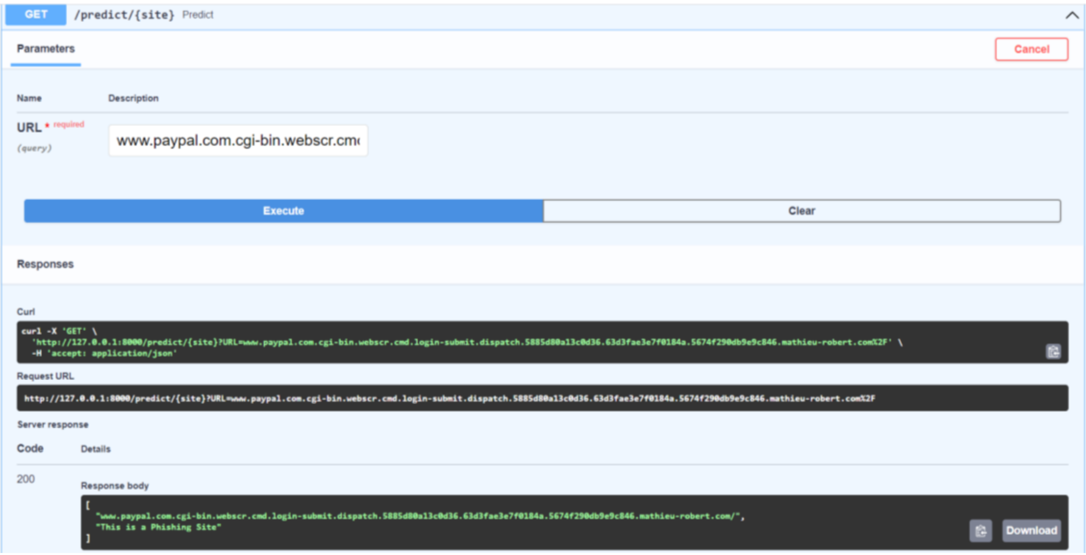


Name	Description
URL * required (query)	URL

Code	Description	Links
200	Successful Response Media type application/json Controls Accept header.	No links

Figure 5.2: User Interface Screen

OUTPUT SCREENS:



URL * required (query): www.paypal.com.cgi-bin.webscr.cm

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/predict/{site}?URL=www.paypal.com.cgi-bin.webscr.cmd.login-submit.dispatch.5885d80a13c0d36.63d3fae3e7f0184a.5674f290db9e9c846.mathieu-robert.com%2F' \
  -H 'accept: application/json'
```

Request URL

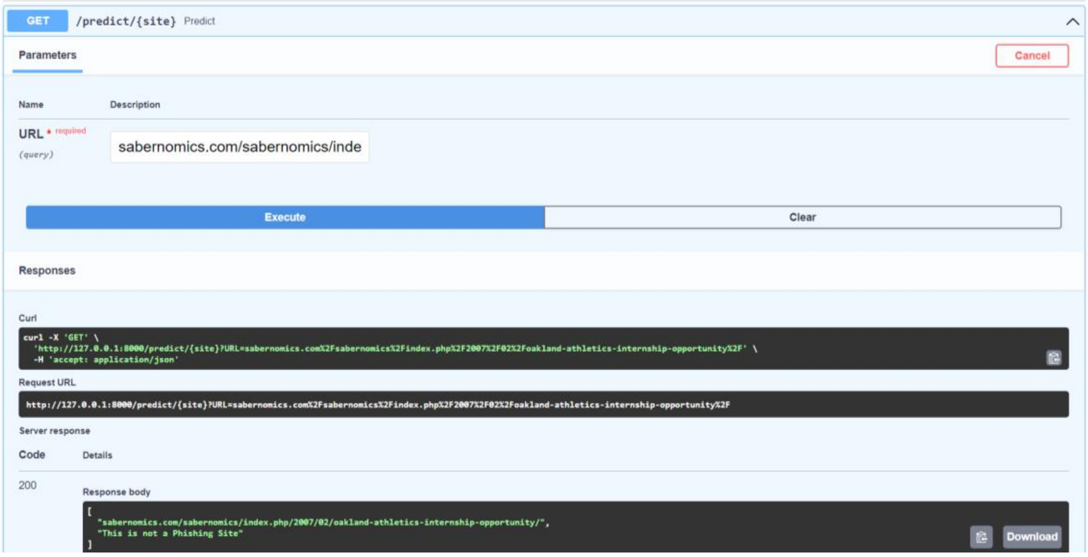
```
http://127.0.0.1:8000/predict/{site}?URL=www.paypal.com.cgi-bin.webscr.cmd.login-submit.dispatch.5885d80a13c0d36.63d3fae3e7f0184a.5674f290db9e9c846.mathieu-robert.com%2F
```

Server response

200 Response body

```
[
  "www.paypal.com.cgi-bin.webscr.cmd.login-submit.dispatch.5885d80a13c0d36.63d3fae3e7f0184a.5674f290db9e9c846.mathieu-robert.com/",
  "This is a Phishing Site"
]
```

Figure 5.3:Sample prediction result -1:



URL * required (query): sabernomics.com/sabernomics/inde

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/predict/{site}?URL=sabernomics.com%2Fsabernomics%2Findex.php%2F2007%2F02%2Foakland-athletics-internship-opportunity%2F' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/predict/{site}?URL=sabernomics.com%2Fsabernomics%2Findex.php%2F2007%2F02%2Foakland-athletics-internship-opportunity%2F
```

Server response

200 Response body

```
[
  "sabernomics.com/sabernomics/index.php/2007/02/oakland-athletics-internship-opportunity/",
  "This is not a Phishing Site"
]
```

Figure 5.4:Sample prediction result-2:

6. TESTING

In this chapter, we check for the working of the proposed system by testing and comparing the result of the algorithm and the actual result. It is basically validating the system. The testing is done for each algorithm with a legitimate and phishing URL and the results are as follows.

Below are the section to be concentrated in testing chapter .

6.1 UNIT TESTING

Unit Testing is a testing approach where the units of the modules are investigated to check regardless of whether they are fit as a fiddle to be utilized.

Table 6.1: Test Case -1

Test case	01
Test Name	Testing of LR -1
Input	http://fraud.hmmmm.com/reroute?dst=www.paypal.com+dxz=hj7880
Expected output	Phishing site
Actual output	Phishing site
Remark	Success

Table 6.2: Test Case -2

Test case	02
Test Name	Testing of LR -2
Input	https://www.amazon.com/
Expected output	Not a Phishing site
Actual output	Not a Phishing site
Remark	Success

CONCLUSION

Due to the growing use of Internet in our daily life, cyber attackers aim their victim over this platform. One of the mostly encountered attack is named as "phishing" which creates a spoofed web page to obtain the users sensitive information such as user-ID and password in financial websites by using social networking facilities. The malicious web page is created as if a legitimate web page, especially copying the original web page one to one. Therefore, detection of these pages is a very trivial problem to overcome due to its semantic structure which takes the advantage of the humans' vulnerabilities. Software tools can only be used as a support mechanism for detection and prevention this type attacks, and these tools especially use whitelist/blacklist approach to overcome this type of attacks. However, they are static algorithms and cannot identify the new type of attacks in the system. Therefore, as an efficient solution, We applied logistic regression and Naïve Bayes algorithms to model and train our model and at the end logistic regression which gave a more accurate prediction was used in our system.

BIBLIOGRAPHY

[1]https://www.researchgate.net/publication/346534722_Phishing_website_detection_using_support_vector_machines_and_nature-inspired_optimization_algorithms

[2] https://ijarsct.co.in/Paper1412

[3] https://www.sciencedirect.com/science/article/pii/S187770581200940X

[4] https://link.springer.com/chapter/10.1007/978-981-33-4299-6_12

[5]https://www.researchgate.net/publication/2614079_Genetic_Algorithms_and_Heuristic_Search

[6] https://arxiv.org/pdf/2009.11116.pdf

[7] https://cybersecurity.springeropen.com/articles/10.1186/s42400-022-00126-9