

Palestine Technical University- Kadoorie

faculty of applied science

Department Applied Computing

**developing a Phishing Detection Tool**

**Prepared by:**

Ahmad Saleh-202012191

Nagham Maraheen-202012658

Muhannad Tomeh-202011744

**Supervisor:**

Dr..Eman Droubi

**2022/2023**

**Abstract**

With the wide spread of the Internet and its entry into all areas of life, until our personal information and bank accounts became stored on the Internet, and with research and statistics, it was found that the greater the use of the Internet, the greater the penetration rate around the world. The Internet has become a place where thieves and intruders aspire to intrude on others and break into their privacy or cause damage or theft.

Therefore, there has become an urgent need for us to preserve our privacy and personal information and protect ourselves from exposure to danger and penetration.

Therefore, in this project, we decided to develop a mechanism, which is a website. This site provides a service that can detect dangerous sites and protect them from the danger they cause from breaching the privacy of the user or the browser on the Internet in general.

In this project, we will build a machine learning model for prediction

if website is phishing or not start fetch data then preprocess data by applying feature scaling then create neural network (mlp multilayer perceptron) and set number of layers and each layer has specific number of neurons

And the last thing that determines the number of repetitions.

Therefore, this project is very necessary for everyone who uses the Internet, especially if he has little experience in the Internet, so it is easy to penetrate, and this service can be used by anyone in the world, as it is restricted to a specific region.

Therefore, this project will help many people, official and unofficial institutions and companies to ensure the safety of links to websites, with an accuracy rate of more than 90% to protect them from hacking.

By making it a reference for them in case they encounter any external links that may harm them.

**Contents:**

1. **Introduction** 
   1. Introduction
   2. Problem statement
   3. Project objectives
   4. Project benefits
   5. Project methodology
   6. Research plan (working plan)
   7. Project constraints (research limitations)
   8. Summary and recommendations
2. **System Analysis** 
   1. Introduction
   2. A brief history of the organization
   3. Project implementation options
   4. The proposed system
   5. System requirements
   6. Feasibility study
      1. Economic feasibility
      2. Technical feasibility
      3. Legal feasibility
      4. Schedule feasibility
   7. Project added values
   8. Project management
   9. Summary and recommendations
3. **System Functional Requirements** 
   1. Introduction
   2. Context or Use-Case Diagrams
   3. Functional requirements description
   4. Non-functional requirements description
   5. Summary and recommendations
4. **System Design and Development**
   1. Introduction
   2. Class diagram
   3. Sequence diagrams
   4. Entity Relationship Diagram (ER-model)
   5. Activity diagrams
   6. System interface (input/ output design)
   7. Summary and recommendations
5. **Coding and Implementation** 
   1. Introduction
   2. Coding programming languages
   3. Database system
   4. Explain algorithms and some technique
   5. Summary and recommendations
6. **Conclusions and Future Works** 
   1. Conclusions
   2. Future works
   3. Closing remarks

**List of figures:**

Figure 1.1. Cyber Phishing websites

Figure 3.2.1 context diagram.

Figure 3.2.2 use case.

Figure 4.2 class diagram

Figure 4.3 Sequence diagrams

Figure 4.4 ER-Model.

Figure 4.5 Activity diagram.

Figure 4.6.1 main page

Figure 4.6.2 tips page.

Figure 4.6.3 check page

Figure 4.6.3.1 check page (not phishing)

Figure 4.6.3.2 check page(phishing)

**List of Tables:**

Table 2.5.1 Functional System requirement

Table 2.5.2 Non-Functional System requirement

Table 2.6.1 Technical Feasibility (Hardware)

Table 2.6.2 Technical Feasibility(software)

Table 2.6.4 Schedule feasibility

Table 3.3 Functional Requirements

**Chapter** **1**

**1.1 Introduction:**

Nowadays, with the spread of the Internet and websites, and the almost daily use of websites, some people are taking advantage of the spread of links in a large way and take advantage of people's inexperience with security on the Internet! They are creating malicious fake links and deceiving people in them, aiming to seize the victim's personal information for personal reasons, either for extortion, stealing money, or just sabotage.

A number of subscribers and users of mobile phones are exposed to an increasing number of fraud attempts in recent times, due to the penetration of their devices as a result of entering malicious or fake links that arrive through WhatsApp messages, text messages (sms) or other electronic programs in which they ask you for an amount of money or some assistance. Then you are surprised that that person's phone device has been hacked and that the person who sent that message is the hacker or the hacked organization of the phone or device.

In this following figures , examples of these malicious link :



Figure 1.1. Cyber Phishing websites [1]

**1.2 Problem Statement:**

Phishing is one of the easiest types of cyberattacks for hackers to do and one of the easiest types of cyberattacks to fall for too! This is often done to steal user data such as login details, credit card numbers, and other personal information. The attacker appears as a trusted facility to trick the person into clicking on a specific link or opening an email or WhatsApp message. And some links come short so that you cannot know their source.

Malware may be installed if the recipient clicks on the link, the system may be frozen (a type of ransomware attack), or sensitive information may be exposed due to a security vulnerability in the operating system.

Statistics reveal that more than 4.1 billion people are email users and about 3 million emails are sent per second which means that more than 50% of the world’s population uses email and the number is expected to rise to 4.5 billion users by 2024. These The high numbers encouraged criminals to focus their attacks on email rather than other sources. And with COVID-19, many organizations have ordered their employees to work remotely for their safety, and this has led to many people relying on email communication a lot, and with employees receiving a large number of messages daily, this makes them more vulnerable to cyberattacks. And when it comes to phishing, all it takes is one click to lose it all.

Phishing attacks, in their most common form, are email messages that urge mail recipients to take action, usually to achieve one of two goals:

- Trick the user into sharing their personal information

- Tricking the user into downloading malicious software

Once you give the software access, hackers can access your bank account, steal your identity, or make fraudulent purchases in your name.

A common example of this deception is that you receive in your e-mail

A fake letter in the name of your bank containing a request

The bank requires you to log in to your account and update your personal information

Clicking on the link in the message, you will be directed to a site

a different electronic mail, the appearance of which is identical to the website of that bank,

And when you enter the username and password as requested

From you, this data is sent to the hacker's address, not to the bank.

Over the past few years, email scams have increased by over 400%. The growth and success of email phishing scams has also led to the frenzy of these scams. We will discuss more about it below:

SMiShing

As the name suggests, SMiShing is similar to phishing email, but it tricks users via text messages. All in all, many people are aware of email phishing, but the level of awareness about SMS fraud is lower than the aforementioned pattern, which increases the possibility of people falling victim to a phishing scam.

Spear Phishing

Spear phishing uses the same methods of tricks that we mentioned above, but the difference is that it targets a specific person. You might see a series of emails designed to entice you to take action. Phishing attacks can also target multiple messaging platforms.

**1.3 Project objectives:**

The following points summarize the main objectives of this project:

\*Investigating the nature of cyberattacks content and its types.

\*Developing a websites for detecting cyberattacks and phishing systems

\* Conducting extensive experiments for evaluating the proposed model.

**1.4 Project benefits:**

**- For users:**

Building a tool for Detecting cyberattacks in websites which may be added to Facebook officially as adds-on may play a role in:

1-Making phishing email or websites detection easy for users.

2-Making internet safer and reducing attacks.

3-Protect people from attackers and save they especially data

**- For developers:**

Enhancing and expanding general knowledge about cyberattacks and improving the experience of developing useful data mining-based programs.

Learning about the many algorithms that are used to solve a problem and which are the building blocks of the advanced digital world we see today. It is an important concept that must be understood, because in machine learning, learning algorithms - not computer programmers - create the rules.

**1.5 Project methodology:**

The software development methodologies are very important which are mostly used for various software development projects. Moreover, all these methodologies work well in certain projects depending upon the nature of the project. None of these methodologies are foolproof as each has its pros and cons. The basic purpose of these methodologies is to provide smooth software development according to the project requirements [3].

Below are some of the methodologies that are primarily used software development methodologies.

In our project, We will adopt **the waterfall model** because :

1 - requirements will be very understood so we don’t need to send to customer proto type for each small update

2 – It is simple and easy to understand and use.

3 - The project status is more easily measured based on a complete schedule and resource plan.

4 - Risk is zero or minimum

**1.6 Research Plan :**

In this project we are going to build machine learning model to predict

if website is phishing or not starting with fetching data and then preprocess the data by apply feature scaling then create neural network (mlp multilayer perceptron) and set number of layers and for each layer set number of neurones  
and last thing set number of iterations .

**1.7 Project constraints:**

There are some big challenge for any machine learning project

and in this project we need to mention some of them :

**1 – Accuracy Of Model :**

In any machine learning model all the time we need to optimize model

to get high accuracy

So we need to maximum accuracy percentile.

**2 – Model Efficiency:**

If Model take very long Time to predict weather website is phishing or not, what do we benefit?

So in project we the performance and effectiveness of the algorithm used have been taken into account .

**1.9 Summary and Recommendation:**

In this chapter, we have discussed the introduction to this project, the problem statement of the project and its solution, and we have mentioned the objectives of the project and how it will benefit both developers and users. We have mentioned some of the mythologies and discussed how the work plan will be done. Finally, we talked about project constraints and limitations.

**Chapter 2**

**{System Analysis}**

**2.1 Introduction :**

In this chapter, we will cover the important topics. will explain what the system analysis is in general mention and discus previous researches that the same our project , explain the project Implementation options and specify the one that we will choose , explain what it is the system requirements , then finally, we will show the feasibility study of this project.

**2.2 Related work:**

According to APWG phishing attack trends reports [13, 14], the number of phishing attacks observed by APWG and its members grew through 2020, doubling over the course of the year. Phishing are spread via e-mail, SMS, instant messaging, social networking etc., but e-mail is a popular way to carry out this attack. The phishing email can lead to financial loss. Attacker always sending email tends to make user believe that they are communicating with trusted entity and deceive them into providing personal credentials in order to access service, such as credit card numbers, account login credential or identity information. In 2019, 293.6 billion emails were sent and received daily. This includes billions of promotional emails sent by merchants every day.

In Sect. 1 of this paper, we applied machine learning on three different data sets where the first two datasets depend on multi features and the third one depends on text feature only. Section 2 we review the Related Work of classifiers used in detecting phishing emails, in Sect. 3 we mentioned the targeted victims in phishing. The methodology that has been followed to do this research has been introduced in Sect. 4. Section 5 presents the experiments for classifying Phishing Email Using Machine Learning, Finally, the work is concluded in Sect.

**2.3 Project implementation options :**

There are several project implementations options, like Desktop Applications, Web Applications, and Mobile application.

**- Desktop Applications**

Desktop applications are stand-alone application which runs on system and laptops.

**- Web Applications**

This type of software application is used through the internet via a web browser. A web browser allows you to access the app and its content and also runs all the scripts responsible for its features. What differentiates a simple static web page from the web application is interactivity. They often allow you to create, edit, or manipulate data and content

**- Mobile application**

It’s a type of application designed to run on a mobile device, which can be a smartphone or tablet computer.

In our project, we are using Python to develop a Web application. Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches Additionally many libraries that support data science and machine learning. Web applications are easy to use for anyone and much enjoyable than mobile application.

**2.4 The proposed system :**

The main objective of the project is to discover phishing sites that help the general public, companies and institutions and protect them from those trap sites that, if they deal with them, cause them harm and expose them to theft or damage to their sites .

In this project we are going to implement a website for phishing detection . the tool will have information about phishing that admin give it as dataset from Kaggle contain alot of website and its result phishing or not , and tool be trained about phishing websites then will be ready to act a new website which checked in future .

and tool use many algorithims like Logistic.Regression and SVM Support Vector Machine ,Naive Bayes

K Nearest Neighbors , Neural Network(MLP), and Multilayer Precptron that tool depend on its to take decesion .

And the tool will have edit text which allow the user to enter a URL .

Then based on a trained ML model it will check if the URL is phishing or not .

**2.5 System requirements:**

The Functional requirements explain how the system must work, while the nonfunctional requirements explain how the system should perform.

**The Functional requirement of the system:**

Table 2.5.1 Functional System requirement

|  |
| --- |
| Data Preprocessing 🡪 Clean Data And Convert Sentence To Vectors |
| Rate Model 🡪 Rate Model From 5 And Write Feedback |
| Import Data 🡪 Extract Data From Kaggle |
| Classification 🡪 Classification Weather URL Phishing OR Not |
| Retrain Model 🡪 Train Model From Imported Data To Improve Results |
| Show Tips🡪 Show Tips For User For Increase Awareness For User . |
| Invite Friend🡪Invited Friend Via Facebook , Linkedin , … |
| Show Results🡪Show Precision , Recall And F1 Score For Model |
| Show Rates & Feedback 🡪 Show People Rates For Model |

**Non Functional Requirements:**

Table 2.5.2 Non Functional System requirement

|  |  |  |
| --- | --- | --- |
| **Description** | **Nonfunctional** | # |
| Clear, simple and easy that can be understood and used by anyone who does not have experience in the field | Understandable &Usability | 1 |
| Be available to all users at any time | Availability | 2 |
| Able to give high F1 Score result | Correctness | 3 |
| Ability to perform speedily and execute efficiently | Performance | 4 |
| Ability to cope with errors during execution and cope with erroneous input | Robustness | 5 |

**2.6 Feasibility study:**

**2.6.1 Technical Feasibility :**

**-Hardware:**

Table 2.6.1 Technical Feasibility (Hardware)

|  |  |
| --- | --- |
| **Components** | **Description** |
| HP Laptop | Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz |
| HP Laptop | Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz 2.00 GHz  RAM : 8.00 GB |
| ASUS | Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz  RAM: 20.0 GB |

**-Software:**

Table 2.6.2 Technical Feasibility(software)

|  |  |
| --- | --- |
| **البرامجComponents** | **استخدام البرنامج Description** |
| PyCharm | To build the Apps |
| Sql Server | To create the data base |
| EdrawMax, VP | To draw the diagrams |
| Internet Connectivity | Available at the client phone |

**2.6.3 Legal feasibility :**

The proposed project ensures that it is legally accept and conform the legal and ethical requirements.

**2.6.4 Schedule feasibility :**

Table 2.6.4 Schedule feasibility

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **First month** | | | | **Second month** | | | | **Third month** | | | | | **Fourth month** | | | |
| **Week 1** | **Week 2** | **Week 3** | **Week 4** | **Week 5** | **Week 6** | **Week 7** | **Week 8** | **Week 9** | | **Week 10** | **Week 11** | **Week 12** | **Week 13** | **Week 14** | **Week 15** | **Week 16** |
| **Find my group** |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |
| **Find supervisor** |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |
| **Find idea** |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |
| **Project introduction** |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |
| **System**  **analysis** |  |  |  |  |  | | |  |  | |  |  |  |  |  |  |  |
| **System functional requirements** |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
| **Documentary** |  |  |  |  |  | | | | | | | | | | | | |

**2.7 Project added values :**

Educate the users of the program about the dangerous and trap links, and increase people's knowledge of them and their dangers, so that knowledge of them spreads by providing them with simple advice for knowledge to be sufficient.

**2.8 Project management :**

- Supervision : Dr. Iman Droubi

The team:

1- Ahmad Saleh : leader / task : backend

2- Nagham Maraheen : member / task : interfaces

3- Muhannad Tomeh : member / task : interfaces

**2.9 Summary and recommendations :**

In this chapter, we talked about system analysis staring from a brief history of organization considering previous project and how they did work and how did their performance evolve over the time, project implementation options and proposed system that illustrate where and how the system is applied, we referred to functional and nonfunctional System requirements, discussed Feasibility study, Project added values and ending with mentioning project management.

**Chapter 3**

**{System Functional Requirements}**

**3.1 Introduction :**

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. In this chapter, we will use the "use case diagram" to show how the user interact s with our application, and we will show the functional and non-functional requirements of this application.

**3.2 Context or Use-Case Diagrams :**

**Context Diagram :**

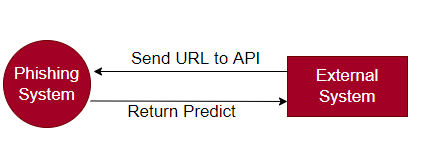
****

Figure 3.2.1 context diagram.

**Use Case Diagram:**

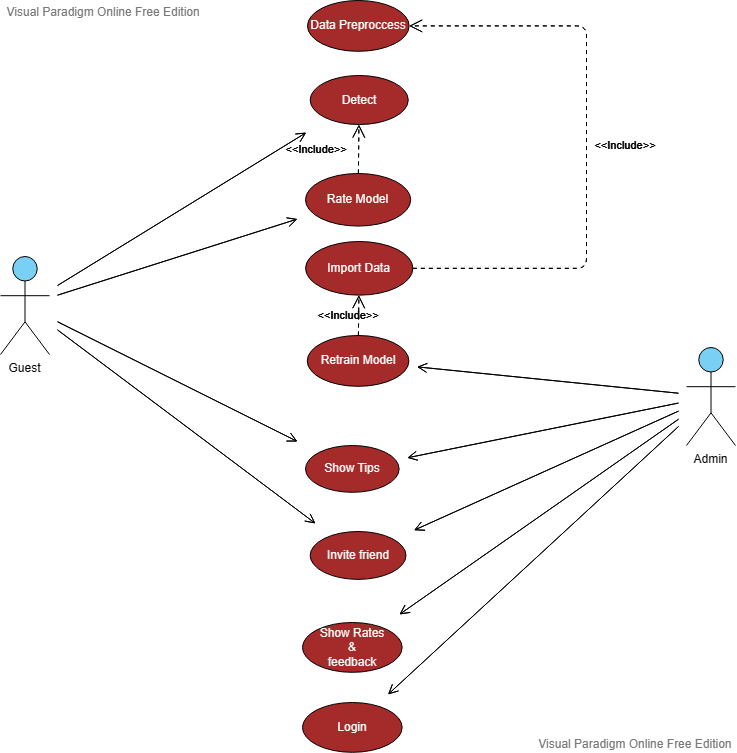


Figure 3.2.2 use case.

**3.3 - Functional Requirements:**

Table 3.3 Functional Requirements

|  |  |
| --- | --- |
| **Functional** | # |
| **Data Preprocessing**  1- Actor🡪Admin.  2- Input🡪Data OR URL.  3- Description🡪 Preprocess Data Like Convert Text To Vector Using Technique Word2Vec And Cleaning Data By Ignoring Missing Values | 1 |
| **Detect**  1- Actor🡪Guest.  2- Input🡪URL.  3- Description🡪 Check Weather URL Is Phishing OR Not  4- Reaction/Output🡪 Alter If Its Phishing OR Its Not Phishing | 2 |
| **Rate Model**  1- Actor🡪Guest.  2- Input🡪Rate From 5 And Feedback.  3- Description🡪Rate Model.  4- Reaction/Output🡪Rated Successfully OR Not | 2 |
| **Import Data**  1- Actor🡪Admin.  2- Input🡪Data  3- Description🡪Import Data From Admin Device To Train Model On It  4- Reaction/Output🡪Imported OR Not | 2 |
| **Retrain Model**  1- Actor🡪Admin.  2- Input🡪Data To Train Model On It  3- Description🡪Retrain Model For Getting Better Results  4- Reaction/Output🡪 Trained Successfully OR Not With New Information About Recall, Precision And F1 Score | 3 |
| **Show Tips**  1- Actor🡪Guest , Admin.  2- Input🡪Press Button Show Tips  3- Description🡪 Show Some Tips For Increase Awareness For User .  4- Reaction/Output🡪 Selection of appropriate and suitable data. | 4 |
| **Invite Friends**  1- Actor🡪Guest, Admin.  2- Input🡪  3- Description🡪Open Small Dialog To Share Website Using Facebook , Linkedin ,..  4- Reaction/Output🡪 | 5 |
| **Show Rates & Feedback**  1- Actor🡪Admin  2- Input🡪 Press Button Show Rates & Feedback  3- Description🡪Show People Ratings And His Feedback  4- Reaction/Output🡪 | 6 |
| **Login**  1- Actor🡪Admin  2- Input🡪 Email And Password  3- Description🡪Login Page For Admin  4- Reaction/Output🡪Login Successfully OR Failed | 7 |

**3.4 Non Functional Requirements:**

Table 3.4 Non Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Description** | **Nonfunctional** | # |
| Clear, simple and easy that can be understood and used by anyone who does not have experience in the field | Understandable &Usability | 1 |
| Be available to all users at any time | Availability | 2 |
| Able to give high F1 Score result | Correctness | 3 |
| Ability to perform speedily and execute efficiently | Performance | 4 |
| Ability to cope with errors during execution and cope with erroneous input | Robustness | 5 |

**3.5 Summary and recommendations:**

In this chapter, we talked about Context , Use-Case Diagrams, Functional requirements description and Non-functional requirements description.

**Chapter 4**

**{System Design and Development}**

**4.1 Introduction:**

In this chapter, we will show several diagrams for our system, which are Class Diagram, Sequence Diagram, ER Diagram and Activity Diagram. And we will show the prototype of our application, which are the expected interfaces.

**4.2 Class diagram:**

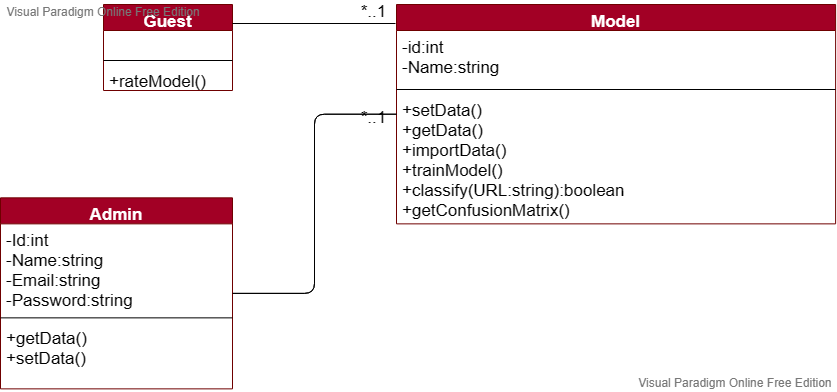
****

Figure 4.2 class diagram

**4.3 Sequence diagrams:**

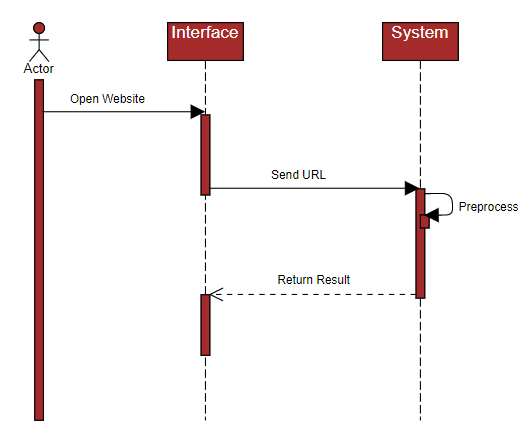


Figure 4.3 Sequence diagrams

**4.4 Entity Relationship Diagram (ER-Model)**

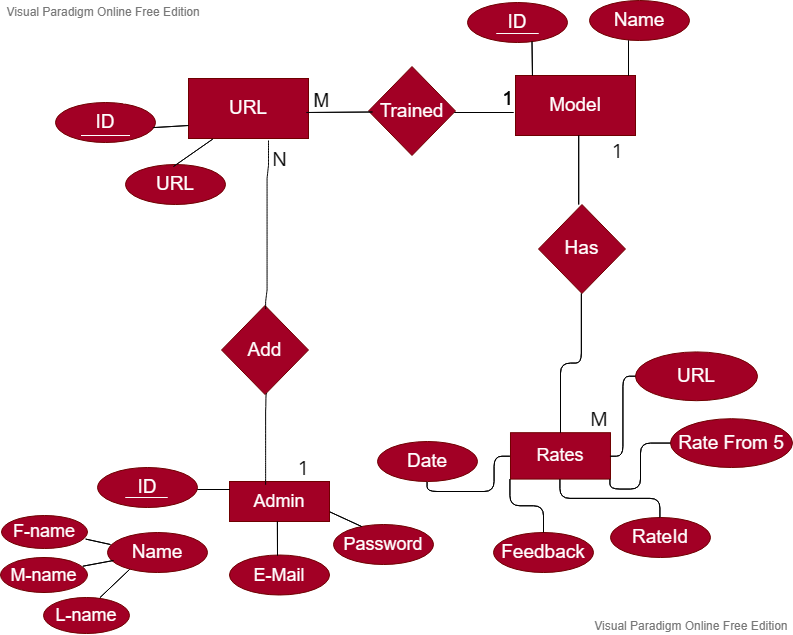
****

Figure 4.4 ER-Model.

**4.5 Activity diagrams:**

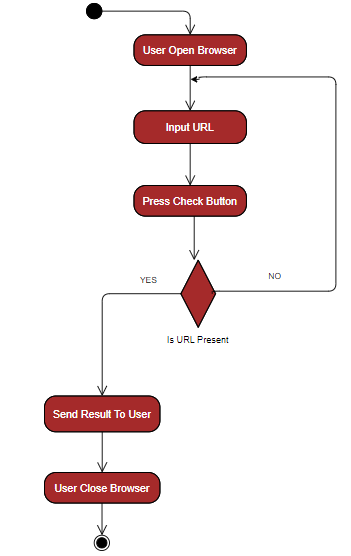
****

Figure 4.5 Activity diagram.

**4.6- System interface (input/ output design) :**

**-Main Page :**

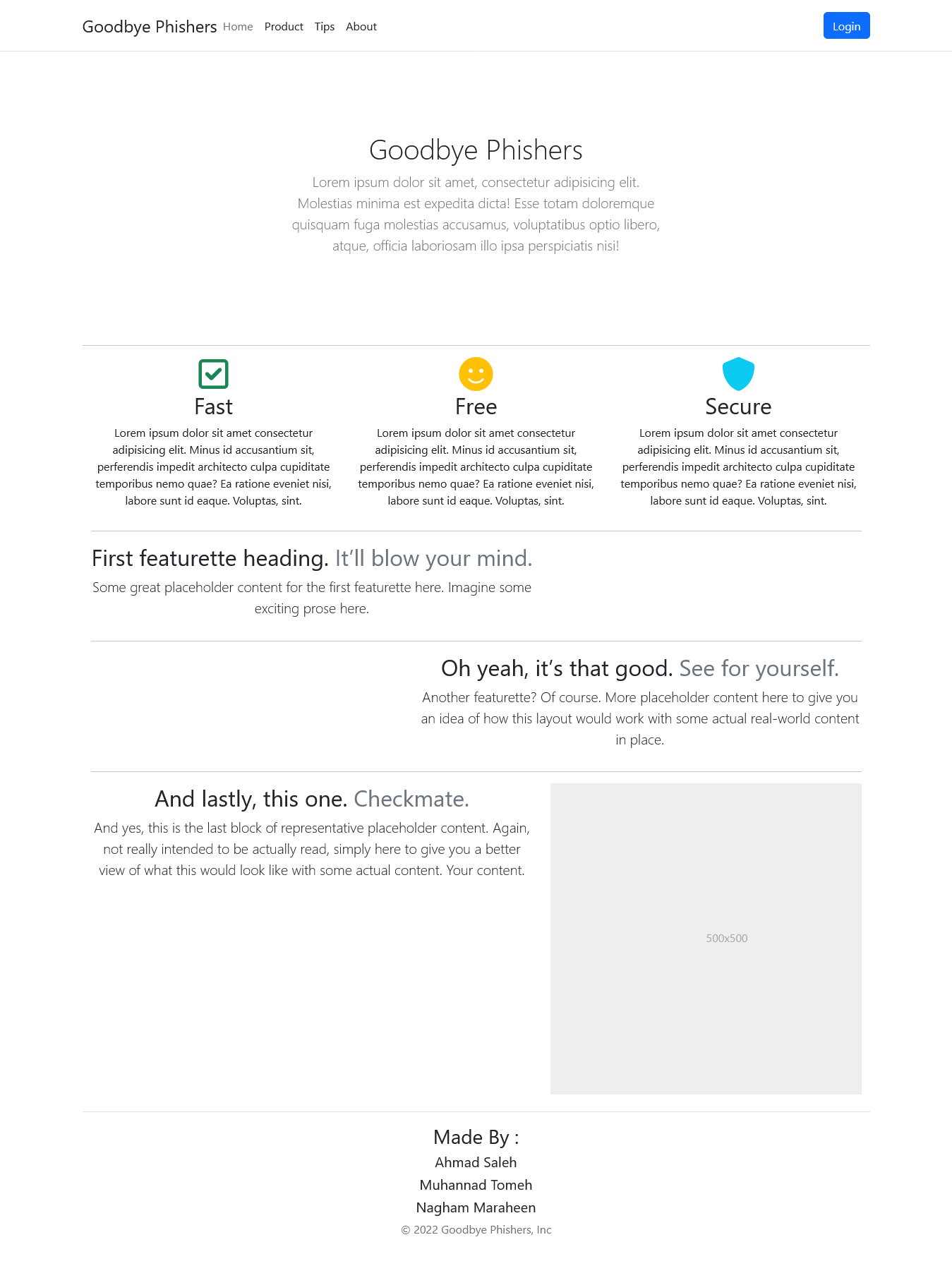
****

Figure 4.6.1 main page

**-Tips Page :**

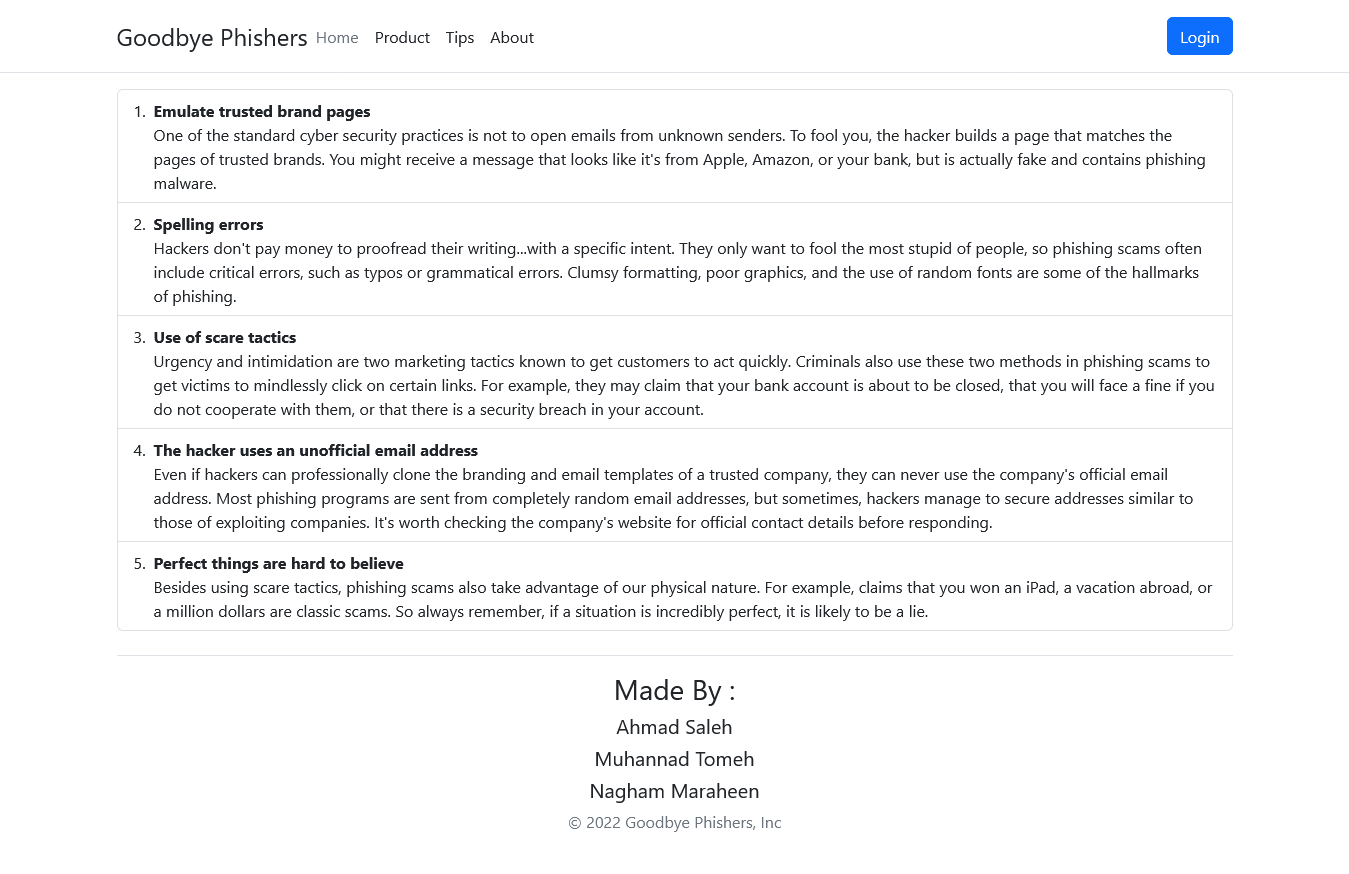


Figure 4.6.2 tips page.

**- Check Page :**

when User Not Test Product At Least One Time  
He Cant Make Rate

So Button Rate Model Is Disabled

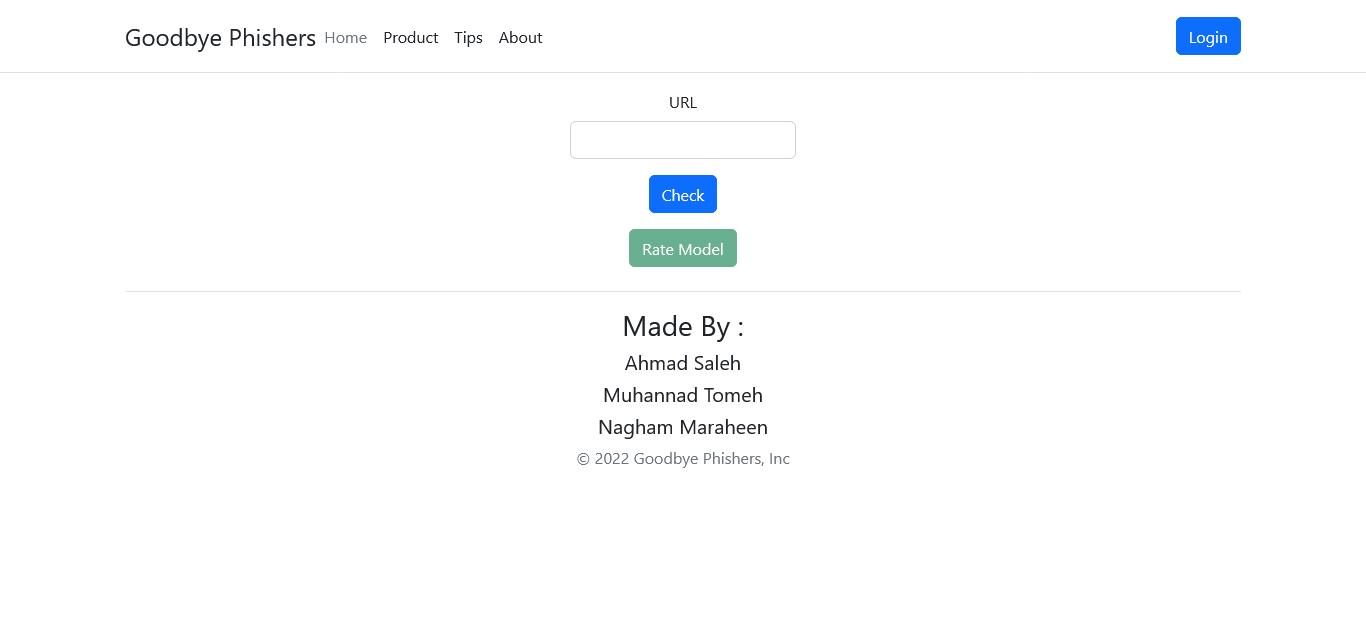


Figure 4.6.3 check page

**-Check Page(URL Is Not Phishing) :**He Can Make Rate Because He Use Model

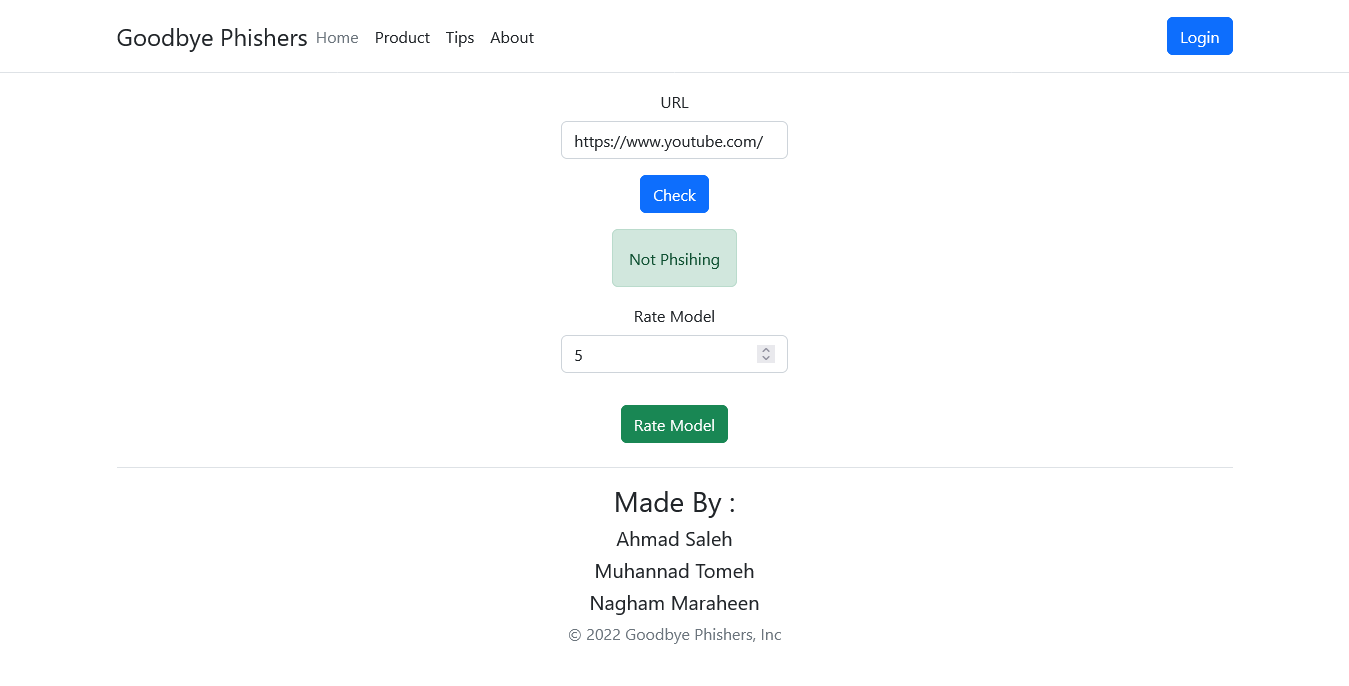
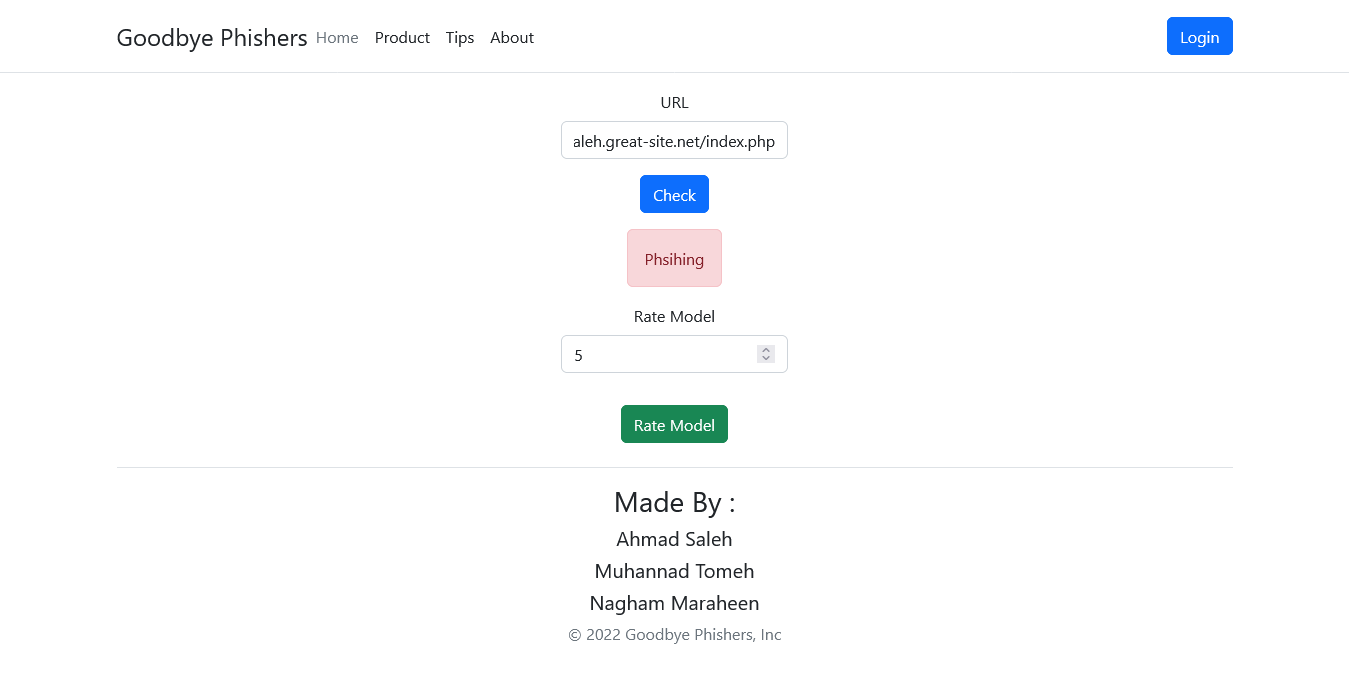
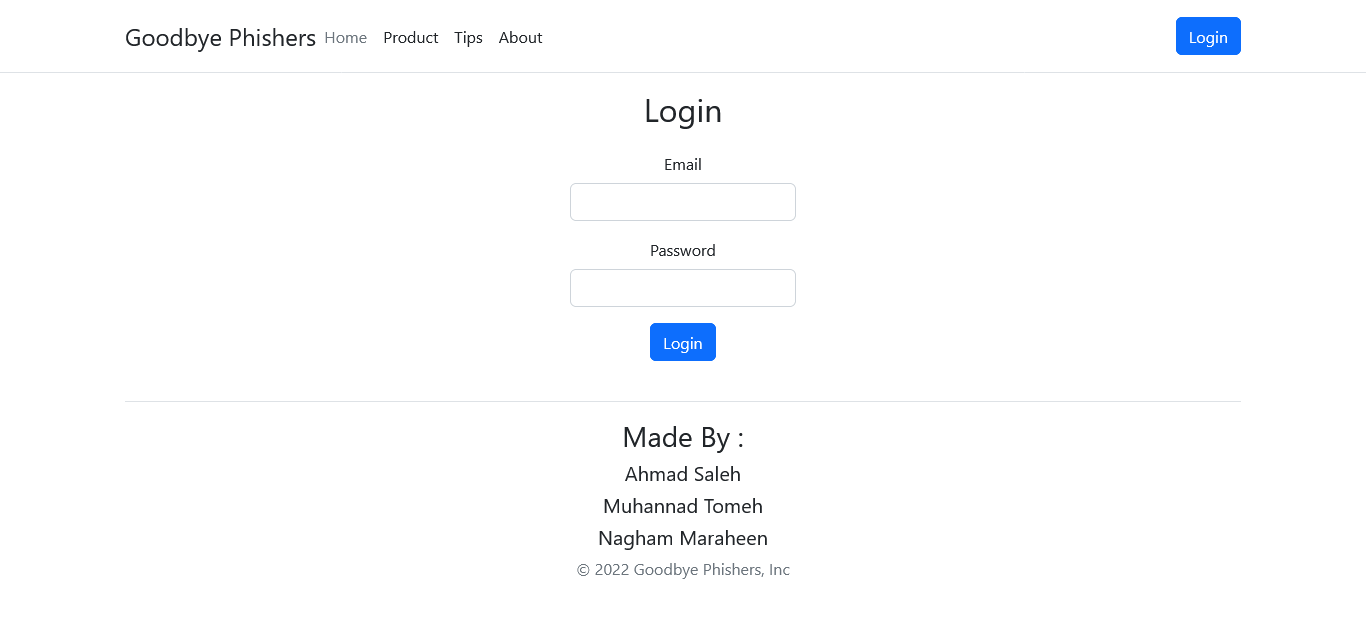


Figure 4.6.3.1 check page (not phishing)

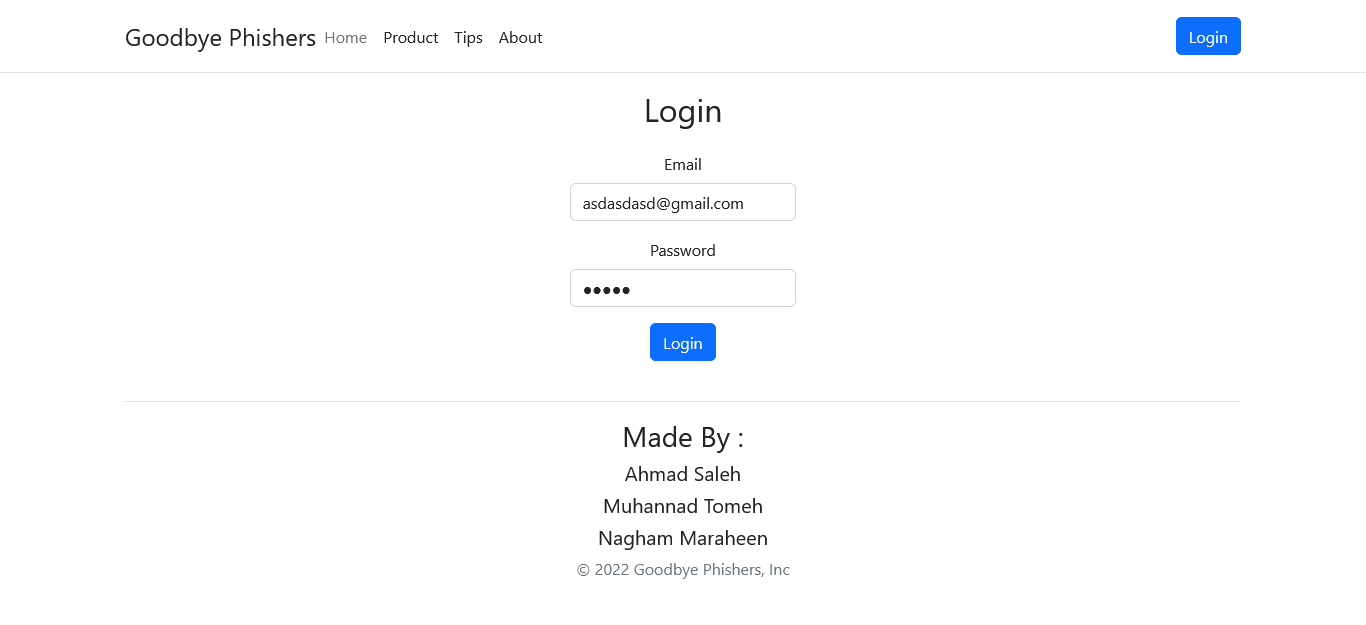
**- Check Page(URL Is Phishing) :**He Can Make Rate Because He Use Model



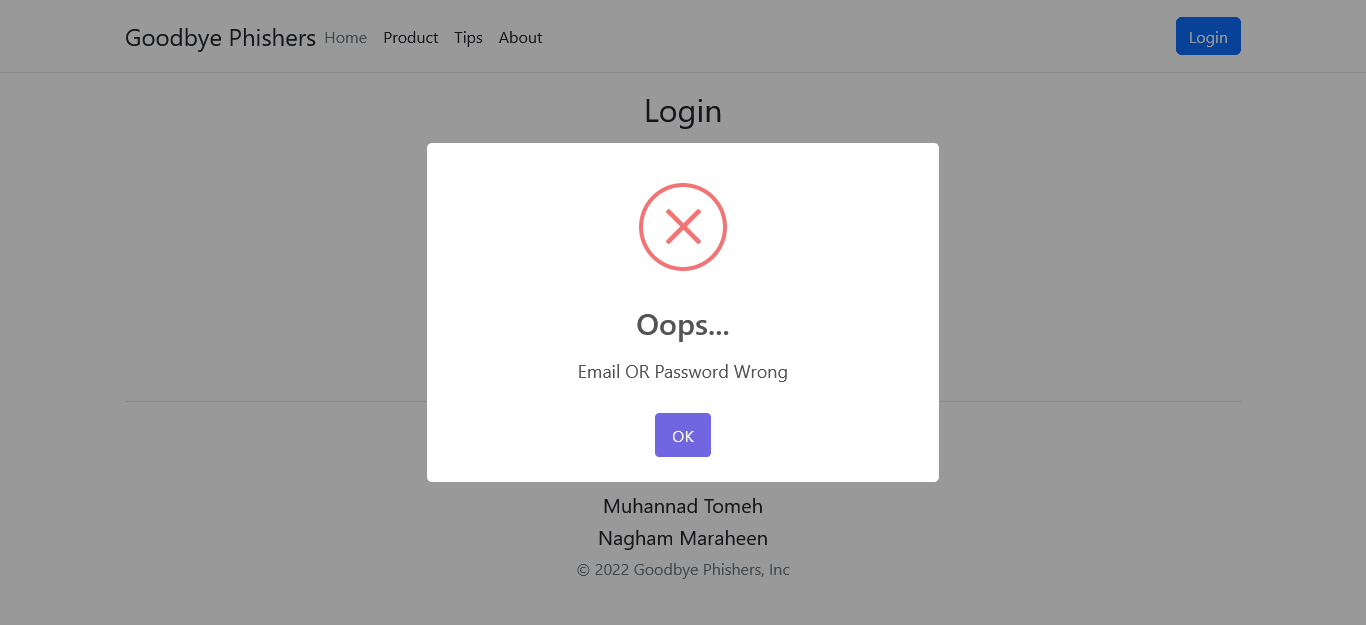
**-Login Page :**



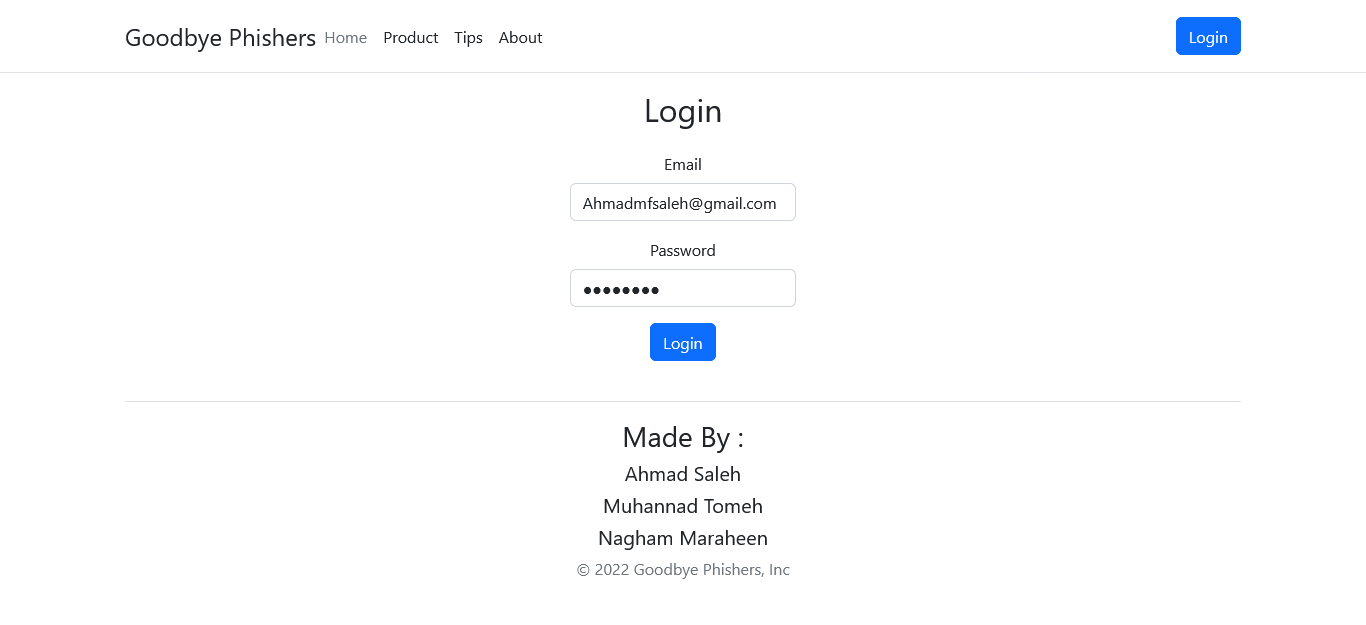
**When Enter Wrong Informations :**

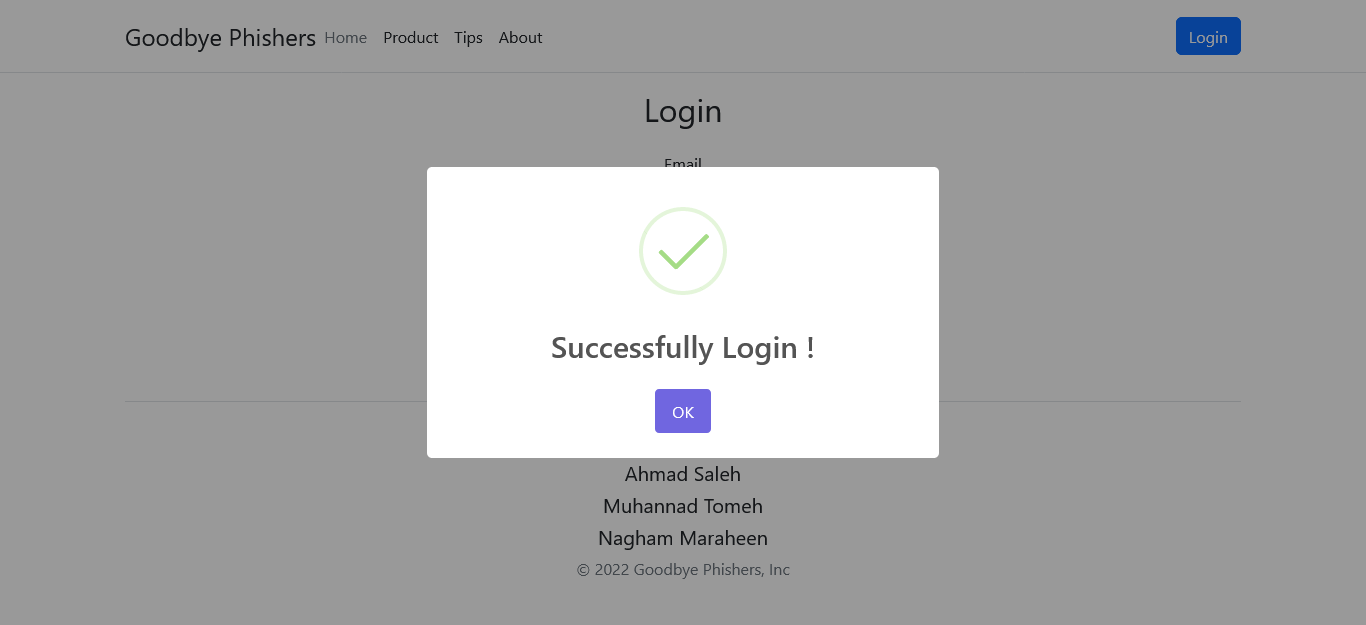


**Reaction :**

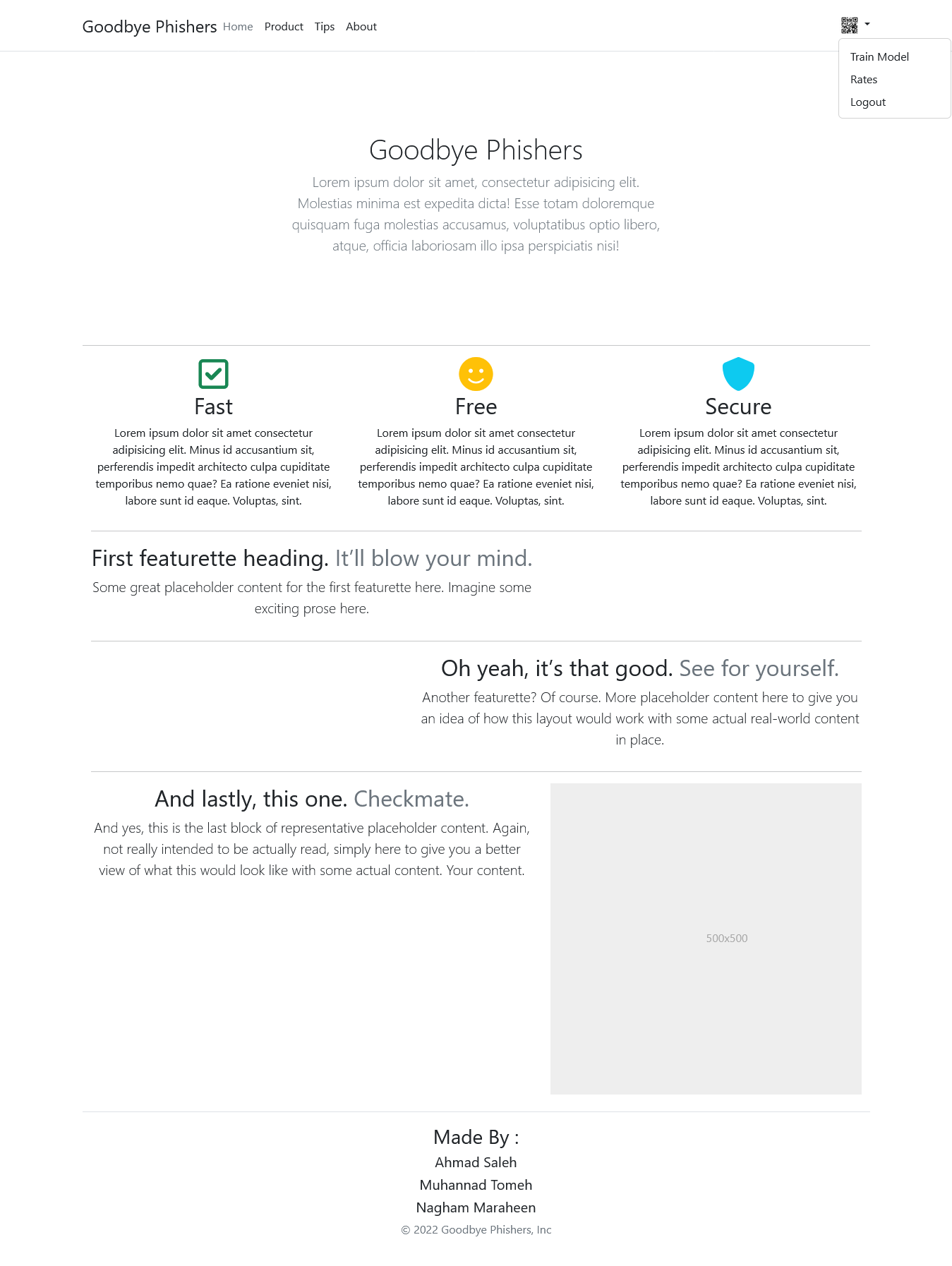


**When Enter Correct Information :**

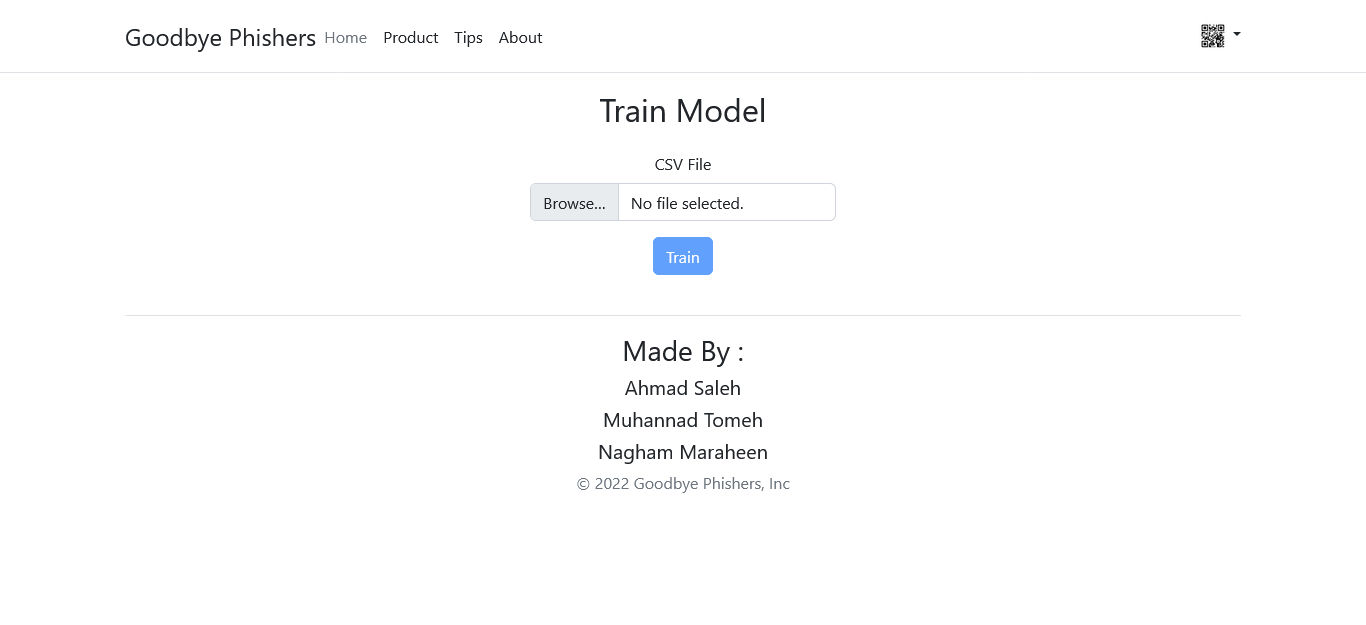


**Reaction :**   


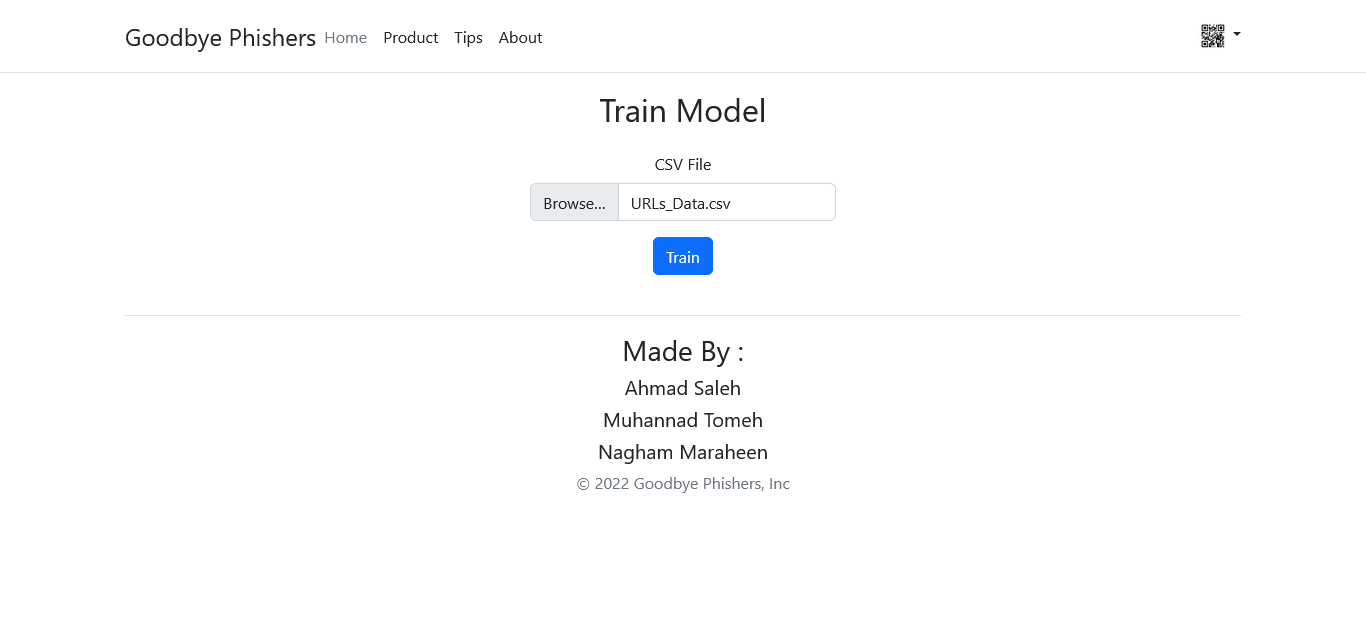
**When Admin Successfully Login :**



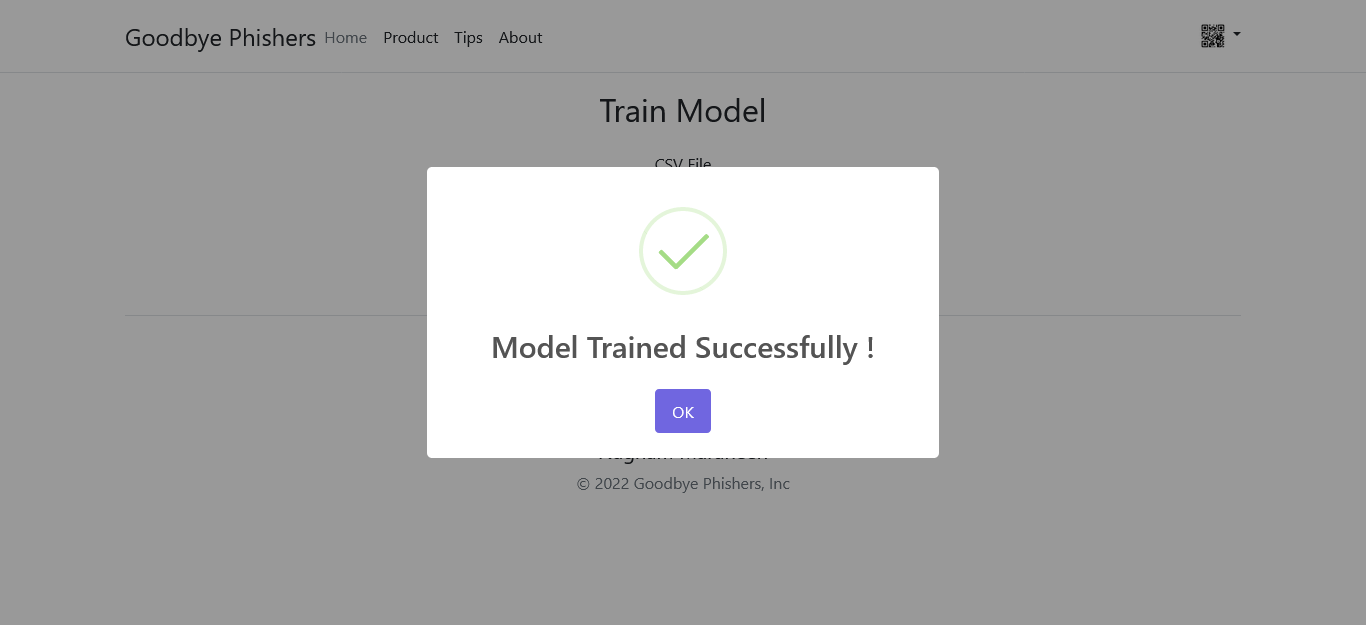
**When Click Train Mode :**



**When Upload Data :**



**Reaction After Train It :**



**When Click Rates :**

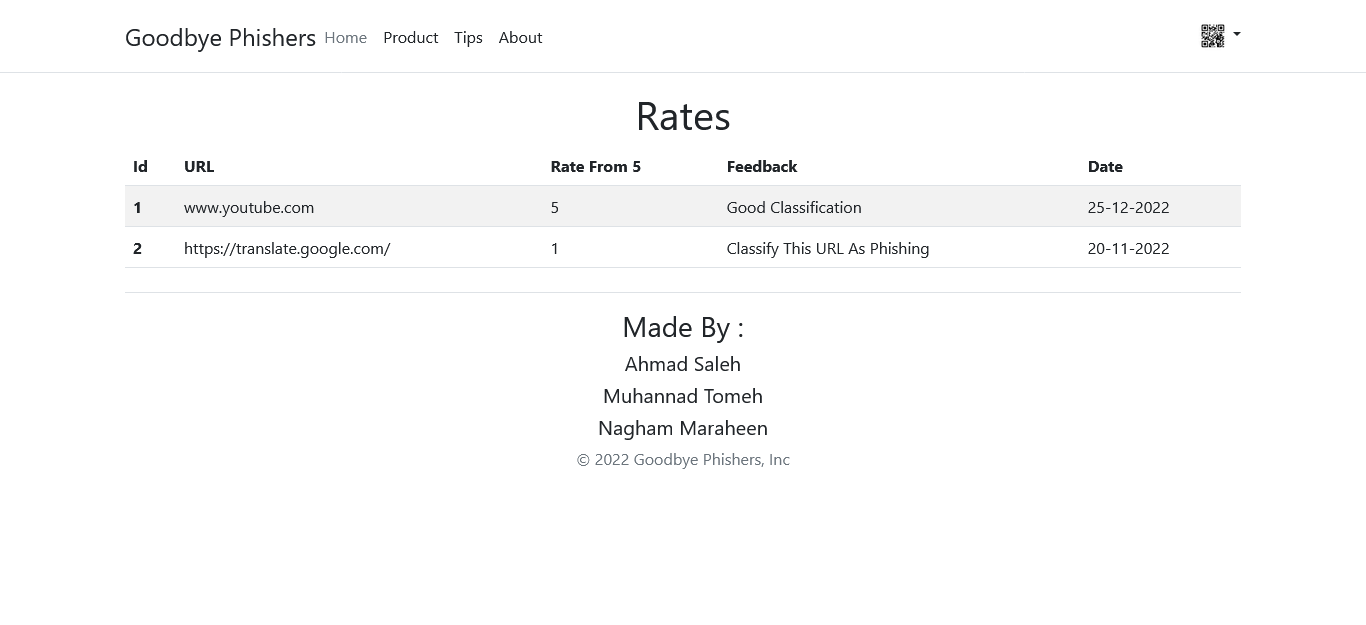


Figure 4.6.3.2 check page(phishing)

**4.7 Summary and recommendations :**

In this chapter, we talked about Class diagram , Sequence diagrams, Entity Relationship Diagram (ER-model) , Activity diagrams and System interface (input/ output design).

**References :**

[ML | Linear Regression - GeeksforGeeks](https://www.geeksforgeeks.org/ml-linear-regression/)

[Introduction to Logistic Regression | by Ayush Pant | Towards Data Science](https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148)

[K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint](https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning)

[Multinomial Naїve Bayes’ For Documents Classification and Natural Language Processing (NLP) | by Arthur V. Ratz | Towards Data Science](https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6)

[Supervised Machine Learning: Regression and Classification - Home | Coursera](https://www.coursera.org/learn/machine-learning/home/week/1)

[Advanced Learning Algorithms - Home | Coursera](https://www.coursera.org/learn/advanced-learning-algorithms/home/week/1)

[Understanding TF-ID: A Simple Introduction (monkeylearn.com)](https://monkeylearn.com/blog/what-is-tf-idf/)

[Understanding TF-IDF for Machine Learning | Capital One](https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/)

[Introduction to Word Embedding and Word2Vec | by Dhruvil Karani | Towards Data Science](https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa)

[Natural Language Processing with Classification and Vector Spaces - Home | Coursera](https://www.coursera.org/learn/classification-vector-spaces-in-nlp/home/welcome)

**Chapter 5**

**{** **Coding and Implementation }**

**5.1 Introduction**

In this chapter, we will discuss the coding and implementation of our phishing URL detection tool using machine learning algorithms and natural language processing techniques. The tool is designed to detect phishing URLs using a machine learning model trained on a dataset of phishing and legitimate URLs.

**5.2 Coding Programming Language**

We chose to use Node.js with the Express framework for creating the backend of our tool. Express is a minimal and flexible Node.js web application framework that provides a set of robust features for web and mobile applications. It allows us to create server-side APIs that enable us to perform various functions such as adding rates to the model, login for admins, and adding new URLs to train the model.

On the frontend, we used HTML, CSS, JavaScript, Bootstrap 5, and Font awesome. We also used the React framework to build a modern and dynamic user interface.

For the machine learning model, we used the Python programming language and the FastAPI library to create an API for accessing the model from the Node.js backend. Python is widely used for machine learning tasks due to its powerful libraries and packages such as NumPy, pandas, and Scikit-learn.

Pros:

* Node.js with Express framework provides a fast and scalable backend
* React framework provides a modern and dynamic frontend
* Python with Scikit-learn provides a wide range of machine learning algorithms and libraries
* FastAPI provides a simple and fast way to create APIs in Python

**5.3 Database System**

We chose to use MySQL as our database system for this project. MySQL is free, easy to use, and widely supported by hosting services. We used MySQL to store the URL dataset and to persist the machine learning model's training data and parameters.

**5.3 Explain algorithms and some technique**

We used several machine learning algorithms to train our phishing URL detection model, including Linear SVC, Logistic Regression, Naive Bayes, Neural Network, and Random Forest Trees.

1. **Linear SVC**: Linear Support Vector Classification is a linear model for classification that uses a hyperplane to separate classes in a high-dimensional feature space. The algorithm finds the best hyperplane that separates the data into two classes by maximizing the margin between the two classes.
2. **Logistic Regression**: Logistic Regression is a classification algorithm that uses a logistic function to model the probability of a binary target variable given one or more independent variables. The algorithm estimates the parameters of the logistic function using the maximum likelihood method.
3. **Naive Bayes**: Naive Bayes is a probabilistic classification algorithm that assumes that the presence or absence of a feature in a class is independent of the presence or absence of other features. The algorithm calculates the probability of each class given the features and selects the class with the highest probability.
4. **Neural Network**: Neural Network is a machine learning algorithm that is inspired by the structure and function of the human brain. The algorithm consists of multiple layers of interconnected nodes, where each node receives input from the previous layer and produces output that is passed to the next layer. The output of the last layer is the prediction of the model.
5. **Random Forest Trees**: Random Forest Trees is an ensemble learning method that uses multiple decision trees to make predictions. The algorithm creates multiple decision trees using random subsets of the data and features and aggregates their predictions to make the final prediction.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Precision | Recall | F1-Score |
| Linear SVC | 0.99 | 0.97 | 0.98 |
| Logistic Regression | 0.98 | 0.95 | 0.96 |
| Naïve Bayes | 0.98 | 0.97 | 0.97 |
| Random Forest Trees | 0.90 | 0.64 | 0.66 |

Now Explain Some Of Technique Used in NLP And Information Reterival  
For getting better accuracy  
F-IDF stands for Term Frequency-Inverse Document Frequency. It is a commonly used technique in Natural Language Processing (NLP) to extract real features from textual data. TF-IDF is used to evaluate the importance of a word or a phrase in a document based on how frequently it appears in the document and how common it is in the entire corpus.

TF-IDF is calculated as follows:

Term Frequency (TF): This measures the frequency of a term in a document. It is calculated as the number of times a term appears in a document divided by the total number of terms in the document.

TF = (Number of times term appears in a document) / (Total number of terms in the document)

Inverse Document Frequency (IDF): This measures the importance of a term across a corpus. It is calculated as the logarithm of the total number of documents in the corpus divided by the number of documents in which the term appears.

IDF = log\_e(Total number of documents / Number of documents with term t in it)

Once we have both the TF and IDF values, we can calculate the TF-IDF value for a term by multiplying the two:

TF-IDF = TF \* IDF

This gives us a measure of how important a term is to a document in the context of the entire corpus. Terms with high TF-IDF values are considered important to the document and can be used as features to train machine learning models.

For example, consider the following two documents:

Document 1: "The quick brown fox jumps over the lazy dog"

Document 2: "The quick brown fox is not lazy"

To calculate the TF-IDF value for the term "fox" in Document 1, we would first calculate the TF value:

TF = 1 / 9 = 0.11

To calculate the IDF value for "fox", we would count the number of documents in which it appears:

Number of documents with term "fox": 2

Total number of documents in the corpus: 2

IDF = log\_e(2 / 2) = 0

Thus, the TF-IDF value for "fox" in Document 1 is:

TF-IDF = 0.11 \* 0 = 0

This tells us that "fox" is not an important term in Document 1. We can perform similar calculations to determine the importance of other terms in the documents.

When using TF-IDF to extract features from URLs, we can treat the URL as a document and the individual terms within the URL as words. We can then calculate the TF-IDF values for each term in the URL and use them as features to train a machine learning model to detect phishing URLs.  
  
Example :

Suppose we have a dataset of URLs, where each URL is labeled as either "phishing" or "legitimate". We want to train a machine learning model to classify new URLs as either phishing or legitimate.

To extract features from the URLs, we can use TF-IDF. Here's how we might do it:

Tokenization: We split each URL into its component parts (e.g., the domain name, path, query string, etc.) and treat each part as a "term".

Term Frequency: For each URL, we count the number of times each term appears in the URL. This gives us a matrix of term frequencies, where each row represents a URL and each column represents a term.

Inverse Document Frequency: We calculate the IDF value for each term based on its frequency across all URLs. This gives us a vector of IDF values, one for each term.

TF-IDF: We multiply the term frequency matrix by the IDF vector to get a matrix of TF-IDF scores, where each row represents a URL and each column represents a term.

Once we have the TF-IDF matrix, we can use it to train a machine learning model to classify new URLs as either phishing or legitimate. We might use a classification algorithm like logistic regression or a decision tree to make predictions based on the TF-IDF scores.

For example, suppose we have the following two URLs:

URL 1: http://www.paypal-login.secure-account.com/login

URL 2: http://www.paypal.com/login

We might tokenize these URLs into the following terms:

URL 1: paypal, login, secure-account

URL 2: paypal, login

Then we might count the number of times each term appears in each URL:

|  |  |  |  |
| --- | --- | --- | --- |
|  | paypal | login | secure-account |
| URL 1 | 1 | 1 | 1 |
| URL 2 | 1 | 1 | 0 |

Next, we would calculate the IDF values for each term:

|  |  |
| --- | --- |
|  | IDF |
| paypal | 0.00 |
| Login | 0.00 |
| Secure-account | 0.69 |

Finally, we would multiply the term frequency matrix by the IDF vector to get the TF-IDF matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | paypal | login | secure-account |
| URL 1 | 0.00 | 0.00 | 0.47 |
| URL 2 | 0.00 | 0.00 | 0.00 |

We could then use these TF-IDF scores as features to train a machine learning model to classify new URLs as either phishing or legitimate.

but In our project, we did not use the above-mentioned technique of tokenizing URLs. Instead, we utilized a library like scikit that automatically selects a large number of features without the need for tokenization. The library selects common or rarely occurring sub-URLs, and we did not follow a systematic approach of tokenization to obtain better accuracy in classifying URLs.

**5.5 Summary and recommendations**

In chapter 5 of the graduation project, we have discussed the coding and implementation details of a tool for phishing URL detection using machine learning algorithms and natural language processing concepts.

We have used Node.js with the Express framework for the backend and HTML, CSS, JS, Bootstrap5, FontAwesome, and React for the front end. We also used Python programming language and the FastAPI library to create an API to use the machine learning model. The advantages of each technology were mentioned.

We used MySQL as our database system because it is free and easy to use and supported by all hosting services.

Five machine learning algorithms were used, including Linear SVC, Logistic Regression, Naive Bayes, Neural Network, and Random Forest Trees, and each algorithm was explained in detail along with a comparison table of their Precision, Recall, and F1 scores.

Finally, we have explained the use of the TF-IDF technique for extracting real features to train the machine learning model. A brief explanation of the TF-IDF technique was provided with examples of normal documents and URLs.

In conclusion, the implementation of the tool for phishing URL detection using machine learning algorithms and natural language processing concepts was discussed in detail. It is recommended to conduct further research and experimentation to improve the accuracy of the classification of URLs using machine learning algorithms. Also, it is recommended to use the latest techniques and technologies in this area, as the field is constantly evolving.

**Chapter 6**

**{** **Conclusions and Future Works** **}**

**6.1 Conclusions**

In this graduation project, we have developed a phishing detection system using machine learning algorithms and natural language processing (NLP) techniques. The system achieved an accuracy rate of 98% using the linear Support Vector Machine (SVM) algorithm.

Our system provides an effective and reliable approach to detecting phishing URLs, which is a critical task in cybersecurity. By analyzing the textual content of URLs and their associated features, our system can identify phishing URLs with high accuracy.

Overall, this project has demonstrated the effectiveness of machine learning and NLP techniques in detecting phishing URLs. The system can provide a valuable tool for organizations and individuals to improve their cybersecurity defenses and protect against phishing attacks.

**6.2 Future Works**

While the developed phishing detection system achieved a high level of accuracy, there are still some opportunities for future improvements.

**First**, we can consider using other machine learning algorithms and comparing their performance with the linear SVM algorithm. We can also explore other NLP techniques, such as sentiment analysis and topic modeling, to improve the accuracy of the system.

**Second**, we can integrate the system into existing cybersecurity systems to provide real-time phishing detection and response capabilities. This can help organizations to proactively detect and respond to phishing attacks before they cause significant harm.

**Finally**, we can explore the application of our system in other domains, such as social media and messaging applications, where phishing attacks are prevalent. This can provide a wider scope for our system and help to improve cybersecurity in various domains.

**6.3 Closing remarks**

In conclusion, this graduation project has developed a phishing detection system using machine learning algorithms and NLP techniques. The system achieved an accuracy rate of 98% using the linear SVM algorithm and demonstrated the effectiveness of these techniques in detecting phishing URLs.

As phishing attacks continue to evolve and become more sophisticated, it is essential to develop effective and reliable detection systems. Our system provides a valuable tool for organizations and individuals to improve their cybersecurity defenses and protect against phishing attacks.

We hope that this project can serve as a foundation for future research in the field of cybersecurity and contribute to the development of more advanced and sophisticated phishing detection systems.