



**KONGU POLYTECHNIC COLLEGE,
PERUNDURAI, ERODE -638060**



**IDENTIFICATION AND DETECTION OF PHISING WEBSITE
(SPOOFING ATTACK)**

SUBMITTED BY:

GOKUL R	(20502326)
JAGADEESH R	(20502332)
NIHIL K K	(20502346)
PRADEEP T	(20502352)
PRAVIN P R	(20502353)
SRI NITHISH V	(20502367)

GUIDED BY : Mr. P. ANAND BABU B.E.,

Of

KONGU POLYTECHNIC COLLEGE

(AN ISO 9001:2008 CERTIFIED INSTITUTION)

PERUNDURAI, ERODE-638060.

A REPORT ON PROJECT WORK

Submitted in partial fulfilment of the requirements for the award of

DIPLOMA IN COMPUTER ENGINEERING

OF

THE DIRECTORATE OF TECHNICAL EDUCATION



DEPARTMENT OF COMPUTER ENGINEERING
(ISO 9001:2008 Quality Certified Institution)
PERUNDURAI-638060

CERTIFICATE OF PROJECT APPROVAL

THIS IS TO CLARIFY THAT THE PROJECT REPORT ENTITLED
IDENTIFICATION AND DETECTION OF PHISING WEBSITE (SPOOFING ATTACK)
IS THE APPROVED WORK DONE BY

NAME :

REGISTER No :

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DIPLOMA
DURING THE YEAR 2019-2022

STAFF IN-CHARGE

HEAD OF THE DEPARTMENT

SUBMITTED FOR THE BOARD EXAMINATION HELD ON _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I wish to express my extreme gratefulness to our beloved Correspondent **Thirumathi. Malathi Elango** and all the trust members of Kongu Vellalar Institute of Technology Trust for providing all the necessary facilities to complete the project work successfully.

I express my deep sense of gratitude to our beloved Principal **Dr.V.Vedhagiri Eswaran**, for providing an opportunity to complete the project work successfully.

I express my gratitude to our Head of the Department **Mr.P.S.Dhanasekar**, Department of Computer Engineering for his valuable suggestions during the tenure of the project work successfully.

I am highly obliged to my supervisor **Mr.P.S.Dhanasekar**, Department of Computer Engineering, for his encouragement, and guidance in my project work successfully.

I extend my sincere thanks to the Teaching and Non-Teaching staffs for their valuable suggestions and help during my project work and also I like to express my sincere thanks to my beloved parents and trustworthy friends for their support towards the successful completion of the project work

TABLE OF CONTENTS

SL.NO	CHAPTERS	PAGE NO
	CERTIFICATE	
	DECLARATION	
	ACKNOWLEDGEMENT	
	ABSTRACT	
1	INTRODUCTION	
2	SYSTEM SPECIFICATION	
	2.1 HARDWARE SPECIFICATION	
	2.2 SOFTWARE SPECIFICATION	
	2.3 FEATURES OF SOFTWARE	
3	SYSTEM STUDY AND ANALYSIS	
	3.1 EXISTING SYSTEM	
	3.2 PROPOSED SYSTEM	
4	SYSTEM DESIGN	
	4.1 INPUT DESIGN	
	4.2 OUTPUT DESIGN	

	4.3 DATABASE DESIGN	
	4.4 DESCRIPTION OF MODULES	
5	SYSTEM TESTING AND IMPLEMENTATION	
	5.1 SYSTEM TESTING	
	5.2 SYSTEM IMPLEMENTATION	
6	CONCLUSION	
	APPENDIX	
	A. DATAFLOW DIAGRAM	
	C. SAMPLE CODING	
	D. SAMPLE FORMS	
	BIBLIOGRAPHY	

ABSTRACT

ABSTRACT

Phishing is a new type of network attack where the attacker creates a copy of an existing web page to fool users in to submitting personal, financial, or password data to what they think is their service provider's website.

This information can be used for future target advertisements or even identity theft attacks (e.g., transfer money from victims' bank account). The frequently used attack method is to send e-mails to potential victims, which seemed to be sent by banks, online organizations, or ISPs. In these e-mails, they will make up some causes, e.g. the password of your credit card had been mis-entered for many times, or they are providing upgrading services, to allure you visit their Web site to conform or modify your account number and password through the hyperlink provided in the e-mail. If you input the account number and password, the attackers then successfully collect the information at the server side, and is able to perform their next step actions with that information (e.g., withdraw money out from your account).

INTRODUCTION

1. INTRODUCTION

Phishing is another sort of organization assault where the aggressor makes a duplicate of a current page to trick clients in to submitting individual, monetary, or secret word information to their thought process is their specialist co-op's site.

This data can be utilized for future objective notices or even wholesale fraud assaults (e.g., move cash from casualties' financial balance). The much of the time utilized assault technique is to send messages to possible casualties, which appeared to be sent by banks, online associations, or ISPs. In these messages, they will make up certain purposes, for example the secret phrase of your Mastercard had been mis-entered for commonly, or they are giving updating administrations, to charm you visit their Web website to adjust or alter your record number and secret key through the hyperlink gave in the email. In the event that you input the record number and secret phrase, the aggressors effectively gather the data at the server side, and can play out their subsequent stage activities with that data (e.g., pull out cash out from your record).

The concept is an end-host based anti-phishing algorithm, called the Link Guard, by utilizing the generic characteristics of the hyperlinks in phishing attacks. The link Guard algorithm is the concept for finding the phishing emails sent by the phisher to grasp the information of the end user. Link Guard is based on the careful analysis of the characteristics of phishing hyperlinks. Each end user is implemented with Link Guard algorithm. After doing so the end user recognizes the phishing emails and can avoid responding to such mails. Since Link Guard is characteristics based it can detect and prevent not only known phishing attacks but also unknown ones.

The common characteristics of the hyperlinks in phishing e-mails. Our analysis identifies that the phishing hyperlinks share one or more characteristics as listed below:

- 1) The visual link and the actual link are not the same;
- 2) The attackers often use dotted decimal IP address instead of DNS name;

SYSTEM SPECIFICATION

2. SYSTEM SPECIFICATION

2.1 HARDWARE SPECIFICATION

PROCESSOR	: I5
HARD DISK CAPACITY	: 500 GB
MONITOR	: 14 “SAMTRON MONITOR
PRINTER	: TVS 80 COLOR
INTERNAL MEMORY CAPACITY	: 128 MB
KEYBOARD	: LOGITECH OF 104 KEYS
CPU CLOCK	: 1.08 GHz
MOUSE	: LOGITECH MOUSE

2.2 SOFTWARE SPECIFICATION

OPERATING SYSTEM	: WINDOWS 10
FRONT END	: ASP.NET
BACK END	: MS SQLSERVER

FEATURES OF SOFTWARES

ACTIVE SERVER PAGES:

Active Server Pages (ASP) is a technology that enables the development of dynamic web pages. ASP was developed by Microsoft to allow server side development. ASP files are HTML files with special tags containing source code that provide the dynamic content

Using Active server pages

- Generate dynamic web pages. An active server page can display different content to different user or display different content at different times of the day.
- Process the contents of HTML forms. We can use an active server page to retrieve and respond to the data entered into an HTML form.
- Can create database-driven web pages. An active server page can insert new data or retrieve existing data from a database such a Microsoft SQL Server, Microsoft Access.

Working of active server pages

An Active server page is a standard HTML file that is extended with additional features. Like a standard HTML file, an active server page contains HTML tag that can be interpreted and displayed by a web browser. Anything we could normally place in an HTML file Java applets, Blinking text, client side scripts, client-side Active X Controls we can place in an active server page. However, Active server page has three important features that make it unique.

- An active server page contains server side scripts.
- Active server page provides several built-in objects.
- Active server page can be extended with additional components.

ASP Objects and components

Active server pages include several built-in objects and installable ActiveX components.

Objects:

An object is something typically has methods, properties or collections. An object method determines the things we can do with the object. An objects property can be read or set to specify the state of the object. An object constitutes different sets of keys and value pairs related to the objects.

Components:

An ActiveX component is similar to an active server page built-in object. However, when we are using V, there is one important distance between components and object. A distance of a component must be explicitly created before it can be used.

ASP Objects:

Active server pages include several built-in objects. The following explain detail how to use each of the built-in objects. Built-in objects of ASP are

- Application Object
- Request object
- Response object
- Server object
- Session object
- Object context object

Application objects:

The application object is used to store and retrieve information that can be shared among all users of an application. For Example, we can use the Application Object to pass information between users of your web site.

Response objects:

The response object is used to send information back to a browser. We can use the response object to send output from our scripts to a browser.

Server objects:

The server object enables we to use various utility functions on the server object to control the length of time a script executes before it times out. We can also use the server object to create instance of other objects.

Session objects:

The session object can be used to store and retrieve information about particular users sessions. We can use the session object to store information that persists over the course of a visit by a user to our web site.

Object context object:

The object context object is used to control Active server page Transactions. The transactions are managed by the Microsoft transaction server (MTS).

The built in objects differ from normal objects. We don't need to create an instance of a built in object before we can use it in a script. The methods, collections and properties of a built in objects are automatically accessible throughout a web site application.

ASP Components:

Active server page components can be used to extend the power of our scripts components differ from the built-in-objects because there are typically used for more specialized tasks. The following list provides a brief overview of some of the components bundled with active server pages.

- Ad rotator component
- Browser capabilities component
- Content linking component
- Counters component
- Content rotator component
- Collaborations data objects
- ActiveX Data objects

VBScript

VBScript stands for Visual Basic Script, a scripting language developed by Microsoft to be

used with Microsoft products, mainly Internet Explorer. It has gone through many changes over the years and is now mainly used as the default scripting language of ASP. VBScript was created to allow web page developers the ability to create dynamic web pages for their viewers who used Internet Explorer. With HTML, not a lot can be done to make a web page interactive, but VBScript unlocked many tools like: the ability to print the current date and time, access to the web servers file system, and allow advanced web programmers to develop web applications.

HYPER TEXT MARKUP LANGUAGE (HTML)

HTML is an application of the Standard Generalized Markup Language (SGML), which was approved as an international standard in the year 1986. SGML provides a way to encode hyper documents so they can be interchanged.

SGML is also a Meta language for formally describing document markup system. Infact HTML uses SGML to define a language that describes a WWW hyper document's structure and inter connectivity.

Following the rigors of SGML, TBL bore HTML to the world in 1990. Since then, many of us have it to be easy to use but sometimes quite limiting. These limiting factors are being addressed but the World Wide Web Consortium (aka W3c) at MIT. But HTML had to start somewhere, and its success argues that it didn't start out too badly.

MS-SQL SERVER 2005

MS SQL Server is a powerful database management system and the user can create application that requires little or no programming. It supports GUI features and an entire programming language, Visual Studio Application which can be used to develop richer and more developed application.

There are quite a few reasons, the first being that SQL is a feature rich program that can handle

any database related task you have. You can create places to store your data build tools that make it easy to read and modify your database contents, and ask questions of your data. SQL is a relational database, a database that stores information about related objects. In MS SQL that database means a collection of tables that hold data. It collectively stores all the other related objects such as queries, forms and reports that are used to implement function effectively.

The MS SQL database can act as a back end database for .NET as a front end, MS SQL supports the user with its powerful database management functions. A beginner can create his/her own database very simply by some mouse clicks. Another good reason to use SQL as backend tool is that it is a component of the overwhelmingly popular Microsoft office software suite.

MS SQL however is a relational database, which means that you can define relationships among the data it contains. Relational database, are superior to flat file databases because you can store discrete information.

Microsoft SQL Server 2005 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. In this article, we'll cover six of the more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online. Let's take a brief look at each:

Enterprise Manager is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.

Service Manager is used to control the MSSQLServer (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQLServerAgent processes. An icon for this service normally resides in the system tray of machines running SQL Server. You can use Service Manager to start, stop or pause any one of these services.

Data Transformation Services (DTS) provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

Books Online is an often overlooked resource provided with SQL Server that contains answers to a variety of administrative, development and installation issues. It's a great resource to consult before turning to the Internet or technical support. Hopefully, this article has provided you with a brief introduction to the various tools available to Microsoft SQL Server users. Now get out there and give them a whirl!

➤ **SQL Server Architecture**

Microsoft SQL Server data is stored in databases. The data in a database is organized into the logical components visible to users. A database is also physically implemented as two or more files on disk. When using a database, you work primarily with the logical components such as tables, views,

procedures, and users. The physical implementation of files is largely transparent. Typically, only the database administrator needs to work with the physical implementation.

Each instance of SQL Server has four system databases (**master**, **model**, **tempdb**, and **msdb**) and one or more user databases. Some organizations have only one user database, containing all the data for their organization. Some organizations have different databases for each group in their organization, and sometimes a database used by a single application. For example, an organization could have one database for sales, one for payroll, one for a document management application, and so on. Sometimes an application uses only one database; other applications may access several databases. It is not necessary to run multiple copies of the SQL Server database engine to allow multiple users to access the databases on a server. An instance of the SQL Server is capable of handling thousands of users working in multiple databases at the same time. Each instance of SQL Server makes all databases in the instance available to all users that connect to the instance, subject to the defined security permissions.

When connecting to an instance of SQL Server, your connection is associated with a particular database on the server. This database is called the **current database**. You are usually connected to a database defined as your default database by the system administrator.

SQL Server allows you to **detach** databases from an instance of SQL Server, then **reattach** them to another instance, or even attach the database back to the same instance. If you have a SQL Server database file, you can tell SQL Server when you connect to attach that database file with a specific database name.

SYSTEM STUDY AND ANALYSIS

3. SYSTEM STUDY AND ANALYSIS

3.1 EXISTING SYSTEM

In the existing system can detect the phishing Web sites in time, we then can block the sites and prevent phishing attacks. It's relatively easy to (manually) determine whether a site is a phishing site or not, but it's difficult to find those phishing sites out in time. Here we list two methods for phishing site detection. It is therefore possible to detect this kind of download at the Web server and trace back to the phisher. Both approaches have shortcomings. For DNS scanning, it increases the overhead of the DNS systems and may cause problem for normal DNS queries, and furthermore, many phishing attacks simply do not require a DNS name. For phishing download detection, clever phishers may easily write tools which can mimic the behavior of human beings to defeat the detection.

DISADVANTAGES

- DNS problem.
- Difficult to detect phishing attack
- Enormous amount of time is consumed
- Manual prediction

3.2 PROPOSED SYSTEM

Our proposed system a user visits a Web site, the anti-phishing tool searches the address of that site in a blacklist stored in the database. If the visited site is on the list, the anti-phishing tool then warns the users. Tools in this category include Scam Blocker from the EarthLink Company, Phish Guard, and Net craft, etc. Though the developers of these tools all announced that they can update the blacklist in time, they cannot prevent the attacks from the newly emerged (unknown) phishing sites

ADVANTAGES

- Dynamic prediction of attack
- Early prediction of attack with user details
- It takes less time consuming

SYSTEM DESIGN

4. SYSTEM DESIGN

.1 INPUT DESIGN

Input design is the process of converting the user-oriented. Input to a computer based format. The goal of the input design is to make the data entry easier, logical and free error. Errors in the input data are controlled by the input design. The quality of the input determines the quality of the system output.

The entire data entry screen is interactive in nature, so that the user can directly enter into data according to the prompted messages. The users are also can directly enter into data according to the prompted messages. The users are also provided with option of selecting an appropriate input from a list of values. This will reduce the number of error, which are otherwise likely to arise if they were to be entered by the user itself.

Input design is one of the most important phases of the system design. Input design is the process where the input received in the system are planned and designed, so as to get necessary information from the user, eliminating the information that is not required. The aim of the input design is to ensure the maximum possible levels of accuracy and also ensures that the input is accessible that understood by the user. The input design is the part of overall system design, which requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify the errors. The objectives considered during input design are:

- Nature of input processing.
- Flexibility and thoroughness of validation rules.
- Handling of properties within the input documents.
- Screen design to ensure accuracy and efficiency of the input relationship with files.
- Careful design of the input also involves attention to error handling, controls, batching and validation procedures.
- Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the production of erroneous information

.2 OUTPUT DESIGN

Output design is very important concept in the computerized system, without reliable output the user may feel the entire system is unnecessary and avoids using it. The proper output design is important in any system and facilitates effective decision-making. The output design of this system includes various reports.

Computer output is the most important and direct source of information the user. Efficient, intelligible output design should improve the system's relationships with the user and help in decision making. A major form of output is the hardcopy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the data flow diagram. Human factors reduce issues for design involved addressing internal controls to ensure readability.

An application is successful only when it can provide efficient and effective reports. Reports are actually presentable form of the data. The report generation should be useful to the management for future reference. The reports are the main source of information for user's operators and management. Report generated are a permanent record of the transaction occurred. After any valid transactions; have commenced the report of the same are generations and: filed for future reference. Great care has been taken when designation the report as it plays an important role in decision-marking.

.3 DATABASE DESIGN

A well database is essential for the good performance of the system .several tables are referenced or manipulated at various instance. The table also knows as relation; provide information pertaining to a specified entity. Normalization of table is carried out to extent possible, while the normalizing tables, care should be taken to make sure that the number of tables do not exceed the optimum level, so that table maintenance. Is convenient and effective. The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Not all of these steps will be necessary in all cases. Usually, the designer must:

- Determine the data to be stored in the database
- Determine the relationships between the different data elements
- Superimpose a logical structure upon the data on the basis of these relationships.

Within the relational model the final step can generally be broken down into two further steps that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects. This step is not necessary with an database. In a majority of cases, the person who is doing the design of a database is a person with expertise in the area of database design, rather than expertise in the domain from which the data to be stored is drawn e.g. financial information, biological information etc. Therefore the data to be stored in the database must be determined in cooperation with a person who does have expertise in that domain, and who is aware of what data must be stored within the system.

DESCRIPTION OF MODULES

- New Customer Registration
- Customer Login
- Purchase Table
- Admin login
- Product details
- Customer Details
- Attack detection

Customer login

New Customer registration:

Used to Enter Customer Details who visit the Company. This module contains the details of customers like customer id, customer name, address, contact no, mail id, password and re enter password.

Customer Login:

The Customer can give their Username and Password to precede login. After the successful login the customer is allowed to view all the All Available Types of s and Sub Category of s Products and its details and Purchase too. The Images and Table representation is very useful for customers to know about the products features and other details.

View Products

This module is used to purchase for various type of s and raw materials required for customer from different location which includes product no, qty, price, purchasing date etc.,

Purchase Table

Purchase Module Shows the Products and its related details which was purchased by the Customer through online. Customer can also add or cancel the Purchased items from the purchase table. Product details like product name, product code, Quantity of each Product, total quantity of products, price of each product, discounts for product, Total price.

Admin Login

Admin Module only used by the Administrator. Administrator uses this module to add new

products, modify the old product Rates etc., Product details Sub category of materials. Product details are must important for sales, customer have an idea about their Product which is they going to buy so product administrator should provide the full details about their Product to the Customers need.

Attack detection

Admin can use this module. In this module he gives some use of the website. It will crawl the links inside that website. If it is an attack detected, it will added to the black list.

SYSTEM TESTING AND IMPLEMENTAION

5. SYSTEM TESTING AND IMPLEMENTATION

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system.

The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input. However the data are created with the intent of determining whether the system will process them correctly without any errors to produce the required output.

5.1 SYSTEM TESTING

- Unit testing
- Integration testing
- Validation testing
- Output testing
- User acceptance testing

Unit Testing

All modules were tested and individually as soon as they were completed and were checked for their correct functionality.

Integration Testing

The entire project was split into small program; each of this single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructors were used. It give a lot of

problem by not functioning is an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in frames are convenient and it is satisfied. when the frames were given for the test, the end user gave suggestion. Based on their suggestions the frames were modified and put into practice.

Validation Testing

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., Validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

Output Testing

After performing the validation testing the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system under consideration. Here the output format is considered in two ways. one is on screen and another one is printed format. The output format on the screen is found to be corrected as the format was designed in the system phase according to the user needs. And for the hardcopy the output comes according to the specifications requested by the user.

User Acceptance System

An acceptance test as the objective of selling the user on validity and reliability of the system. It verifies that the procedures operate to system specification and that the integrity of vital is maintained.

5.2 SYSTEM IMPLEMENTATION

The purpose of **System Implementation** can be summarized as follows:

It making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the Performing Organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the Consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support responsibilities involves changing from a system *development* to a system *support and maintenance* mode of operation, with ownership of the new system moving from the Project Team to the Performing Organization.

List of System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover Planning

Planning is the first task in the system implementation. Planning means deciding on the method and the time scale to be adopted. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments. The line managers controlled through an implementation coordinating committee. The committee considers ideas, problems and complaints of user department, it must also consider;

- The implication of system environment

- Self-selection and allocation from implementation tasks
- Consultation with unions and resources available
- Standby facilities and channels of communication

The following roles are involved in carrying out the processes of this phase. Detailed descriptions of these roles can be found in the Introductions to Sections I and III.

- Project Manager
- Project Sponsor
- Business Analyst
- Data/Process Modeler
- Technical Lead/Architect
- Application Developers
- Software Quality Assurance (SQA) Lead
- Technical Services (HW/SW, LAN/WAN, TelCom)

- **Information Security Officer**

CONCLUSION

7. CONCLUSION

Thus the concept is an end-host based anti-phishing algorithm, called the Link Guard, by utilizing the generic characteristics of the hyperlinks in phishing attacks. The link Guard algorithm is the concept for finding the phishing attack done by the phisher to grasp the information of the end user. One greater advantage of hosting web is to make the resource to know all and to become familiar which paves a way for development of organization as well as for profit. While hosting the software it should be attractive, user-friendly, and easier to access. This auction warehouse software constitutes all this characteristics

Some special features of this project are

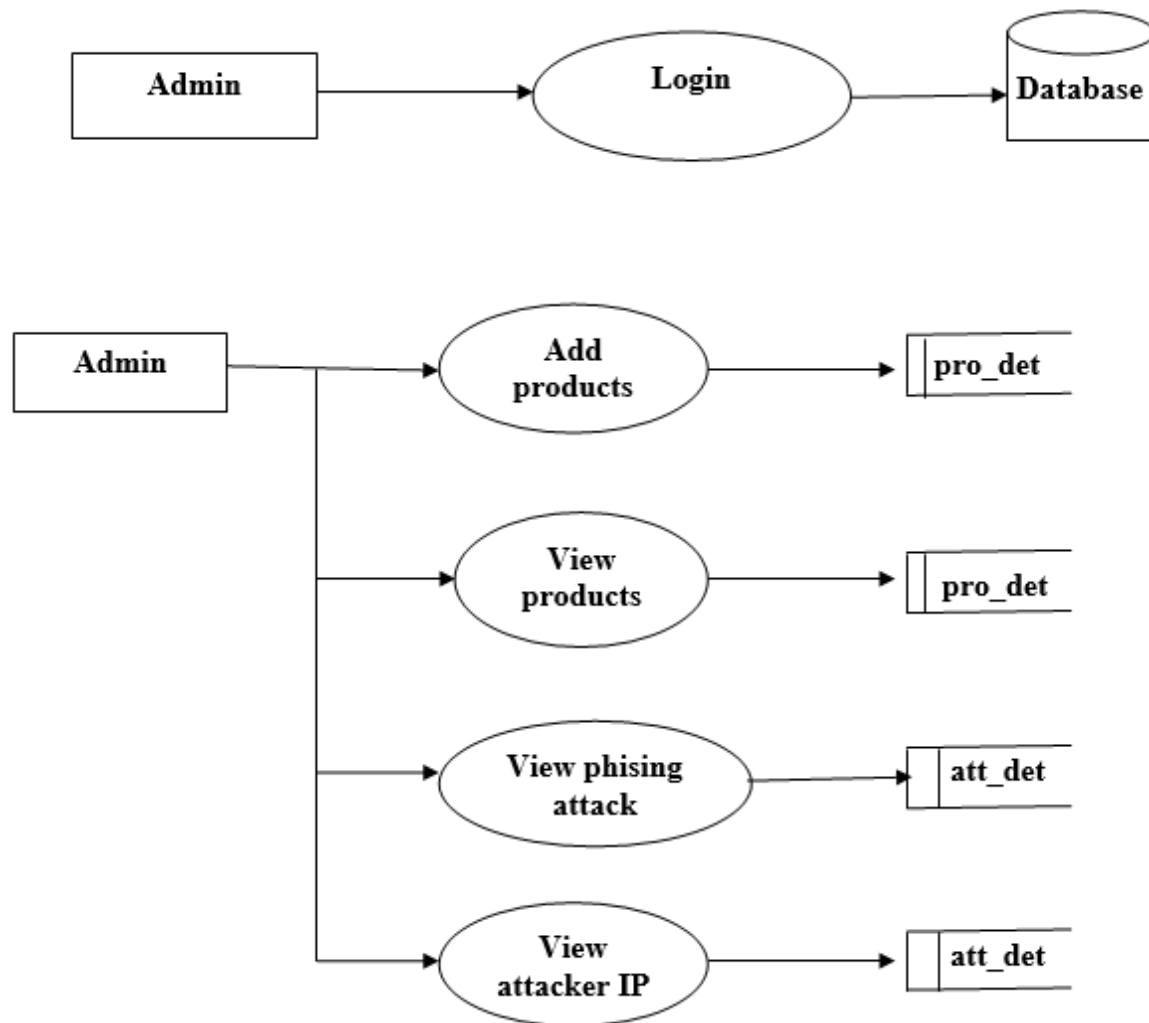
- It reduces the manpower to some extent
- It reduces the time cost.

APPENDIX

8. APPENDIX

DATA FLOW DIAGRAM

Level 0:



SAMPLE CODING

```
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
```

```

using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class _Default : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data Source=ADMIN\\SQLEXPRESS;Initial
    Catalog=ElectronicShopping;user id=sa;pwd=sql");
    SqlCommand cmd, cmd1 = new SqlCommand();
    SqlDataAdapter da, da1 = new SqlDataAdapter();
    DataSet ds, ds1;
    string str, mail, pwd, str2;
    string ipadd, logTime;
    protected void Page_Load(object sender, EventArgs e)
    {
        logTime = System.DateTime.Now.ToLongTimeString();
        if (!IsPostBack)
        {
            Label12.Visible = false;
            Label13.Visible = false;
            TextBox3.Visible = false;
            Button2.Visible = false;
            Label14.Visible = false;
            //Label11.Text = "WELCOME ";

        }
        ipadd = IpAddress();

    }
    public string IpAddress()
    {
        string strHostName = System.Net.Dns.GetHostName();
        string clientIPAddress = System.Net.Dns.GetHostAddresses(strHostName).GetValue(0).ToString();
    }
}

```

```

return clientIPAddress;
}
protected void Button1_Click(object sender, EventArgs e)
{
try
{
Session.Clear();
SqlCommand cmd = new SqlCommand("select email,password,uCnt,fname from users where email='" +
TextBox1.Text + "' and password='" + TextBox2.Text + "'", con);
SqlDataAdapter da = new SqlDataAdapter(cmd);
DataSet ds = new DataSet();
da.Fill(ds, "users");
int exec=ds.Tables[0].Rows.Count;
if(exec>0)
{
Session["userid"] = ds.Tables["users"].Rows[0][0].ToString();
Session["password"] = ds.Tables["users"].Rows[0][1].ToString();
Session["Uname"] = ds.Tables["users"].Rows[0][3].ToString(); ;
Session["inTime"] = System.DateTime.Now.ToLongTimeString();
Session["ipAddress"] = IPAddress();
}
else if (TextBox1.Text=="Admin" && TextBox2.Text=="Admin")
{
Response.Redirect("AdminViewProdRank.aspx");
}
else
{
string strHostName = null;
strHostName = System.Net.Dns.GetHostName();
cmd1.CommandText = "insert into HackerTry values('" + System.DateTime.Now.ToLongDateString() + "','" +
TextBox1.Text + "','" + TextBox2.Text + "','" + strHostName + "','" + ipadd + "','" + logTime + "' ,'" +
System.DateTime.Now.ToLongTimeString() + "')";
con.Open();
cmd1.Connection = con;

```

```

cmd1.ExecuteNonQuery();
con.Close();

}

Response.Redirect("useracc.aspx");

}
catch (Exception ex)
{
//MessageBox.Show(ex.ToString());
}
}
protected void Button2_Click(object sender, EventArgs e)
{
try
{
SqlCommand cmd1 = new SqlCommand("select password from users where email='" + TextBox3.Text + "'",
con);
con.Open();
SqlDataAdapter da1 = new SqlDataAdapter(cmd1);
DataSet ds1 = new DataSet();
da1.Fill(ds1, "users");
Label13.Text = ds1.Tables["users"].Rows[0][0].ToString();
}
catch
{
}
}
protected void Button3_Click(object sender, EventArgs e)
{

}

```

```

protected void LinkButton6_Click(object sender, EventArgs e)
{
    Response.Redirect("newuser.aspx");
}
protected void LinkButton7_Click(object sender, EventArgs e)
{

}
//private void UpdateTimer()
//{{
// // Label15.Text = System.DateTime.Now.ToLongTimeString();
//}}
//protected void Timer1_Tick(object sender, EventArgs e)
//{{
// // UpdateTimer();
//}}

protected void tmrLive_Tick(object sender, EventArgs e)
{
    Label15.Text = System.DateTime.Now.ToLongTimeString();
}
}

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;

```

```

using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;
using System.Windows.Forms;

public partial class _Default : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data Source=ADMIN\\SQLEXPRESS;Initial
    Catalog=ElectronicShopping;user id=sa;pwd=sql");
    SqlCommand cmd, cmd1 = new SqlCommand();
    SqlDataAdapter da, da1 = new SqlDataAdapter();
    DataSet ds, ds1;
    string str;
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {

    if (TextBox1.Text == "" || TextBox3.Text == "" || TextBox5.Text == "" || TextBox8.Text == "" ||
    TextBox10.Text == "")
    {
        MessageBox.Show("Dont Let Blank Fields");
    }
    else if (TextBox2.Text.Length <= 6)
    {
        MessageBox.Show("Password Must Be More Then 7 Chars");
    }
    else

```



```

{
try
{
str = "insert into users values('" + TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text + "','" +
TextBox4.Text + "','" + TextBox5.Text + "','" + TextBox6.Text + "','" + TextBox7.Text + "','" + TextBox8.Text
+ "','" + TextBox9.Text + "','" + TextBox10.Text + "','" + TextBox11.Text + "','" + TextBox12.Text + "','" +
DropDownList1.Text + "','" + DropDownList2.Text + "','" + TextBox13.Text + "',0)";
con.Open();
cmd = new SqlCommand(str, con);
cmd.ExecuteNonQuery();
MessageBox.Show("Registered Successfully");
Response.Redirect("login.aspx");
}
catch
{

}

}
}
protected void Button2_Click(object sender, EventArgs e)
{
TextBox1.Text = "";
TextBox2.Text = "";
TextBox3.Text = "";
TextBox4.Text = "";
TextBox5.Text = "";
TextBox6.Text = "";
TextBox7.Text = "";
TextBox8.Text = "";
TextBox9.Text = "";
TextBox10.Text = "";
TextBox11.Text = "";
TextBox12.Text = "";

```

```

TextBox13.Text = "";
DropDownList1.Text = "Select";
DropDownList2.Text = "Select";
}
protected void LinkButton6_Click(object sender, EventArgs e)
{
Response.Redirect("login.aspx");
}

}

```

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Windows.Forms;
public partial class Default3 : System.Web.UI.Page
{
SqlConnection con = new SqlConnection("Data Source=ADMIN\\SQLEXPRESS;Initial
Catalog=ElectronicShopping;user id=sa;pwd=sql");
SqlCommand cmd, cmd1 = new SqlCommand();
SqlDataAdapter da, da1 = new SqlDataAdapter();
DataSet ds, ds1;
string str,str2,pr,code,ss, cost;

```

```

protected void Page_Load(object sender, EventArgs e)
{
    Session["text1"] = "tt";
}

protected void ImageButton4_Click(object sender, ImageClickEventArgs e)
{
    Session["url"] = "";
    Session["url"] = "laptop.aspx";
    code = Label8.Text;
    cost = Label9.Text;

    sales(code);

    if (ss == code)
    {

    }

    else
    {
        link(code, cost);
        Session["imginfo"] = code;
        Response.Redirect("prodinfo.aspx");
    }
}

protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    Session["url"] = "";
    Session["url"] = "laptop.aspx";
    code = Label2.Text;
    cost = Label3.Text;
    sales(code);
    if (ss == code)
    {

```

```

    }
else
{
    link(code, cost);
    Session["imginfo"] = code;
    Response.Redirect("prodinfo.aspx");
}
}

protected void ImageButton2_Click(object sender, ImageClickEventArgs e)
{
    Session["url"] = "";
    Session["url"] = "laptop.aspx";
    code = Label4.Text;
    cost = Label5.Text;

    sales(code);

    if (ss == code)
    {

    }
else
{
    link(code, cost);
    Session["imginfo"] = code;
    Response.Redirect("prodinfo.aspx");
}
}

protected void ImageButton3_Click(object sender, ImageClickEventArgs e)
{
    Session["url"] = "";

```

```

Session["url"] = "laptop.aspx";
code = Label6.Text;
cost = Label7.Text;

sales(code);

if (ss == code)
{

}
else
{
link(code, cost);
Session["imginfo"] = code;
Response.Redirect("prodinfo.aspx");
}
}
protected void ImageButton5_Click(object sender, ImageClickEventArgs e)
{
Session["url"] = "";
Session["url"] = "laptop.aspx";
code = Label10.Text;
cost = Label11.Text;
sales(code);

if (ss == code)
{

}
else
{
link(code, cost);

```

```

Session["imginfo"] = code;
Response.Redirect("prodinfo.aspx");
}
}
protected void ImageButton6_Click(object sender, ImageClickEventArgs e)
{
Session["url"] = "";
Session["url"] = "laptop.aspx";
code = Label12.Text;
cost = Label13.Text;
sales(code);

if (ss == code)
{

}
else
{
link(code, cost);
Session["imginfo"] = code;
Response.Redirect("prodinfo.aspx");
}
}
protected void ImageButton7_Click(object sender, ImageClickEventArgs e)
{
Session["url"] = "";
Session["url"] = "laptop.aspx";
code = Label14.Text;
cost = Label15.Text;
sales(code);

if (ss == code)
{

```

```

    }
else
{
    link(code, cost);
    Session["imginfo"] = code;
    Response.Redirect("prodinfo.aspx");
}
}

protected void ImageButton8_Click(object sender, ImageClickEventArgs e)
{
    Session["url"] = "";
    Session["url"] = "laptop.aspx";
    code = Label16.Text;
    cost = Label17.Text;
    sales(code);
    sales(code);

    if (ss == code)
    {

    }
else
{
    link(code, cost);
    Session["imginfo"] = code;
    Response.Redirect("prodinfo.aspx");
}
}

protected void ImageButton9_Click(object sender, ImageClickEventArgs e)
{
    Session["url"] = "";

```

```
Session["url"] = "laptop.aspx";
```

```
code = Label18.Text;
```

```
cost = Label19.Text;
```

```
sales(code);
```

```
sales(code);
```

```
if (ss == code)
```

```
{
```

```
}
```

```
else
```

```
{
```

```
link(code, cost);
```

```
Session["imginfo"] = code;
```

```
Response.Redirect("prodinfo.aspx");
```

```
}
```

```
}
```

```
private void link(string code, string cost)
```

```
{
```

```
str = "insert into cart1 values('laptop','" + code + "','" + cost + "','0')";
```

```
str2 = "insert into purchase values('laptop','" + code + "','" + cost + "','0')";
```

```
//con.Open();
```

```
cmd = new SqlCommand(str, con);
```

```
cmd.ExecuteNonQuery();
```

```
cmd = new SqlCommand(str2, con);
```

```
cmd.ExecuteNonQuery();
```

```
}
```

```
private void sales(string code)
```

```
{
```



```

con.Open();
SqlCommand cmd2 = new SqlCommand("select prodid from cart1 where prodid='" + code + "'", con);
SqlDataAdapter da2 = new SqlDataAdapter(cmd2);
DataSet ds2 = new DataSet();
da2.Fill(ds2, "cart1");

if (ds2.Tables[0].Rows.Count > 0)
{
pr = ds2.Tables["cart1"].Rows[0][0].ToString();
ss = pr;
if (code == pr)
{
MessageBox.Show("Product Already In Cart");
}
}

}
}

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

```

```

public partial class _Default : System.Web.UI.Page
{
    SqlConnection con=new SqlConnection("Data Source=ADMIN\\SQLEXPRESS;Initial
Catalog=ElectronicShopping;user id=sa;pwd=sql") ;
    SqlCommand cmd,cmd1=new SqlCommand();
    SqlDataAdapter da,da1=new SqlDataAdapter();
    DataSet ds,ds1;
    string str;

    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            SqlCommand cmd = new SqlCommand("select imgname,name,price,imginfo from info where imgname ='' +
            Session["imginfo"] +''",con );
            con.Open();
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            con.Close();
            DataSet ds =new DataSet() ;
            da.Fill(ds, "info");
            Label4.Text=ds.Tables["info"].Rows[0][0].ToString();
            Label2.Text=ds.Tables["info"].Rows[0][1].ToString();
            Label3.Text = ds.Tables["info"].Rows[0][2].ToString();
            string imge = ds.Tables["info"].Rows[0][3].ToString();
            Image2.ImageUrl = "~/products/" + imge + "";
            Image3.ImageUrl = "~/Feacture/"+imge + "";

        }
        catch(Exception )
        {

        }
    }
}

```

```

}
protected void Button2_Click(object sender, EventArgs e)
{
    Response.Redirect("CART.aspx");
}
protected void Button3_Click(object sender, EventArgs e)
{

}
protected void Button1_Click(object sender, EventArgs e)
{
    string ses = Session["url"].ToString();

    Response.Redirect (ses );
}

private void link(string code, string cost)
{
    str = ("insert into cart values(\Laptop\','"
    + (code + ("'\','"
    + (cost + "'\','0\')"))));
    cmd = new SqlCommand(str, con);
    cmd.ExecuteNonQuery();
}
}

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;

```

```

using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class ProductRanking : System.Web.UI.Page
{
    SqlConnection conn = new SqlConnection();
    protected void Page_Load(object sender, EventArgs e)
    {
        conn.ConnectionString = "Data Source=ADMIN\\SQLEXPRESS;Initial Catalog=ElectronicShopping;user
id=sa;pwd=sql";
        Load_GridData(); // call method below
    }

    void Load_GridData()
    {
        conn.Open(); // open the connection
        SqlDataAdapter Sqa = new SqlDataAdapter("select top 10 name,price,imginfo from info order by rank desc",
conn);
        DataSet ds = new DataSet();
        Sqa.Fill(ds); // fill the dataset
        GridView1.DataSource = ds; // give data to GridView
        GridView1.DataBind();
        conn.Close();
    }
}

using System;
using System.Collections;
using System.Configuration;

```

```

using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;
using System.Windows.Forms;

public partial class _Default : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data Source=ADMIN\\SQLEXPRESS;Initial
    Catalog=ElectronicShopping;user id=sa;pwd=sql");
    string str,v1,v2;
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        try
        {
            SqlCommand cmd = new SqlCommand("select
            fname,lname,cname,vatno,mobile,phone,address,address2,city,state,country,zip from users where fname='" +
            TextBox4.Text + "'", con);
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            da.Fill(ds, "users");

            TextBox4.Text = ds.Tables["users"].Rows[0][0].ToString();
            TextBox5.Text = ds.Tables["users"].Rows[0][1].ToString();

```

```

TextBox6.Text = ds.Tables["users"].Rows[0][2].ToString();
TextBox7.Text = ds.Tables["users"].Rows[0][3].ToString();
TextBox8.Text = ds.Tables["users"].Rows[0][4].ToString();
TextBox9.Text = ds.Tables["users"].Rows[0][5].ToString();
TextBox10.Text = ds.Tables["users"].Rows[0][6].ToString();
TextBox11.Text = ds.Tables["users"].Rows[0][7].ToString();
TextBox12.Text = ds.Tables["users"].Rows[0][8].ToString();
DropDownList1.Text = ds.Tables["users"].Rows[0][9].ToString();
DropDownList2.Text = ds.Tables["users"].Rows[0][10].ToString();
TextBox13.Text = ds.Tables["users"].Rows[0][11].ToString();

}
catch (Exception)
{
}

}

protected void Button2_Click(object sender, EventArgs e)
{
if ( TextBox4.Text == "" || TextBox5.Text == "" || TextBox8.Text == "" || TextBox10.Text == "" )
{
MessageBox.Show("Dont Let Blank Fields");
}
else

try
{
SqlCommand cmd = new SqlCommand("update users set lname=" + TextBox5.Text + ",cname=" +
TextBox6.Text + ",vatno=" + TextBox7.Text + ",mobile=" + TextBox8.Text + ",phone=" + TextBox9.Text
+ ",address=" + TextBox10.Text + ",address2=" + TextBox11.Text + ",city=" + TextBox12.Text +
",state=" + DropDownList1.Text + ",country=" + DropDownList2.Text + ",zip=" + TextBox13.Text + "
from users where fname=" + TextBox4.Text + """, con);
con.Open();

```

```

SqlDataAdapter da = new SqlDataAdapter(cmd);
cmd.ExecuteNonQuery();
MessageBox.Show("Updated Successfully");
}
catch
{

}

}

protected void Button3_Click(object sender, EventArgs e)
{
if (TextBox4.Text == "" || TextBox5.Text == "" || TextBox8.Text == "" || TextBox10.Text == "")
{
MessageBox.Show("Dont Let Blank Fields");
}
else

try
{
Response.Redirect("payment.aspx");
}
catch
{

}

}
protected void LinkButton6_Click(object sender, EventArgs e)
{

```

```
Response.Redirect("Login.aspx");  
}  
}
```

SAMPLE FORMS

Firefox | Electronic Shopping | localhost:62778/phishing attacks/Login.aspx

PHISING ATTACK

Home | New arrivals | My Account

PRODUCTS

- LAPTOP
- MOBILE
- PENDRIVE
- MOUSE
- SPEAKER
- PRODUCT RANKING

CREATE A NEW ACCOUNT WITH US

[CLICK HERE TO CREATE A NEW ACCOUNT](#)

SIGN IN TO YOUR ACCOUNT

USER ID

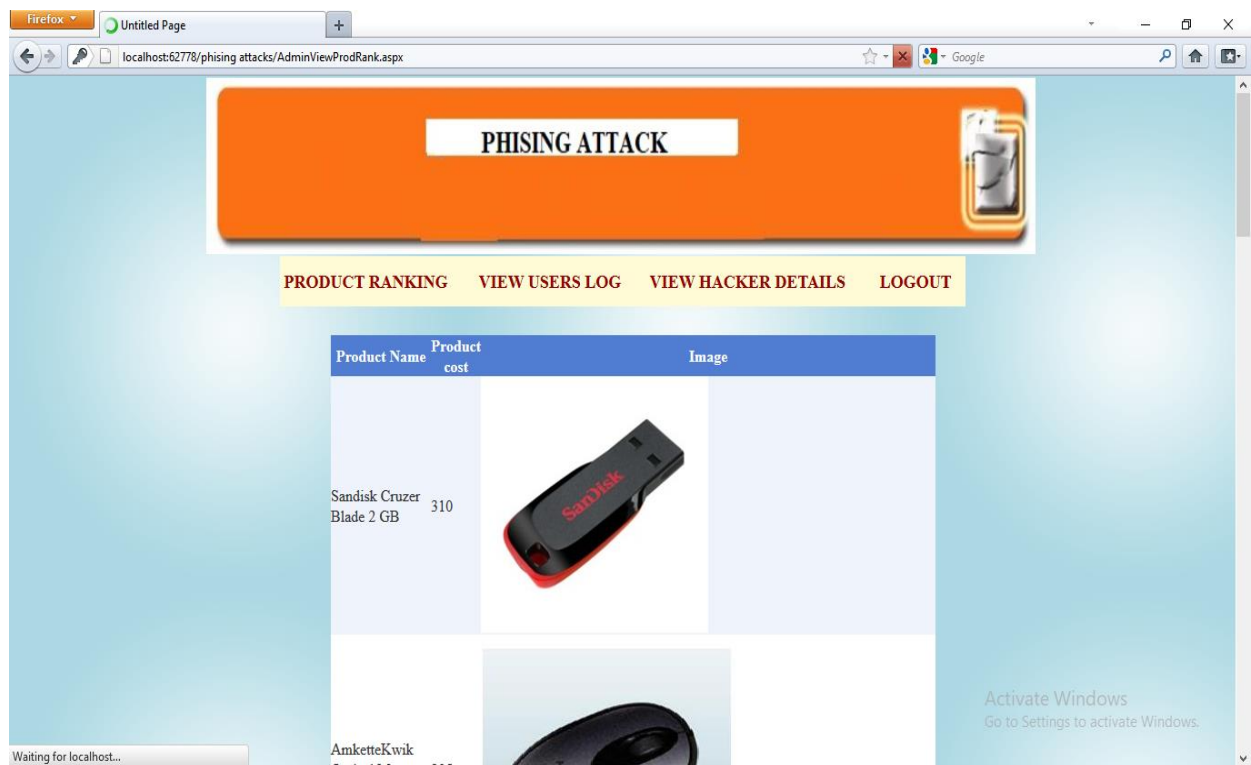
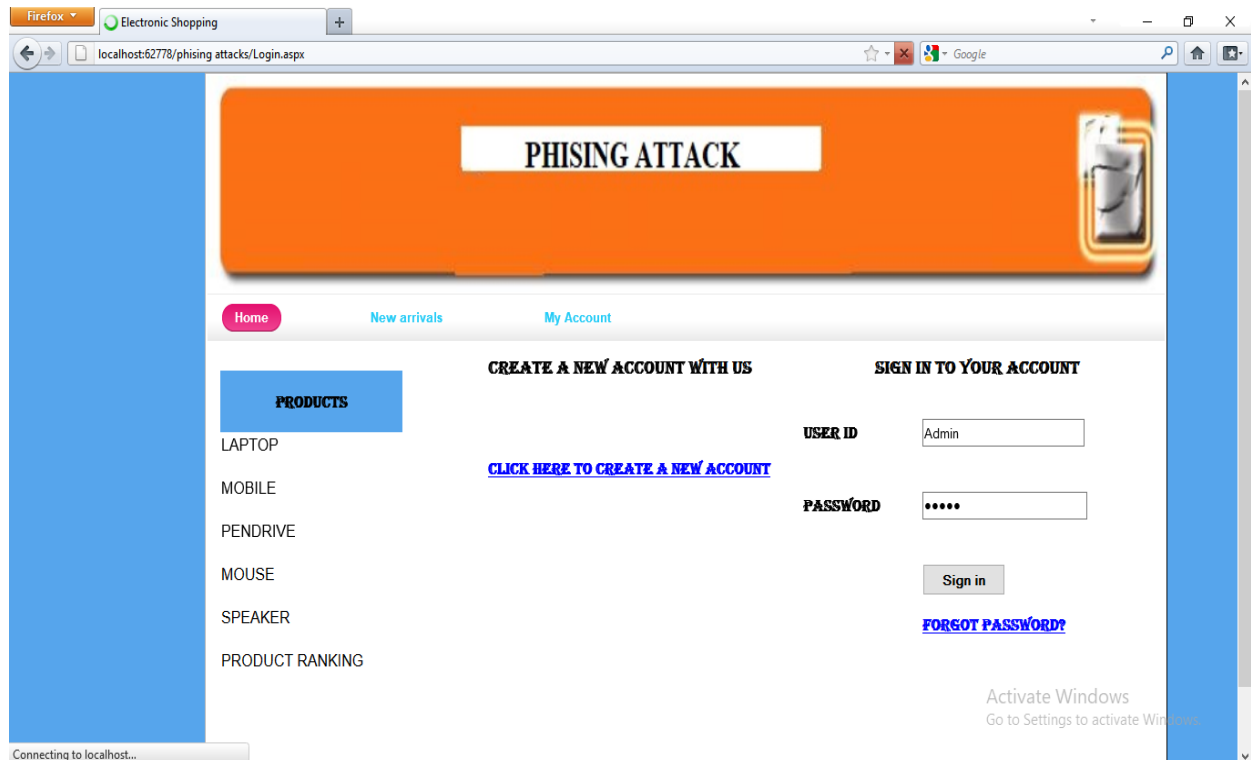
PASSWORD

Sign in

[FORGOT PASSWORD?](#)

Activate Windows
Go to Settings to activate Windows.

Connecting to localhost...



Firefox | Untitled Page | localhost:62778/phising attacks/AdminUserLog.aspx

PHISING ATTACK

[PRODUCT RANKING](#)
[VIEW USERS LOG](#)
[VIEW HACKER DETAILS](#)
[LOGOUT](#)

E-Mail Address	User Name	User IP	TimeIn	TimeOut	User Count
a@gmail.com	a	192.168.1.31	4:19:32 PM	4:19:45 PM	3
crispsystem@gmail.com	Gowtham	192.168.1.31	1:09:19 PM	1:09:45 PM	1

Activate Windows
Go to Settings to activate Windows.

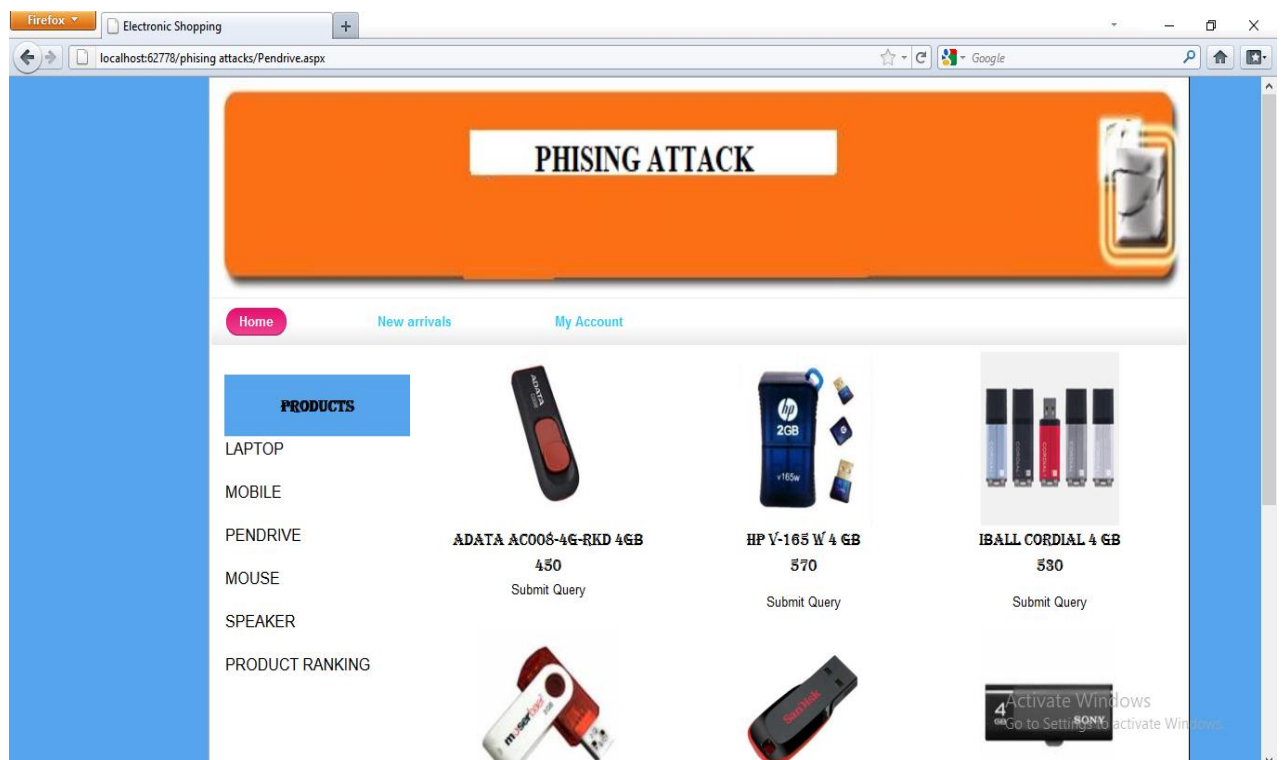
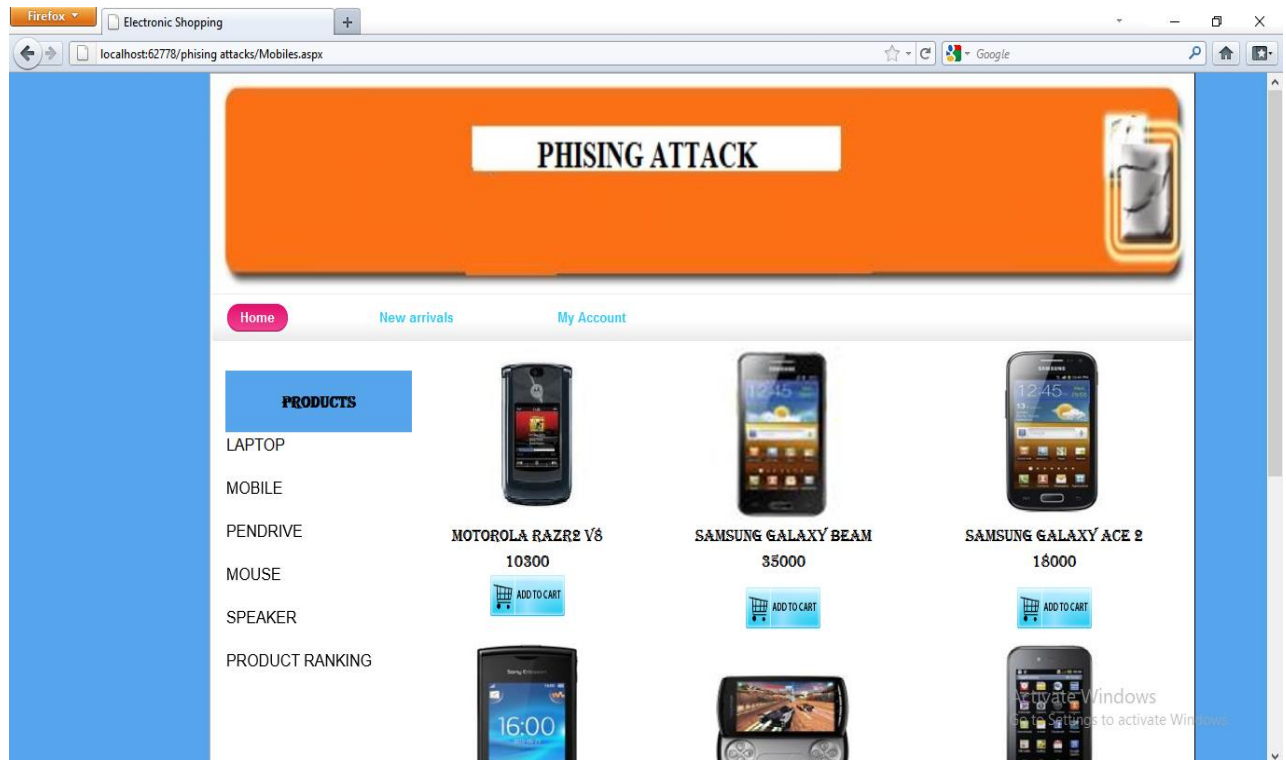
Firefox | Untitled Page | localhost:62778/phising attacks/AdminHackerDetail.aspx

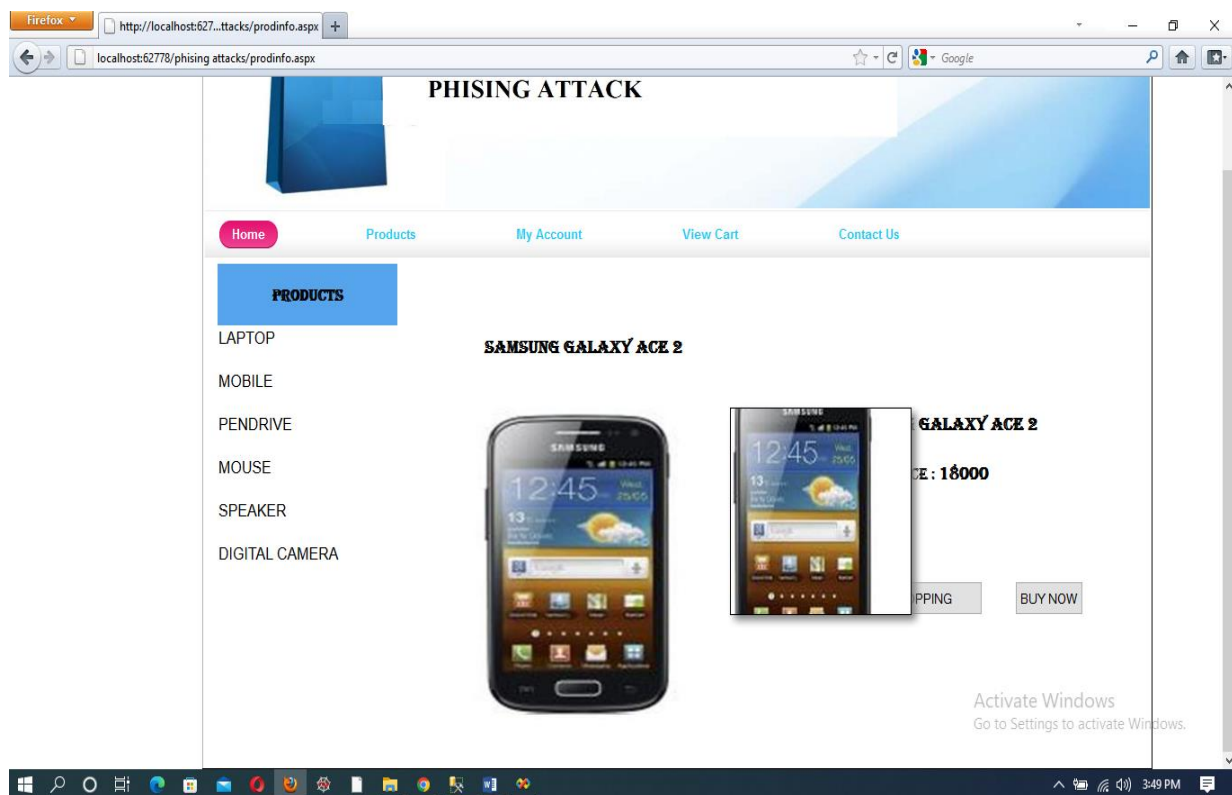
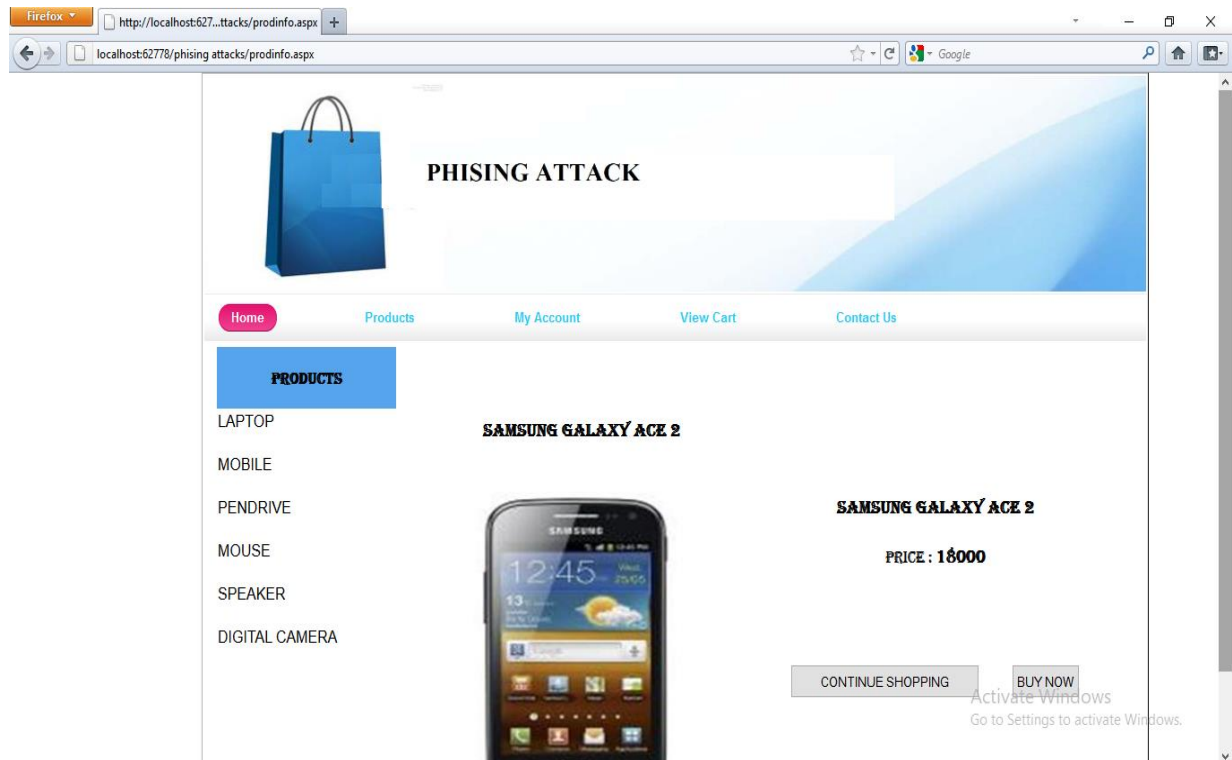
PHISING ATTACK

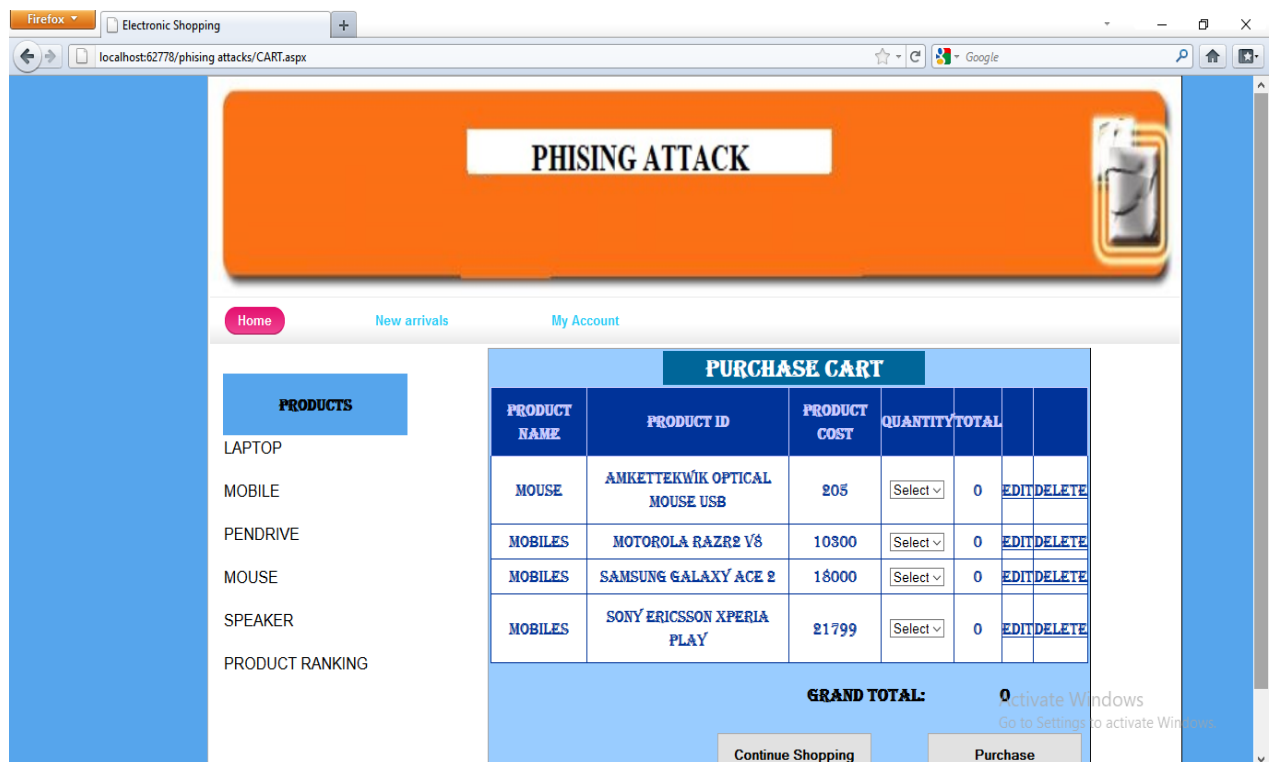
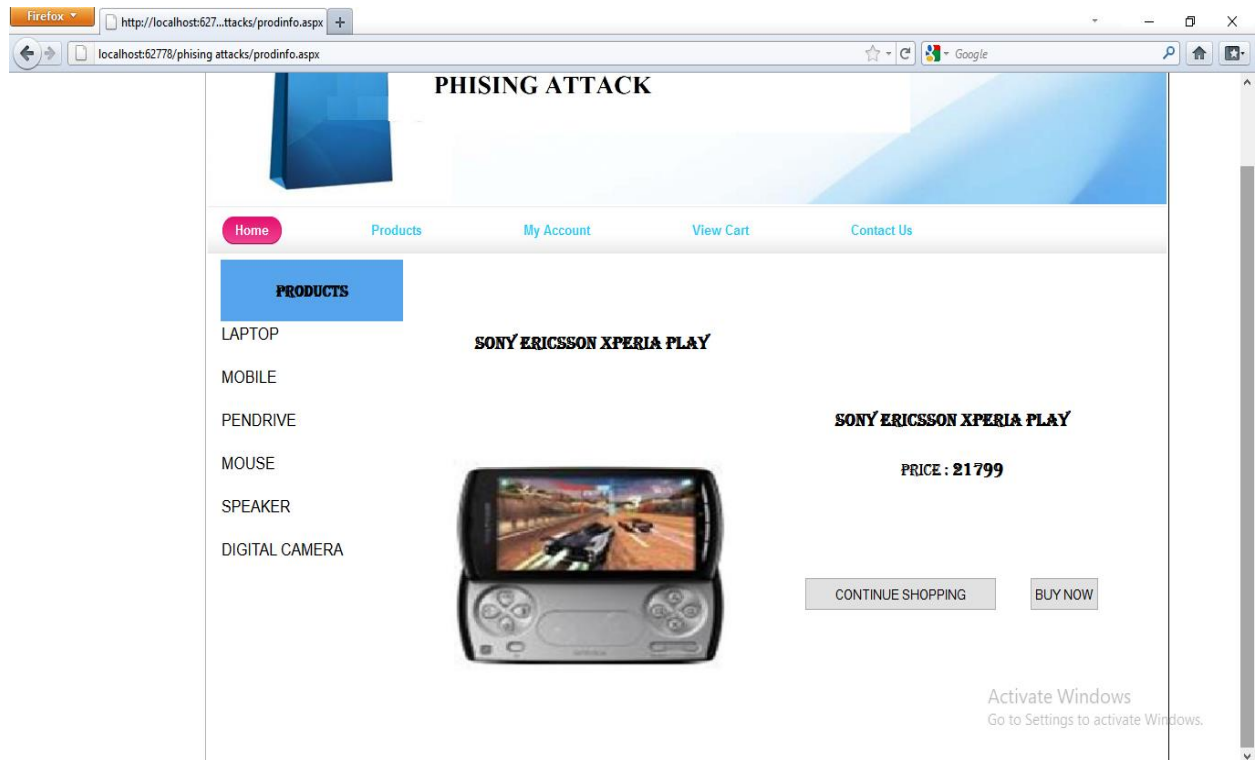
[PRODUCT RANKING](#)
[VIEW USERS LOG](#)
[VIEW HACKER DETAILS](#)
[LOGOUT](#)

T_Day	T_ID	Tried Password	Server Name	IP Address	Log Time
Wednesday, April 30, 2022	admin	ADMIN	crisp-5	192.168.1.31	11:09:16 AM
Wednesday, April 30, 2022	Admin	Admin	crisp-5	192.168.1.31	11:09:28 AM

Activate Windows
Go to Settings to activate Windows.







Firefox Untitled Page
localhost:62778/phising attacks/Login1.aspx

PHISING ATTACK

Create a New Account With Us

[Click Here To Create a New Account](#)

Sign in To Your Account

User ID

Password

[Forgot Password?](#)

Activate Windows
Go to Settings to activate Windows.

Firefox Untitled Page
localhost:62778/phising attacks/shipping1.aspx

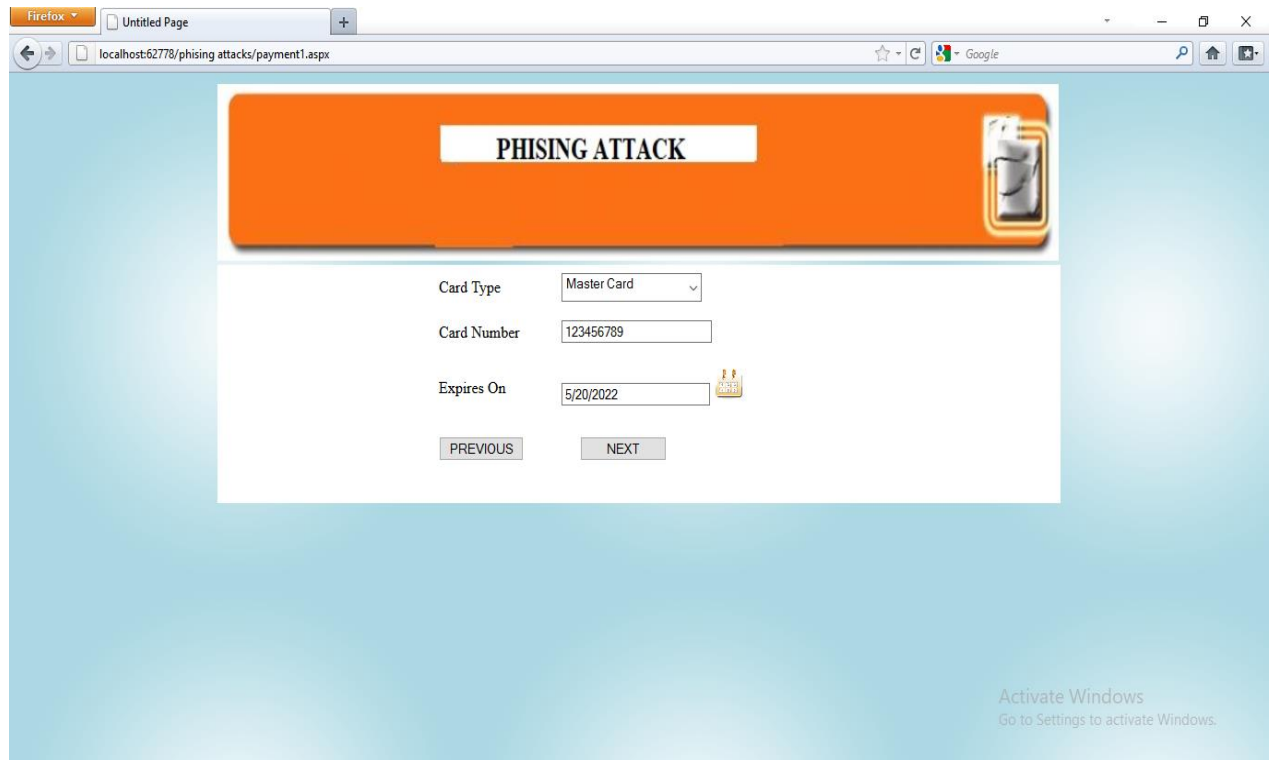
[BACK...](#)

SHIPPING DETAILS

First Name:	<input type="text" value="RAM"/>
Last Name:	<input type="text" value="KUMAR"/>
Company Name:	<input type="text" value="tcs"/>
Company VAT No:	<input type="text" value="12222"/>
Mobile No.:	<input type="text" value="9898989898"/>
Phone No.:	<input type="text" value="7676767676"/>
Street:	<input type="text" value="east"/>
Nearby Locality:	<input type="text" value="gg hospital"/>
City:	<input type="text" value="salem"/>
State/Province:	<input type="text" value="Tamil Nadu"/>
Country:	<input type="text" value="India"/>
Zip/Postcode:	<input type="text" value="545443"/>

* Edit and Click Save to save address to your Profile

Activate Windows
Go to Settings to activate Windows.



BIBLIOGRAPHY

BOOKS:

1) Software engineering

By

Roger S.Pressman

Fifth Edition,

Mc.Graw Hill International Edition 2001

2) System Analysis And Design

Elais M.Awad, Second Edition

Galgotia Publications (P) Ltd

Published by Sunil Galgotia, 2002

WEBSITES

- www.asptutorial.com
- www.pscore.com
- www.w3schools.com
- www.hotscripts.com
- www.freesoft.in