SCHOOL OF VOCATIONAL EDUCATION AND TRAINING, IGNOU, MAIDAN GARHI, NEW DELHI – 110 068

II. PROFORMA FOR THE APPROVAL OF PGDIS PROJECT PROPOSAL (MSEP-028)

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

Project Proposal No :.....
(for office use only)

Enrolment No.: 172047194
Study Centre: L.D. Arts College
Regional Centre: 09 **RC Code:** 09-01
E-mail: yamini.ce.24.@gmail.com
Mobile/Tel No.: 756.7.804550

1. Name and Address of the Student: Yamini Dineshkumar Rathod
B 202, Smita Apartment, Radio Mirchi Road
Satellite, Ahmedabad - 380015

2. Title of the Project: Security analysis and security implementation on Inventory management-web application & case study on related cyber crimes.

Signature of the Student
Date: 12-02-2018

For Office Use Only

☑ Approved , ☐ Not Approved

Signature, Designation, Stamp of the Project Proposal Evaluator
Date: .....

Suggestions for reformulating the Project: A well written proposal. It is, however, advised that security policy should be comprehensively defined beforehand.

Ensure that you include the following while submitting the Project Proposal:

1. **Proforma for Approval of Project Proposal duly filled and signed by the student with date.**
2. **Synopsis of the project proposal (12-15 pages).**
3. **A self-addressed envelope with duly affixed postage stamps (to send it by speed post) on it.**

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

MSEP - 028

TITLE OF THE PROJECT

"Security analysis and security implementation on Inventory management web application & case study on related cyber crimes".

by

Yamini Dineshkumar Rathod

Enrolment No: 172047194

Submitted to the School of Vocational Education and Training, IGNOU

in partial fulfillment of the requirements

for the award of the degree

PG Diploma in Information Security (PGDIS)

June 2018

Indira Gandhi National Open University

Maidan Garhi

New Delhi – 110068

CERTIFICATE OF ORIGINALITY

This is to certify that the project report entitled "Security analysis and security implementation on Inventory management web application & case study on related cyber crimes" submitted to **Indira Gandhi National Open University** in partial fulfillment of the requirement for the award of the degree of **PG DIPLOMA IN INFORMATION SECURITY(PGDIS),** is an authentic and original work carried out by me.

The matter embodied in this project is genuine work done by me and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirements of any course of study.

…….

Signature of the Student

Date: ……..

Name and Address of the student

……..

……..

……..

……..

Enrolment No……

Table of Contents

Chapter 1

Introduction

1.1 Methodology

1.1.1 Rational Unified Process

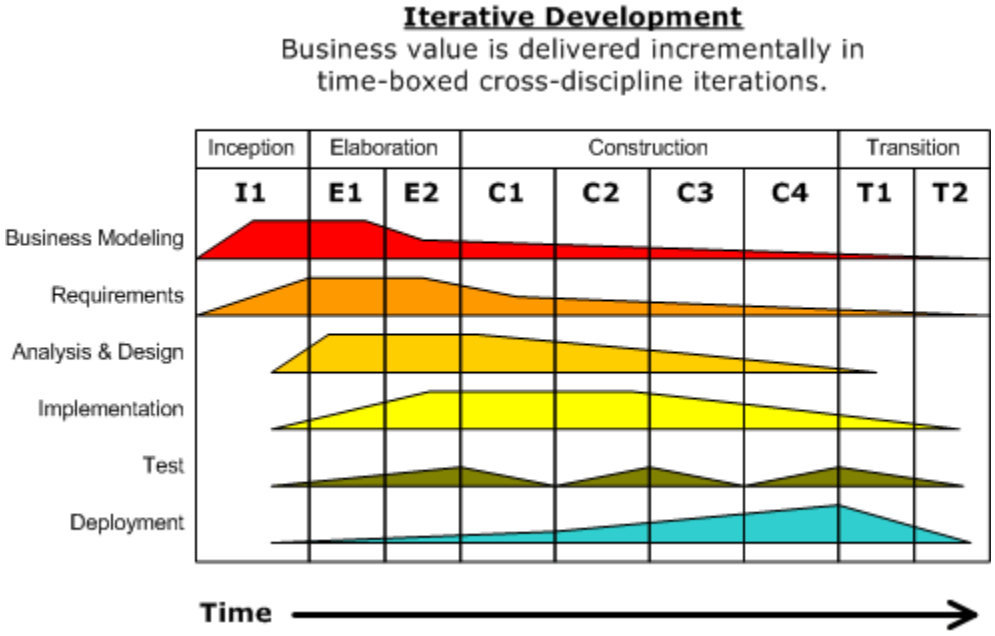
This software engineering process has three branches from all of the generic process models that a standard software engineering project would requires or is entitled to, it also supports regular revisions and it also provides guidelines of "standard practices" in system analysis and design.

This model is best described by following three perspectives:

Static | This phase shows the set of actions that are carried out over development period

Dynamic | This phase shows an overview of phases that covers major aspects of development

Practice | These phases are guidelines that provides the bases of standards that should be followed



Inception | The objective of this phase is to provide the outline to the organization for the system. This includes identification of all the factors and entities that will be the part of communication system either to the system or from the system. Collection of this information will help create the business case for the proposed system.

Elaboration | This phase mainly encloses the entities related to the bases of understanding the problem, its domain, the framework of architecture and the project plan and most importantly the stakeholder and risk associated with each of them.

Construction |Although previous phase may include the activities related to establishing the grounding framework of this project, this phase is actually dedicated to the overall system design, implementation and testing. Parts of the system may be developed in small sized units and finally are coupled together during this phase.

Transition |This last phase is concerned with deploying the system to the customer base from the testing and development environment.

1.1.2 Purpose

The main purpose of this project is to focus on real life application scenario in terms of information security. The agenda would be to build an application and implement the security by identifying the "security loop holes" and analysis on cyber crimes.

The reason behind taking this project is to combine the theoretical knowledge and practical knowledge of real life scenario.

Supermarket Inventory System is to facilitate our customers to track their products as and when they are transported from the vendor to the warehouse and from the warehouse to the retail location to the customers.

It also aims to acquaint the user with the position of the supermarket currently by producing various sale graphs of items based on what quantity of a certain item is sold on a daily, weekly, monthly and yearly basis.

It is necessary to keep our resources safe and protected. In order to implement security in application it would be done by implementing encryption, keeping secure session base password, implementing two level authentications, observing system logs and security faults, analyzing network flow using wireshark, implementing wireshark, preventing the application validation from un-necessary inputs, session management, session hijacking, hacking, cross site scripting and implementing code to prevent from SQL injection and many more.

1.1.3 Scope

This Supermarket Inventory System creates purchase orders once the inventory level reaches to a pre-defined level. Supermarkets and the vendor's warehouse use this system to create receipt and invoice. The accounting department uses this system to match invoice and receipt so that the payment can be recorded accurately.

By this project we will be able to focus on both small and big retail stores in helping manage their Inventory of their store with security implementation. If taken in more general form it can be used to manage inventory of even Production House's and Warehouse's.

The Inventory Management System is an application designed to allow the supermarket staff to create, maintain and view the contents and value of its inventory of items in a categorized way.

It also aims to analyze the position of the supermarket in the market and help it know what items to order in what quantity by producing graphs depicting sale of different items on different basis such as monthly, yearly, brand type etc.

The main goal as of now is to implement application by considering security loop holes. We will analyze and implement web security in this project followed by case study on cyber crimes in India.

By security inputs, it will decrease the vulnerability of application being from hacked and attacks. It provides the surety and trust to the customers that their resources are safe and secure. Customers trust should be an organizations job.

It also determines the use of information security at its pick level to resolve the real world problems.

1.1.4 Overview

The Inventory Management System is an application designed to allow the supermarket staff to create, maintain and view the contents and value of its inventory of items in a categorized way.

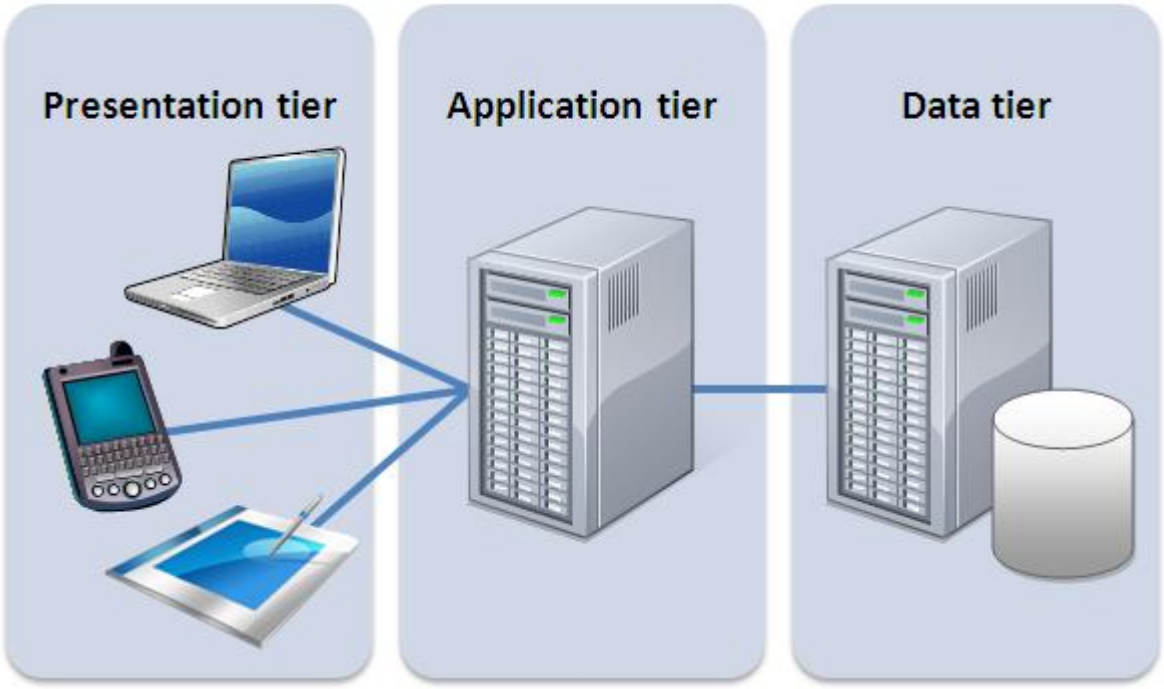
It also aims to analyze the position of the supermarket in the market and help it know what items to order in what quantity by producing graphs depicting sale of different items on different basis such as monthly, yearly, brand type etc.

1.2 Tools Used

1.2.1 Application Architecture: Three Tier Architecture

Three tier architecture

Three-tier architecture is a client–server software architecture pattern in which the user interface(presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.



The three tiers in three-tier architecture are:

1. Presentation Tier: Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.

2. Application Tier: Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

Data Tier: Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

1.2.2 Application Programming Architecture: HTML, CSS, PHP, JavaScript

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code can be simply mixed with HTML code, or it can be used in combination with various template engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable.

After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

JavaScript, often abbreviated as **JS**, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the ECMA Script specification, with some engines not supporting the spec fully, and with many engines supporting additional features beyond ECMA.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents. These pages contain information on how to learn and use CSS and on available software. They also contain news from the CSS working group.

1.2.3 Web server – WAMP

WAMP is an acronym for an archetypal model of web service solution stacks, originally consisting of largely interchangeable components: Windows, the Apache HTTP Server, the MySQL relational database management system, and the PHP programming language. As a solution stack, LAMP is suitable for building dynamic web sites and web applications.

1.2.4 Development tool – RAD

Jetbrains PHP Storm is an IDE that can support the development of PHP application by providing code hinting debugging, database visualization and many more.

1.2.5 Database platform – MySQL

MySQL the world's second most widely used open-source Relational Database Management System (RDBMS).

1.2.6 Network Security Platform – WireShark

Wireshark is a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.

Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows. There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License.

1.2.7 Web Application Security – Encryption using Linux Turbo C

In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it. Encryption does not itself prevent interference, but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm – a cipher – generating ciphertext that can only be read if decrypted.

For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a well-designed encryption scheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.

1.2.8 System Security Logs – Fiddler

Fiddler is an HTTP debugging proxy server application written by Eric Lawrence, formerly a Program Manager on the Internet Explorer development team at Microsoft.

Fiddler captures HTTP and HTTPS traffic and logs it for the user to review (the latter by implementing man-in-the-middle interception using self-signed certificates).

Fiddler can also be used to modify ("fiddle with") HTTP traffic for troubleshooting purposes as it is being sent or received. By default, traffic from Microsoft's WinINET HTTP(S) stack is automatically directed to the proxy at runtime, but any browser or Web application (and most mobile devices) can be configured to route its traffic through Fiddler.

1.3 Technologies to be used

PHP: Server side scripting

HTML : Responsive website deign

JQuery: Sophisticated JavaScript library

CSS : Adding style (e.g., fonts, colors, spacing) to Web documents

MySQL: Relational database system

Wamp Server: Rapid application developer

Linux Turbo C : Encryption algorithm survey

Wire-Shark : Source packet analyzer

Fiddler : An HTTP debugging proxy server application

1.4 Overview

1.4.1 Existing System

There might be possible existing system but there is lack of security. We are trying to build efficient system with security prevention.

1.4.2 Proposed System

Supermarket Inventory management system is a tool for tracking asset levels, order management, safety stock, sales and deliveries. It would help to avoid product overstock and outages.

It is a system that contains a list of orders to be received and then prompts workers to pick the necessary items, and provides them with packaging and shipping with high level of security.

1.4.3 Our plan

Release the proposed system at the completion of major project under semester 2 for course of Post Graduate Diploma in Information Security.

Chapter 2

Research Methodology

2.1 Study of different software development lifecycle model

Studied below software models as a part of research methodology.

2.1.1 The Linear Sequential Model

2.1.2 The Prototyping Model

2.1.3 The RAD Model

2.1.4 Evolutionary Software Process Models

2.1.4.1 The Incremental Model

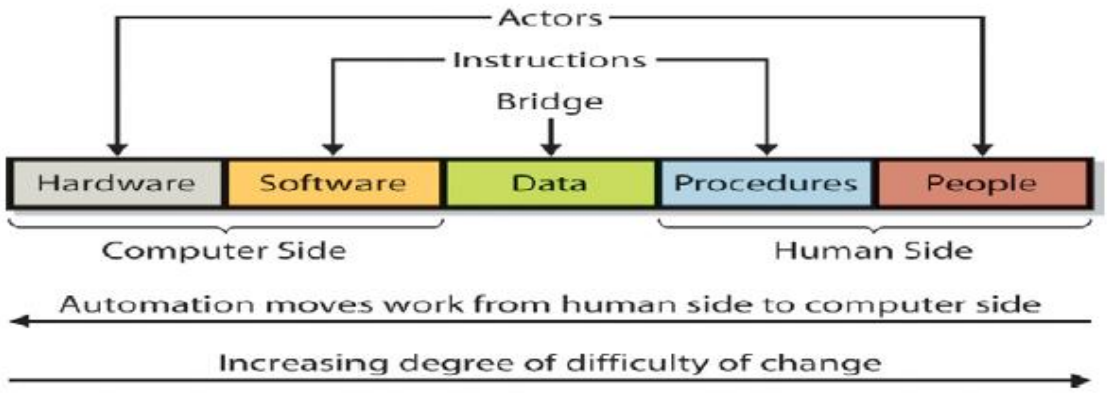
2.1.4.2 The Spiral Model

2.2. Study of inventory system

Inventory Management Systems is a key instrument for businesses when tracking their inventory.

Typically, Inventory Management Systems are used by firms that either sell a product or manufacture a product for purposes of accounting for all the tangible goods that allow for a sale of a finished product, or parts for making a product. The size and volume of a firm help dictate whether or not a firm is in need of such a system as they can be quite extensive and costly. Large firms that have thousands of components must have a system in place for the primary objective of tracking their assets.

There are three main reasons why an Inventory Management System is needed such as timing/lead time, forecasting, and utilizing economies of scale.



There are various ways companies control their inventory, finding the correct program/software that suits the need of each business takes careful consideration. However, from our perspective the outcome of inventory control systems are heavily influenced by the users, for example employees, as well management.

2.3. Study of information security concepts

Studied encryption algorithms as a part of research methodology.

2.3.1 Comparison of different Encryption algorithms

2.3.1.1 Basics of Encryption

2.3.1.2 Ciphers

2.3.1.3 History

2.3.1.4 Importance

2.3.1.5 Types of Cryptography

2.3.1.6 Importance of Encryption

2.3.1.7 AES Algorithm

2.3.1.8 Symmetric key DES

2.3.1.9 Asymmetric key RSA

2.3.1.10 Algorithm comparison criteria

I have implemented DES and RSA algorithms in C language on platform Linux, that has been demonstrated in detail in upcoming chapters.

Comparison Criteria

1. Input data size
2. Time
3. Throughput
4. Key Used
5. Scalability
6. Avalanche Effect
7. Power consumption
8. Security
9. Confidentiality

2.3.1.11 Comparison between DES and RSA

Comparison between DES and RSA

Features	DES	RSA
Key Used	Same key is used for encryption and decryption purpose	Different keys are used for encryption and decryption purpose
Scalability	It is scalable algorithm due to varying the key size and block list	No scalability occurs
Avalanche Effect	No more effected	More effected
Power consumption	Low	High
Throughput	Very High	Low
Confidentiality	High	Low

2.3.1.12 Hash function MD2 Algorithm

2.3.1.13 Hash function MD4 Algorithm

2.3.1.14 Hash function MD5 Algorithm

2.3.1.15 Hash function SHA1 Algorithm

2.3.1.16 Hash function Crypt Algorithm

2.3.1.17 Hash function SHA256 Algorithm

2.3.1.18 Cryptanalytic attacks

Cryptanalytic Attacks

It can be classified by how much information needed by the attacker:

1. Ciphertext-only attack

Given: Encryption Algorithm, Ciphertext

2. Known-plaintext attack

Given: Encryption Algorithm, Ciphertext, One or more plaintext-ciphertext pairs formed with the secret key

3. Chosen-plaintext attack

Given: Encryption Algorithm, Ciphertext, Plaintext message chosen by cryptanalyst together with its corresponding ciphertext generated with the secret key

4. Chosen-ciphertext attack

Given: Encryption Algorithm, ciphertext, ciphertext chosen by cryptanalyst together with its corresponding decrypted plaintext generated with the secret key.

5. Chosen Text

Given: Encryption Algorithm, ciphertext, plaintext message chosen by cryptanalyst together with its corresponding ciphertext generated with the secret key, ciphertext chosen by cryptanalyst together with its corresponding decrypted plaintext generated with the secret key

2.3.2 Study of Wireshark

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable.

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed.

Wireshark is perhaps one of the best open source packet analyzers available today.

Here are some examples people use Wireshark for:

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

Beside these examples Wireshark can be helpful in many other situations too.

The following are some of the many features Wireshark provides:

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.

- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

2.3.3 Study of Fiddler logs

2.3.3.1 Web Session Manipulation

It allows to edit web sessions easily, just set a breakpoint to pause the processing of the session and permit alteration of the request/response. Compose your own HTTP requests and run them through Fiddler.

2.3.3.2 Performance Testing

HTTP caching and compression at a glance. Isolate performance bottlenecks with rules such as “Flag any uncompressed responses larger than 25kb.”

2.3.3.3 Security Testing

Using fiddler we can decrypt HTTPS traffic and display and modify web application requests using a man-in-the-middle decryption technique. Configure Fiddler to decrypt all traffic, or only specific sessions.

2.3.3.4 HTTP/HTTPS Traffic Recording

To log all HTTP(S) traffic between your computer and the Internet. Debug traffic from virtually any application that supports a proxy (IE, Chrome, Safari, Firefox, Opera and more).

2.3.3.5 Web Debugging

Debug traffic from PC, Mac or Linux systems and mobile (iOS and Android) devices. Ensure the proper cookies, headers and cache directives are transferred between the client and server. Supports any framework, including .NET, Java, Ruby, etc.

2.3.4 Study of code security to prevent software piracy

2.3.4.1 Secure code generation

A large percentage of recent security problems, such as Cross-site Scripting, Session Hijacking or SQL injection, is caused by string-based code injection vulnerabilities. These vulnerabilities exist because of implicit code creation through string serialization. Based on an analysis of the vulnerability class underlying mechanisms, we propose a general approach to outfit modern programming languages with mandatory means for explicit and secure code generation which provide strict separation between data and code. Using an exemplified implementation for the languages Java and HTML/JavaScript respectively, we show how our approach can be realized and enforced. In order to prevent from such attacks it is mandatory to design an application in such a way that the vulnerability of the attacks gets minimum as possible. Programming code should follow certain rules and standards. We have implemented the secure code in this project and will be demonstrated in upcoming chapters.

2.3.4.2 Complexity of code

Code should be written in well manner and it must be very specific and clear.

It is good to comment it well so that it become understandable by third party.

It should be easy to debug and easy to fix.

Complexity of codes can be judged on bases of many parameters.

1. Throughput

2. Reliability

3. Timeliness

4. Performance

5. Time taken to execute code

6. Space

7. Number of users logged-in

8. Load balancing

These parameters are helpful to build an application to prevent the attacks such as MIM, DDOS if the code is designed to handle the complexity of inputs.

2.3.5 Study on Cyber Attacks

2.3.5.1 Session Hijacking

In computer science, session hijacking, sometimes also known as cookie hijacking is the exploitation of a valid computer session—sometimes also called a session key—to gain unauthorized access to information or services in a computer system.

Session hijacking, also known as TCP session hijacking, is a method of taking over a Web user session by surreptitiously obtaining the session ID and masquerading as the authorized user. Once the user's session ID has been accessed (through session prediction), the attacker can masquerade as that user and do anything the user is authorized to do on the network.

2.3.5.2 SQL Injection

SQL injection is a code injection technique that might destroy your database. IT is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements, via web page input.

SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

Look at the following example which creates a SELECT statement by adding a variable (myUserId) to a select string. The variable is fetched from user input (getRequestString):

myUserId = getRequestString("UserId");

SQL = "SELECT * FROM Users WHERE UserId = " + myUserId;

SQL Injection Based on 1=1 is Always True

Look at the example above again. The original purpose of the code was to create an SQL statement to select a user, with a given user id.

If there is nothing to prevent a user from entering "wrong" input, the user can enter some "smart" input like this:

UserId : 101 OR 1=1

Then, the SQL statement will look like this:

SELECT * FROM Users WHERE UserId = 105 OR 1=1;

The SQL above is valid and will return ALL rows from the "Users" table, since OR 1=1 is always TRUE.

Does the example above look dangerous? What if the "Users" table contains names and passwords?

The SQL statement above is much the same as this:

SELECT UserId, Name, Password FROM Users WHERE UserId = 105 or 1=1;

A hacker might get access to all the user names and passwords in a database, by simply inserting 105 OR 1=1 into the input field.

2.3.5.3 Cross site scripting

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

2.3.5.4 Characteristics of Viruses

Boot Sector Virus: A Boot Sector Virus infects the first sector of the hard drive, where the Master Boot Record (MBR) is stored. The Master Boot Record (MBR) stores the disk's primary partition table and to store bootstrapping instructions which are executed after the computer's BIOS passes execution to machine code.

File Deleting Viruses: A File Deleting Virus is designed to delete critical files which are the part of Operating System or data files.

Mass Mailer Viruses: Mass Mailer Viruses search e-mail programs like MS outlook for e-mail addresses which are stored in the address book and replicate by e-mailing themselves to the addresses stored in the address book of the e-mail program.

Macro viruses: Macro viruses are written by using the Macro programming languages like VBA, which is a feature of MS office package.

Polymorphic Viruses: Polymorphic Viruses have the capability to change their appearance and change their code every time they infect a different system. This helps the Polymorphic Viruses to hide from anti-virus software.

Stealth viruses: Stealth viruses have the capability to hide from operating system or anti-virus software by making changes to file sizes or directory structure.

Retrovirus: Retrovirus is another type virus which tries to attack and disable the anti-virus application running on the computer.

2.3.5.5 Validation

1. Prospective Validation.

This type of validation is performed before production, during a product's development stage. A risk analysis is performed to assess the production process by breaking it down into separate steps.

2. Concurrent Validation.

We should monitor the first three batches produced on a production-scale as closely as possible. The data gathered through this step can provide an in-depth insight of the fundamentals, which greatly impacts the effectiveness of concurrent validation.

3. Retrospective Validation.

As the name suggests, retrospective validation is rather like validation in hindsight. It involves examining the past experiences of the process and evaluating the final control tests.

4. Revalidation.

Revalidation is essential for ensuring that any changes made to the process or its environment have not resulted in adverse effects on product quality or process characteristics.

5. Client side and Server side Validation.

Validation performed for the inputs entered by users and it are validated at server end before storing it on database server.

Validation is required to prevent web form abuse by malicious users. Improper validation of form data is one of the main causes of security vulnerabilities. It exposes your website to attacks such as header injections, cross-site scripting, and SQL injections.

2.3.5.6 Authentication faults

The lack of strong authentication method sometimes leads attacks like Man In Middle, Session Hijacking.

Detailed demo and description is illustrated in upcoming chapters.

2.3.5.7 Absence of permission policy

In computer systems security, role-based access control (RBAC) is an approach to restricting system access to authorized users. It is used by the majority of enterprises with more than 500 employees, and

can implement mandatory access control (MAC) or discretionary access control (DAC). RBAC is sometimes referred to as role-based security.

Role-based-access-control (RBAC) is a policy neutral access control mechanism defined around roles and privileges. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments.

Detailed demo and description is illustrated in upcoming chapters.

2.4 Conclusion

In this chapter I studied different software development lifecycle models. The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. This is really important in order to build a better planned project.

Also, we reviewed different encryption techniques as a part of information security. This project is more based and focused on security. Encryption is a key concept in information security. I have tried to build an application that more secure in nature as compared to other applications.

The more details information about encryption algorithms are provided in next chapters, I have compare all the techniques and decided the best algorithm that we can build with our application.

I have also reviewed the cryptographic attacks that can be done on network end and the techniques to prevent from them.

Implementation and prevention of all possible security attacks such as validation, SQ injection, XSS, Cross Site Request Forgery, Session Hijacking, Session Management, Authentication policy and many more has been illustrated in detail in upcoming chapter.

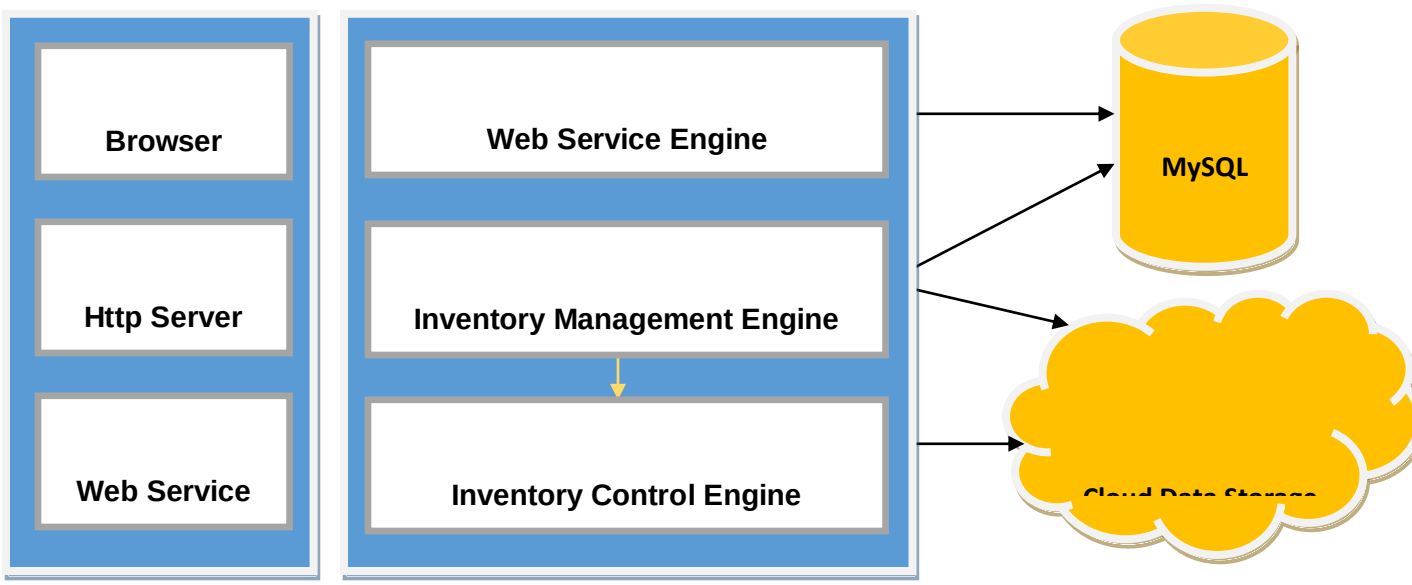
Wireshark and fiddler are used to established network level security and will elaborate this in next upcoming chapters.

Chapter 3

System Design

3.1 Product perspective

3.1.1 Software Interface



3.1.2 Client on Internet

1. Web Browser
2. Operating System (any)

3.1.3 Web Server

1. WAMP
2. Operating System (Windows)

3.1.4 Data Base Server

1. MySQL
2. Operating System (windows)

3.1.5 Development End

1. RAD (PHP, HTML, XML, AJAX, JQuery)
2. MYSQL (Database Server)
3. OS (Windows)
4. WAMP (Web Server)

3.1.6 Information Security

1. WireShark
2. Encryption
3. Validation
4. Session Management
5. SQL Injection Prevention
6. Session Hijacking Prevention
7. Cross site Scripting Prevention
8. Cross site Request Forgery Prevention
9. Two Level Authentication
10. Strong Password Validation
11. Secured Digest Authentication
12. Cookie Management
13. Access Permission Authorization
14. Fiddler

3.2 Hardware Interface

3.2.1 Minimum requirements

Client Side			
	Processor	Ram	Disk Space
Internet explorer 8	Intel Pentium III or AMD - 800 MHz	128 MB	100 MB

Server Side

	Processor	Ram	Disk Space
RAD	Intel Pentium III or AMD - 800 MHz	1 GB	3.5 GB
MySQL	Intel Pentium III or AMD - 800 MHz	256 MB	150 MB

3.2.2 Recommended requirements

Client Side			
	Processor	Ram	Disk Space
RAD	All Intel or AMD - 1 GHZ	256 MB	100 MB

Server Side			
	Processor	Ram	Disk Space
Internet explorer 8	All Intel or AMD - 2 GHZ	2 GB	3.5 GB
MySQL	All Intel or AMD - 2 GHZ	512 MB	500 MB

3.3 Communication Interface

Client (customer) on Internet will be using HTTP/HTTPS protocol.
Client (system user) on Internet will be using HTTP/HTTPS protocol

3.4 Constraints

GUI is only in English.
Login and password is used for the identification of users.
Only registered users will be authorized to use the services.
Limited to HTTP/HTTPS.

3.5 Use Case Model Survey

3.5.1 Owner

Representative	Owner
Description	The main personnel taking care of the entire Super market System
Type	The person is the technically sound and knows everything about the Hardware and Software System
Responsibilities	The key responsibility of the user is to handle the login account and the inventory activity of the other users. It also has to take care of the entire database.
Success Criteria	How does the user define success? How is the user rewarded?
Involvement	He is the one using the entire system and also giving its review and requirement as maybe needed.
Deliverables	User will produce the login account and can give other permission rights.
Comments / Issues	Problems that interfere with success and any other relevant information. Trends that make the user's job easier or harder

3.5.2 Manager

Representative	Manager
Description	To simply login the system and get the knowledge about the Inventory available.
Type	The person will be an expert having knowledge of using the system and other features.
Responsibilities	The key responsibilities is to keep information about the graphs and data
Success Criteria	The success is to get the output of the fired query
Involvement	No involvement
Deliverables	N/A
Comments / Issues	Problems that interfere with success and any other relevant information. Trends that make the user's job easier or harder

3.5.3 Store Keeper

Representative	Store Keeper
Description	To simply login the system and get the knowledge about the Inventory available.
Type	The person can be a novice having just basic knowledge of using the system
Responsibilities	No key responsibilities
Success Criteria	The success is to get the output of the fired query
Involvement	No involvement
Deliverables	N/A

Comments / Issues	Problems that interfere with success and any other relevant information. Trends that make the user's job easier or harder

3.6 Overall Description

3.6.1 Product Perspective

The software requires a connection to a database server containing the inventory database. The program will be executed as a standalone application on a single machine. The application may be executed on multiple machines simultaneously. The user will interact with the program via a GUI. The user will use both the mouse and keyboard for input and all information will be outputted to the monitor.

3.6.2 Problem Statement

The problem of	Inventory management in supermarkets
Affects	Customers, Manufacturers, Retailers
The impact of which is	Overstock, Outages, ignorance of profitable goods
A successful solution would be to	Design of a database for storing the entire inventory, a front end for user interaction with the system, and analysis of daily, weekly, monthly or yearly sales on different types and brands.

3.6.3 Product Functions

3.6.3.1 Add Function

This function will be used to add new merchandize.

3.6.3.2 Delete Function

This function will be used to delete merchandize. This may be necessary if a user inputs wrong information.

3.6.3.3 Sell Function

This function will remove an item from the active inventory by marking it as sold.

3.6.3.4 Sales Function

This function will be used to generate statistics about sales of merchandize. The function will be able to calculate the sales within a given time period. The function will determine the number of sales and the total amount.

3.6.3.5 Connect Function

3.7 Product Positioning System

For	Manager, store-keeper and owner
The (product name)	Inventory Management System
That	Avoids tedious hand-keeping format for storing various information and provides efficient, user-friendly computer base management system

3.8 User Summary

[Present a summary list of all the identified users:]

Name	Description	Stakeholder
Owner Manager Store Keeper	To keep check on the whole working of the super market (both market and inventory). To keep track record and market analysis of the entire inventory.	

	To keep track record of the currently available quantity.	

3.9 User Environment

The total number of user depends on the employee given authority to check the inventory. The time required to check the inventory using system is less than a minute. The system setup is purely indoor and can be used remotely if internet facility is used.

3.10 Database Design

Database

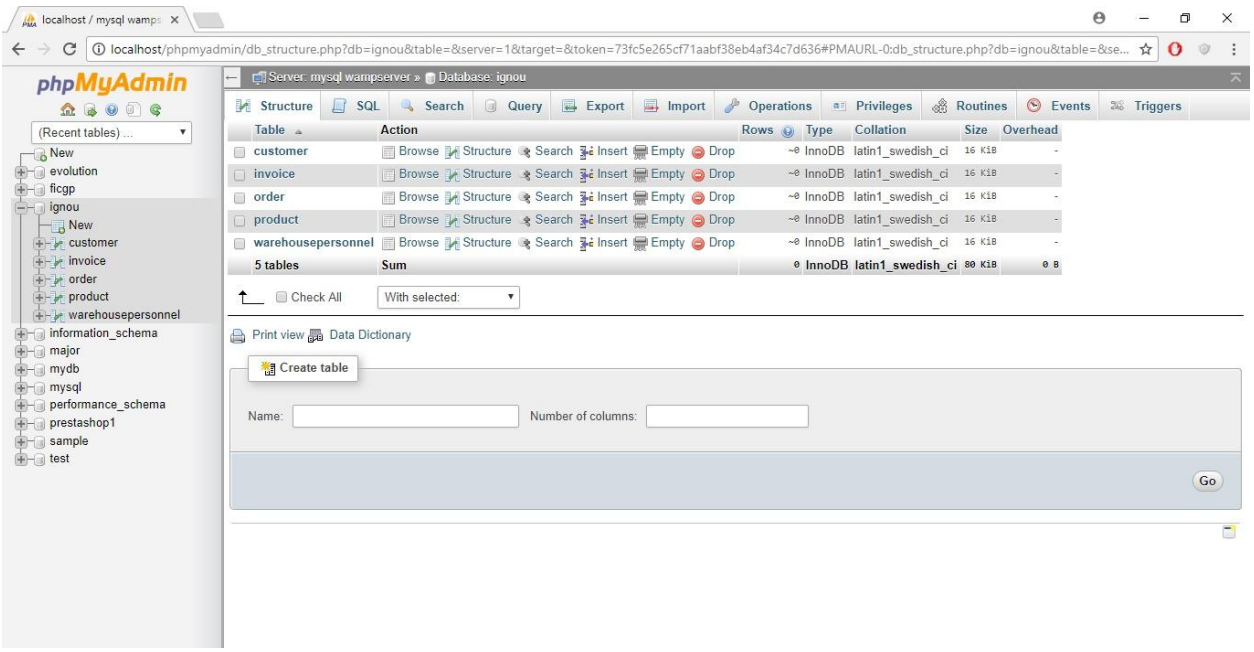
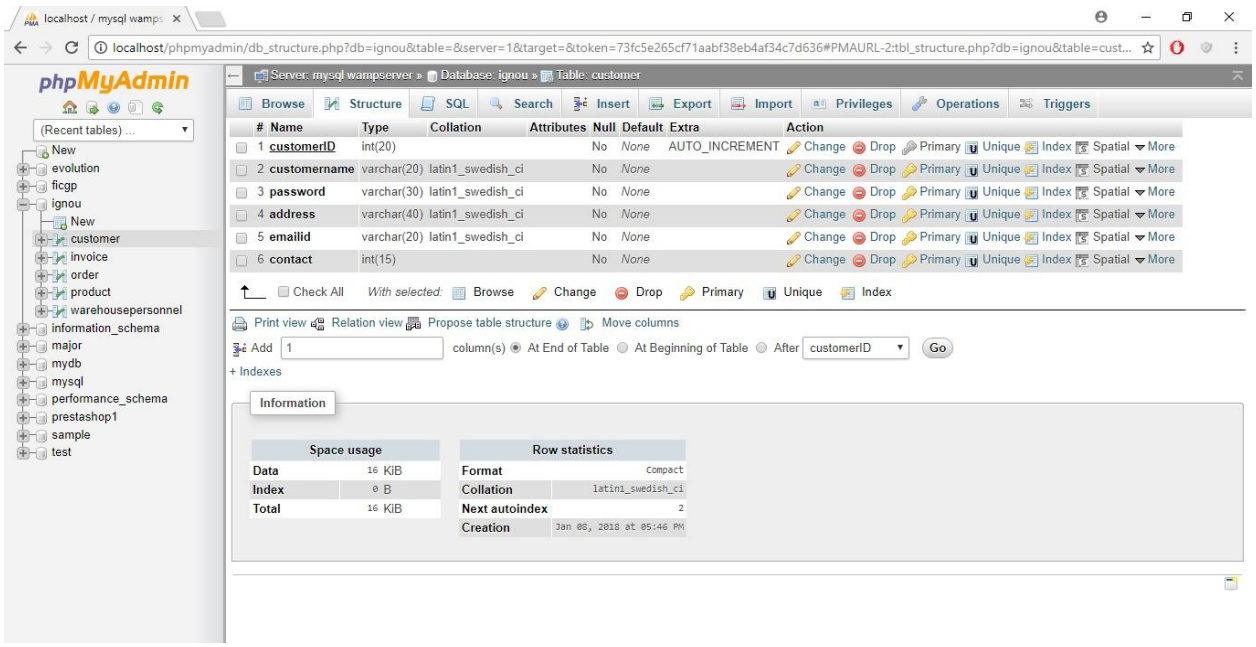
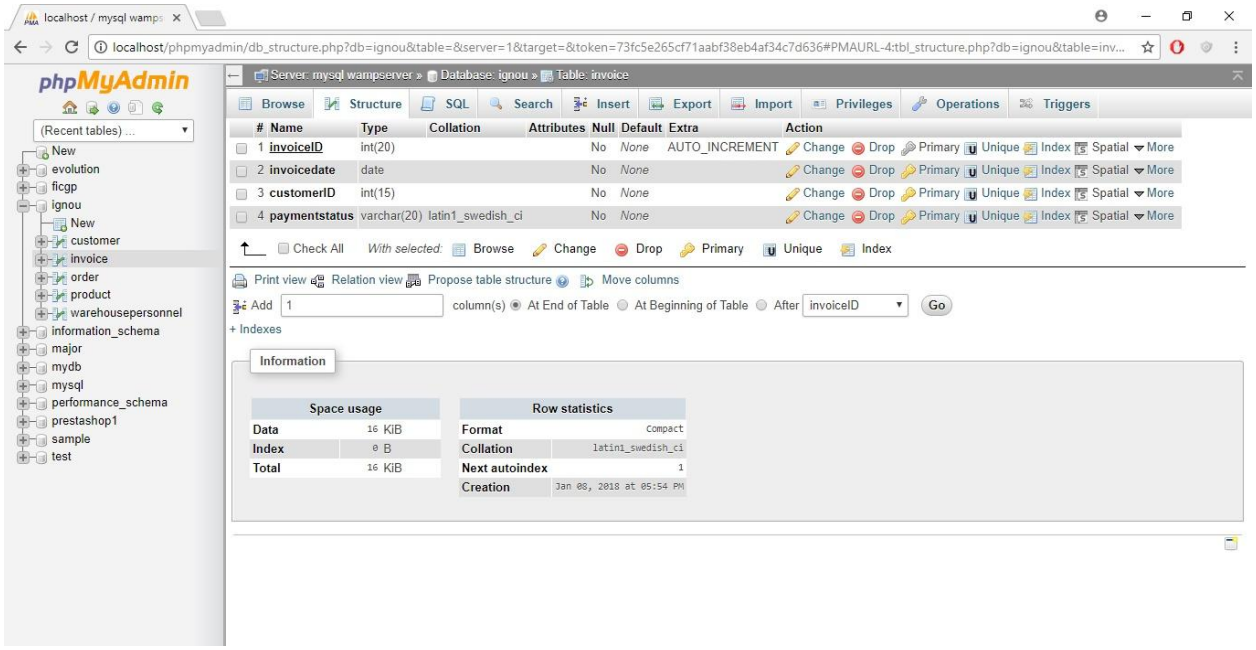


Table structure

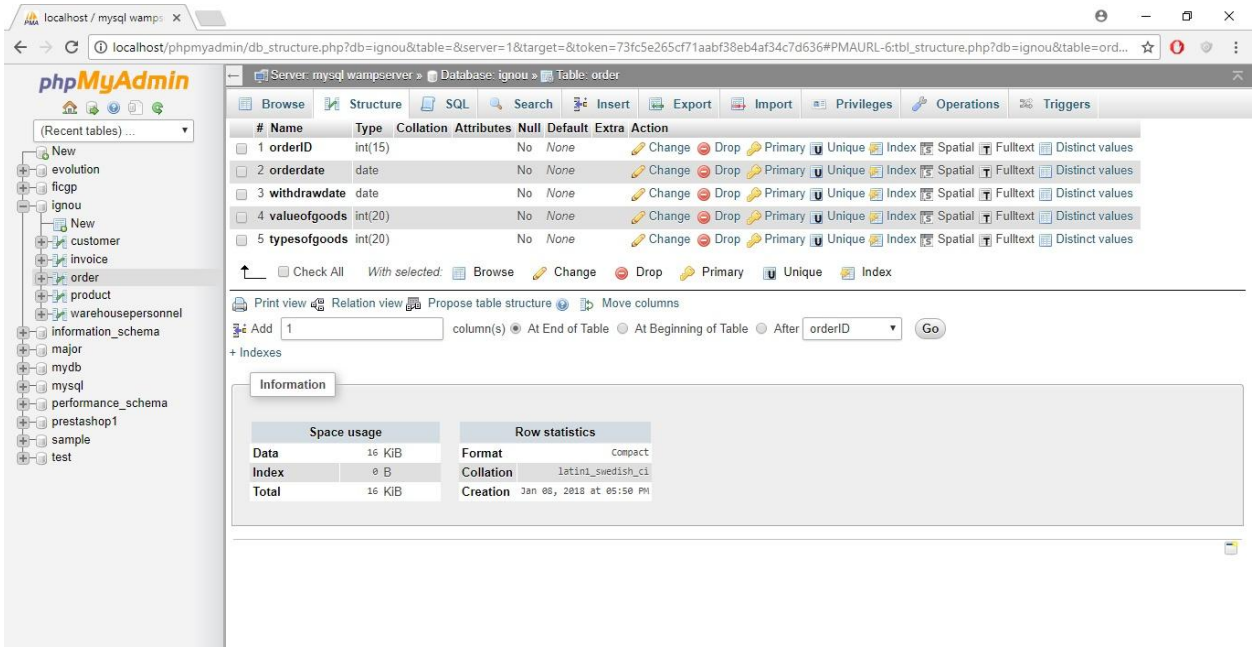
Customer



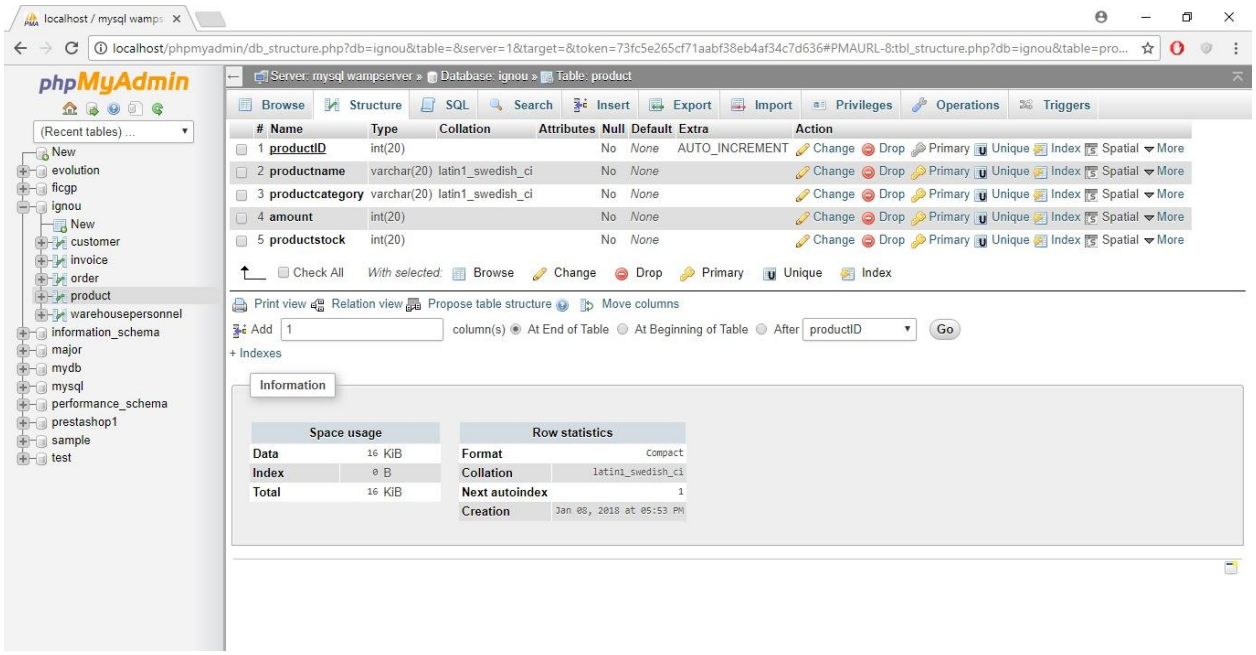
Invoice



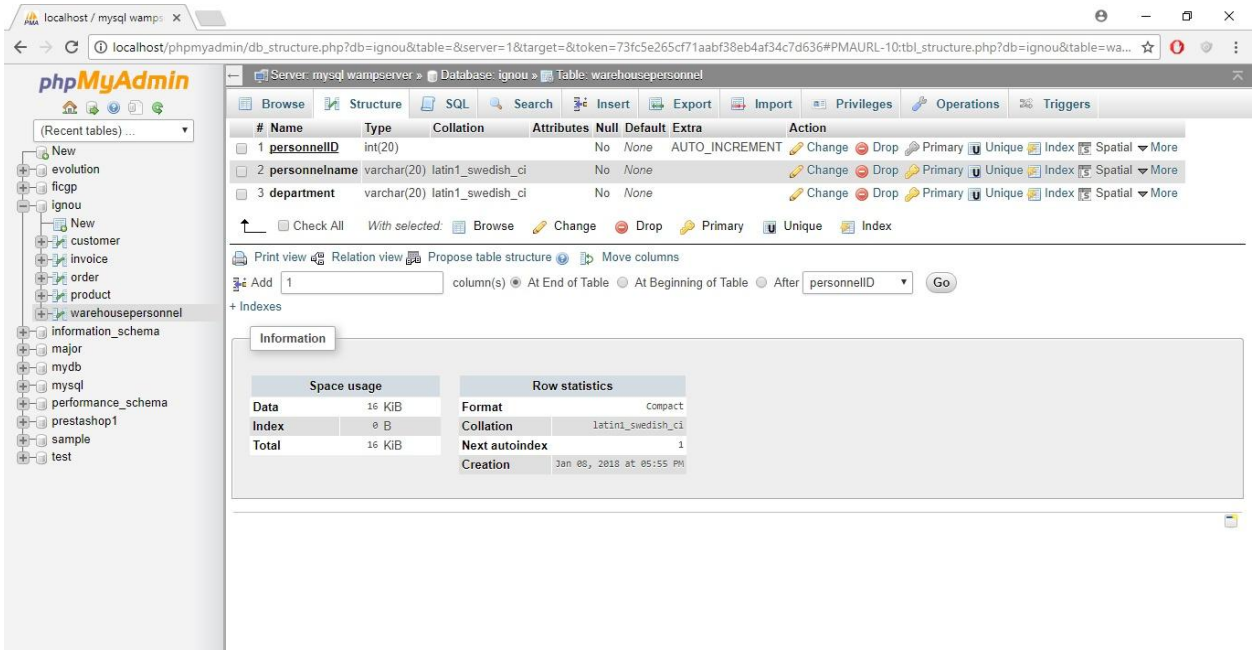
Order



Product



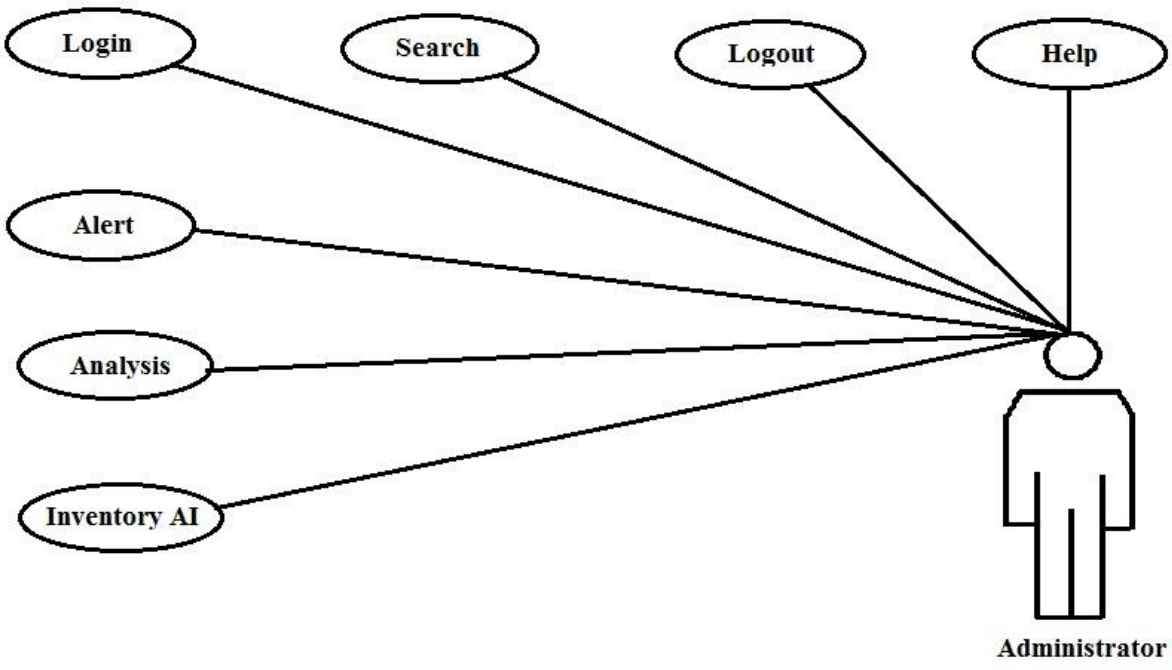
Warehousepersonnel



3.11 Specific Requirements

3.11.1 Use case reports

3.11.1.1 Owner/Administrator



- Analysis

Provides a graphical analysis of products sold in the terms of weeks, months or year on the basis of type of product.

- Search

Helps the user search for items from the database on the bases of item number or item type or item name.

- Alert

Alerts the user when it is time to restock any product.

- Help

Provides help to the user about the software and how to use it.

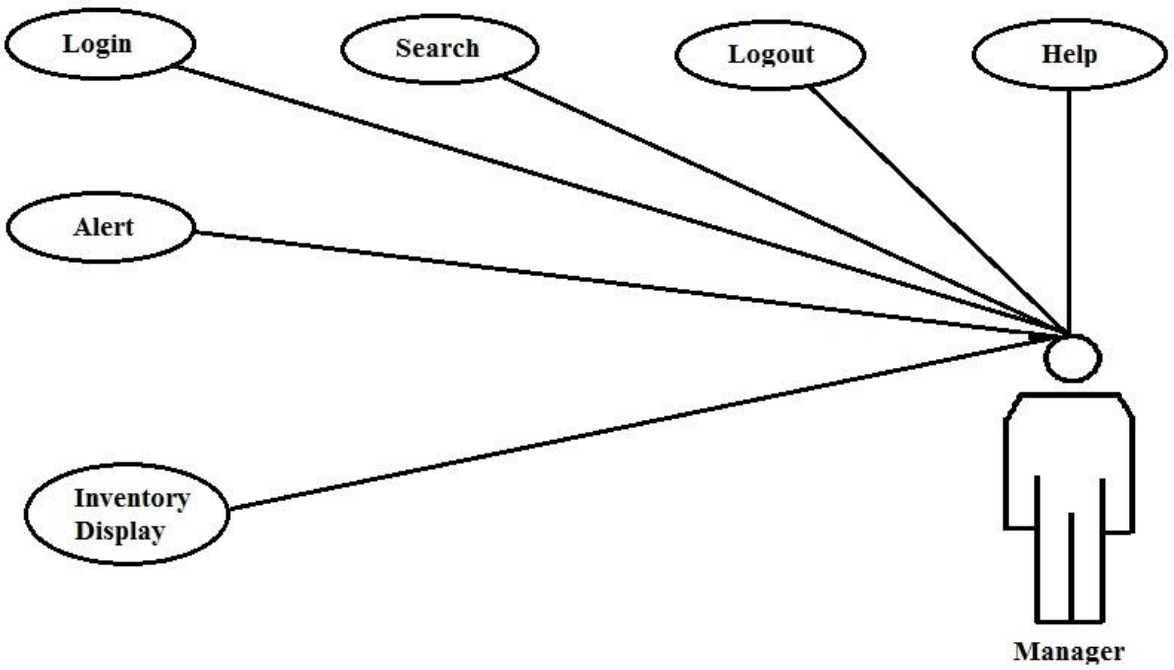
- Log in

  Provides data security.

- Inventory (All)

  This module provides access to central database and data manipulation authority.

3.11.1.2 Manager



Provides a graphical analysis of products sold in the terms of weeks, months or year on the basis of type of product.

- Search

  Helps the user search for items from the database on the bases of item number or item type or item name.

- Alert

  Alerts the user when it is time to restock any product.

- Help

Provides help to the user about the software and how to use it.

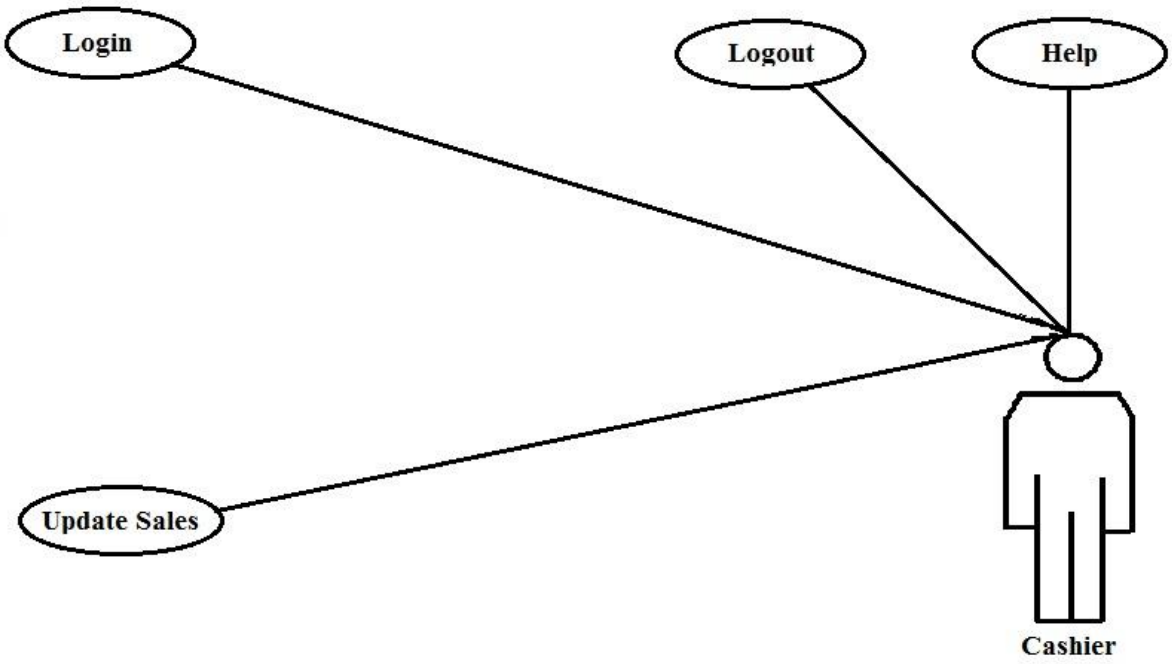
- Log in

  Provides data security.

- Inventory (display)

  This module provides the view to central database.

3.11.1.3 Cashier



- Log in

  Provides data security.

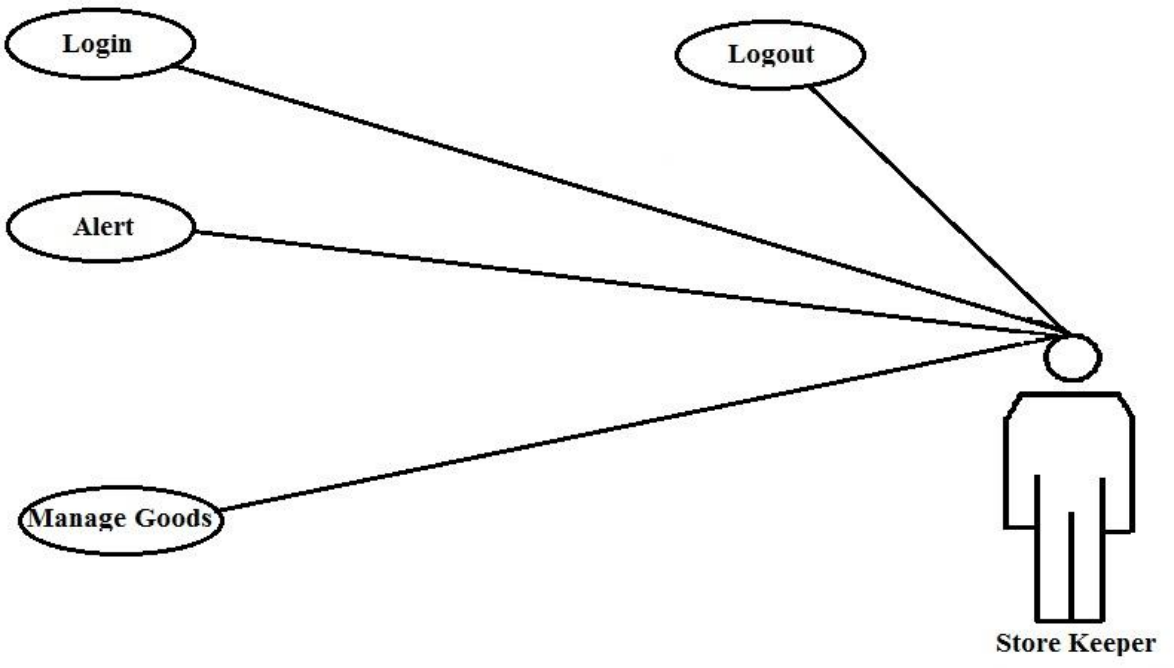
- Help

  Provides help to the user about the software and how to use it.

- Update at sales

  This module updates the main database at the time of sales of a product.

3.11.1.4 Store Keeper



- Log in

  Provides data security.

- Alert

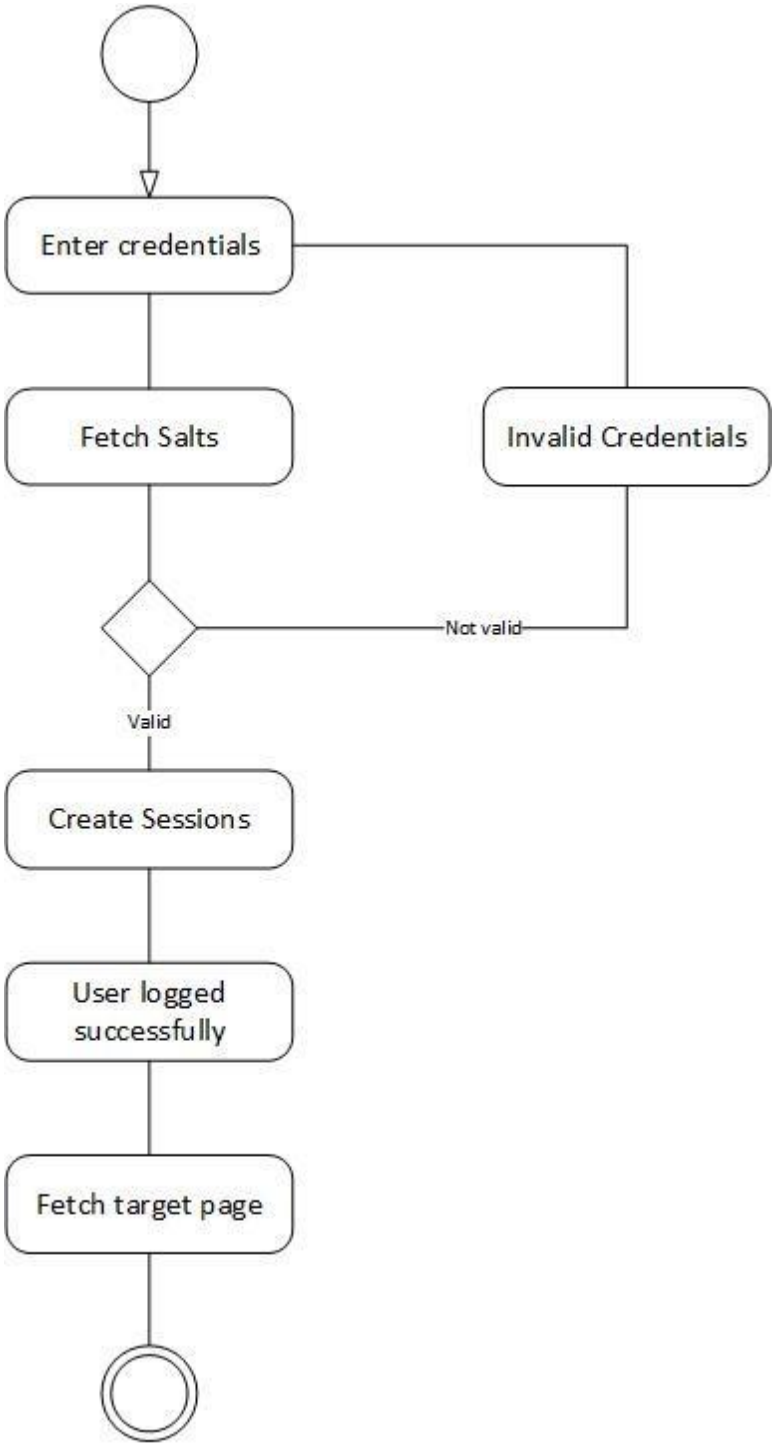
  Alerts the user when it is time to restock any product.

- Manage Goods

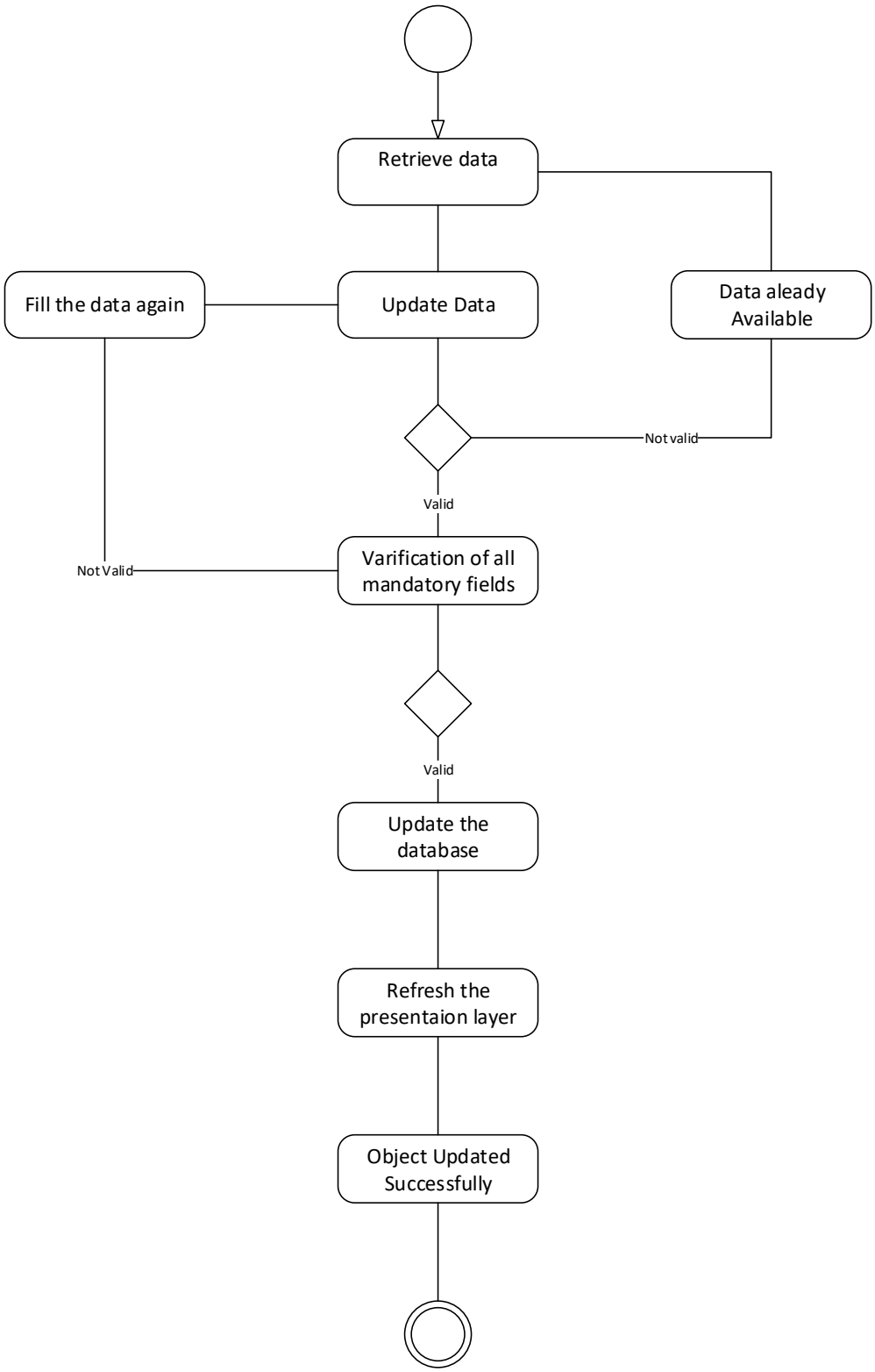
  Manages the goods stock in the store.

3.11.2 Activity Diagrams

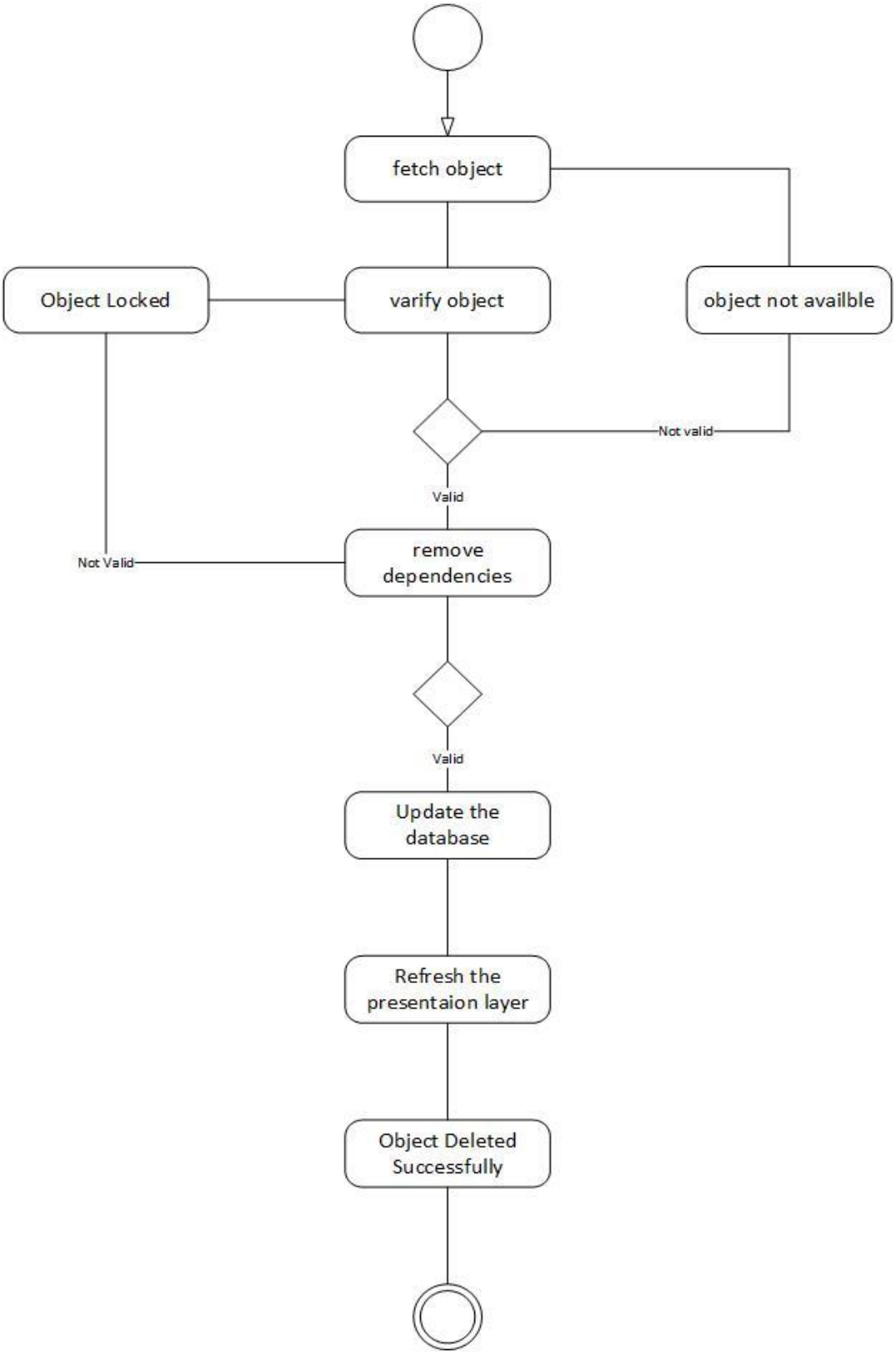
3.11.2.1 Generalized user login Diagram



3.11.2.2 Generalized Object Editing diagram

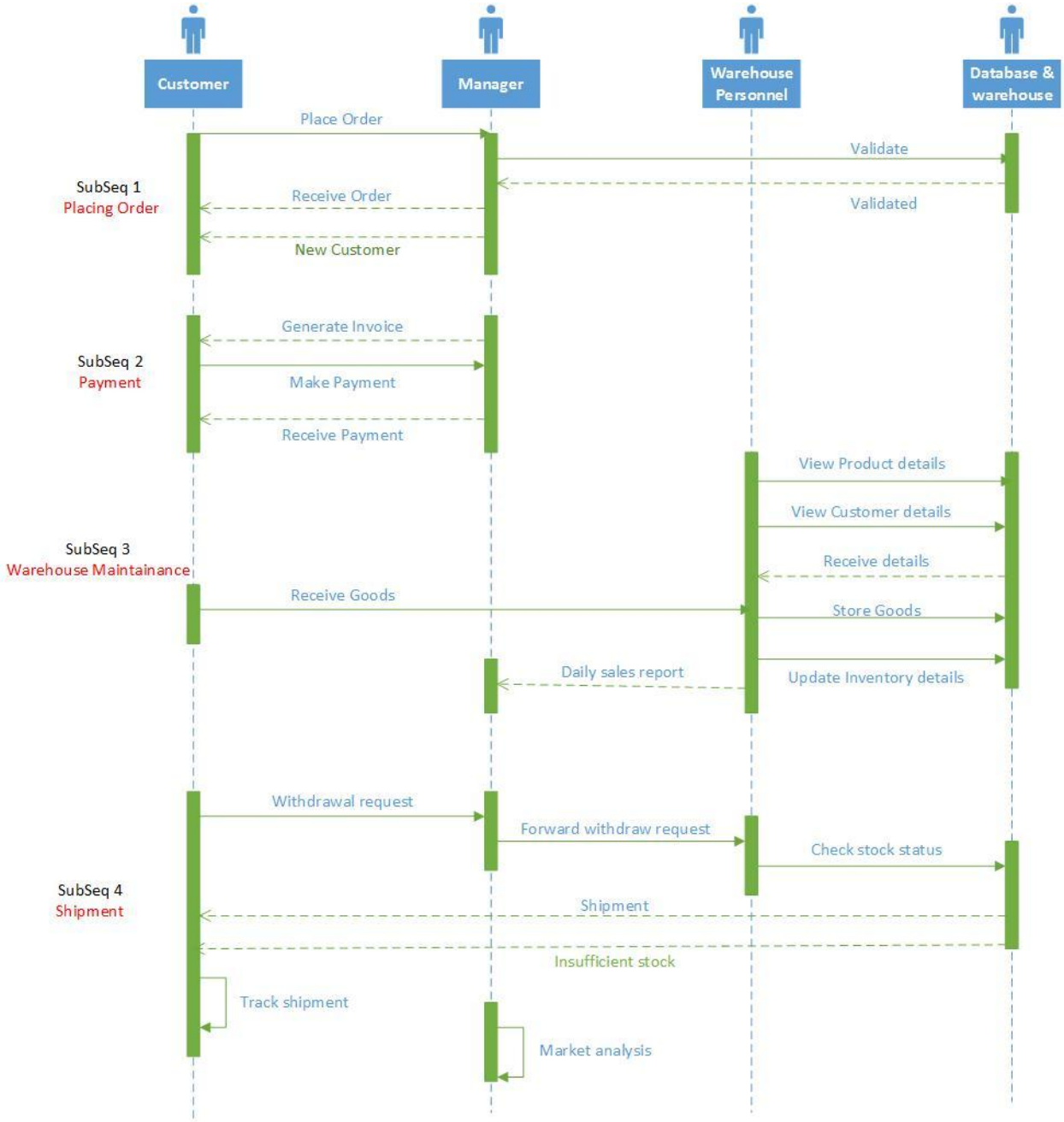


3.11.2.3 Generalized Object Deleting diagram

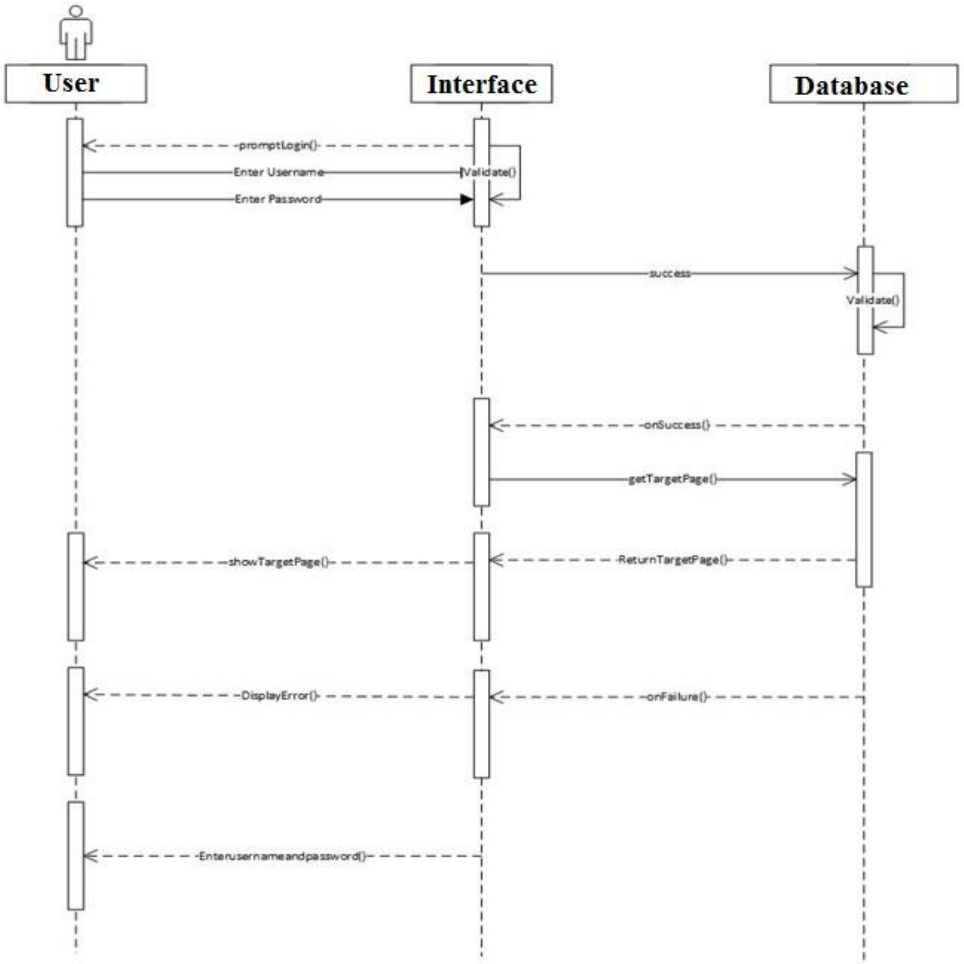


3.11.2.4 Generalized Object Insertion diagram

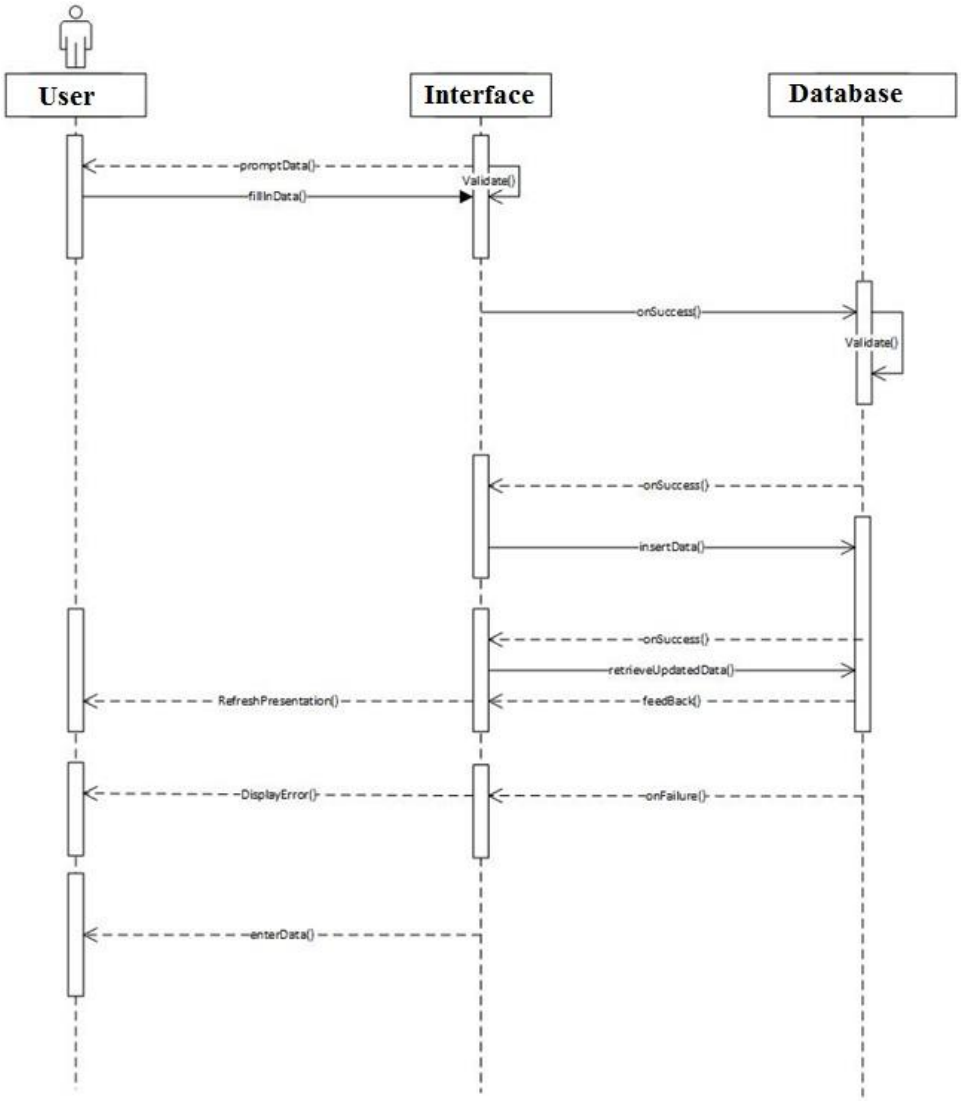
3.11.3 Sequence Diagrams



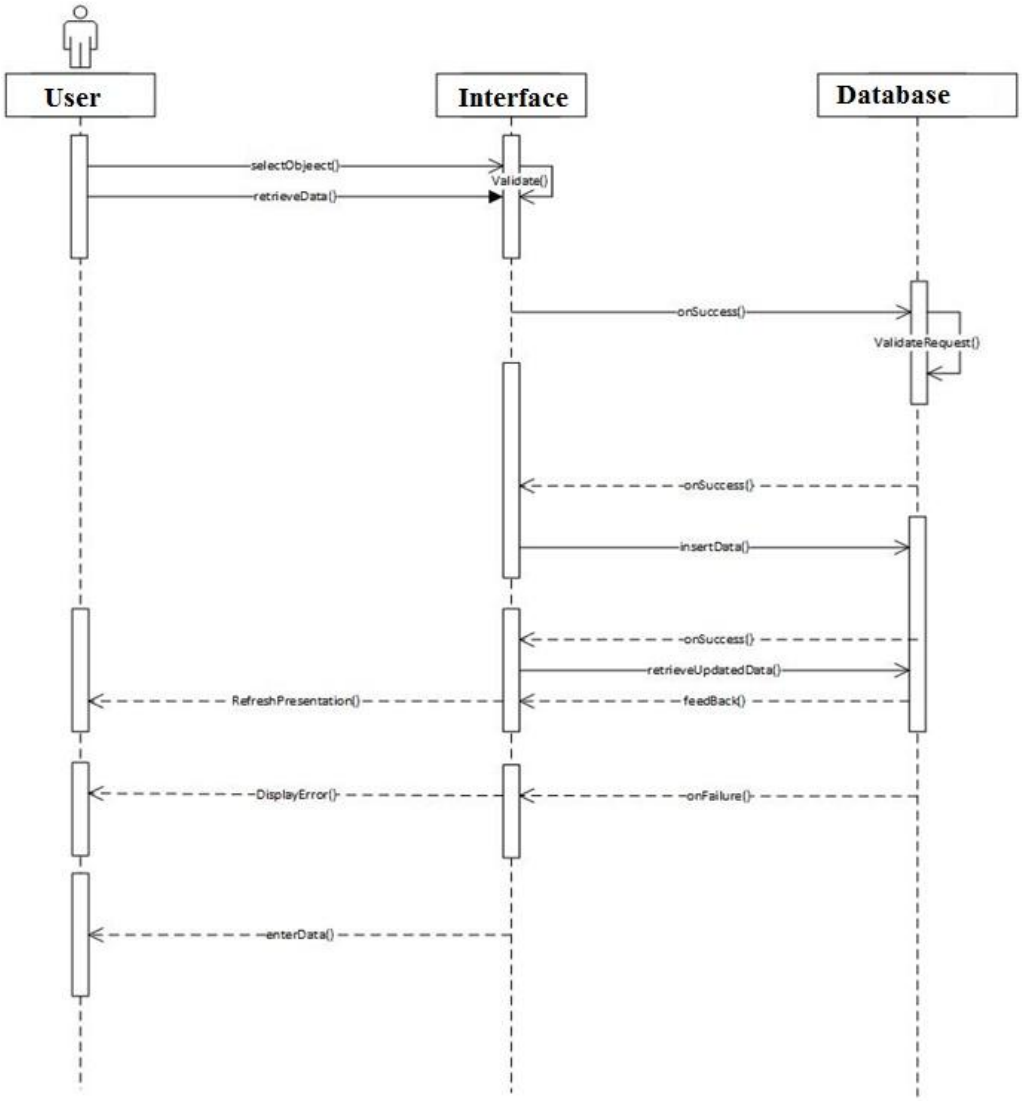
3.11.3.1 User login sequence diagram



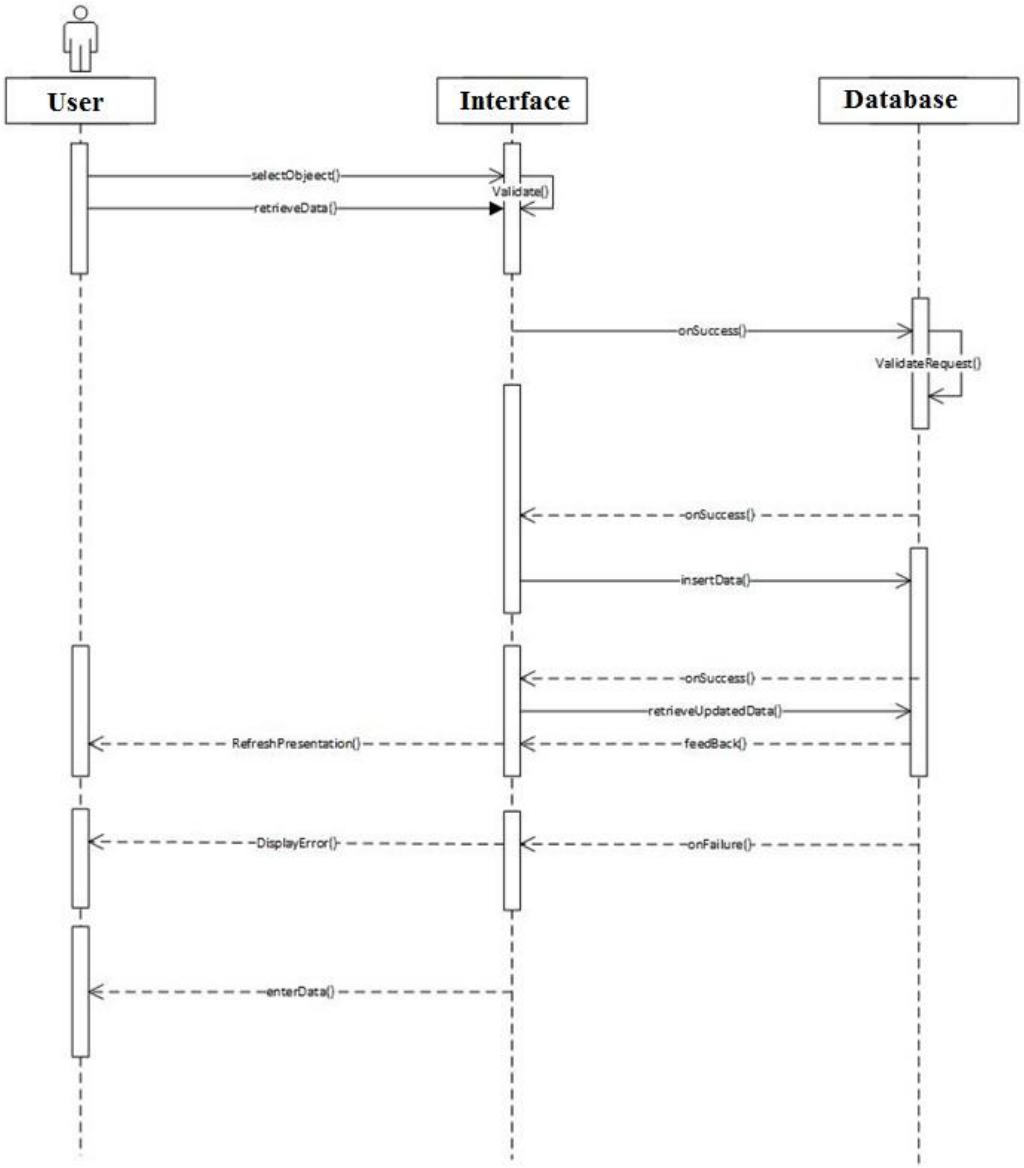
3.11.3.2 Add object sequence diagram



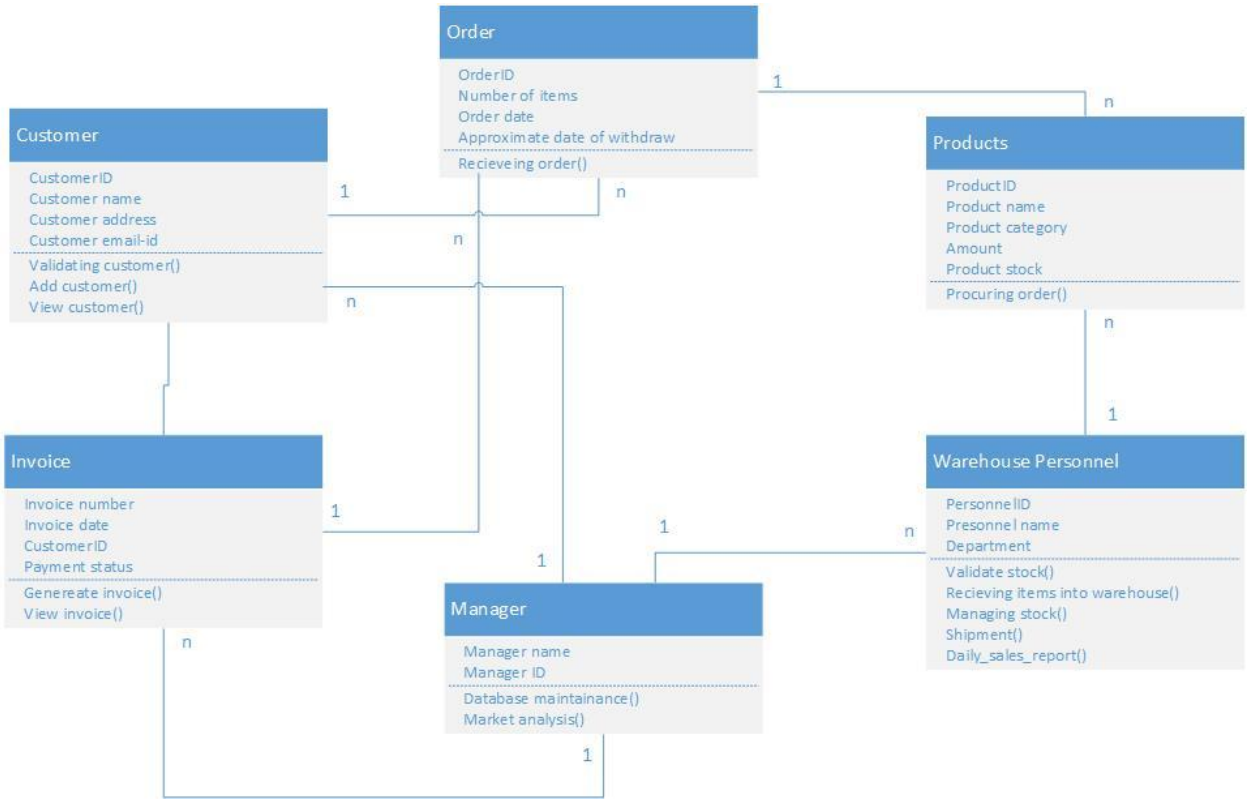
3.11.3.3 Edit Object sequence diagram



3.11.3.4 Delete Object sequence diagram



3.11.4 Class Diagram



3.11.5 CRC Index Cards

Customer

Class Name :	**Approve**
Class Type :	Interaction
Class Characteristics :	Tangibility : Intangible Sequential : Concurrent Persistence : Temporary Inclusiveness : Aggregate Integrity : Guarded
Responsibility :	Collaborations :
Validating Customer() Add Customer() View Customer()	

Order

Class Name :	**Approve**
Class Type :	Interaction
Class Characteristics :	Tangibility : Intangible Sequential : Concurrent Persistence : Temporary Inclusiveness : Aggregate Integrity : Guarded
Responsibility :	Collaborations :

Receiving Order()	Publishing Node

Product

Class Name :	**Approve**
Class Type :	Interaction
Class Characteristics :	Tangibility : Intangible Sequential : Concurrent Persistence : Temporary Inclusiveness : Aggregate Integrity : Guarded
Responsibility :	Collaborations :
Processing Order()	Publishing Node

Invoice

Class Name :	**Approve**
Class Type :	Interaction
Class Characteristics :	Tangibility : Intangible Sequential : Concurrent Persistence : Temporary Inclusiveness : Aggregate Integrity : Guarded
Responsibility :	Collaborations :
Generate Invoice()	Publishing Node

View Invoice()

Warehouse Personnel

Class Name :	**Approve**
Class Type :	Interaction
Class Characteristics :	Tangibility : Intangible Sequential : Concurrent Persistence : Temporary Inclusiveness : Aggregate Integrity : Guarded
Responsibility :	Collaborations :
Validating Stock() Receiving Items Into Warehouse() Managing Stock() Shipment() Daily_Sales_Reprot()	Publishing Node

Chapter 4

Case study on Implementation of Information Security

4.1 Network Security Platform - WireShark

4.1.1 Security analysis of trusted sites using WireShark

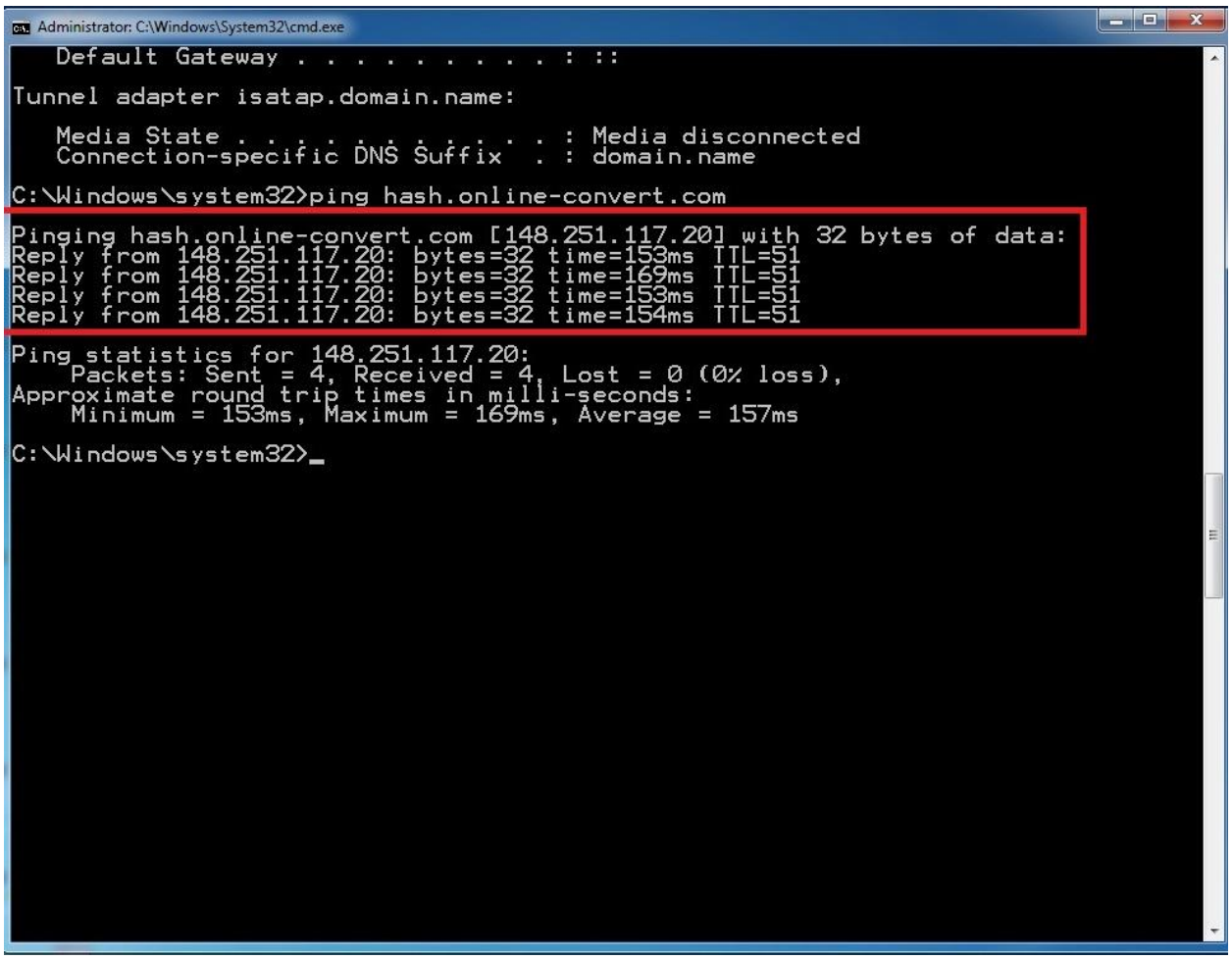
In order to demonstrate this attack I have used three URLs listed as below.

> http://hash.online-convert.com

> https://www.facebook.com

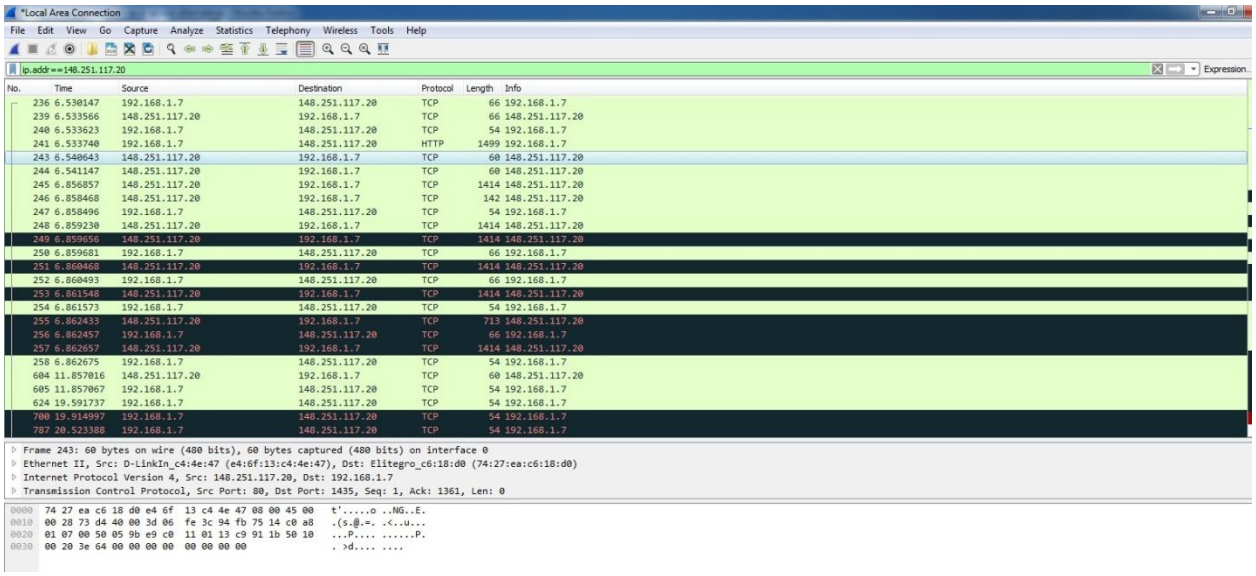
> https://www.ssh.com

Get IP address of website using ping and try to look the content from Wireshark. Please refer below screenshot. Img 4.1.1.1



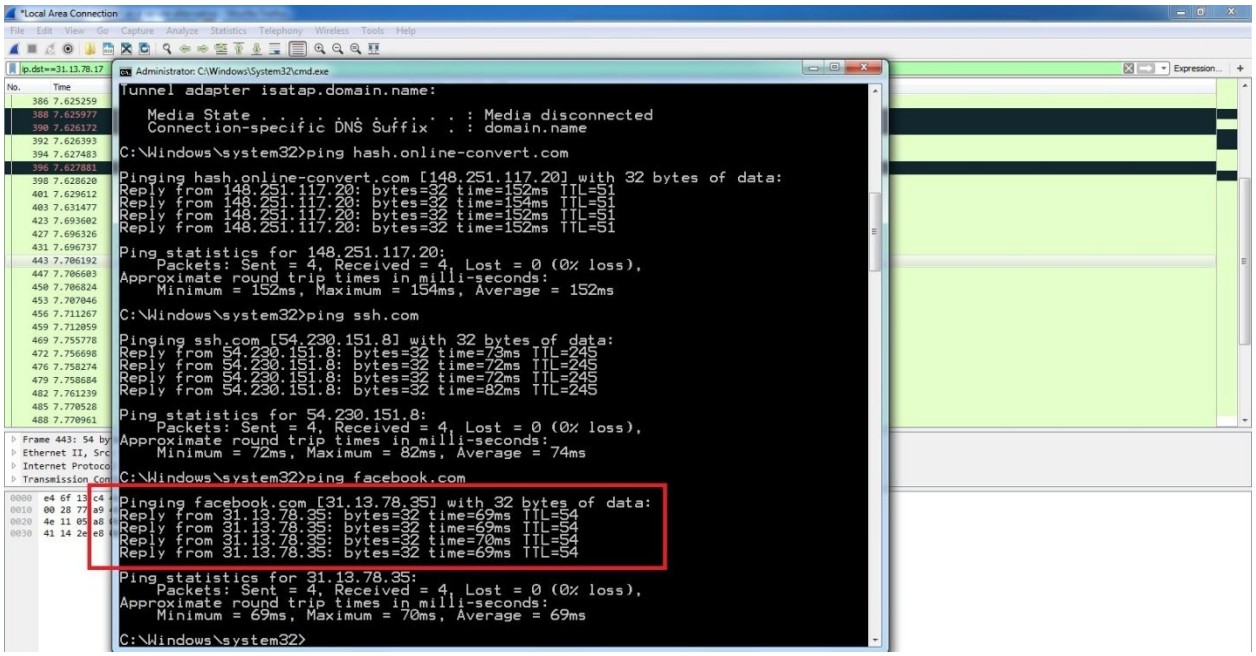
Img 4.1.1.1

Filter IP Address on WireShark. Please refer Im 4.1.1.2

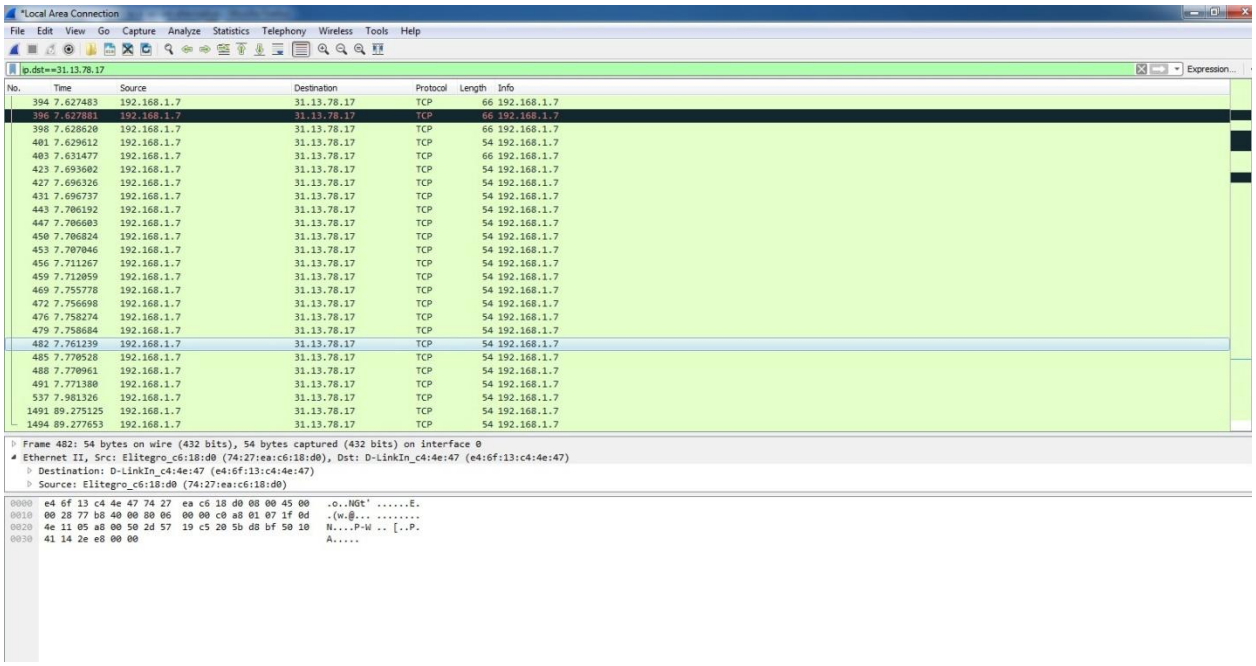


Img 4.1.1.2

Now try to lookup secure SSL website and try to see the content on Wireshark. Img 4.1.1.3, 4.1.1.4



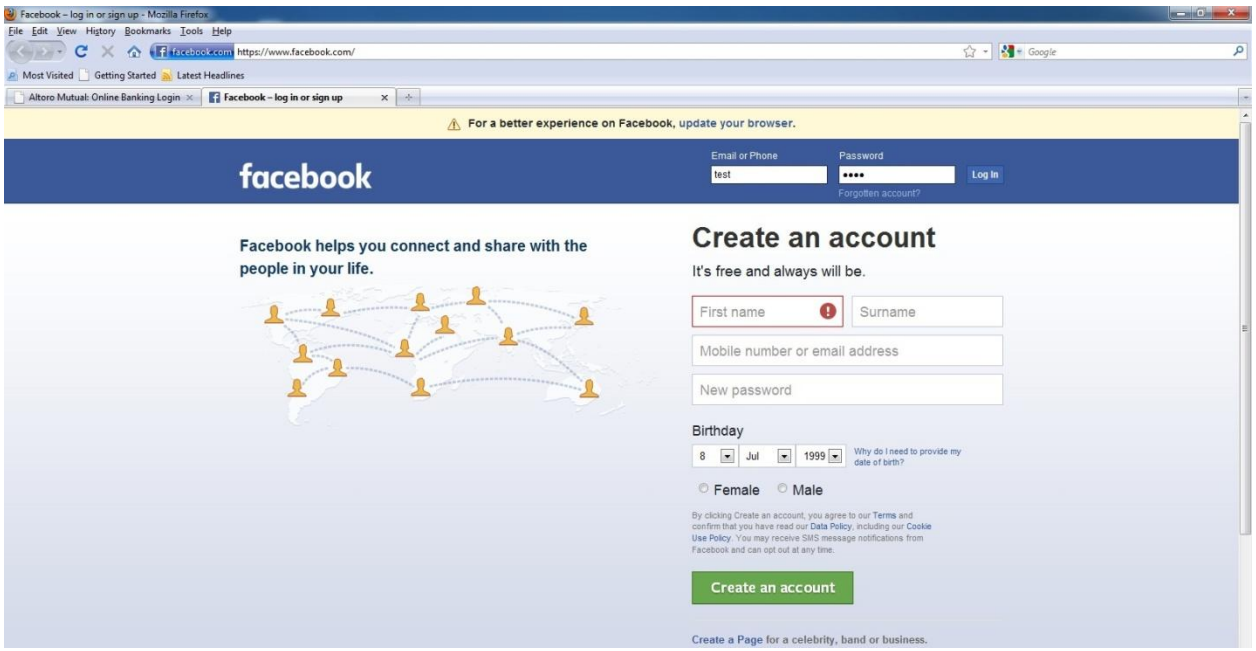
Img 4.1.1.3



Img 4.1.1.4

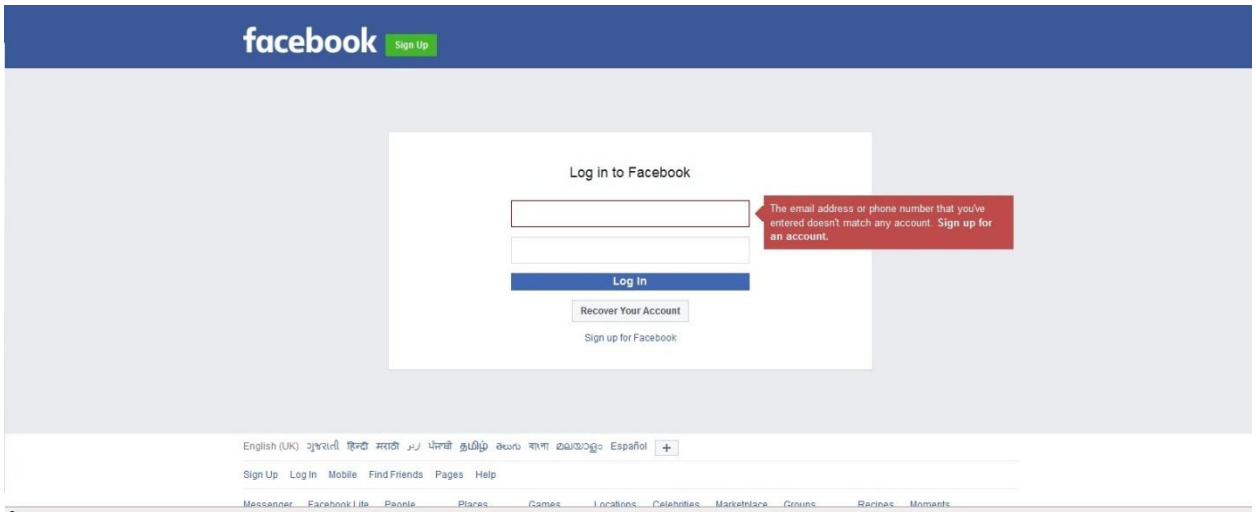
Repeat this activity for 4-5 different parts.

Enter some fake information and try to look the entered details using Wireshark. Img 4.1.1.5



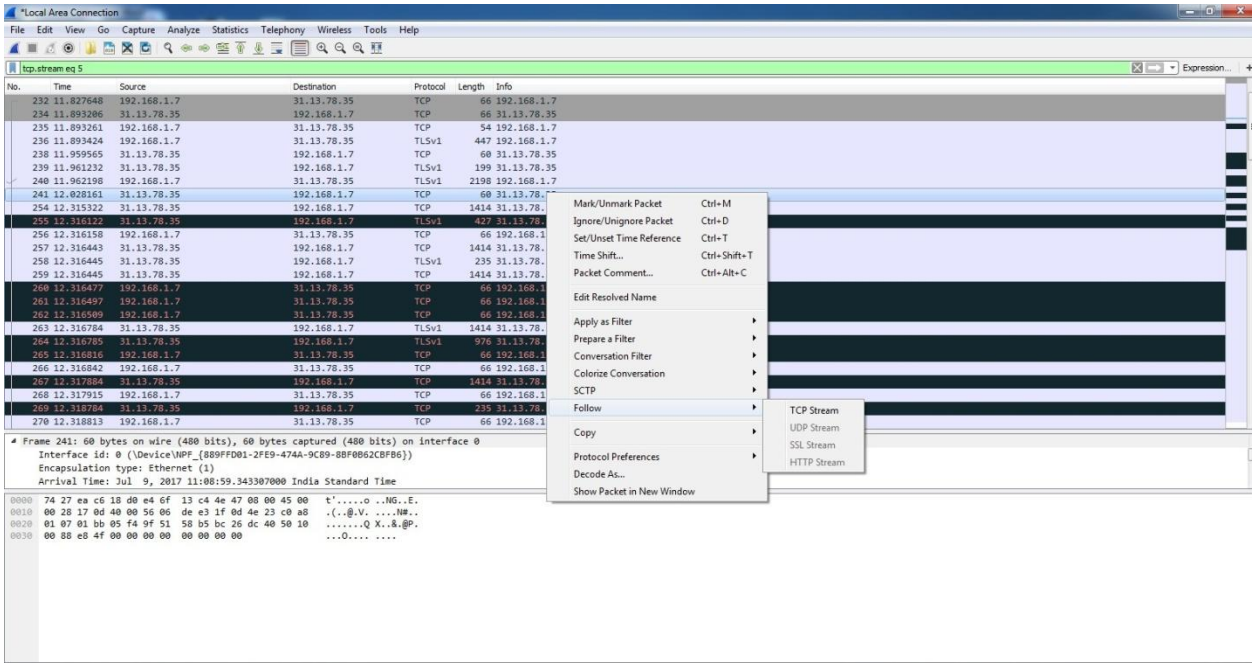
Img 4.1.1.5

Now, Enter fake username and password as shown in below screenshot. Img 4.1.1.6



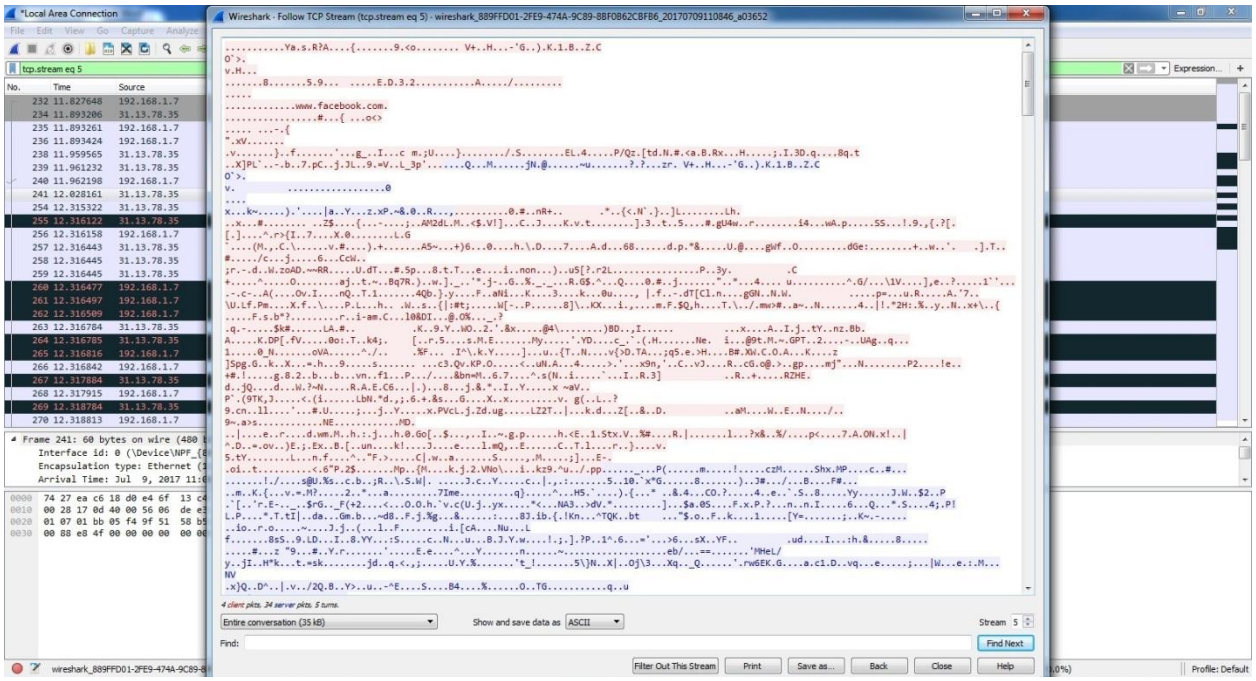
Img 4.1.1.6

Open WireShark, while looking at the content, use the filter for TCP stream. Img 4.1.1.7



Img 4.1.1.7

Observe, this URL has SSL certificate so the content will not be in readable format. It will show in encrypted format. Please refer Img 4.1.1.8



Img 4.1.1.8

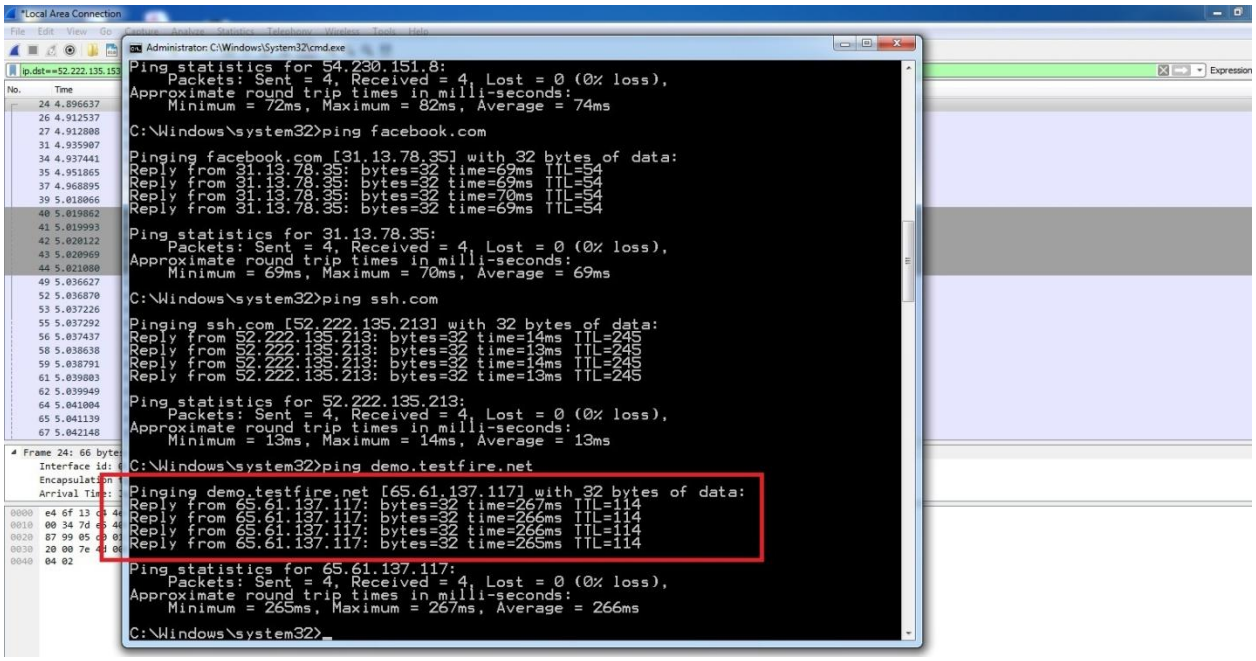
I have followed same scenario in case of non SSL URLs. Please review next chapter to get details on same.

4.1.2 Security analysis of non trusted sites using WireShark

In order to demonstrate this attack I have used URL listed as below.

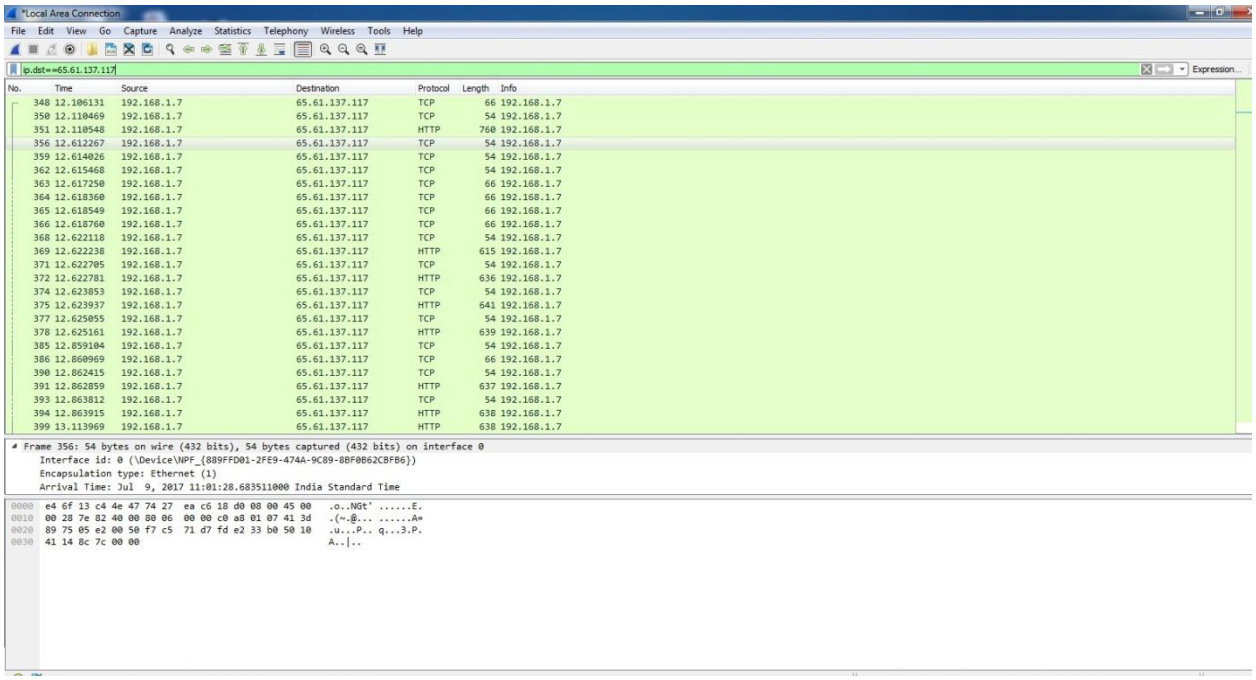
> http://demo.testfire.net

Open cmd and get the IP address of the mentioned URL. Please refer Img 4.1.2.1



Img 4.1.2.1

Start Wireshark capturing. Img 4.1.2.2

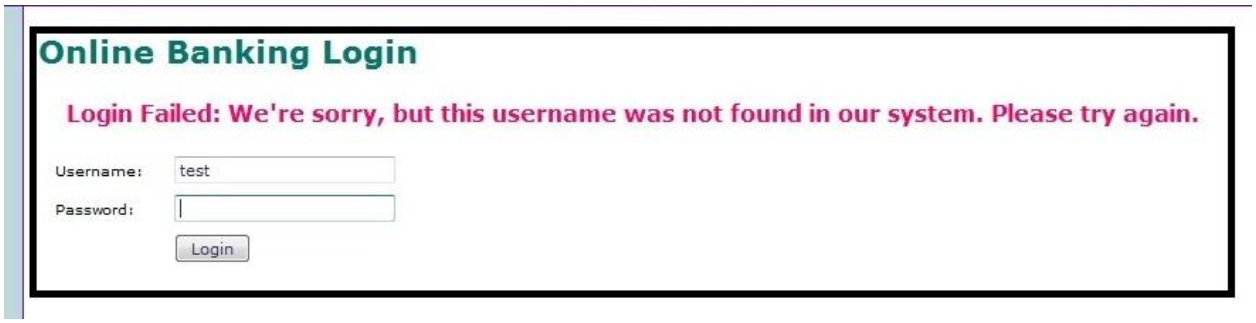


Img 4.1.2.2

Go to this URL i.e. http://demo.testfire.net and enter fake information for username and password. Img 4.1.2.3

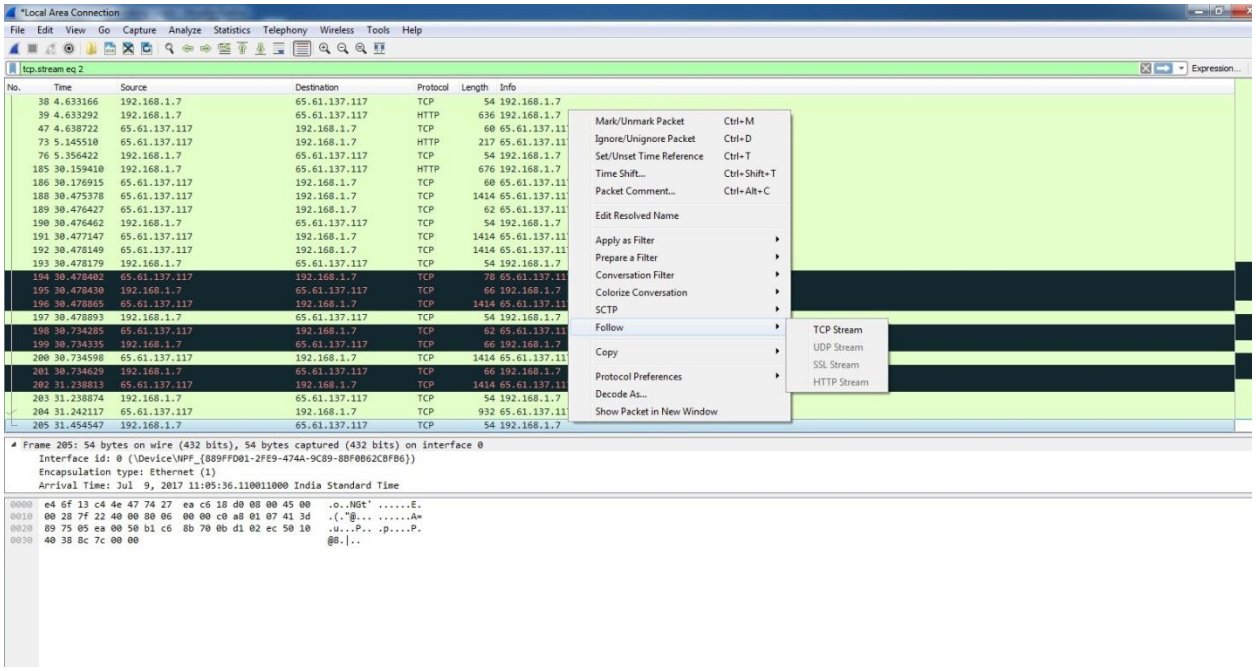
Img 4.1.2.3

The username and password details will show as invalid but data went though the network and it is being captured by Wireshark. Img 4.1.2.4



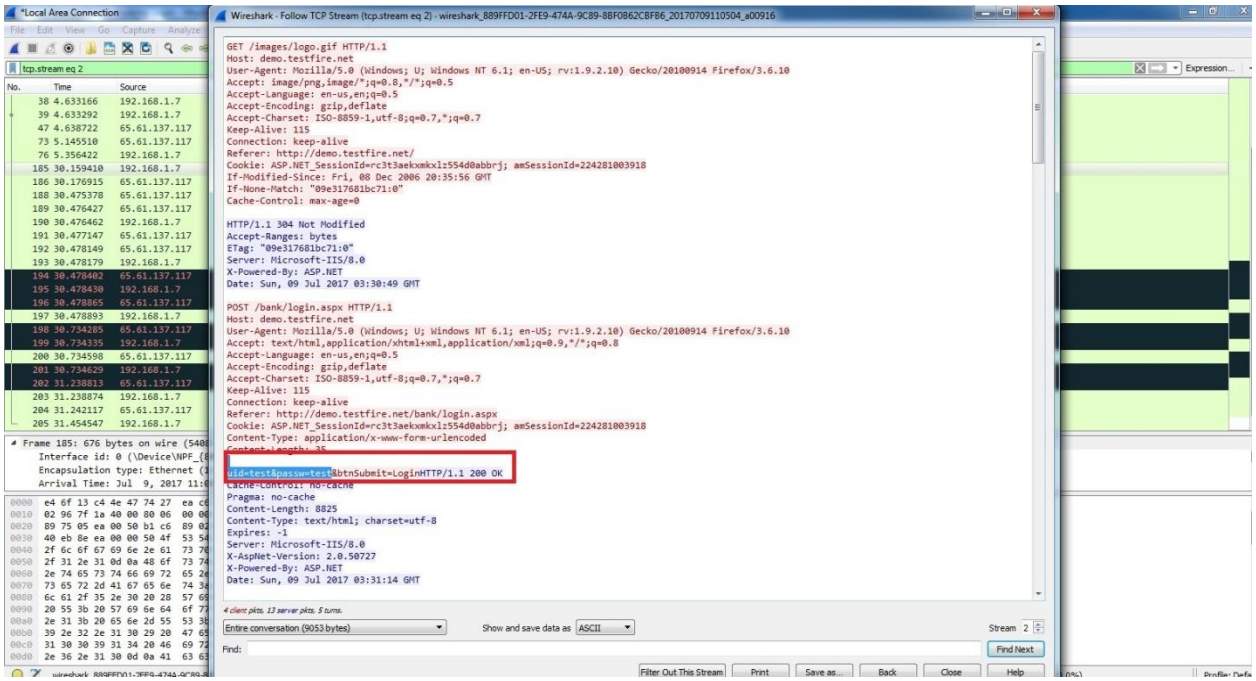
Img4.1.2.4

Go to Wireshark and start the TCP stream only. Img 4.1.2.5



Img 4.1.2.5

This URL is non SSL so username and password can be easily cracked by sniffer tool. Img 4.1.2.6



Img 4.1.2.6

As per the above analysis, it can be sure that SSL sites are more secured; we will not be able to see information easily. It will be in encrypted format. But non SSL sites can be easily cracked. Using any sniffer tools we can capture the sensitive credential details and other information.

Conclusion : Try to use secured SSL certified websites in order to prevent hacking attack on network.

I prepared a comparison report of above activity based on their security level and conformance to guidelines. SSL websites are more secured than non SSL, it is very easy to hack the non trusted sites.

4.2 Application Security – Encryption

The process of secretly encoding messages has been used for centuries. Data Encryption is a process of creating secret message formats for data that is stored on computer files. Within computer software there are multiple encryption techniques available for data files. These techniques are typically known as data encryption algorithms.

4.2.1 Security analysis of Vernam algorithm

The Vernam Cipher is based on the principle that each plaintext character from a message is 'mixed' with one character from a key stream. If a truly random key stream is used, the result will be a truly 'random' ciphertext which bears no relation to the original plaintext.

The ciphertext is generated by applying the logical XOR operation to the individual bits of plaintext and the key stream. The advantage of using the XOR operation for this, is that it can be undone by carrying out the same operation again. In other words:

plaintext + key = ciphertext ⇒ ciphertext + key = plaintext

In mathematics, the XOR operation is known as modulo-2 addition. In our case, the individual bits of the plaintext are XOR-ed with the individual bits of the key.

Implemented vernam algorithm in PHP and analyzed results. Please review Img 4.2.1.1

Newfile.txt

This will store all encrypted and decrypted data inside

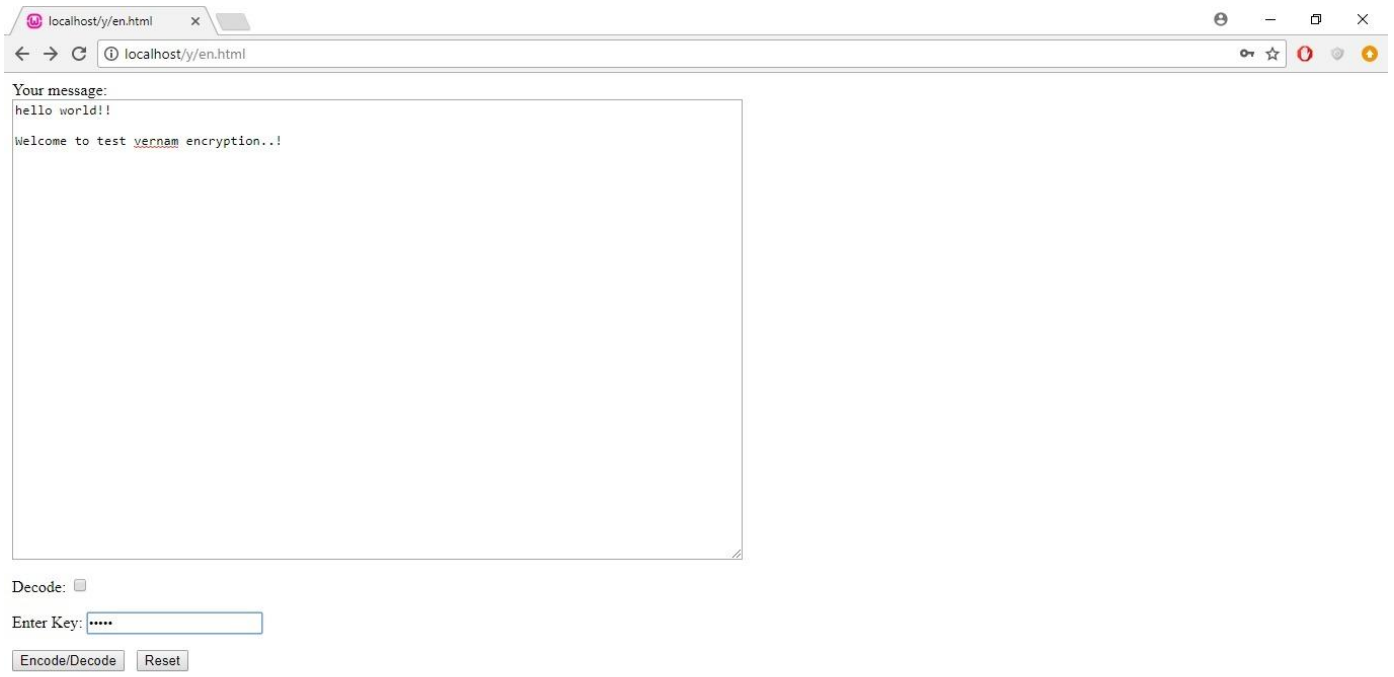
i.e
CQcPCApBFQwWCQVDQmlvbGg0AQkCDQ4BRRUNQxAAEhZDEgATDAIJRQQMABYcERYKC
wtPTEI=

Result screenshot

>> Enter plain text

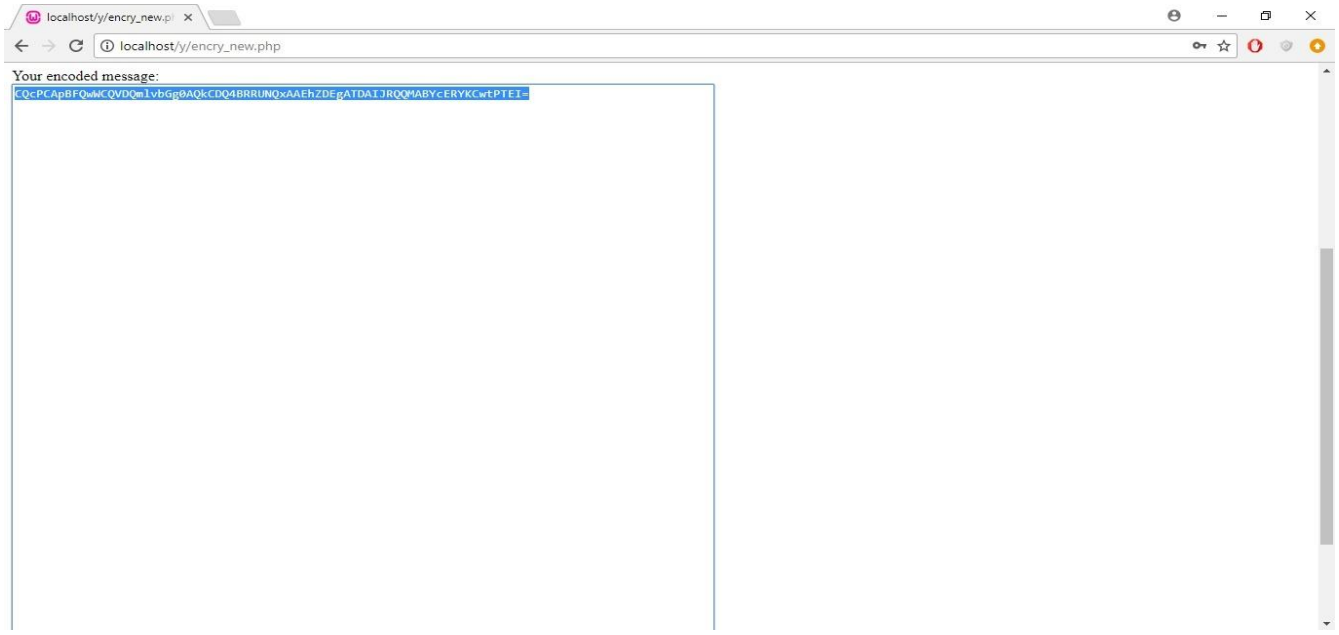
>> Enter Key

>> Encode it



Img 4.2.1.1

It will generate encrypted message. Img 4.2.1.2



Img 4.2.1.2

>> Enter decoded/cipher message

>> Use decode option

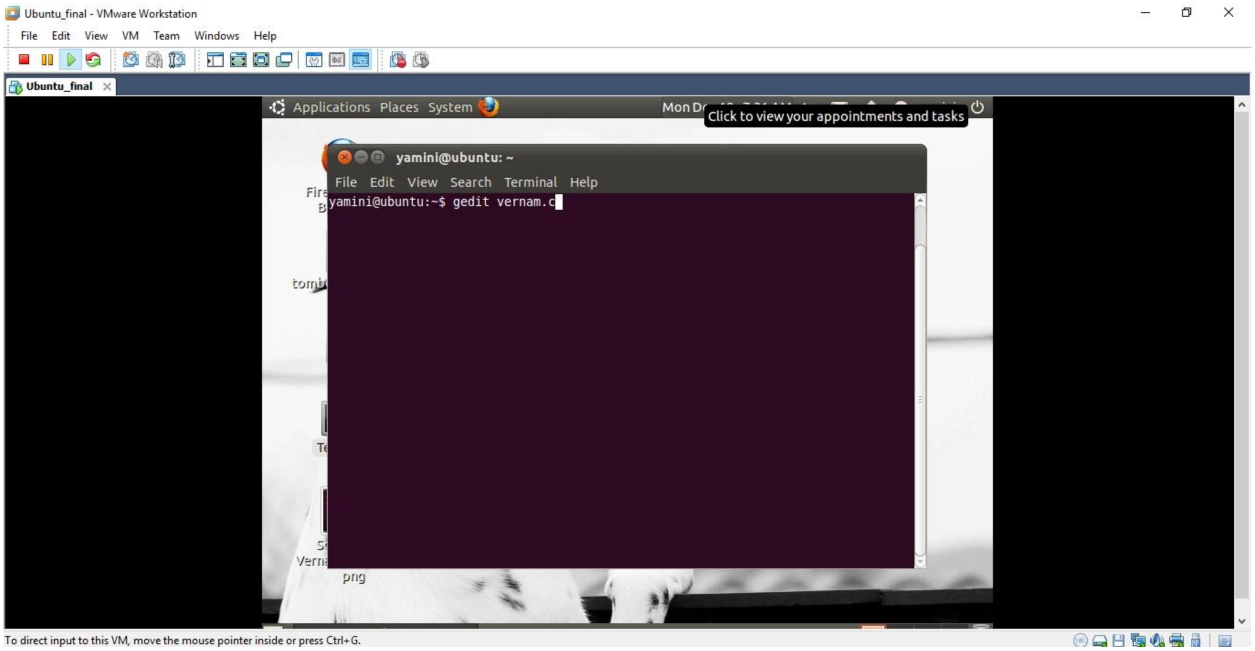
>> Use same Key used while encoding plain text

Img 4.2.1.3

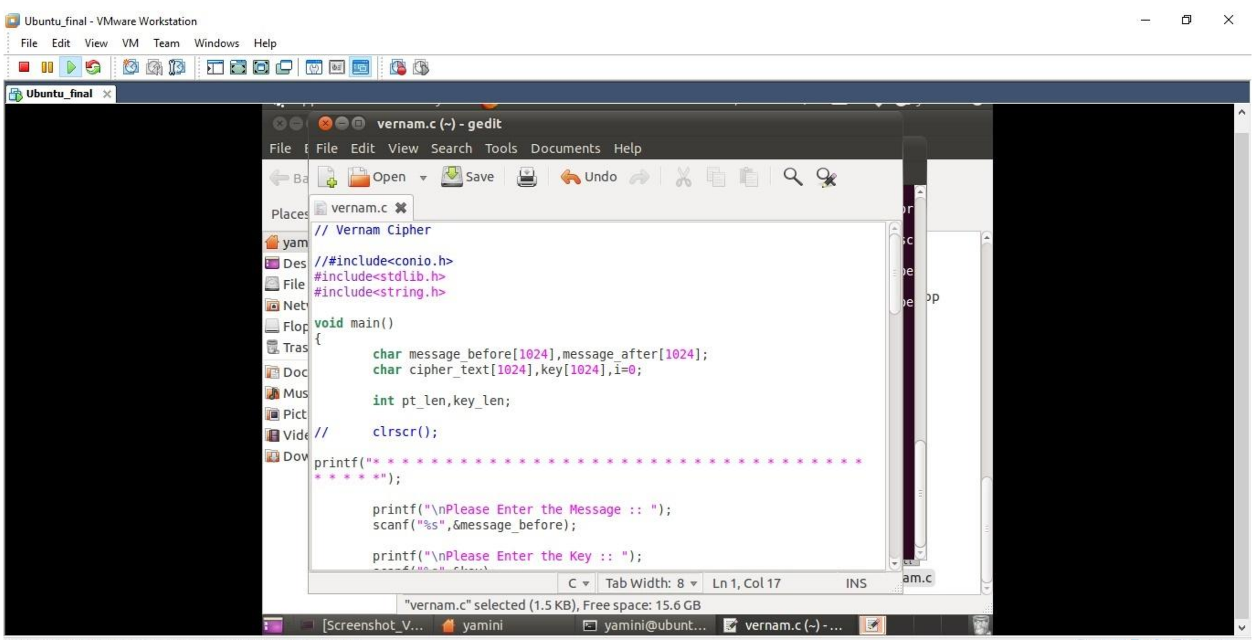
It will convert cipher text into plain text. Img 4.2.1.4

Img 4.2.1.4

Implementation and comparison of Vernam encryption algorithm in C language on Linux platform. Img 4.2.1.5

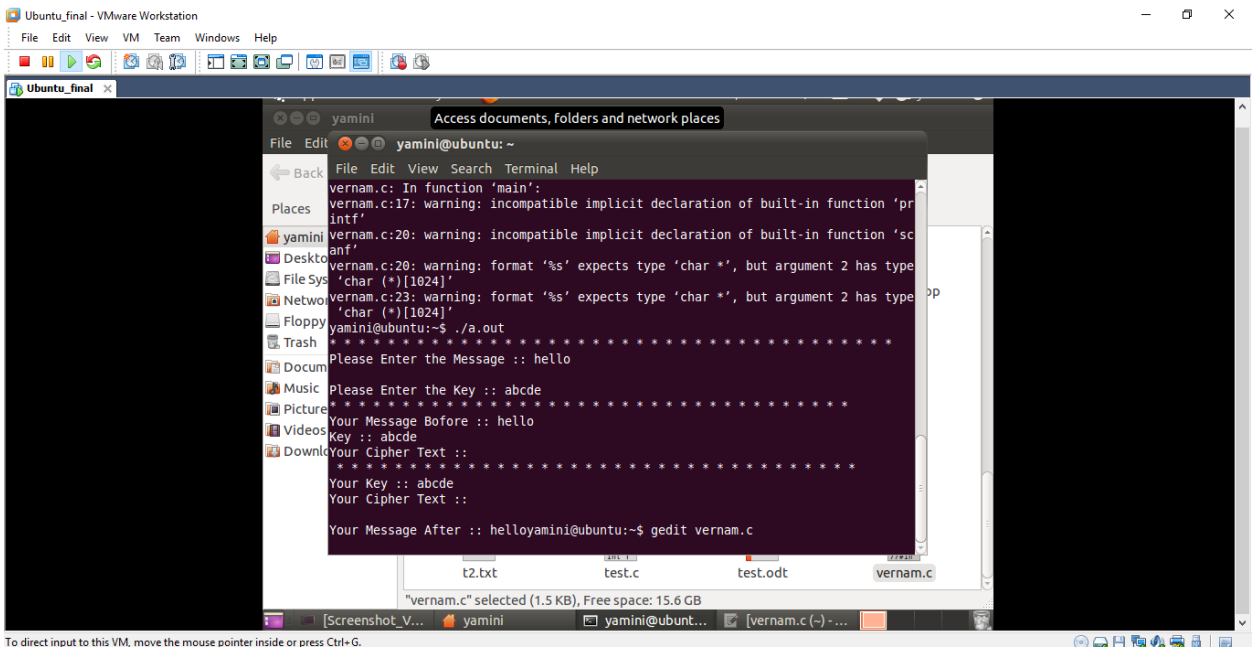


Img 4.2.1.5



Img 4.2.1.6

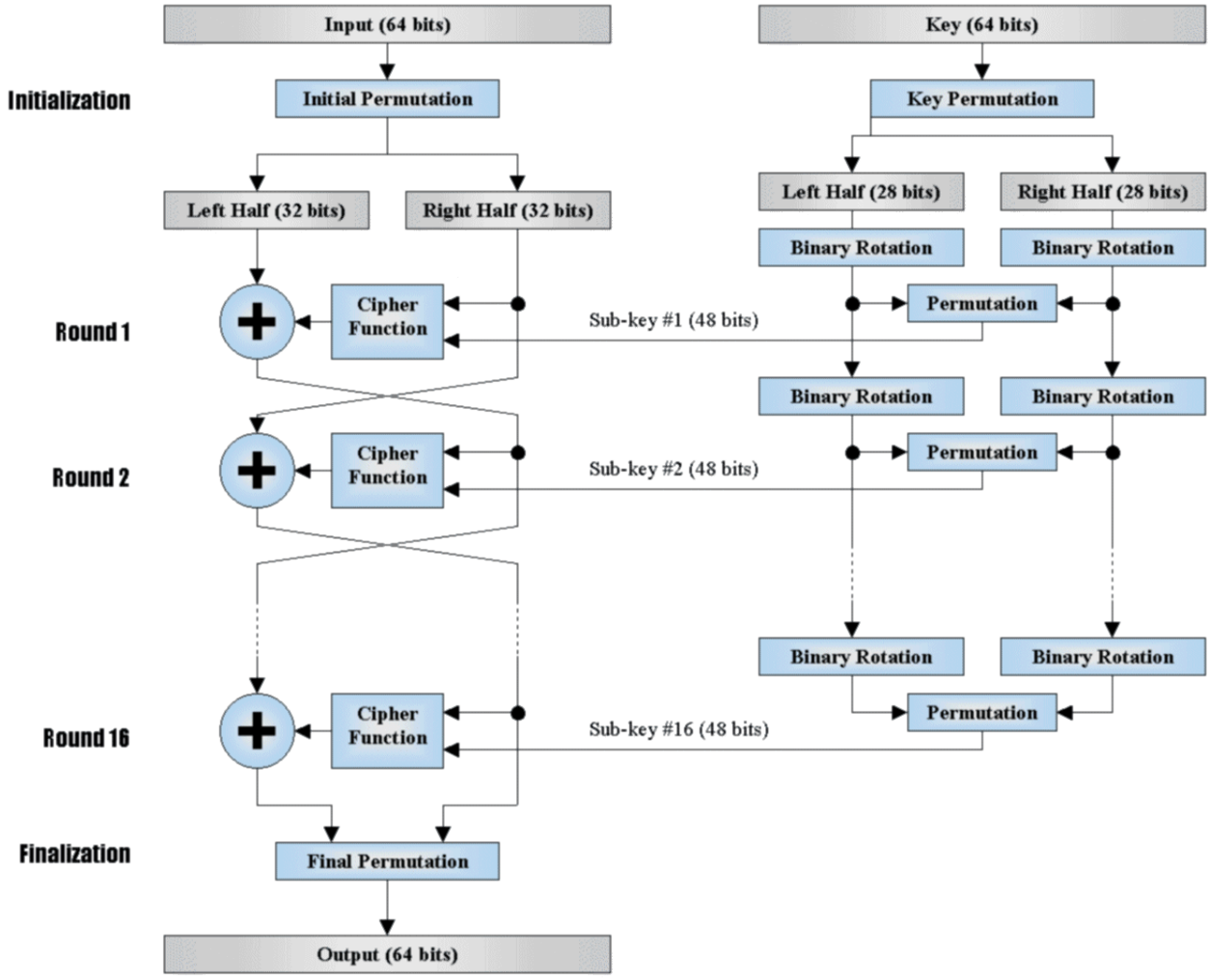
Analyze the output of program. The comparison table has been described in detail in upcoming chapter. Img 4.2.1.7



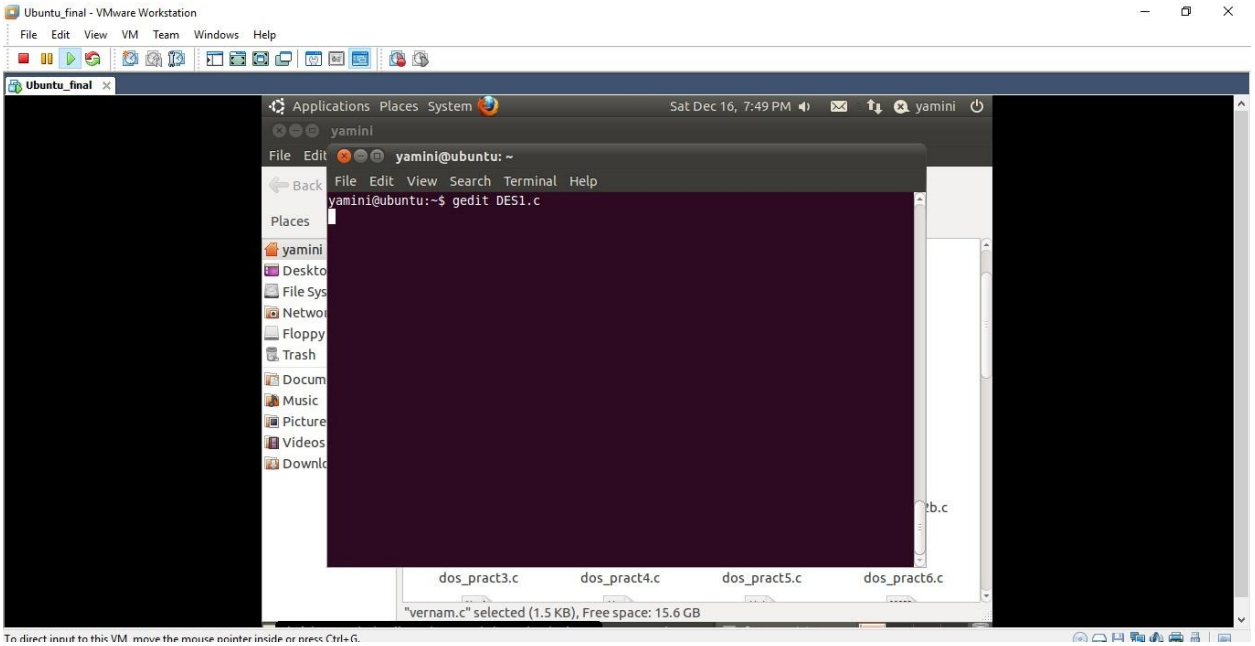
Img 4.2.1.7

4.2.2 Security analysis of DES algorithm

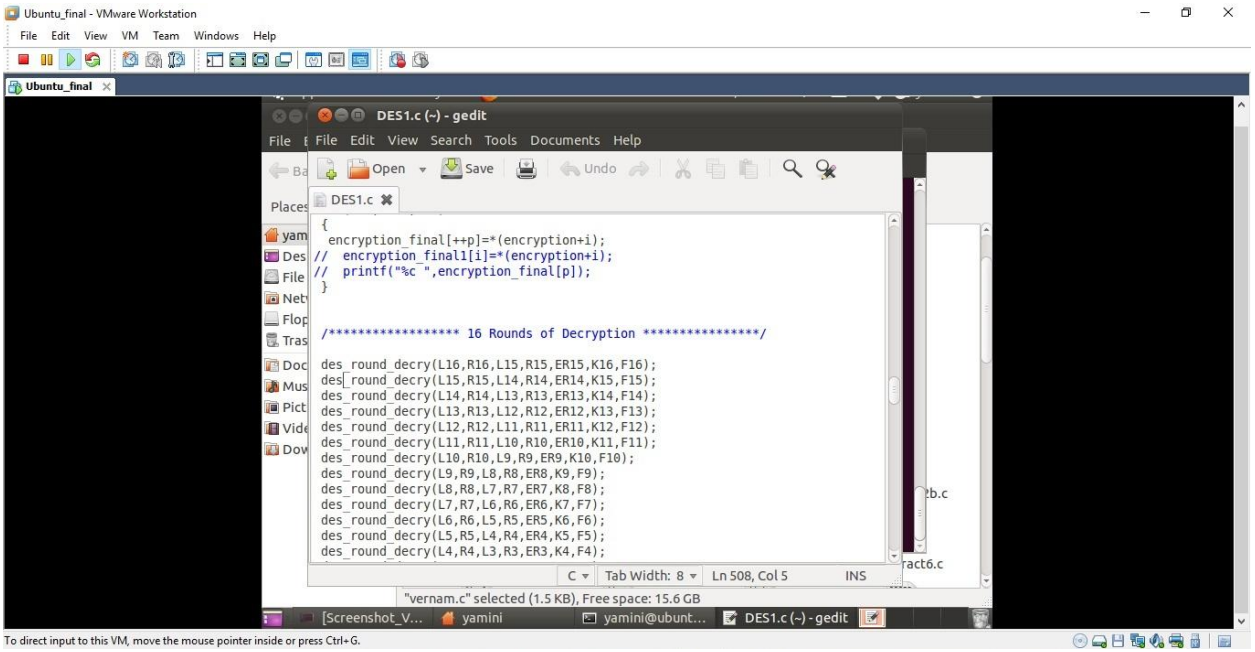
The crypt() function in PHP returns a string encrypted using DES. The DES Algorithm can be implement in PHP using function crypt(str, salt).



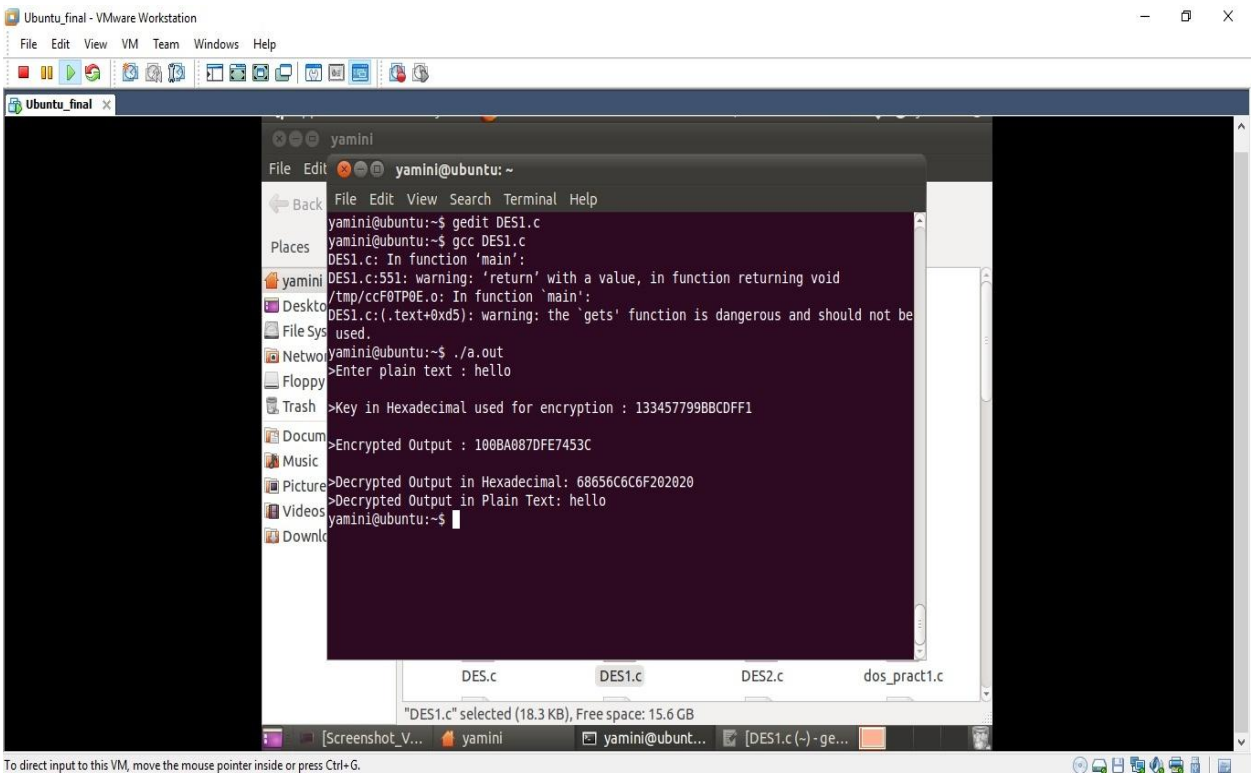
Implementation and comparison of DES encryption algorithm in C language Linux platform. Img 4.2.2.1, 4.2.2.2, 4.2.2.3



Img 4.2.2.1



Img 4.2.2.2


Ubuntu_final - VMware Workstation
File Edit View VM Team Windows Help
Ubuntu_final
yamini
File Edit
yamini@ubuntu: ~
File Edit View Search Terminal Help
yamini@ubuntu:~$ gedit DES1.c
yamini@ubuntu:~$ gcc DES1.c
DES1.c: In function 'main':
DES1.c:551: warning: 'return' with a value, in function returning void
/tmp/ccF0TP0E.o: In function `main':
DES1.c:(.text+0xd5): warning: the `gets' function is dangerous and should not be used.
yamini@ubuntu:~$ ./a.out
>Enter plain text : hello
>Key in Hexadecimal used for encryption : 133457799BBCDFF1
>Encrypted Output : 100BA087DFE7453C
>Decrypted Output in Hexadecimal: 68656C6C6F202020
>Decrypted Output in Plain Text: hello
yamini@ubuntu:~$
Back
Places
yamini
Desktop
File Sys
Networ
Floppy
Trash
Docum
Music
Picture
Videos
Downlc
DES.c
DES1.c
DES2.c
dos_pract1.c
"DES1.c" selected (18.3 KB), Free space: 15.6 GB
[Screenshot_V...
yamini
yamini@ubunt...
[DES1.c (~) - ge...
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Img 4.2.2.3

4.2.3 Security analysis of RSA algorithm

Key Generation

Select p, q	p, q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1) \times (q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

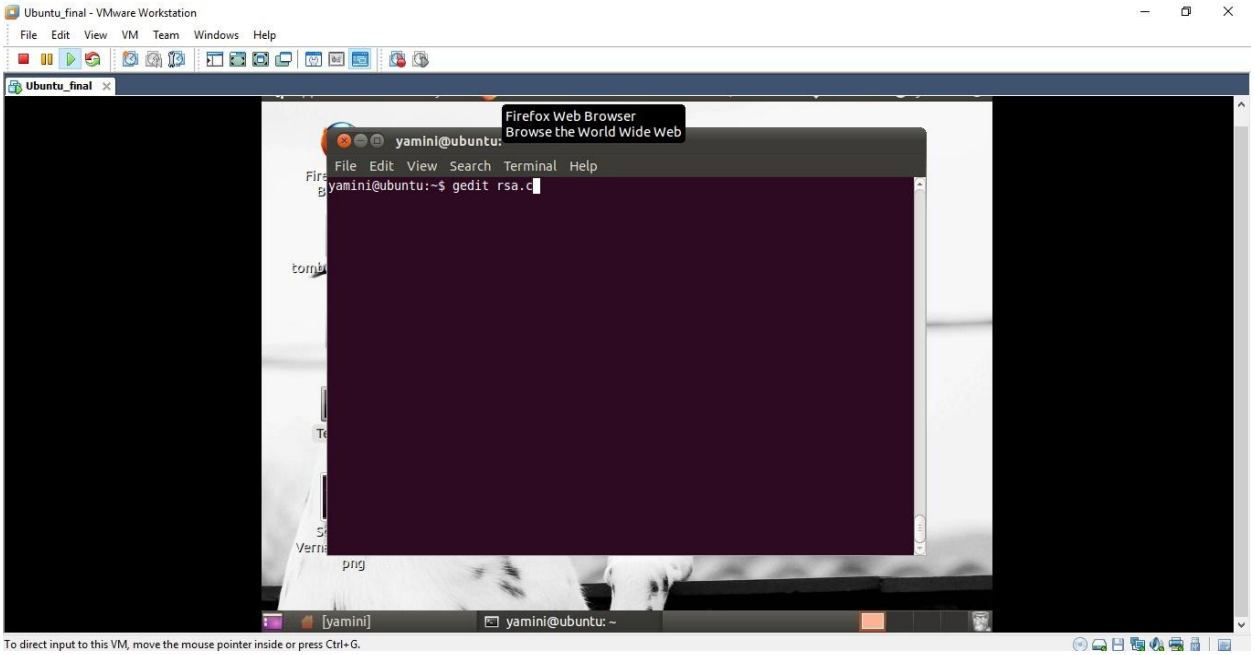
Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

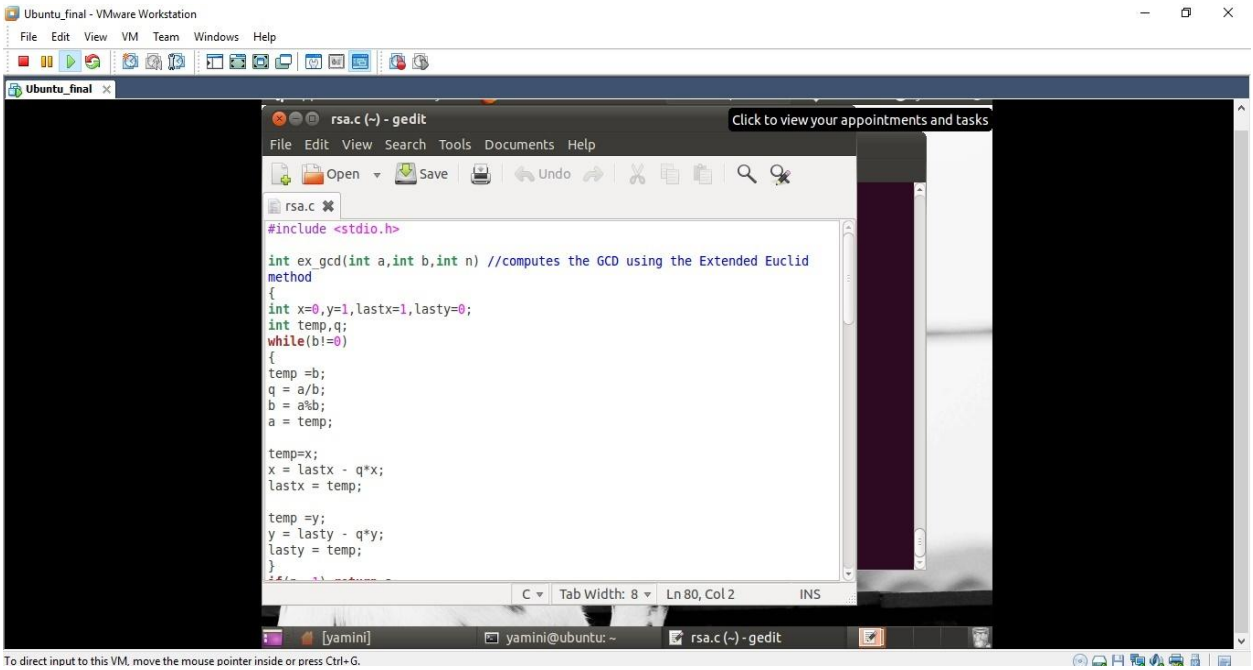
Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod n$

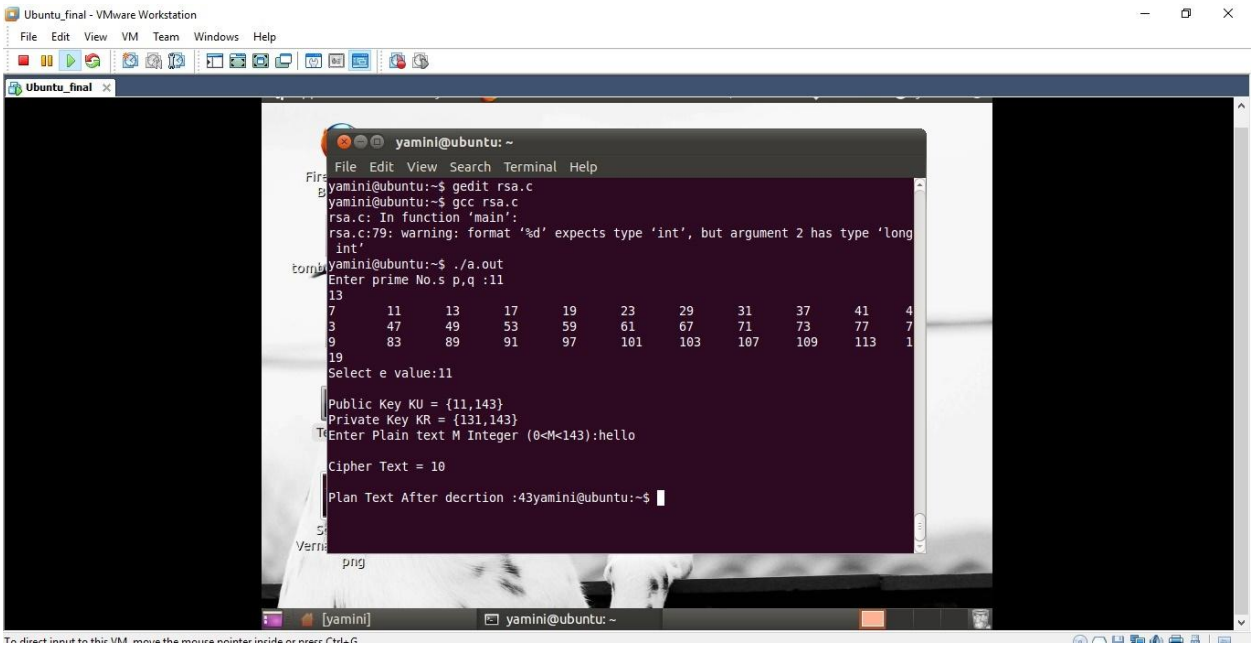
Implementation and comparison of RSA encryption algorithm in C language Linux platform. Img 5.2.3.1, 5.2.3.2, 5.2.3.3



Img 5.2.3.1



Img 5.2.3.2


Ubuntu_final - VMware Workstation
File Edit View VM Team Windows Help
Ubuntu_final
yamini@ubuntu: ~
File Edit View Search Terminal Help
yamini@ubuntu:~$ gedit rsa.c
yamini@ubuntu:~$ gcc rsa.c
rsa.c: In function 'main':
rsa.c:79: warning: format '%d' expects type 'int', but argument 2 has type 'long int'
yamini@ubuntu:~$ ./a.out
Enter prime No.s p,q :11
13
7 11 13 17 19 23 29 31 37 41 4
3 47 49 53 59 61 67 71 73 77 7
9 83 89 91 97 101 103 107 109 113 1
19
Select e value:11
Public Key KU = {11,143}
Private Key KR = {131,143}
Enter Plain text M Integer (0<M<143):hello
Cipher Text = 10
Plan Text After decrtion :43yamini@ubuntu:~$
[yamini]
yamini@ubuntu: ~

Img 5.2.3.3

4.2.4 Security analysis of MD2 algorithm

The algorithm is optimized for 8-bit computers. Although MD2 is no longer considered secure, even as of 2014, it remains in use in public key infrastructures as part of certificates generated with MD2 and RSA. The 128-bit hash value of any message is formed by padding it to a multiple of the block length (128 bits or 16 bytes) and adding a 16-byte checksum to it.

4.2.5 Security analysis of MD4 algorithm

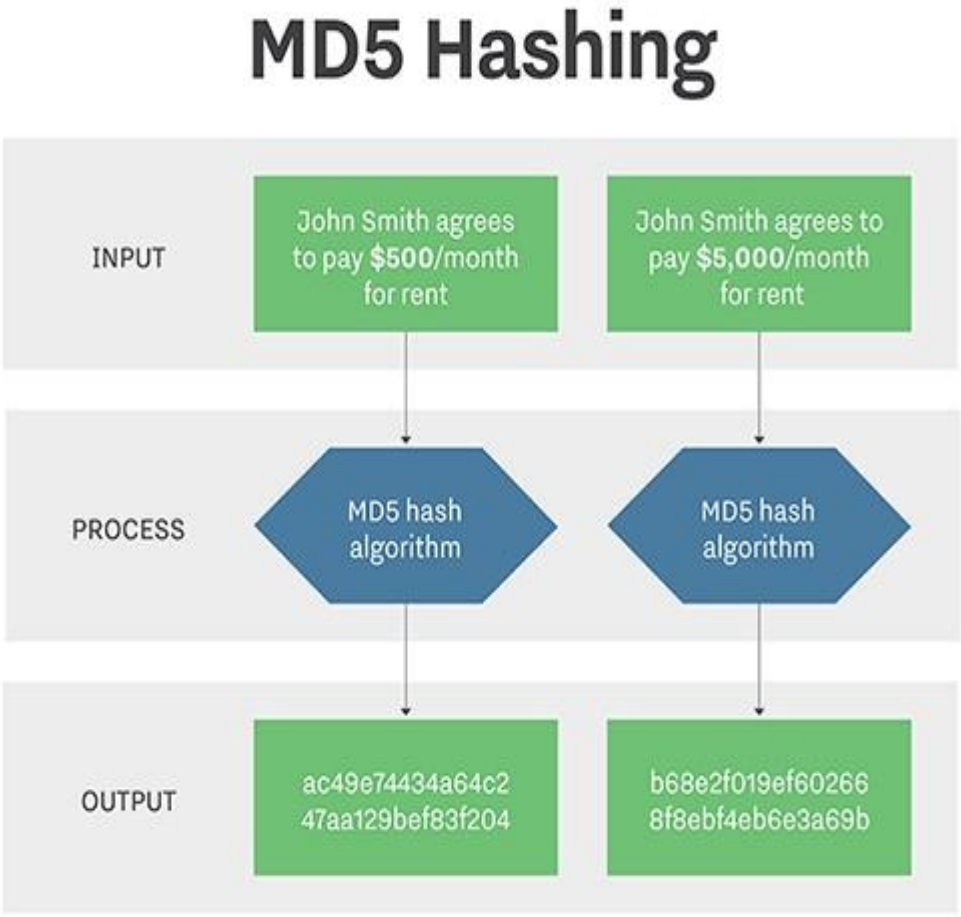
The digest length is 128 bits. The algorithm has influenced later designs, such as the MD5, SHA-1 and RIPEMD algorithms.

4.2.6 Security analysis of MD5 algorithm

The MD5 hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

The MD5 hash function was originally designed for use as a secure cryptographic hash algorithm for authenticating digital signatures. MD5 has been deprecated for uses other than as a non-cryptographic checksum to verify data integrity and detect unintentional data corruption.

Although originally designed as a cryptographic message authentication code algorithm for use on the internet, MD5 hashing is no longer considered reliable for use as a cryptographic checksum because researchers have demonstrated techniques capable of easily generating MD5 collisions on commercial off-the-shelf computers.



MD5 security

The goal of any message digest function is to produce digests that appear to be random. To be considered cryptographically secure, the hash function should meet two requirements: first, that it is impossible for an attacker to generate a message matching a specific hash value; and second, that it is impossible for an attacker to create two messages that produce the same hash value.

4.2.7 Security analysis of SHA1 algorithm

The Secure Hash Algorithms are a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS), including:

SHA-0: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name "SHA". It was withdrawn shortly after publication due to an undisclosed "significant flaw" and replaced by the slightly revised version SHA-1.

SHA-1: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm. Cryptographic weaknesses were discovered in SHA-1, and the standard was no longer approved for most cryptographic uses after 2010.

SHA-2: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. There are also truncated versions of each standard, known as SHA-224, SHA-384, SHA-512/224 and SHA-512/256. These were also designed by the NSA.

SHA-3: A hash function formerly called Keccak, chosen in 2012 after a public competition among non-NSA designers. It supports the same hash lengths as SHA-2, and its internal structure differs significantly from the rest of the SHA family.

SHA-256 : SHA-256 Cryptographic Hash Algorithm.

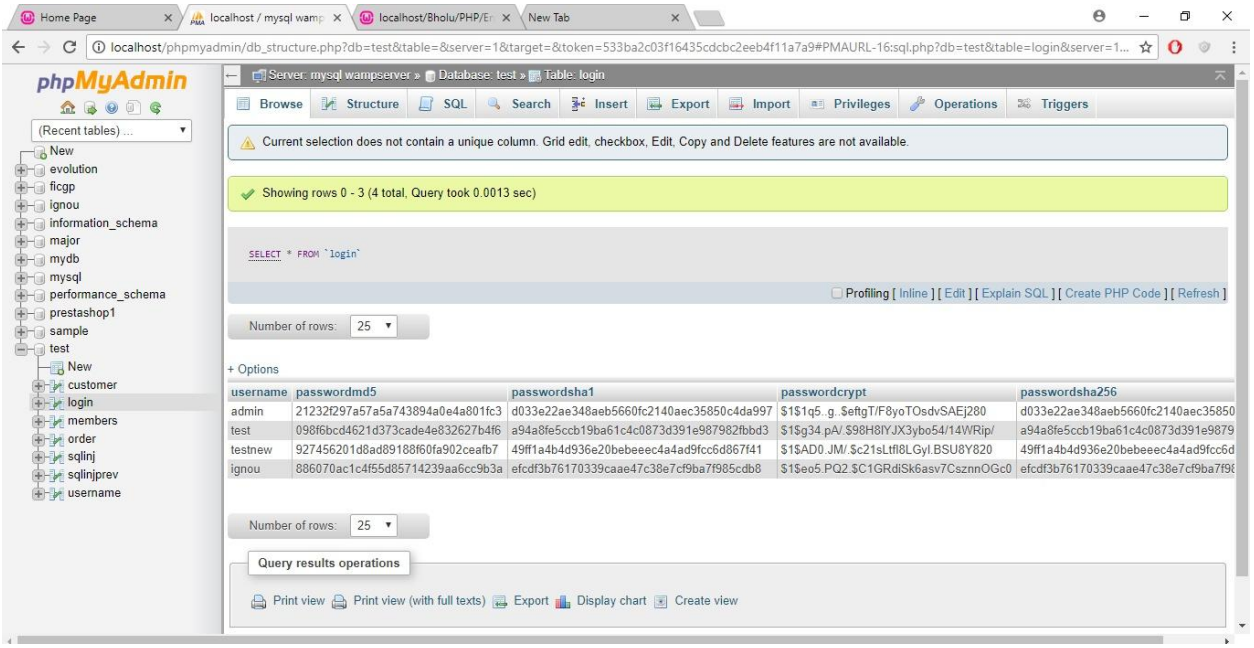
SHA-256 is one of the successor hash functions to SHA-1 (collectively referred to as SHA-2), and is one of the strongest hash functions available. SHA-256 is not much more complex to code than SHA-1, and has not yet been compromised in any way.

4.2.8 Security analysis of Crypt algorithm

In Unix computing, crypt is a utility program used for encryption. Due to the ease of breaking it, it is considered to be obsolete.

4.2.9 Implementation of MD5, SHA-1, Crypt and SHA-256 in PHP

Img 4.2.9.1



Img 4.2.9.1

4.2.10 Comparison of encryption algorithm

Passwords play an important role in daily life in various computing applications and play a critical role in online authentication. The main aim for using passwords is to restrict unauthorized users to access the system.

A technique to obtain secure online passwords is password hashing, where hashed passwords are sent to databases or remote websites. Hashing is an important technique used for secure communication in the presence of eavesdroppers. It provides all the paramount aspects of information security such as integrity, authentication and confidentiality.

Password hashing is lightweight and convenient to use and can defend against phishing attacks.

I have implemented all encryption algorithm and hashing algorithms in different languages to identify the best algorithm.

Post implementation compared it using different parameters

Comparison between RSA and DES algorithms on general properties basis. Below comparison table is taken from one of my published research paper. Img 4.2.10.1

Table 1: Comparison Table

Features	DES	RSA
Execution Time	Low	High
Throughput	High	Low
Key Used	Same key is used for encryption and decryption	Different key is used for encryption and decryption
Scalability	It is scalable	Not scalable
Avalanche Effect	No more effected	More effected
Power Consumption	High	Low
Security	High	Low
Confidentiality	High	Low

Img 4.2.10.1

Comparison between MD5 and SHA hash algorithms on general properties basis. Img 4.2.10.2

Name Of The Algorithm	Size Of Output	Rounds	Collision Status
MD5	128	60	YES
SHA	160	80	YES
SHA-1	160	80	YES
SHA-2	256/512	60/80	THEORITICAL
SHA-192	192	80	NO
SHA-192	192	64	No
SHA-3	256/512	24	NO

Img 4.2.10.2

Features	MD5	SHA
Security	Less Secure than SHA	More Secure
Length Of Message Digest	128 Bits	160 Bits
No. Of Attacks Needed To Find Original Message	2^{128} bit operations Required	2^{160} bit operations required
Attacks to try and find two messages producing the same MD	2^{64} bit operations Required	2^{80} bit operations required
Speed	Faster, 60 iterations	Slower, 80 iterations
Successful attacks so far	Attacks reported some extend	No such attack reported

Img 4.2.10.3

This comparative study helped to understand that the SHA algorithm plays a very important role in comparison to MD5 because SHA algorithms' performance rate is comparatively better than other cryptographic hash algorithm functions.

As a project work, I propose to implement hashing SHA-256 to store passwords so as to get the best of it.

4.3 Application Security - Validation

4.3.1 Types of Validation

1. Prospective Validation.

This type of validation is performed before production, during a product's development stage. A risk analysis is performed to assess the production process by breaking it down into separate steps.

2. Concurrent Validation.

We should monitor the first three batches produced on a production-scale as closely as possible. The data gathered through this step can provide an in-depth insight of the fundamentals, which greatly impacts the effectiveness of concurrent validation.

3. Retrospective Validation.

As the name suggests, retrospective validation is rather like validation in hindsight. It involves examining the past experiences of the process and evaluating the final control tests.

4. Revalidation.

Revalidation is essential for ensuring that any changes made to the process or its environment have not resulted in adverse effects on product quality or process characteristics.

5. Client side and Server side Validation.

Validation performed for the inputs entered by users and it are validated at server end before storing it on database server.

4.3.2 Characteristics of Validation

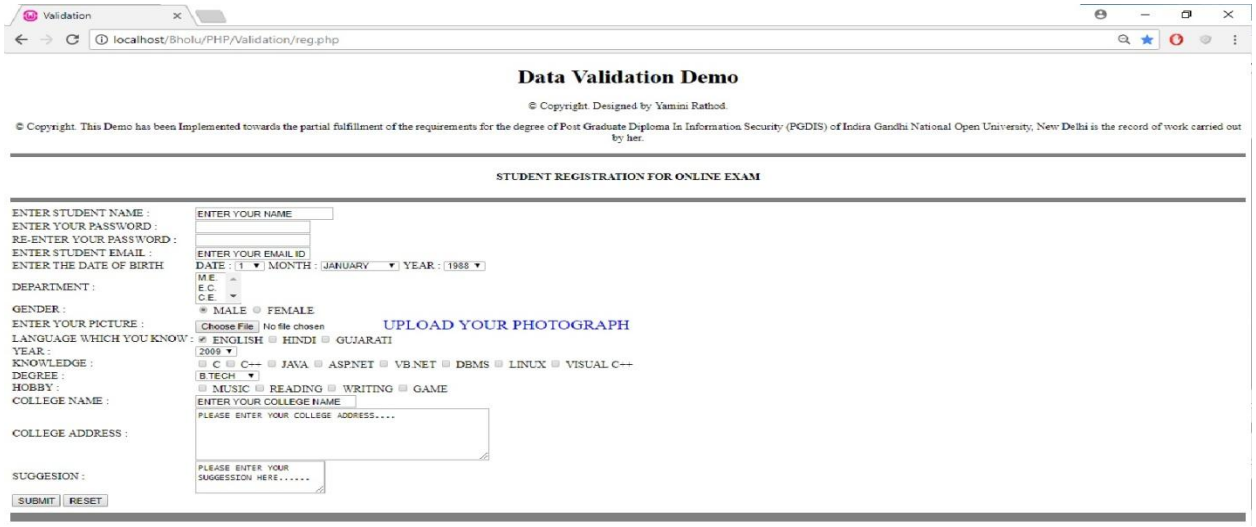
I have performed above mentioned validation during the project life cycle, along with that implemented the client side and server side validation.

4.3.3 Advantages of Validation

Validation is required to prevent web form abuse by malicious users. Improper validation of form data is one of the main causes of security vulnerabilities. It exposes your website to attacks such as header injections, cross-site scripting, and SQL injections.

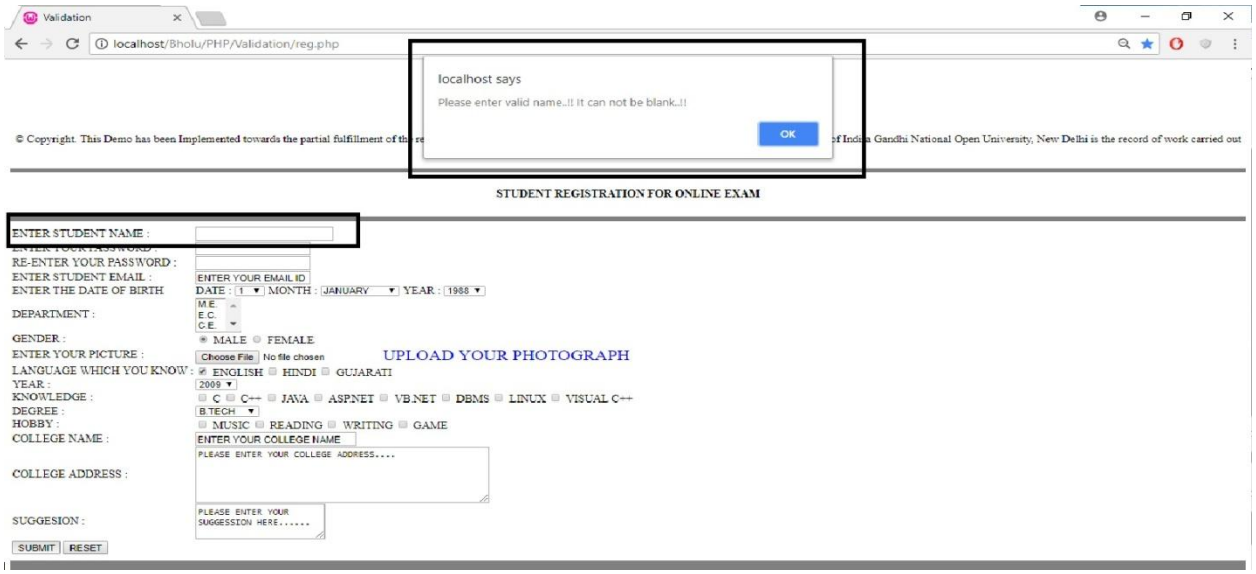
4.4.4 Implementation of Validation

In order to demonstrate the validation in an application, I have designed the sample registration form. Img 4.4.4.1



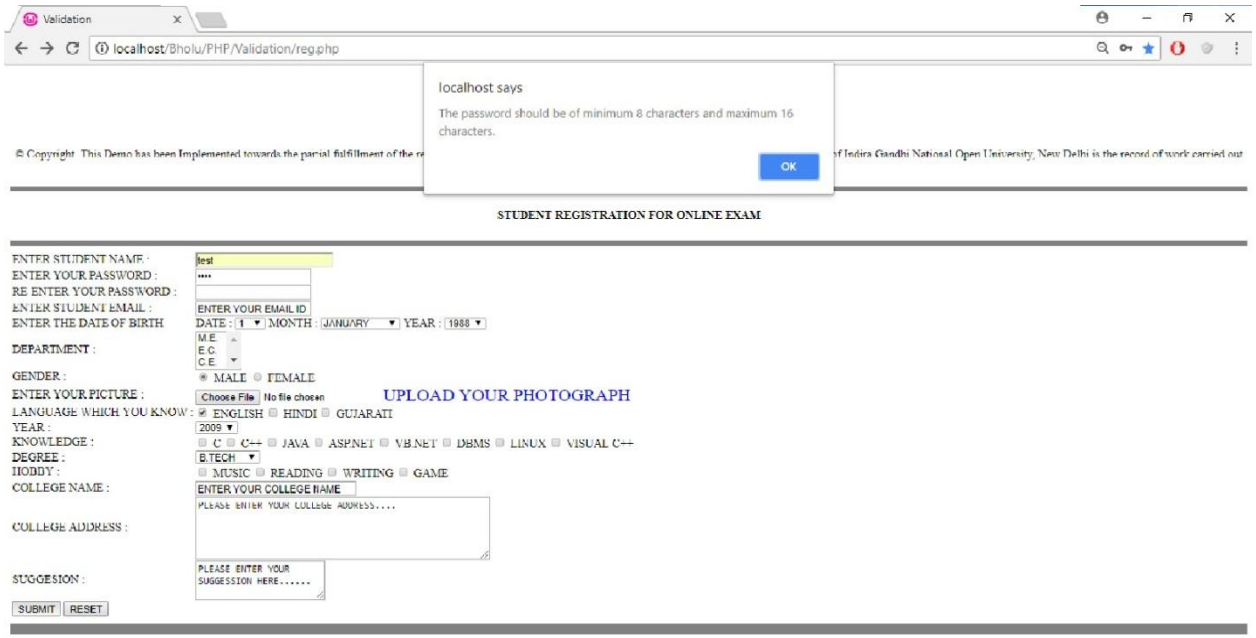
Img 4.4.4.1

Type of Validation : Null data checks Img 4.4.4.2



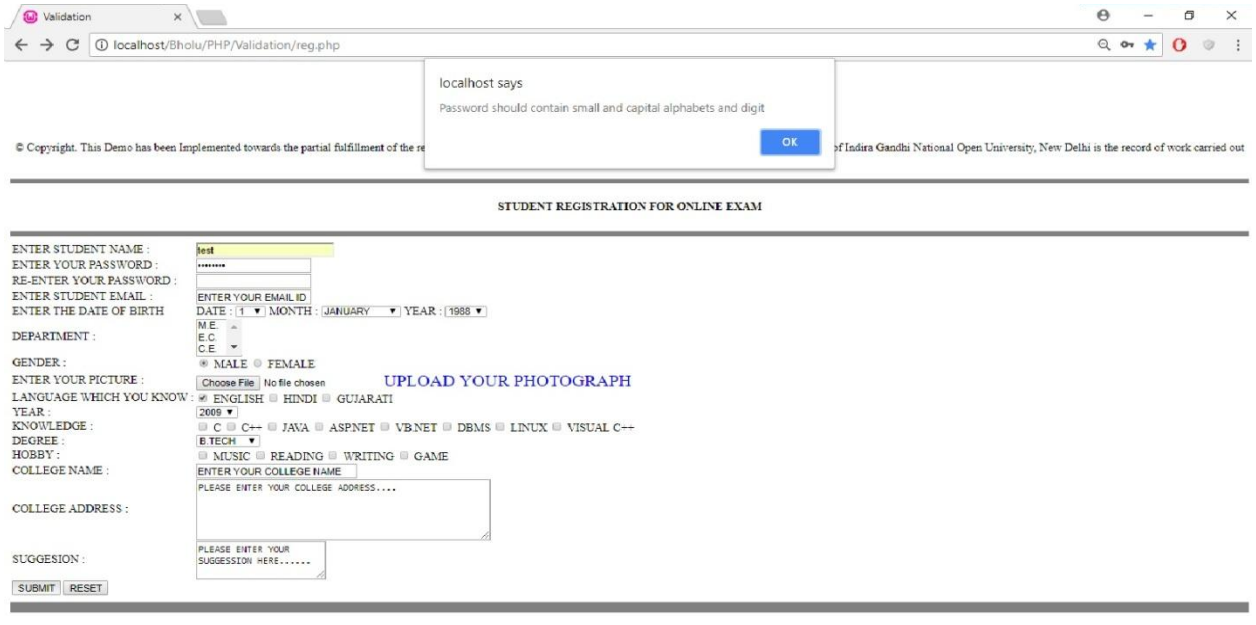
Img 4.4.4.2

Type of Validation : Allowed character checks Img 4.4.4.3



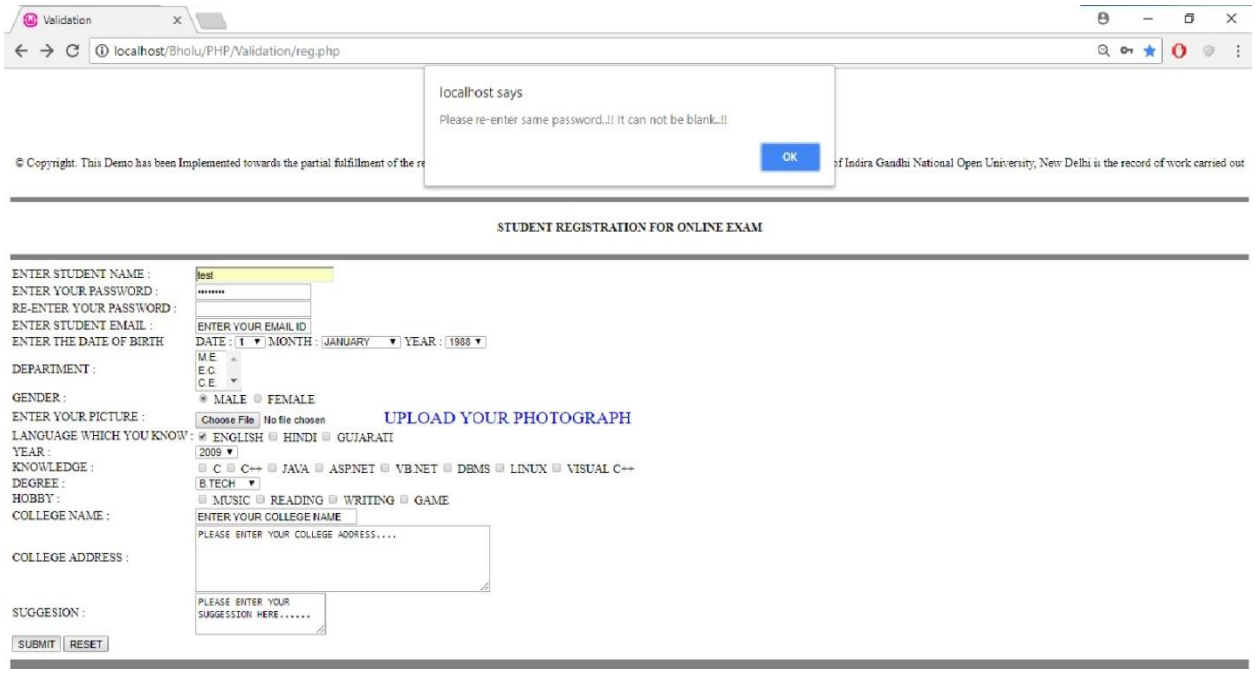
Img 4.4.4.3

Type of Validation : Check digits Img 4.4.4.4



Img 4.4.4.4

Type of Validation : Consistency checks Img 4.4.4.5



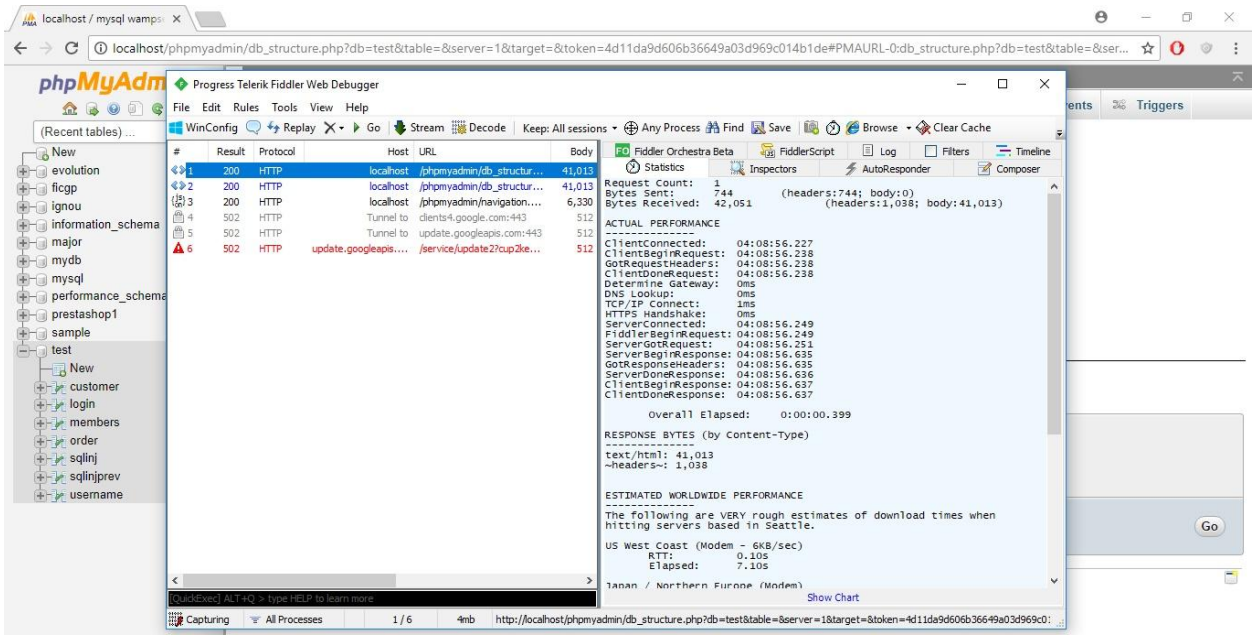
Img 4.4.4.5

There are many other things can be implemented such as File existence check, Limit check, Logic check, Presence check, Uniqueness check which can be implemented as per requirements.

I have included all possible validation techniques in project and will be demonstrate in upcoming chapters.

4.4 System Security Logs - Fiddler

Fiddler is a free web debugging proxy which logs all HTTP(s) traffic between your computer and the Internet. Use it to debug traffic from virtually any application that supports a proxy like IE, Chrome, Safari, Firefox, Opera and more.
Please refer below screenshot, I testing fiddler logs simply by accessing PHP MY Admin screen of Wamp server. Img 4.4.1



Img 4.4.1

Advantages of Fiddler Tool :

- A freeware tool to capture HTTP and HTTPS traffic
- Can also be used to modify traffic as a troubleshooting measure
- Useful in cases of :

1. Performance issues
2. Connectivity problems
3. Incorrect output in web pages

It is good habit to use such tools to keep track on network traffic, malicious user access on network, monitor the application performance, identify the fault path, and many more.

I have used this tool in my application to utilize above mentioned features.

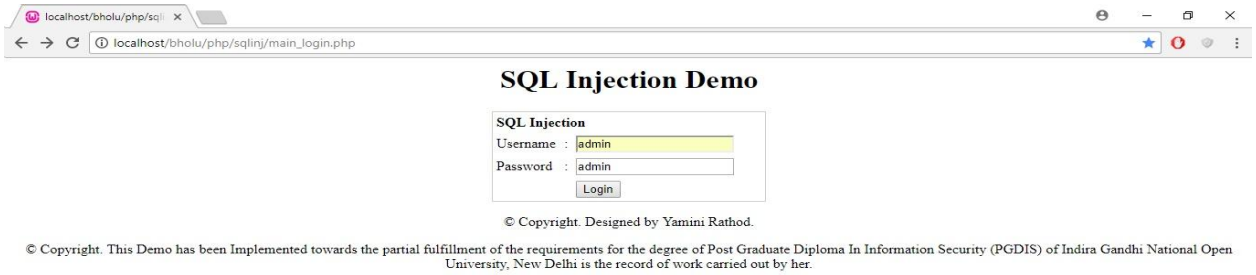
4.5 SQL Injection

4.5.1 SQL Injection attacks

In order to demonstrate this attack I have designed sample login page. Img 4.5.1.1

Img 4.5.1.1

No, try to enter some fake username and password.

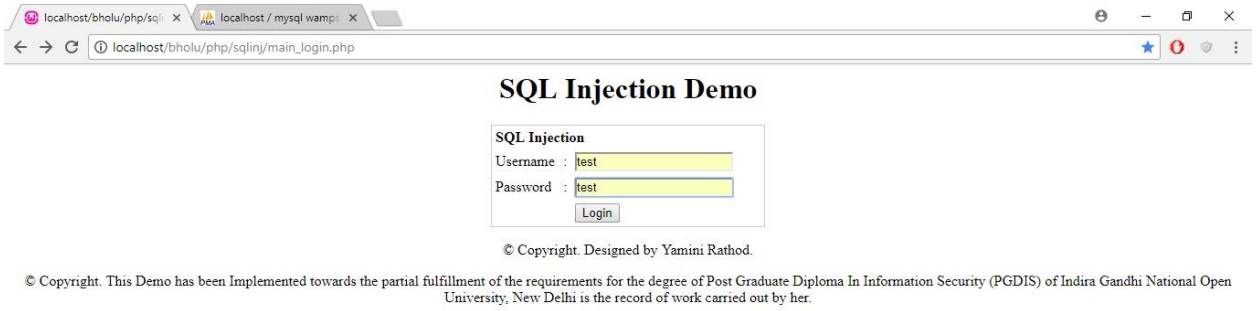


Img 4.5.1.2

I designed the redirect page in such way that we can track the query that is executing in background to get the better understanding on how actually SQL Injection works. Img 4.5.1.3

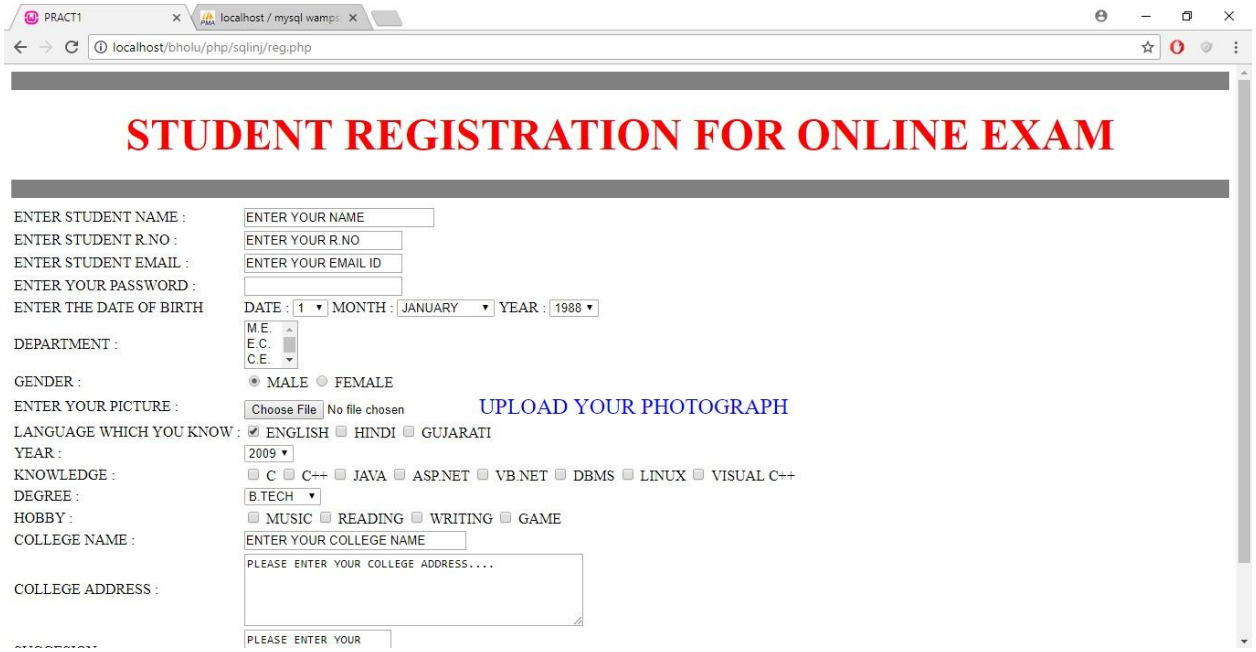
Img 4.5.1.3

Now, add valid username and password stored in database. Img 4.5.1.4



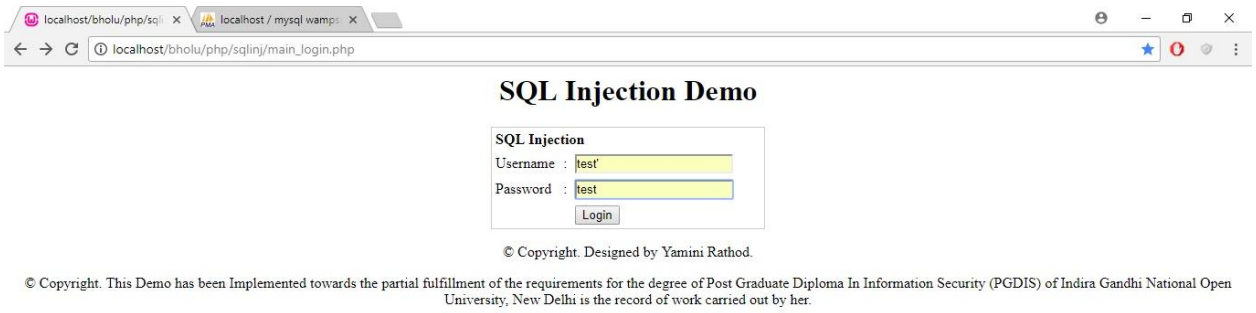
Img 4.5.1.4

Post login, system will redirect us on targeted registration page. Img 4.5.1.5



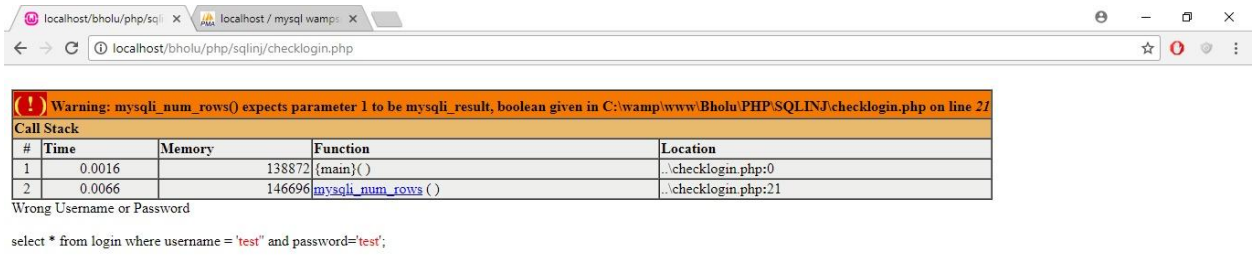
Img 4.5.1.5

Now, add special characters in authentication info like ‘ Img 4.5.1.6



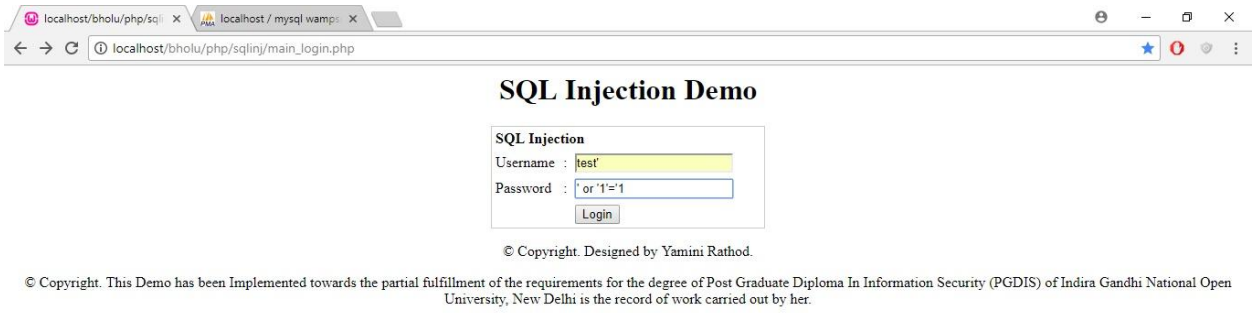
Img 4.5.1.6

Observe the query running in back ground how the added special character is processing behind the scene. Img 4.5.1.7



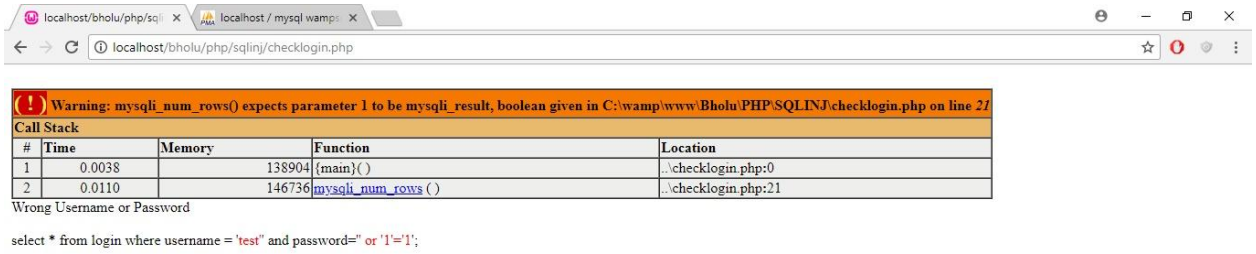
Img 4.5.1.7

As per the query executing in background, we can now try to login using password : ‘ or ‘1’=’1 Img 4.5.1.8



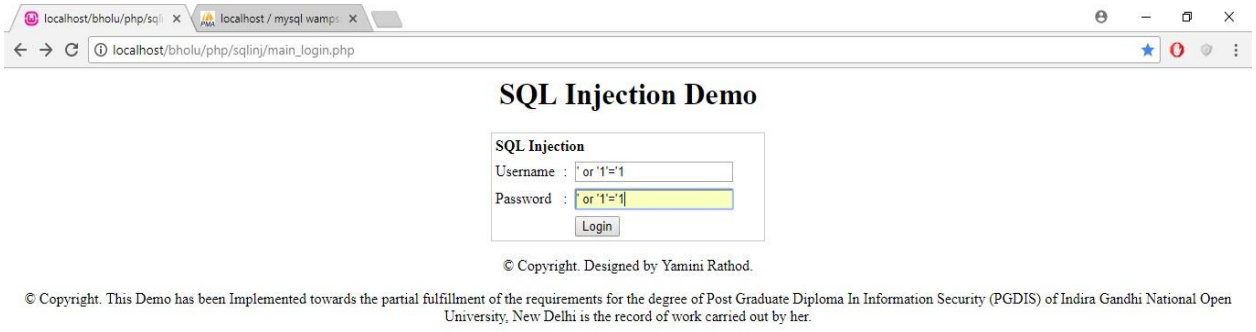
Img 4.5.1.8

Observe the query executing in background. Img 4.5.1.9



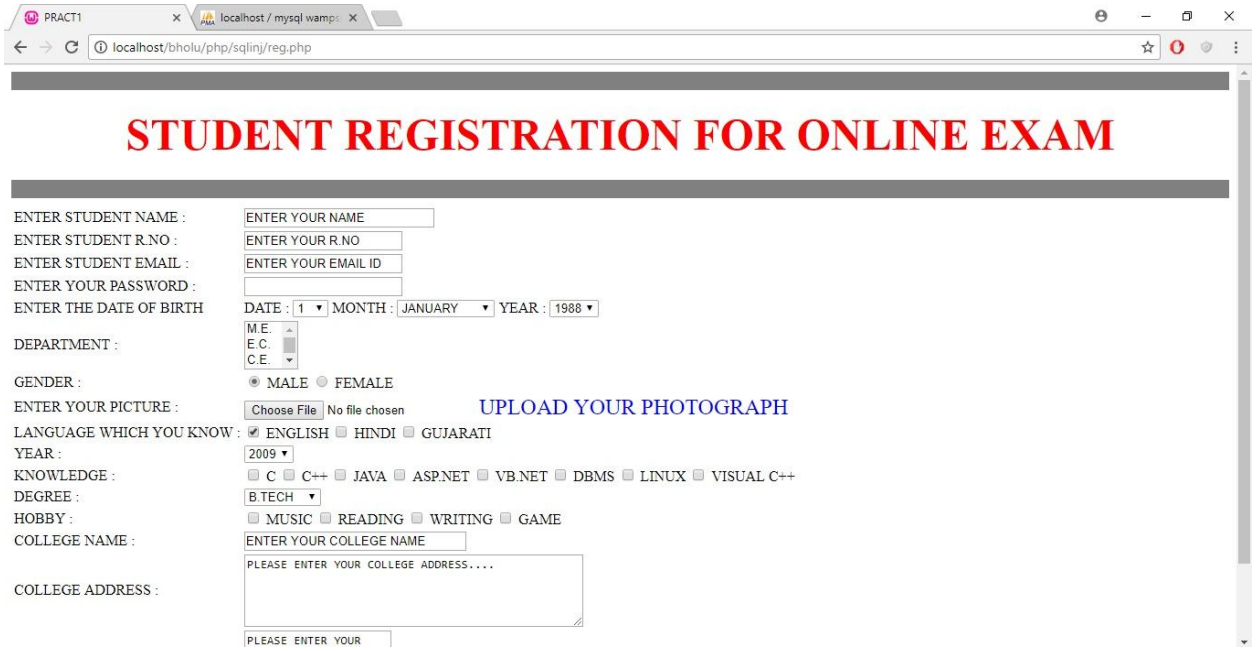
Img 4.5.1.9

Try to login using ‘ or ‘1’=’1 which will be always true. Img 4.5.1.10



Img 4.5.1.10

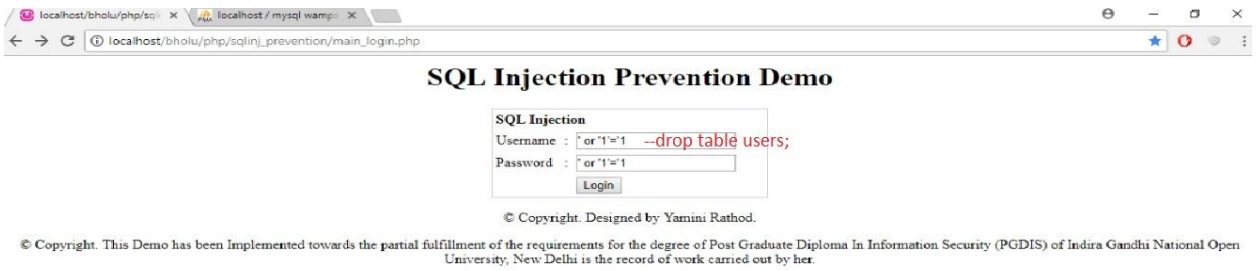
Check that we logged-in and redirected to targeted registration page using SQL Injection. Img 4.5.1.11



Img 4.5.1.11

SQL Injection attack was done on above application due to lack of encryption and lack of SQL Injection prevention code.

It is however not limited to just authentication, smart attacker can even hack the sensitive information or just delete the database itself. Img 4.5.1.12



Img 4.5.1.12

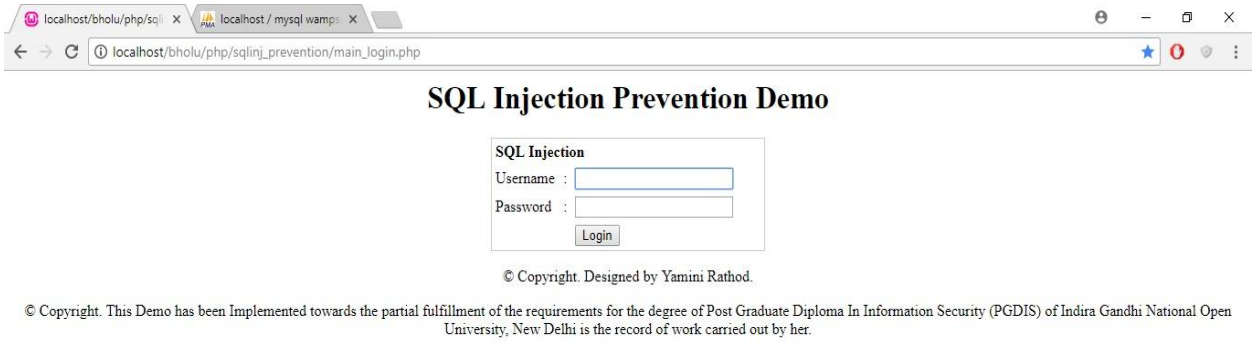
4.5.2 SQL Injection prevention

SQL Injection Prevention can be implemented by including,

1. Encryption.
2. Implement SQL Injection Prevention code.
3. Implement the validation.
4. Implement SSL.
5. Use parameterized queries when dealing with SQL queries that contains user input.
6. A parameterized query allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
7. Do not display SQL errors to the user. If you need to show the user an error, use a generic error message that does not give away sensitive information.

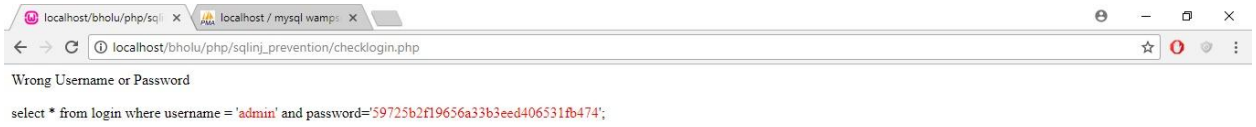
Here, I am able to prevent the SQL Injection by including Encryption and changing query structure as parameterized query.

Designed sample Demo application to demonstrate the SQL Injection Prevention. Img 4.5.2.1



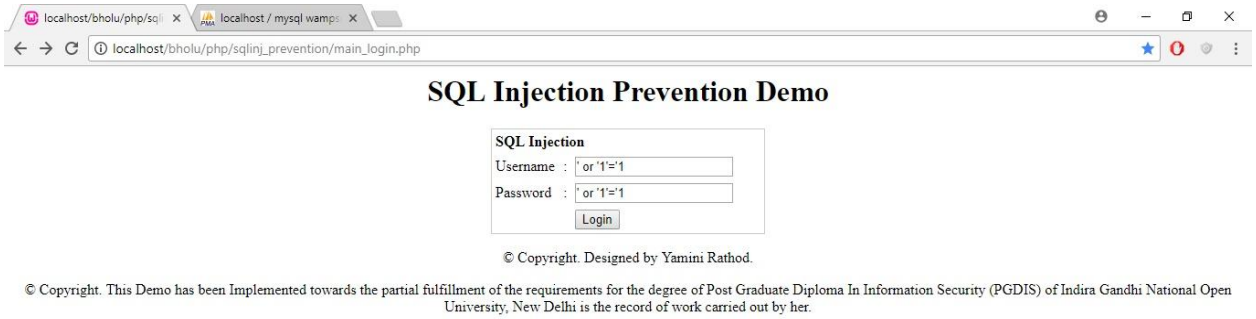
Img 4.5.2.1

Observe the query running in background to validate the user authentication. Img 4.5.2.2



Img 4.5.2.2

Try to attack using injection password ‘ or ‘1’=’1 Img 4.5.2.3



Img 4.5.2.3

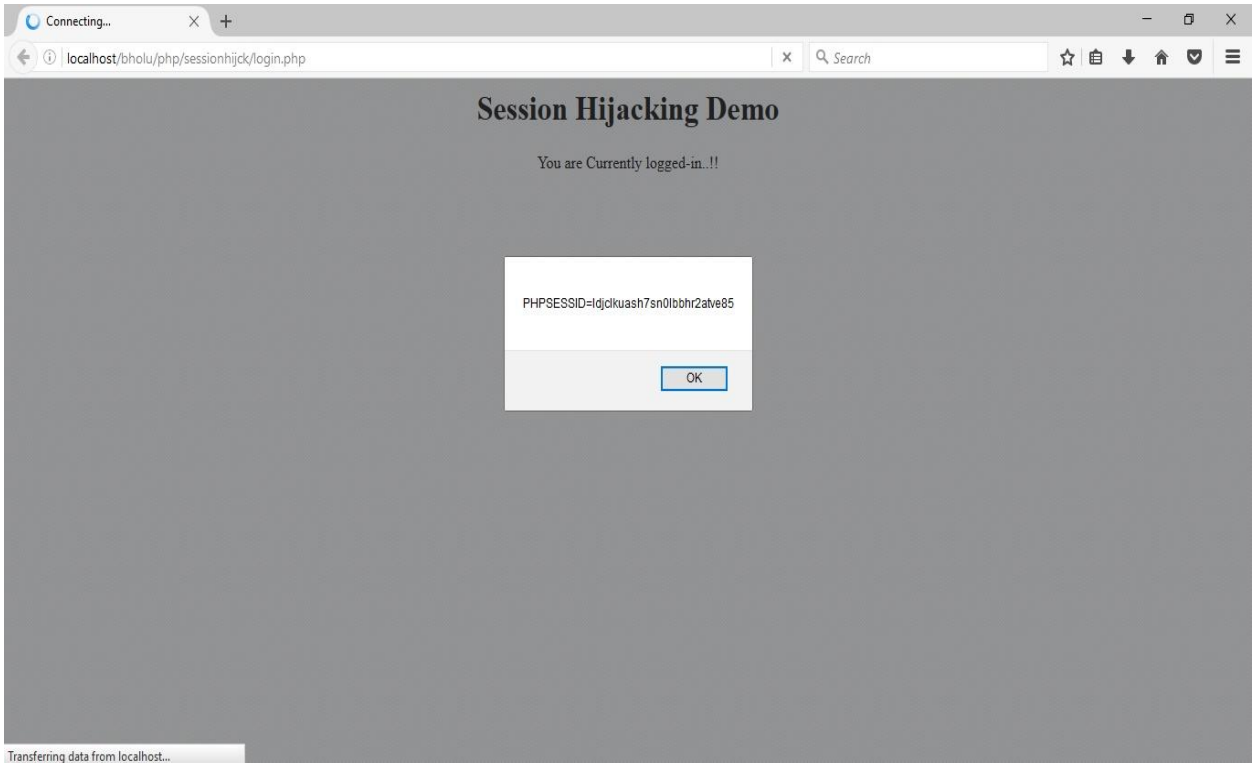
Observe the query running in back ground, it won't allow to get into targeted page due to prevention code and encryption. Img 4.5.2.4

Img 4.5.2.4

4.6 Session Hijacking

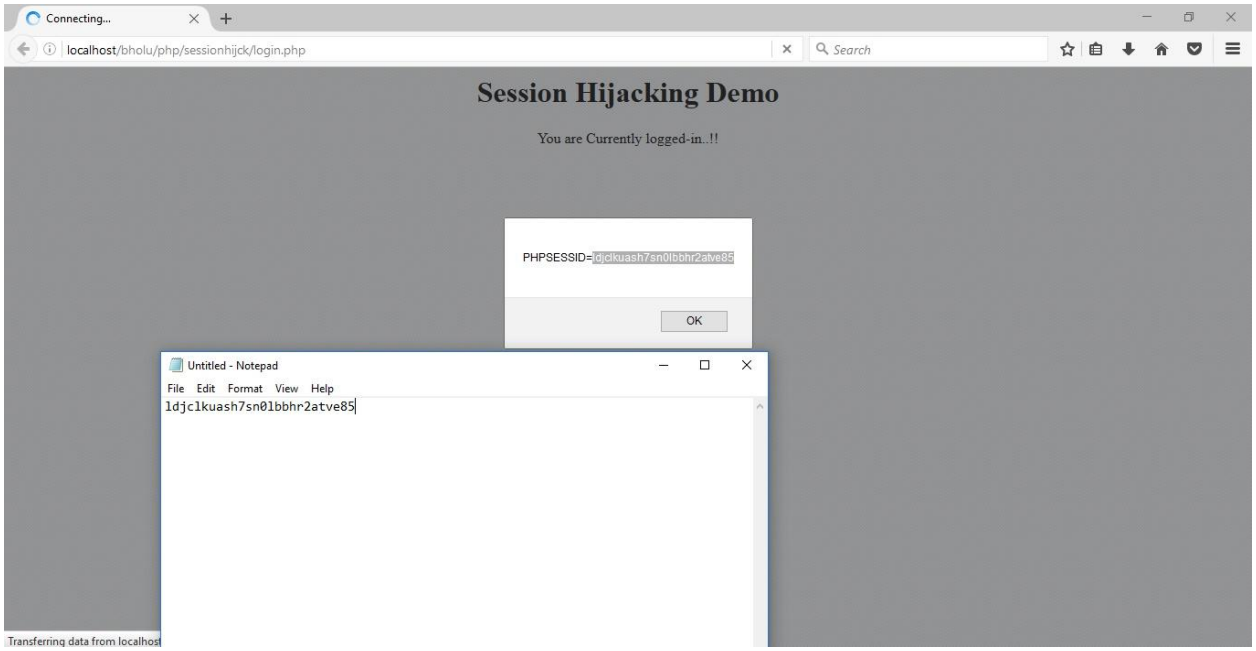
4.6.1 Session Hijacking attacks

Implemented sample demo application to demonstrate the Session Hijacking. I have designed PHP code to get the cookie value from login page. Img 4.6.1.1



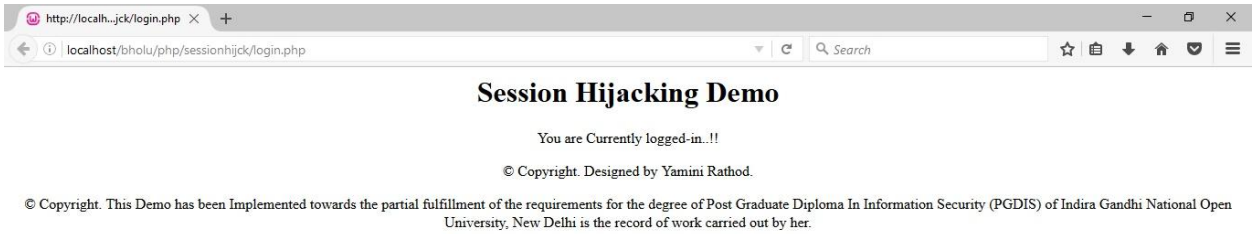
Img 4.6.1.1

Now, note down the cookie value fetched from login page. Img 4.6.1.2



Img 4.6.1.2

Click on Ok and get away from pop up, we are now on main login page. Img 4.6.1.3

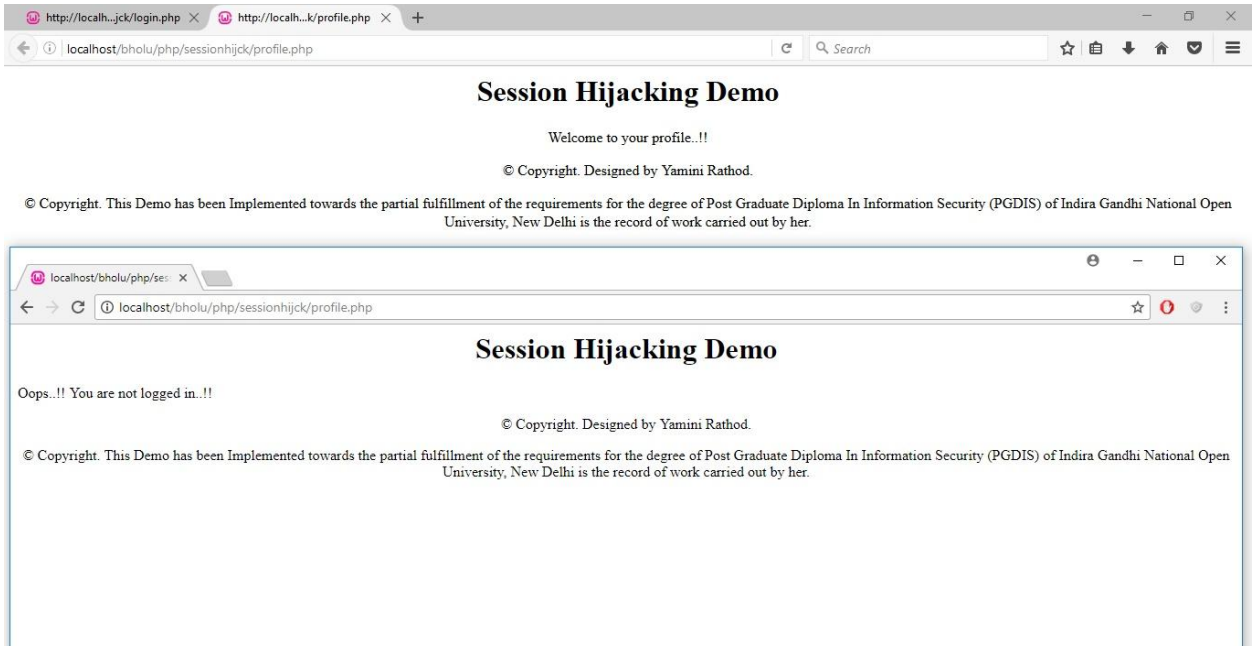


Img 4.6.1.3

As we are already logged-in on first page, as per the session id, application gives welcome message on our targeted profile.php Img 4.6.1.4

Img 4.6.1.4

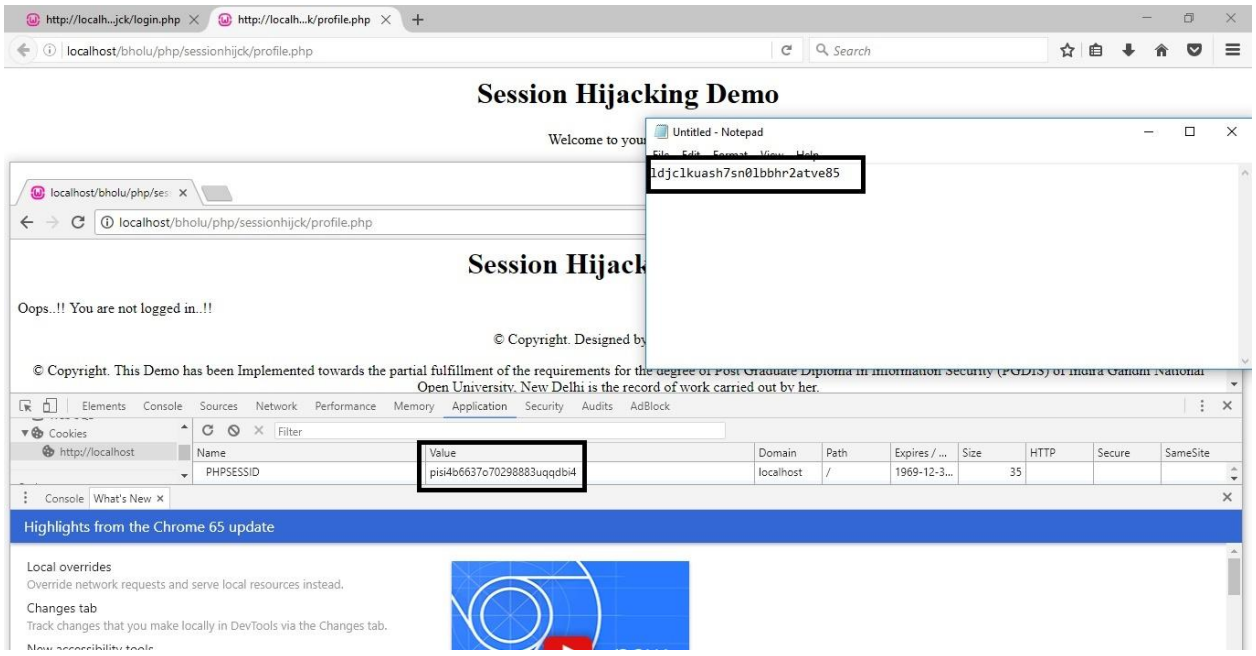
Now try to open the profile.php link from different browser where the session is not running for first main home page. We will get the message stating you are not logged-in as per sessionid. Img 4.6.1.5



Img 4.6.1.5

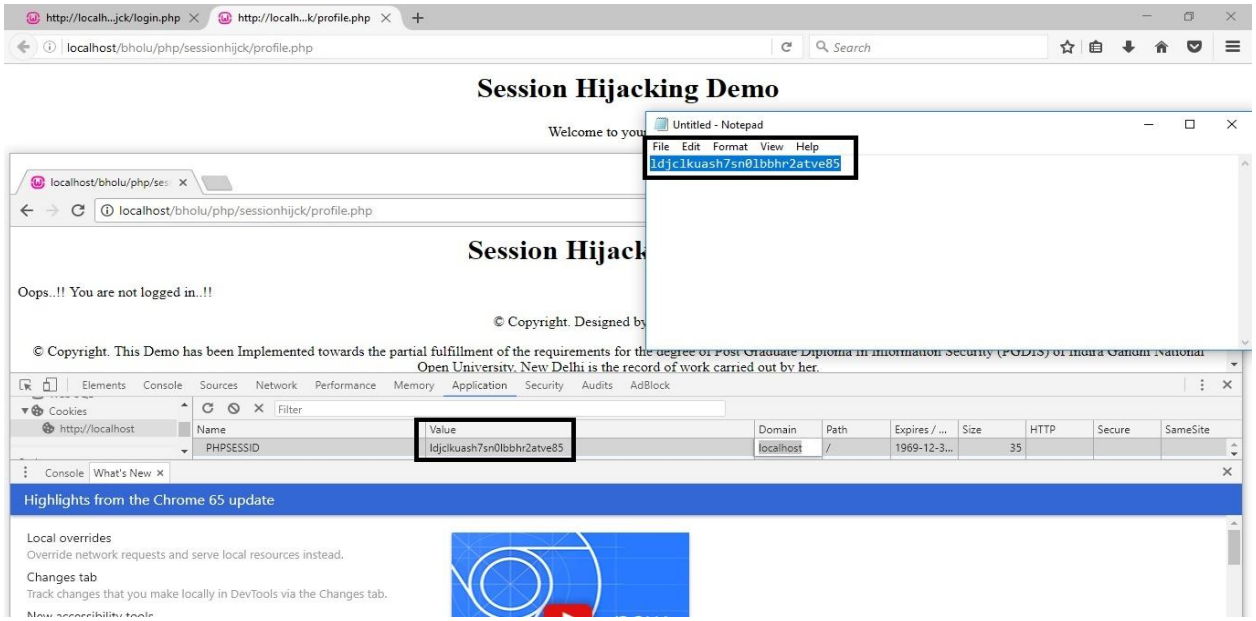
Now, open inspect elements from second browser where the session is not running for main login page.

Compare the value of cookie that we noted from first browser and which is displaying on second browser inspect element. Img 4.6.1.6



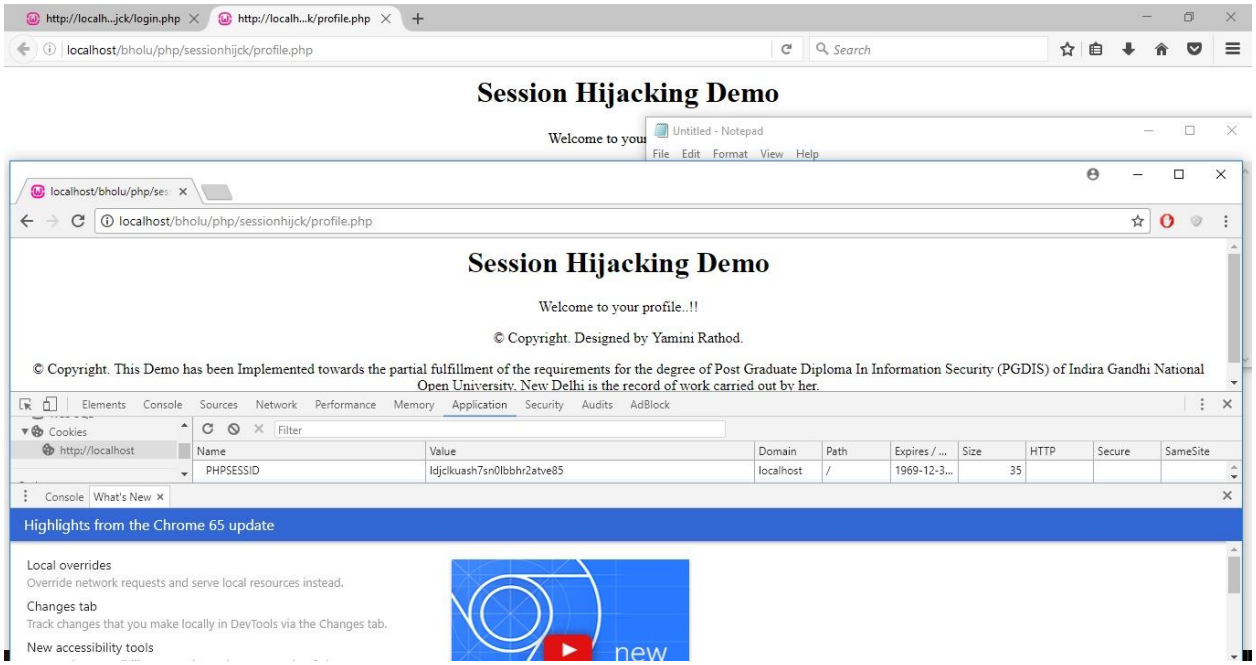
Img 4.6.1.6

Copy the value of Cookie from Notepad and modify it on second browser. Img 4.6.1.7



Img 4.6.1.7

Refresh the page and observe that we are getting logged-in message now. We authenticated using value of cookie. Img 4.6.1.8



Img 4.6.1.8

It is just not limited to the browsers, in case of client server architecture, attackers use to fetch the seesionid/cookieid by means of Man In Middle attacks.

4.6.2 Session Hijacking prevention

We can prevent the session hijacking by implementing SSL.

It was not feasible to implement the SSL on Demo application hence we have done case study on it.

Use of a long random number or string as the session key. This reduces the risk that an attacker could simply guess a valid session key through trial and error or brute force attacks.

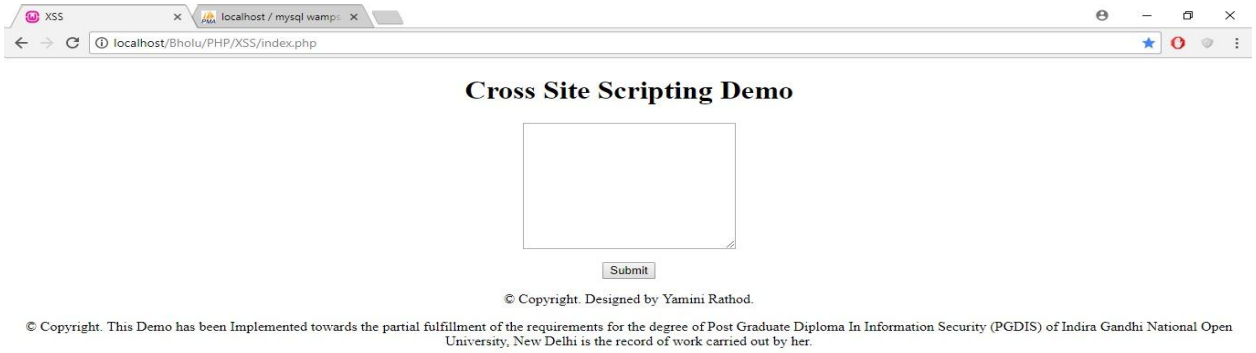
It is always suggested to use the session timeout feature and two level authentication to avoid this attack, session timeout is one of the best solution to prevent the session hijacking.

It has been implemented and demonstrated in upcoming chapters.

4.7 Cross site scripting

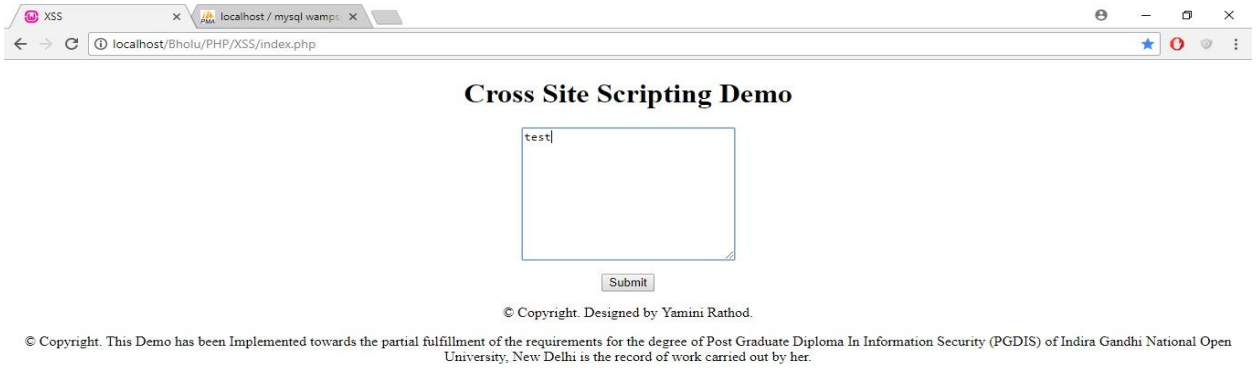
4.7.1 Cross site scripting attacks

I have implemented demo application to describe the cross site scripting attack. Img 4.7.1.1



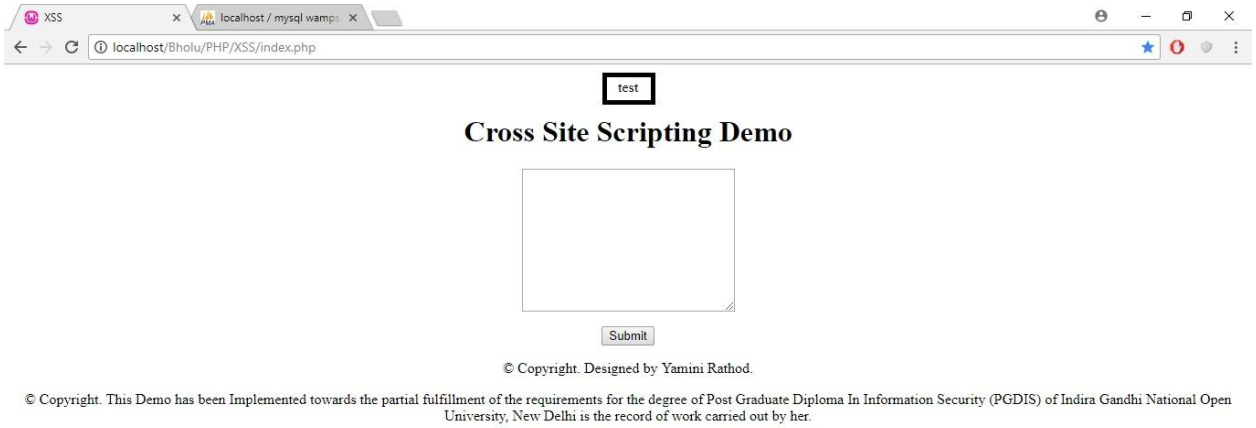
Img 4.7.1.1

Enter some text, the code has been implemented in such a way that it redirects the added information to the top of the php scripts by filtering out the content. Img 4.7.1.2



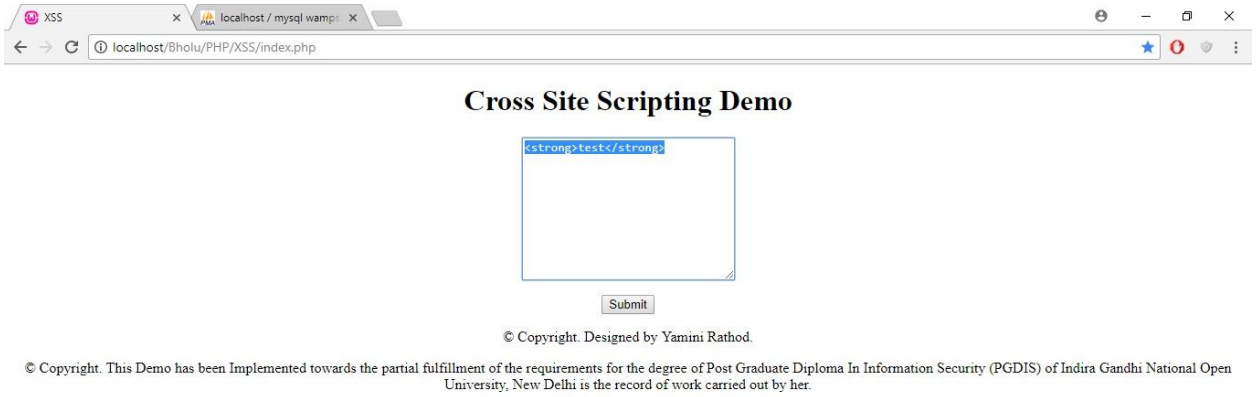
Img 4.7.1.2

Observe the added text reflecting at the top of the page. Img 4.7.1.3



Img 4.7.1.3

Now, add some value in form of html tags. Img 4.7.1.4

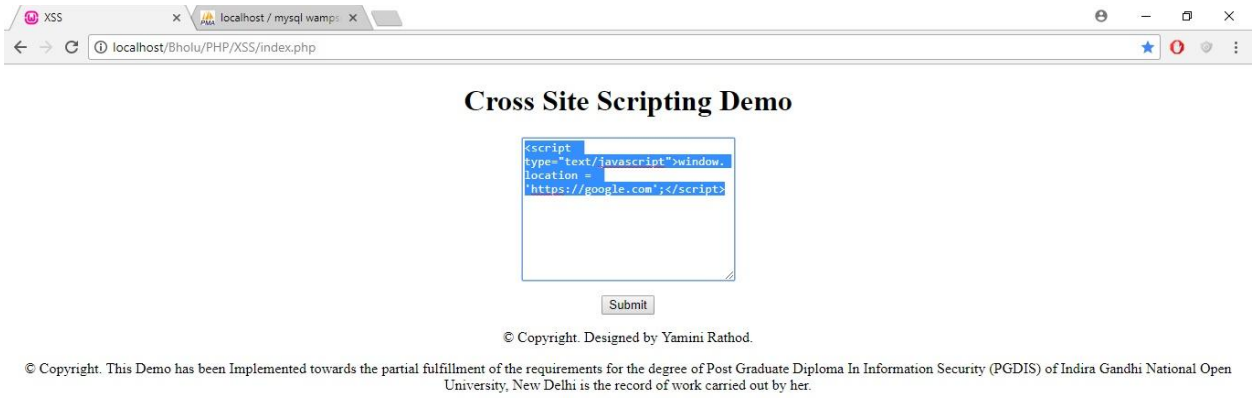


Img 4.7.1.4

Observe the output. We added contents in tag strong, it is reflecting the contents in bold format. Img 4.7.1.5

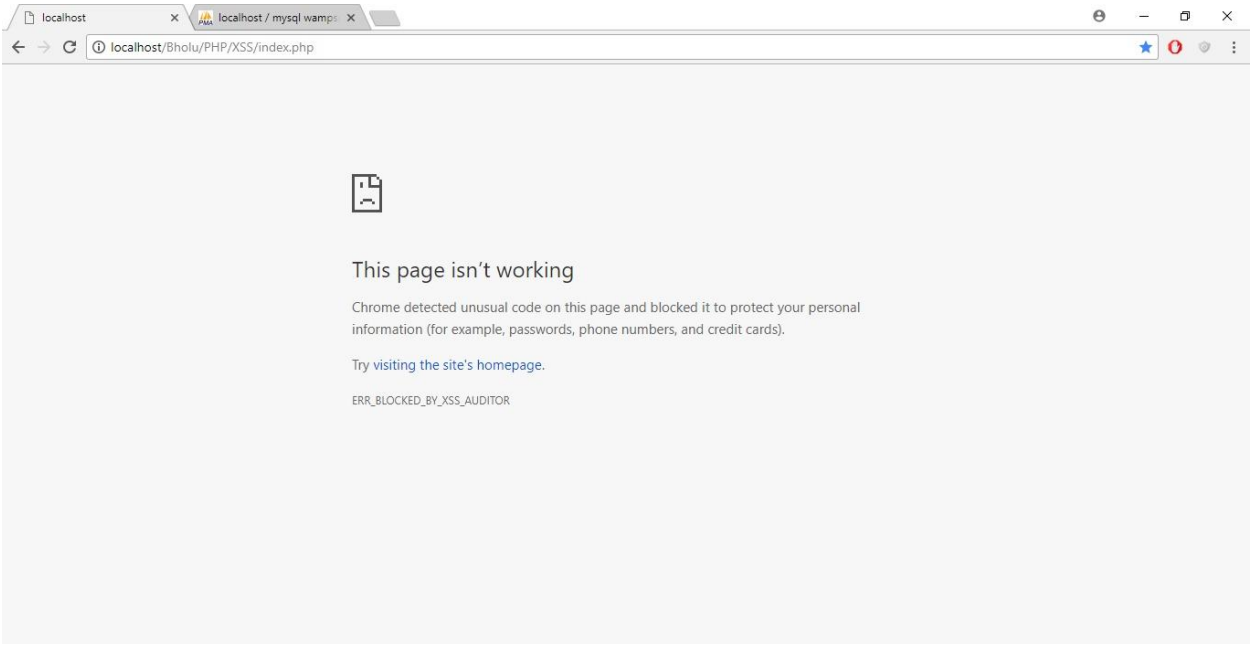
Img 4.7.1.5

Add some malicious code which will redirect to some other page. Img 4.7.1.6



Img 4.7.1.6

The output depends upon the network, security in machine, and versions of your browsers. The attack was unsuccessful in my local environment due to security. It runs successfully in non protected environment. Img 4.7.1.7



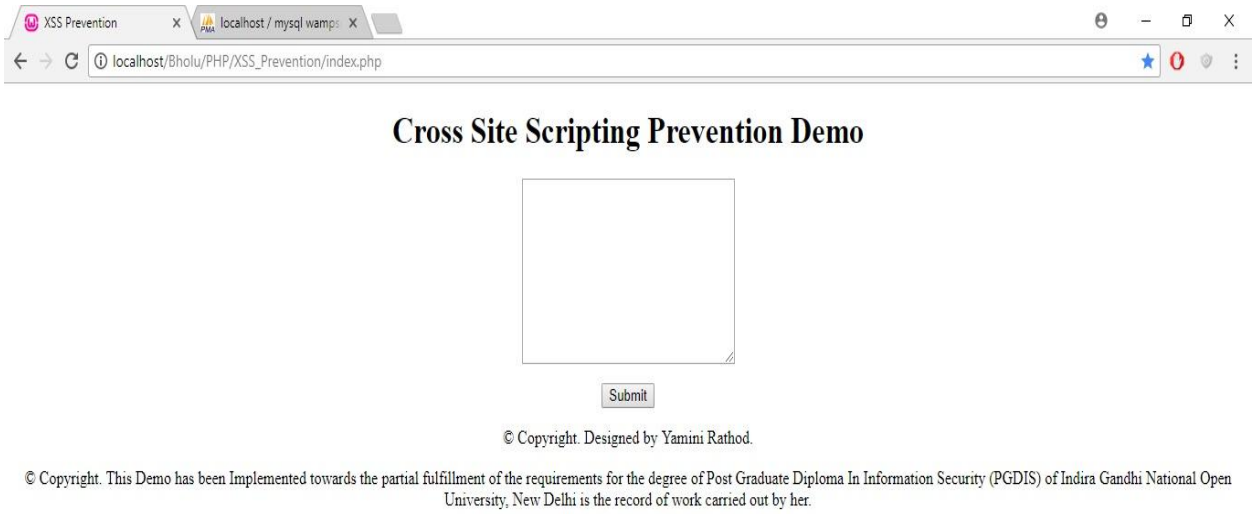
Img 4.7.1.7

4.7.2 Cross site scripting prevention

Cross site scripting has been implemented as demo application.
I have included below code with tag “htmlentities” to prevent the application from XSS.

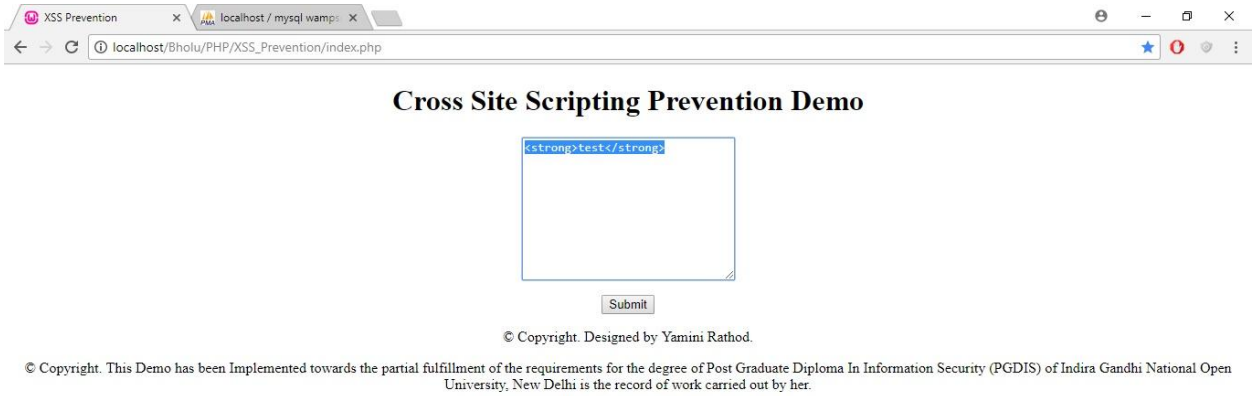
```
if(isset($_POST['input']))
        {
                $data = htmlentities($_POST['input']);
                echo "<p align='center'>$data</p>";

        }
```

Let us observer the demo application. Img 4.7.2.1



Img 4.7.2.1

Try to add malicious html code as content. Img 4.7.2.2



Img 4.7.2.2

Observe the added html strong tag is not getting filtered. Img 4.7.2.3

Img 4.7.2.3

Click on View Inspect element to see the data stored in back ground. Img 4.7.2.4

Img 4.7.2.4

The html code is being converted into &lt and &gt due to htmlentities tag. Img 4.7.2.5

Img 4.7.2.5

XSS can be prevented by adding the prevention tags in an application programming.

4.8 Two level Authentication

4.8.1 Authentication related attacks

The lack of strong authentication method sometimes leads attacks like Man In Middle, Session Hijacking.

4.8.2 Implementation of secured Two level authentication

Two level authentications is basically adding the extra security to the application. It also prevents the attacks related session hijacking.

This feature has been implemented and demonstrated in project.

4.9 Session management

4.9.1 Benefits of session management

It is good habit to implement an application with session timeout feature.

If user is not using an application since certain amount of duration then application should destroy the session and user should not be able to work on it further until he/she logs in again.

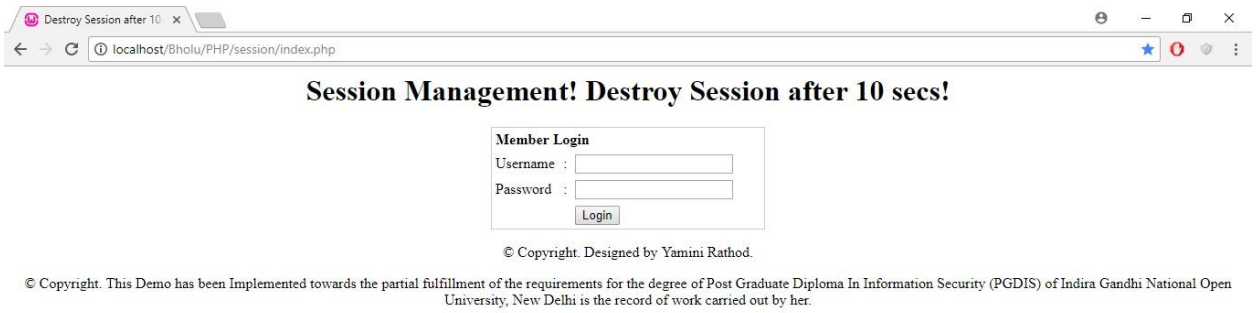
It is key feature to prevent attacks such as session hijacking, Man In Middle attacks, false authentication related crimes etc.

Idle timeout, where the user is away from the system leading to inactivity, e.g. if the page hasn't received an user activity, or the mouse hasn't triggered any on-mouse events. This timeout countdown will reset whenever the user interacts with the web page. Ensuring idle users are logged out quickly significantly reduces system exposure to data breech.

Force-response timeout, triggered after a certain period of time. This ignores the user activity and automatically prompts the user to remain logged in.

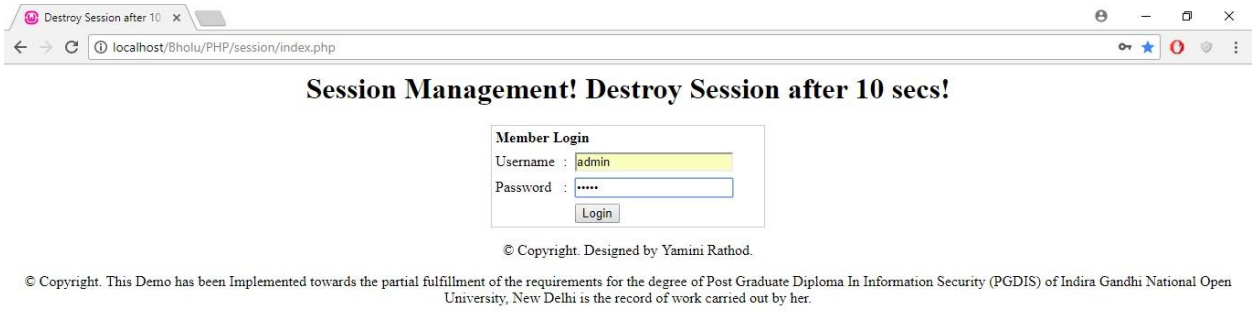
4.9.2 Implementation of session management

The implementation of session management is done using demo application. Img 4.9.2.1



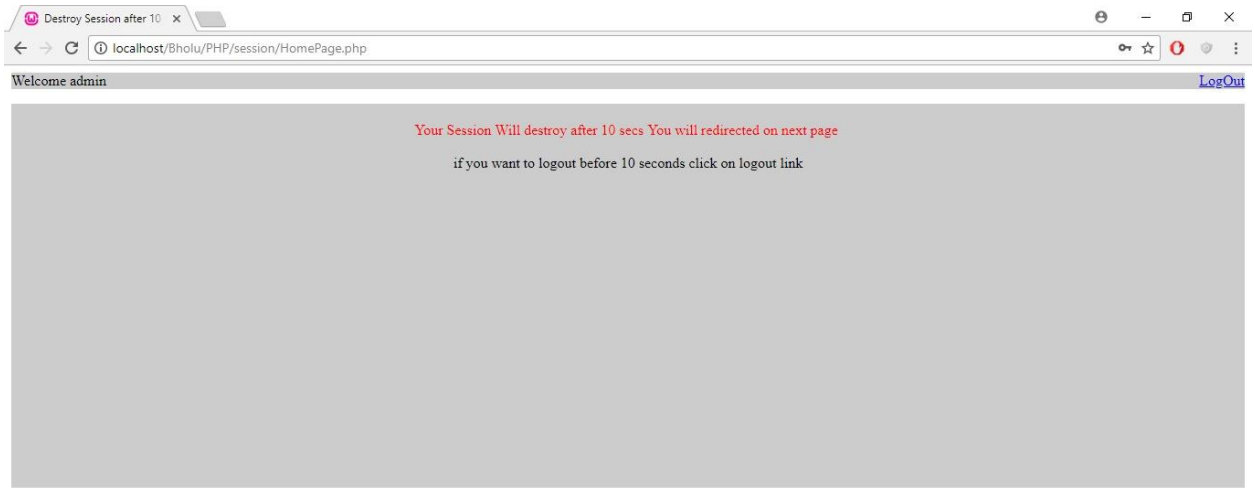
Img 4.9.2.1

Let’s pretend we are doing login in XYZ application as valid authenticated user. Img 4.9.2.2



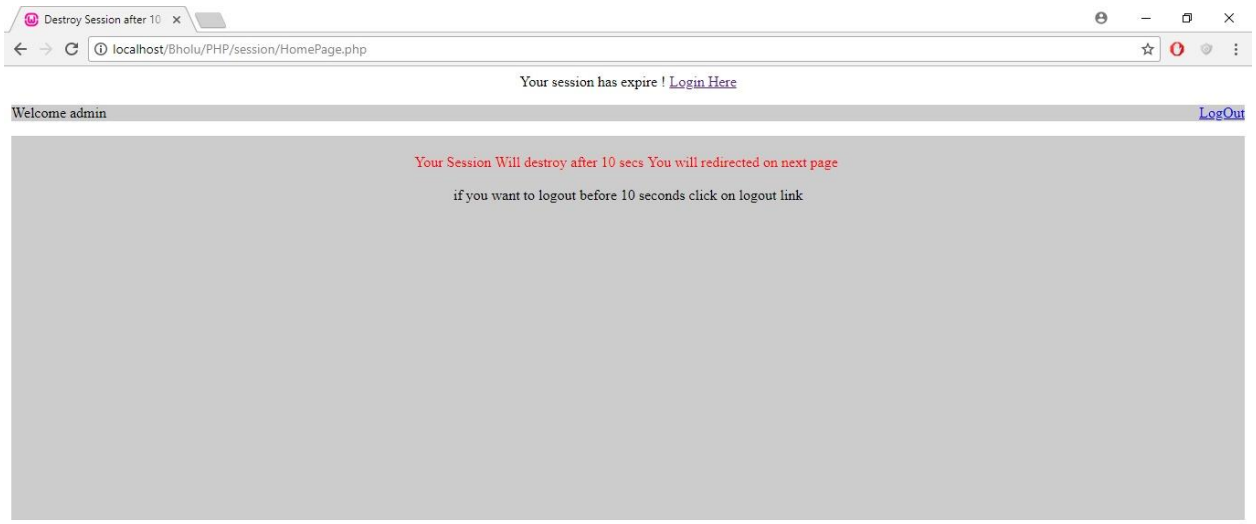
Img 4.9.2.2

Now, we are authenticated and have been redirected to demo page for session timeout. Img 4.9.2.3



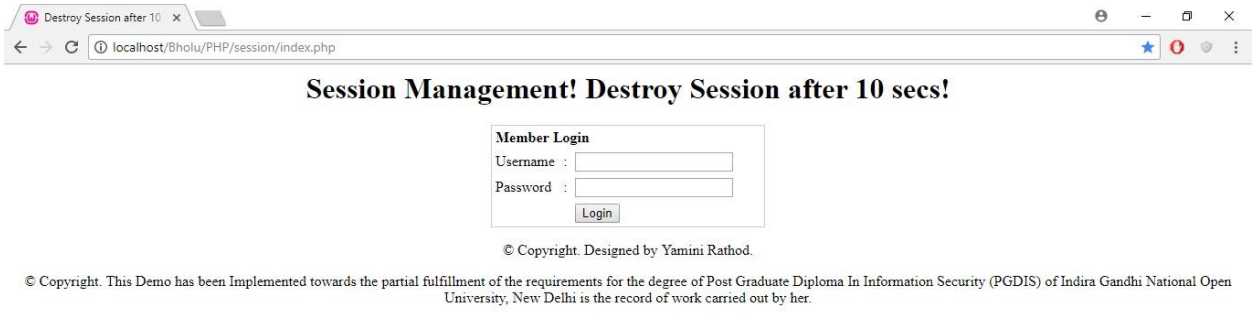
Img 4.9.2.3

Wait for 10seconds as implemented in application demo programming, the session has expired now, hence it will now take us on login page again. Img 4.9.2.4



Img 4.9.2.4

It is redirected now on login screen. Img 4.9.2.5



Img 4.9.2.5

4.10 Authorization

4.10.1 Benefits of authorization

Authorization is the process of giving someone permission to do or have something. In multi-user computer systems, a system administrator defines for the system which users are allowed access to the system and what privileges of use (such as access to which file directories, hours of access, amount of allocated storage space, and so forth). Assuming that someone has logged in to a computer operating system or application, the system or application may want to identify what resources the user can be given during this session. Thus, authorization is sometimes seen as both the preliminary setting up of permissions by a system administrator and the actual checking of the permission values that have been set up when a user is getting access.

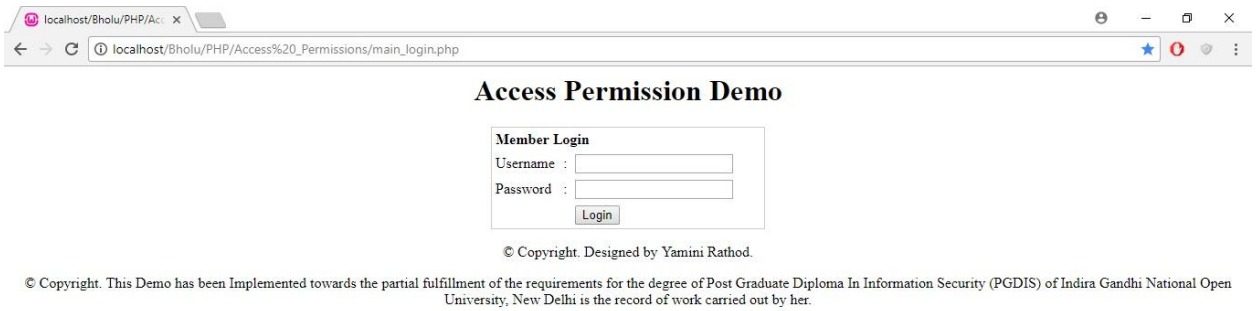
4.10.2 Implementation of authorization policy

The implementation of authorization permission policy is done using demo application.

It is always suggested to maintain access policy in an application; it should be either role based or attribute based.

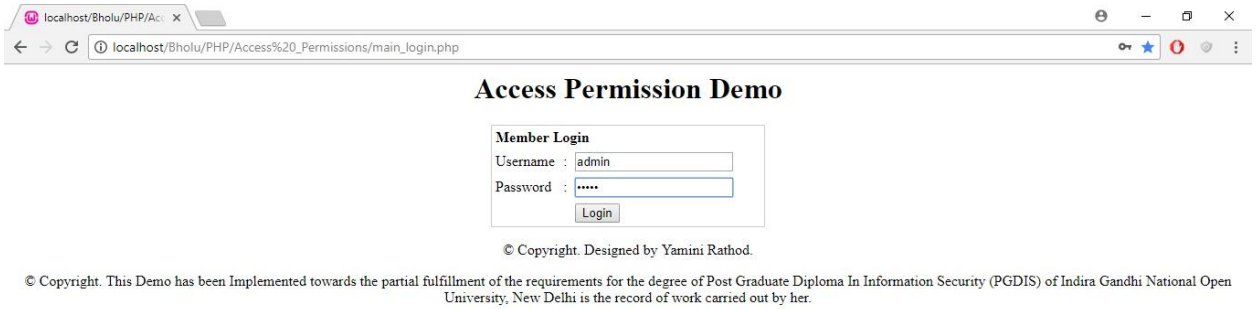
Only authorized user should have control on certain functions in an application, it should be abstract and emphasized on read, write and update access.

Access permission demo sample page. Img 4.10.2.1



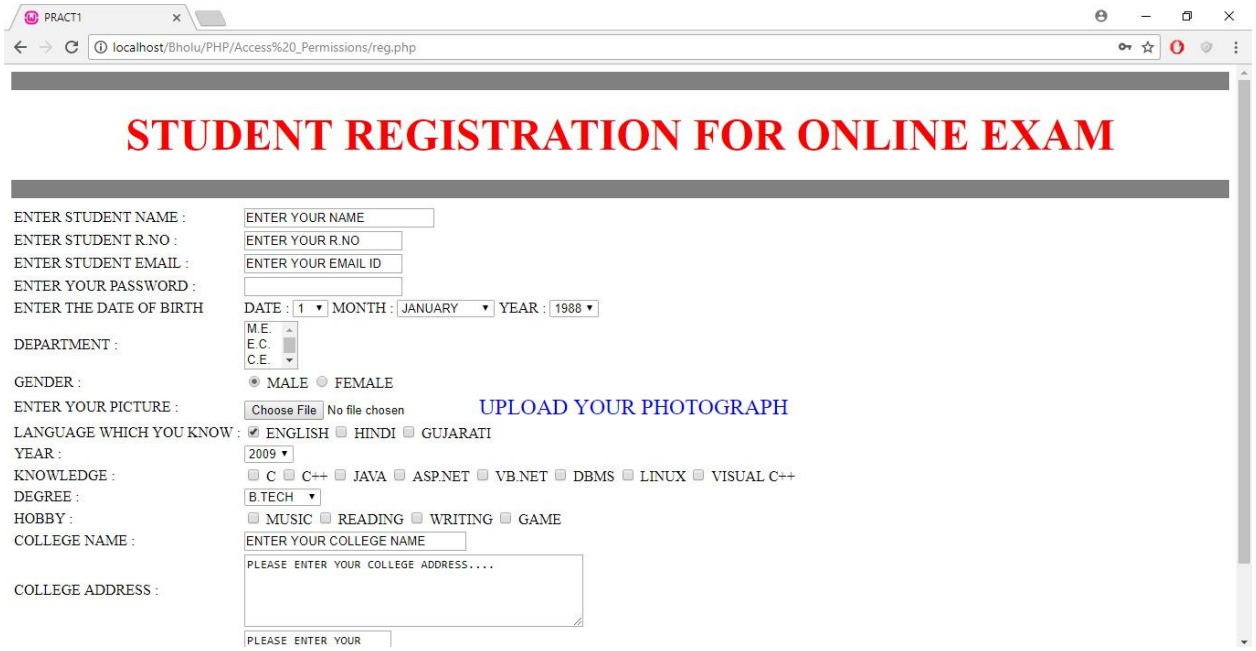
Img 4.10.2.1

Login as admin user. Img 4.10.2.2



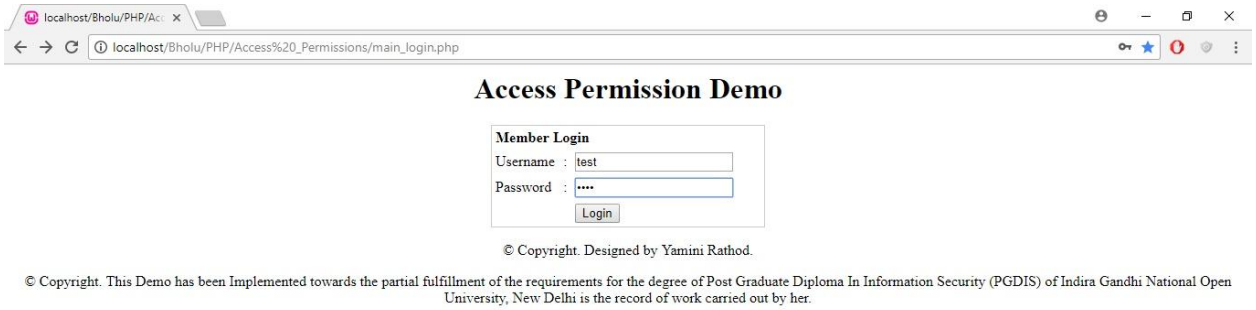
Img 4.10.2.2

Application validates the credentials on back end and redirects user on destination page. Img 4.10.2.3



Img 4.10.2.3

Now, try to login as test user who is restricted to login in an application. Img 4.10.2.4



Img 4.10.2.4

Observe message that denotes the permission policy. Img 4.10.2.5

Img 4.10.2.5

Note : Permission is not limited to this, it is divided into two parts.

1. Authorization
2. Access Control

Authorization signifies whether user is authorized to have access on an application or not, access control is second level of access policy post authorization. It allows user to have access on read, write, update components. We have restrict access to have only read only access, some user should not be able to modify critical contents in an application.

It is implemented in my application IMS, it has been demonstrated in detail in upcoming chapters.

4.11 Security enhancement in Inventory Management

With reference to case study of above illustrated security issues, I have implemented below security features in real time application IMS.

1. Implementation of Encryption
2. Implementation of Validation
3. Implementation of Session Management, Session Lock-Out, Session Time-Out, Session Redirect
4. Implementation of SQL Injection Prevention
5. Implementation of Session Hijacking Prevention
6. Implementation of Cross site Scripting Prevention
7. Implementation of Cross site Request Forgery Prevention
8. Implementation of Two Level Authentication
9. Implementation of Strong Password Validation
10. Implementation of Secured Digest Authentication
11. Implementation of Cookie Management, Session Hijacking Prevention
12. Implementation of Access Permission Authorization
13. Monitoring using WireShark
14. Monitoring using Fiddler

These features have been described in detail in chapter 5.

4.12 Case study on Top Cyber Crimes in the World

1. WannaCry virus hits the NHS, 2017

The most widespread cyber attack ever, hackers managed to gain access to the NHS' computer system in mid-2017, causes chaos among the UK's medical system. The same hacking tools were used to attack world-wide freight company FedEx and infected computers in 150 countries. Ransomware affectionately named "WannaCry" was delivered via email in the form of an attachment. Once a user clicked on the attachment, the virus was spread through their computer, locking up all of their files and demanding money before they could be accessed again. As many as 300,000 computers were infected with the virus. It was only stopped when a 22-year-old security researcher from Devon managed to find the kill switch, after the NHS had been down for a number of days.

2. Hackers steal £650 million from global banks, 2015

For a period of two years, ending in early 2015, a grow up of Russian-based hackers managed to gain access to secure information from more than 100 institutions around the world. The cyber criminals used malware to infiltrate banks' computer systems and gather personal data, They were then able to impersonate online bank staff to authorize fraudulent transfers, and even order ATM machines to dispense cash without a bank card. It was estimated that around £650 million was stolen from the financial institutions in total.

3. JP and Morgan Chase & Co target of giant hacking conglomerate, 2015

Late in 2015, three men were charged with stealing date from millions of people around the world, as part of a hacking conglomerate that spanned the best part of a decade. The trio themselves allegedly described the incident as “one of the largest thefts of financial-related data in history”. Thought to have been operating out of Israel, the trio targeted major corporations, including major US bank JP Morgan Chase & Co, stealing personal data and then selling it on to a large network of accomplices. The group stole information from more than 83 million customers from JP Morgan alone, and are thought to have made hundreds of millions of dollars in illegal profits. Along with personal data, the hacking group also stole information related to company performance and news, which allowed them to manipulate stock prices and make enormous financial gain. Using more than 200 fake identity documents, they were able to facilitate large scale payment processing for criminals, an illegal bitcoin exchange, and the laundering of money through approximately 75 shell companies and accounts globally.

4. One billion user accounts stolen from Yahoo, 2013

In one of the largest cases of data theft in history, Yahoo had information from more than one billion user accounts stolen in 2013. Personal information including names, phone numbers, passwords and email addresses were taken from the internet giant. Yahoo claimed at the time that no bank details were taken. Releasing information of the breach in 2016, it was the second time Yahoo had been targeted by hackers, after the accounts of nearly 500 million users were accessed in 2014.

Chapter 5

Observation

5.1 Home Page

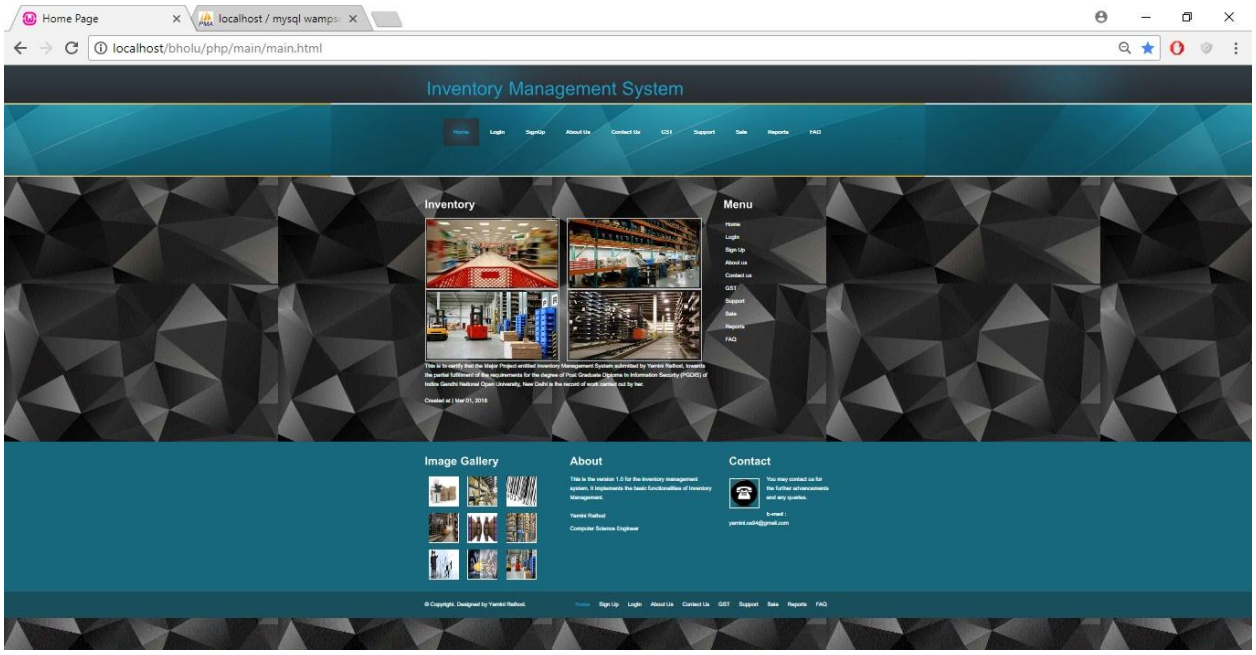
Supermarket Inventory System is to facilitate our customers to track their products as and when they are transported from the vendor to the warehouse and from the warehouse to the retail location to the customers.

It is necessary to keep our resources safe and protected. In order to implement security in application it would be done by implementing encryption, keeping secure session base password, implementing two level authentications, observing system logs and security faults, analyzing network flow using wireshark, implementing wireshark, preventing the application validation from un-necessary inputs, session management, session hijacking, hacking, cross site scripting and implementing code to prevent from SQL injection and many more.

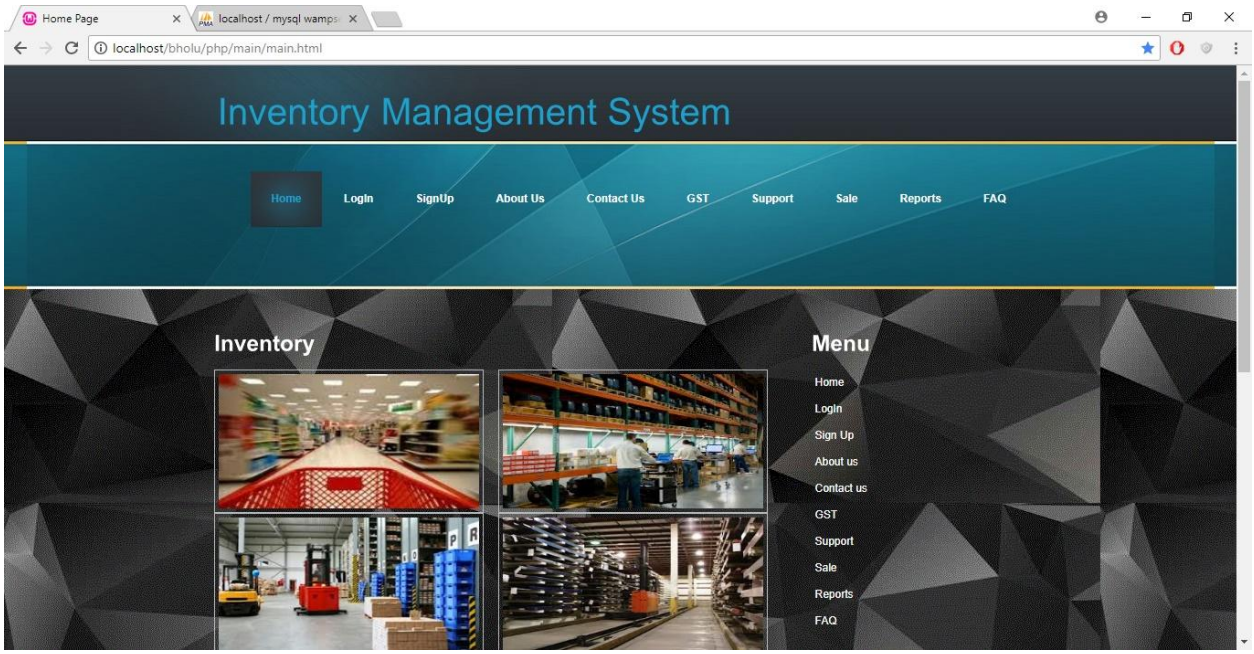
Sample Home page emphasizes on the basic details about IMS organization. Authorized and registered customers would be able to login in IMS system to track and place their orders. They can look at GST module, can contact our customer support team for any queries, review frequently asked questions FAQ, latest sales available and many more. New customers can visit our portal to know about IMS organization and can contact us.

Demonstration of the full layout of Inventory Management System Home page.

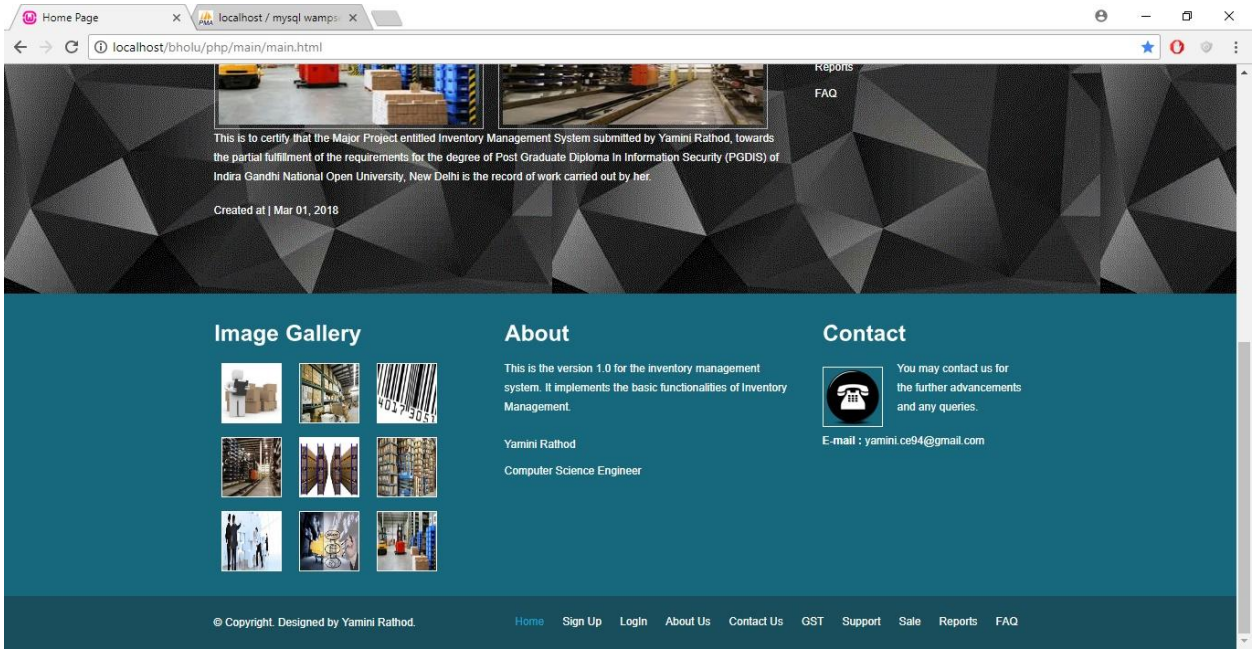
Below screen captures were taken with small font size to cover whole screen, please review below images respectively. Img 5.1.1, 5.1.2, 5.1.3



Img 5.1.1



Img 5.1.2



Img 5.1.3

5.2 Validation

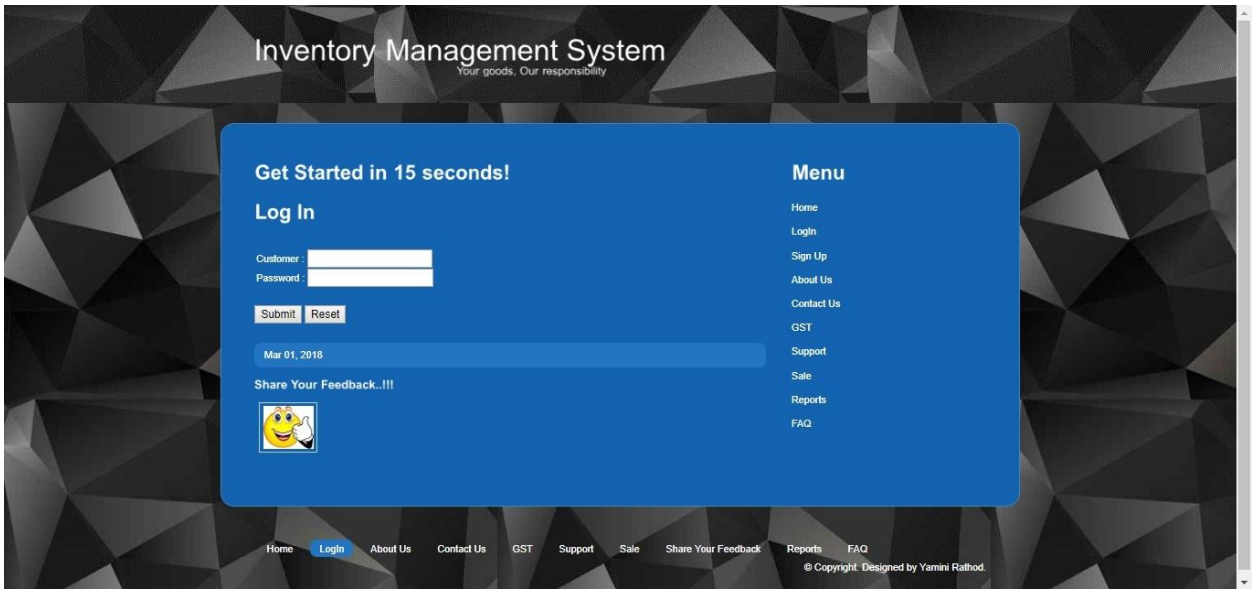
Validation is required to prevent web form abuse by malicious users.

Improper validation of form data is one of the main causes of security vulnerabilities. It exposes your website to attacks such as header injections, cross-site scripting, and SQL injections, MIM Man In Middle attack, DOS Denial of Service Attack, Session Hijacking, Cookie Hijacking, and many more.

I have implemented validation on each and every component in IMS application.

Validation is very essential and critical in login page.

Demonstrating validation on customer login screen. Please review screenshot Img 5.2.1.



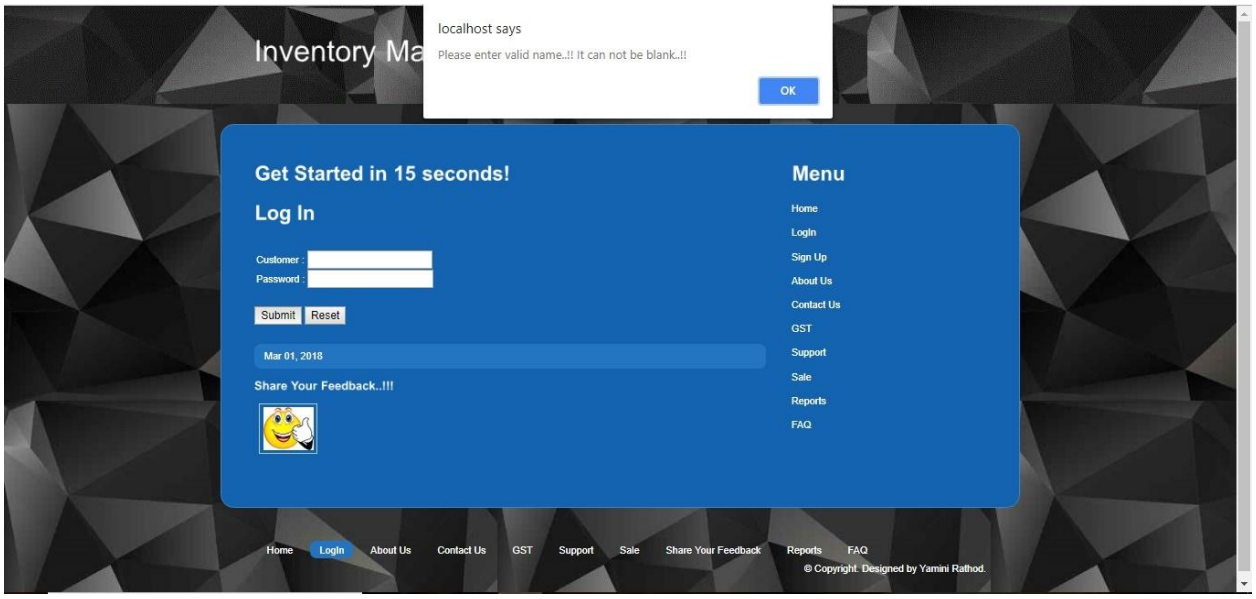
Img 5.2.1

System will not allow invalid username and password. This is just stage 1 validation which will be present in all applications. Higher level of validation such as character check, keyword check, character limits, digit numbers, upper case lower case, special characters check, rematch, formatting check etc has been implemented in upcoming pages.

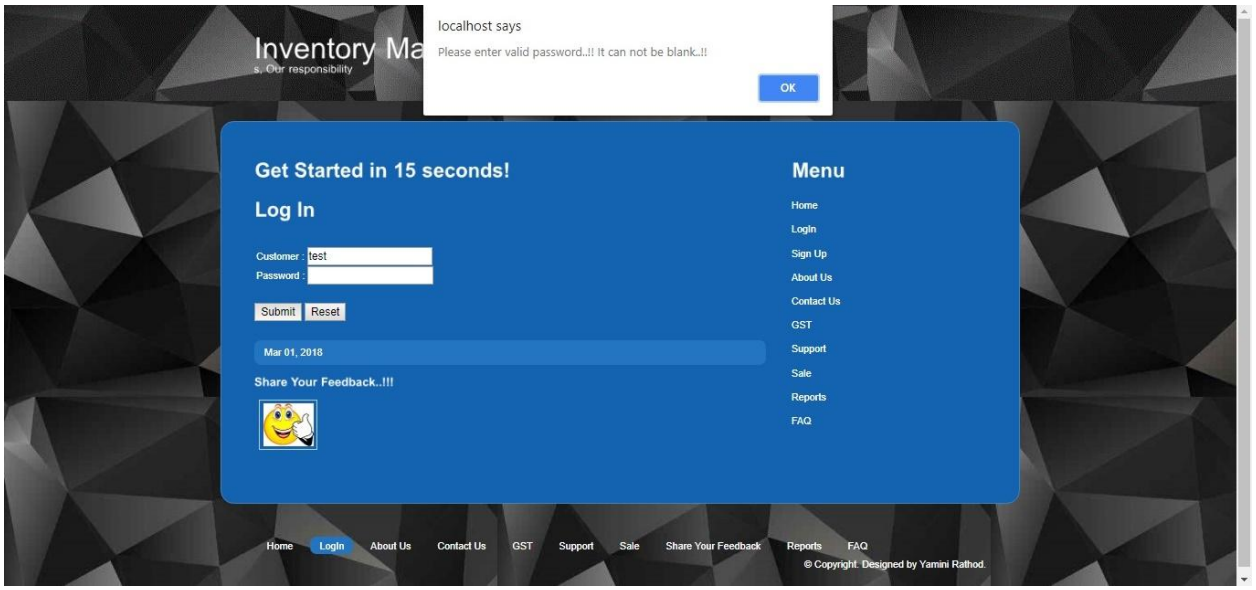
Implemented validation using Javascript and Jquery functions in PHP. System will prompt for empty data inputs. Please refer below screenshots. Img 5.2.2, 5.2.3

Implemented two level authentications to provide more security feature to an application. Most secured websites such as Net-banking websites, Amazon shopping sites etc has feature of two level authentications to prevent from malicious attack. Here, I have implemented the security question as TLA,

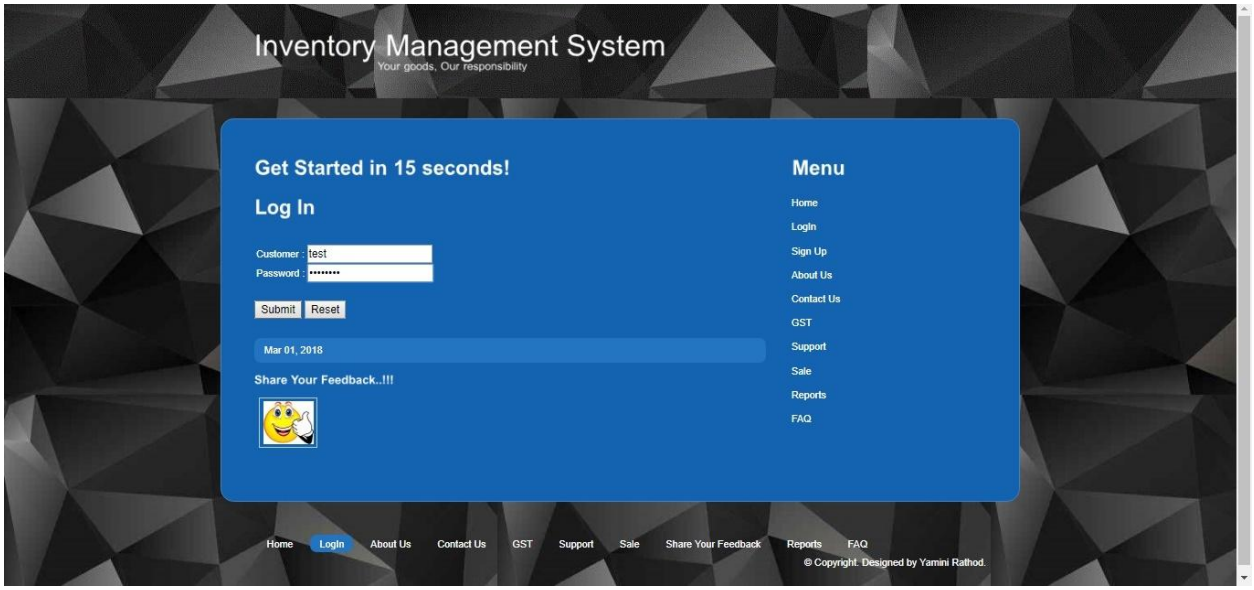
once website goes live, we can implement the feature of RSA token or email token. Token will be sent to registered email id. User will be able to get into application by accessing token from his/her registered email account. Img 5.2.4, 5.2.5



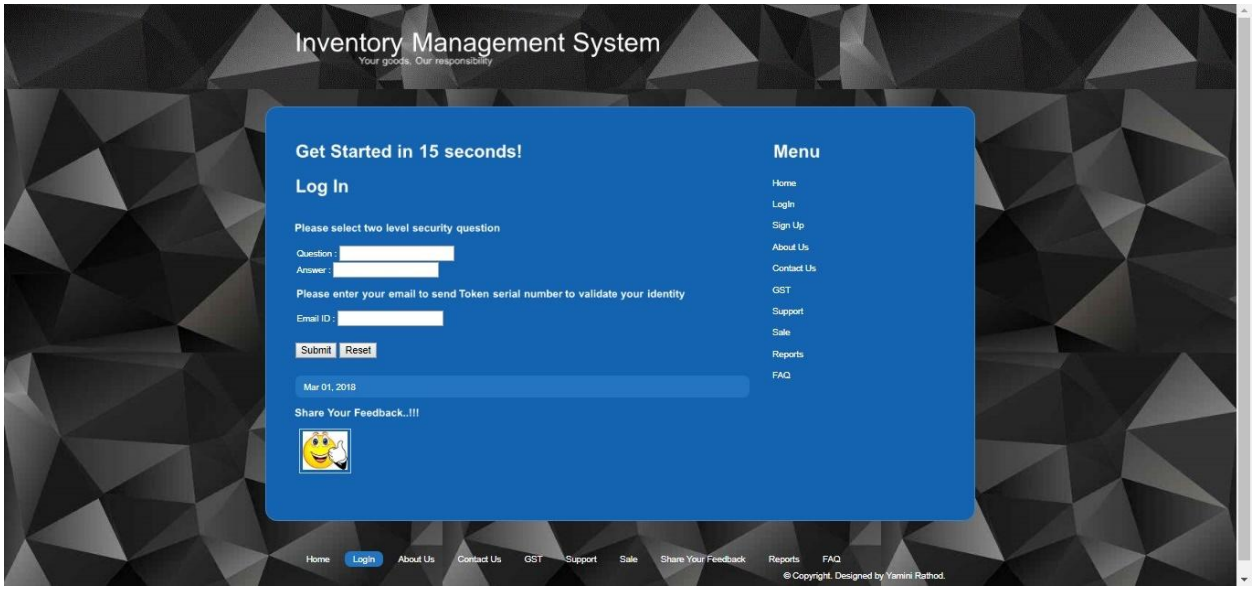
Img 5.2.2



Img 5.2.3

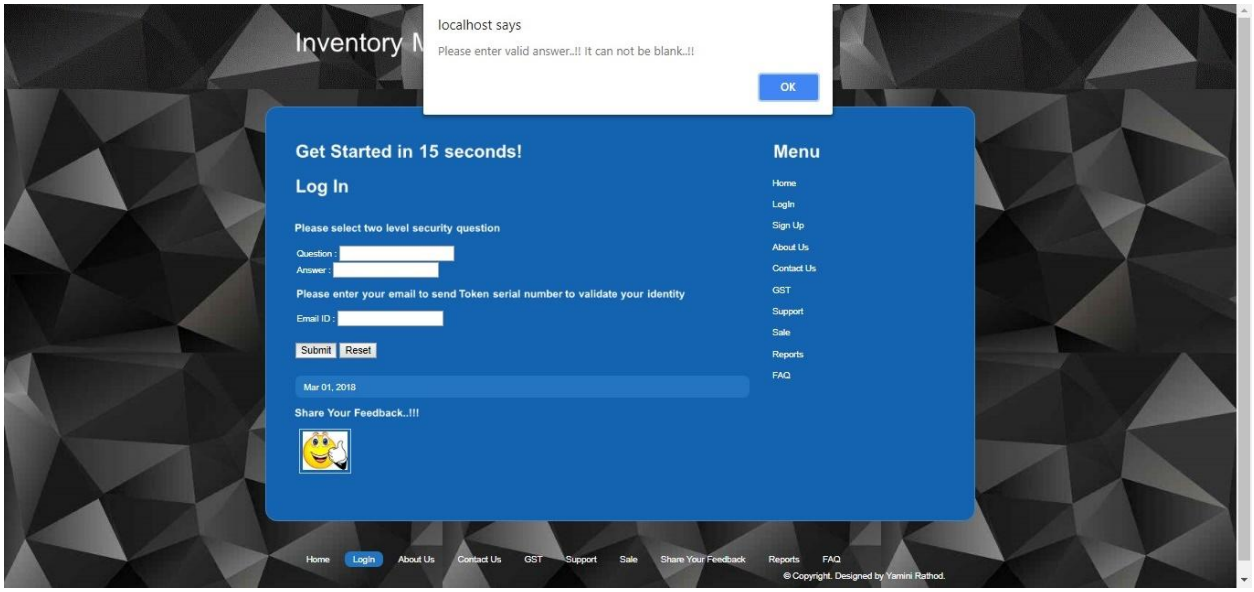


Img 5.2.4

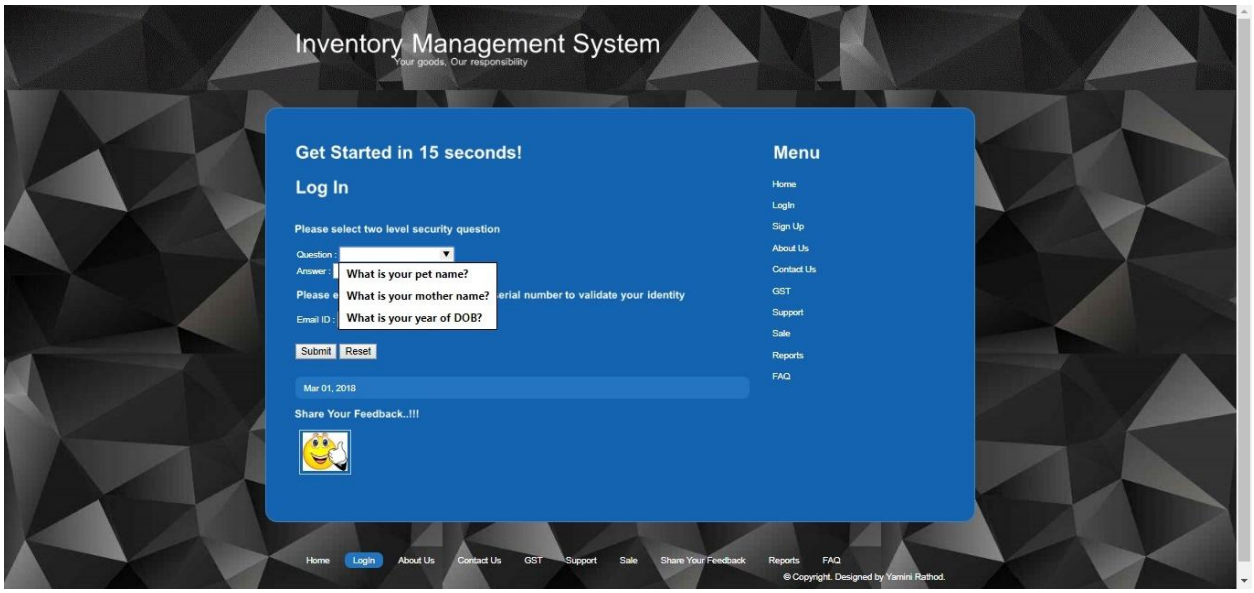


Img 5.2.5

IMS application has been implemented with three questions as two level security authentications. Organization may input many more questions as per the complexity of application and nature of clients. Please review below screenshot Img 5.2.6, 5.2.7.

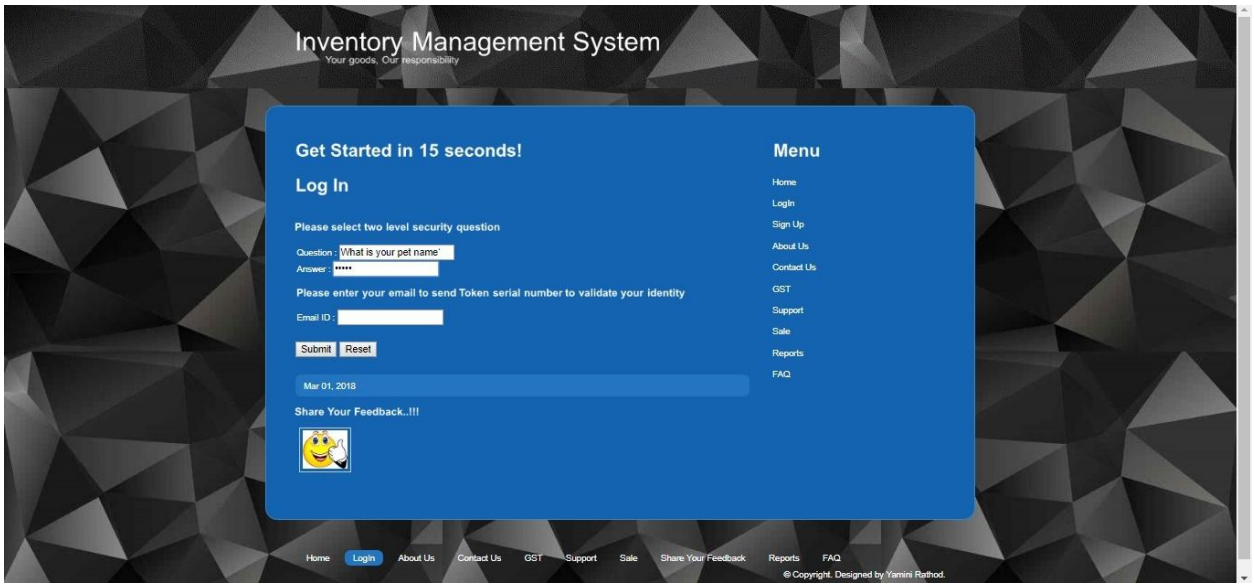


Img 5.2.6

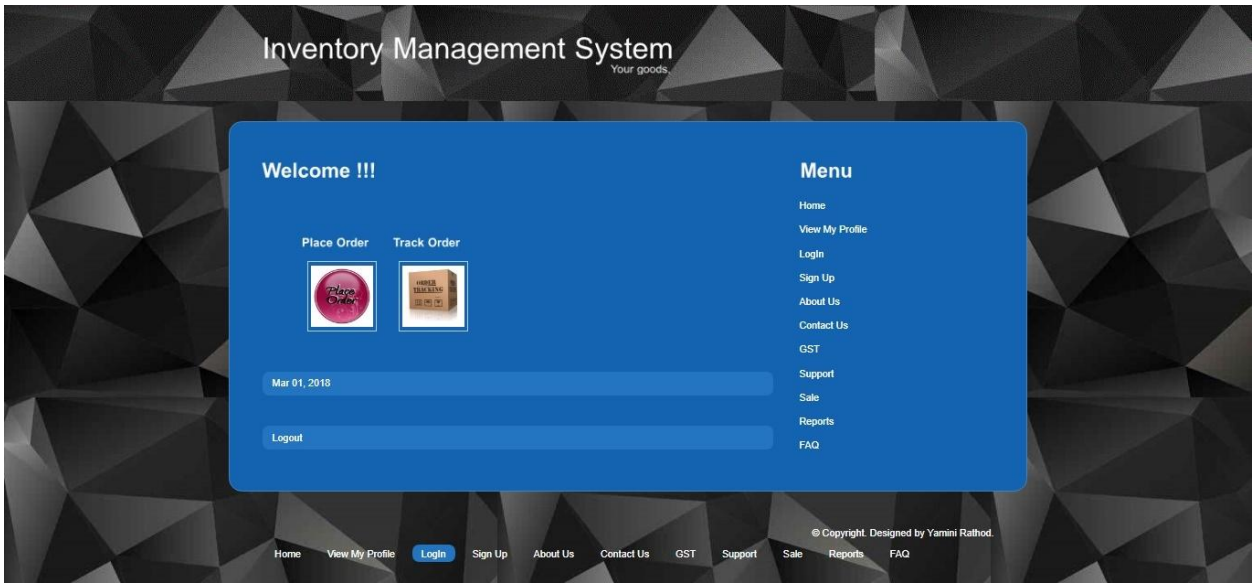


Img 5.2.7

Post two level authentications, system will redirect the customer on the place order and track order screen. Customer can place the order by selecting the type of order, approximate date of withdrawal, date of order, and estimated amount. At same time they can see the status of their order, whether it is valid or it is in consideration. Placed order will get reviewed by manager and admin and then will update the customer on same basis. Please review screen Img 5.2.8, 5.2.9



Img 5.2.8

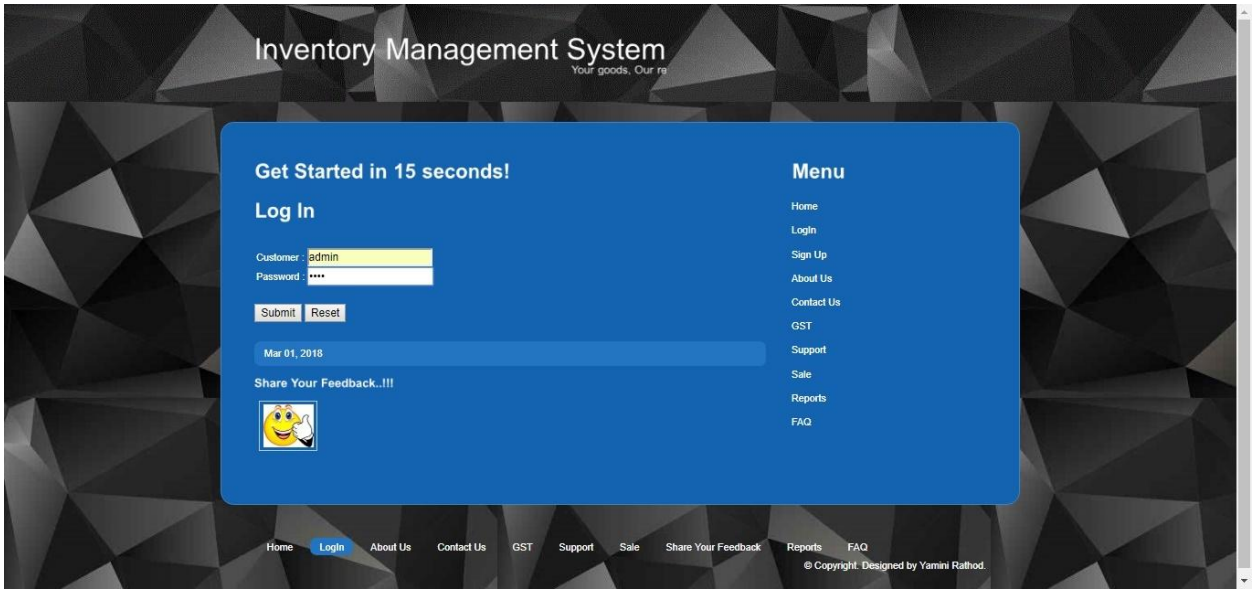


Img 5.2.9

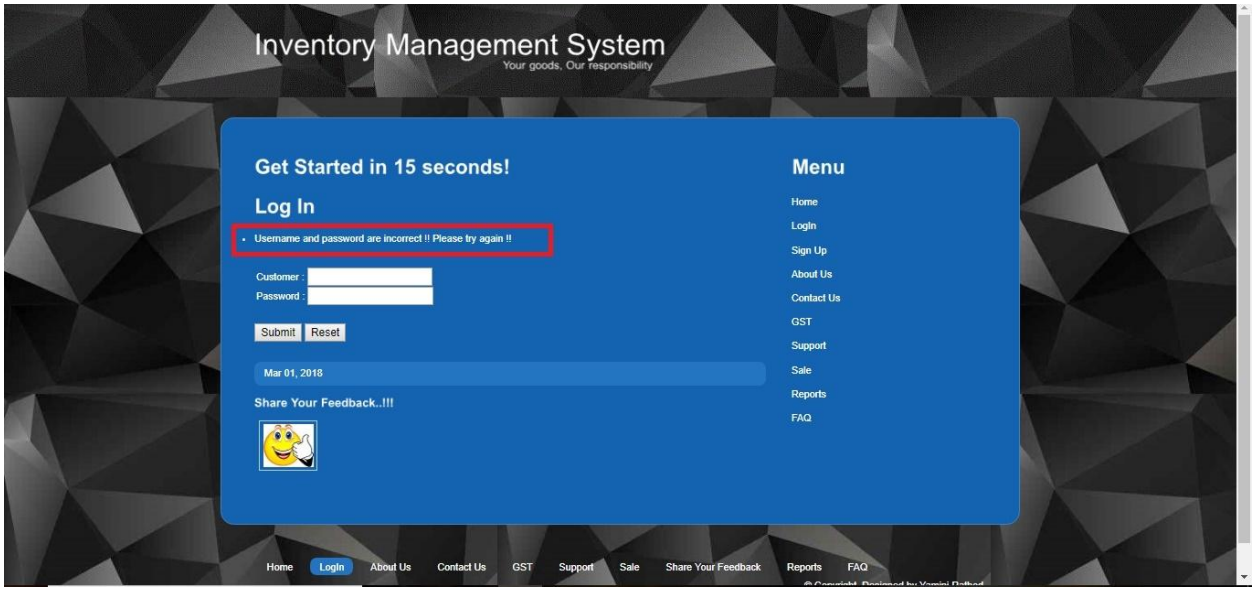
5.3 Authentication

The main purpose of this feature is to validate the user's right to access the system and information, and protect against identity theft and fraud. Please refer screenshot Img 5.3.1, 5.3.2

I have implemented encryption algorithm MD5 in application coding, which is forming digest authentication. Application redirects the password in encrypted format and stores it in MySQL database server.



Img 5.3.1

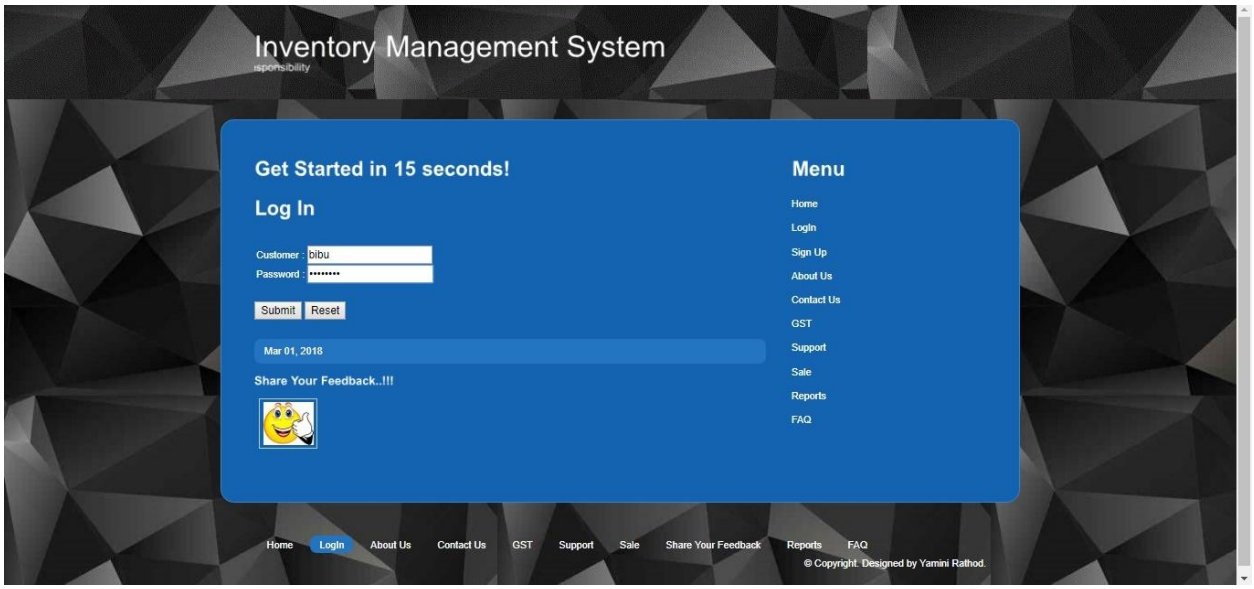


Img 5.3.2

5.4 Authorization

Authorization is the process of giving someone permission to do or have something. In multi-user computer systems, a system administrator defines for the system which users are allowed access to the system and what privileges of use (such as access to which file directories, hours of access, amount of allocated storage space, and so forth). Assuming that someone has logged in to a computer operating system or application, the system or application may want to identify what resources the user can be given during this session. Thus, authorization is sometimes seen as both the preliminary setting up of permissions by a system administrator and the actual checking of the permission values that have been set up when a user is getting access.

In IMS application, Admin has privileges to restrict access to customers if they are inactive in application since past 6months. Customer who are marked inactive in system won't be able to access the IMS system. They will get prompt to contact admin as illustrated in below screenshots. Img 5.4.1, 5.4.2



Img 5.4.1

Img 5.4.2

5.5 Session Timeout

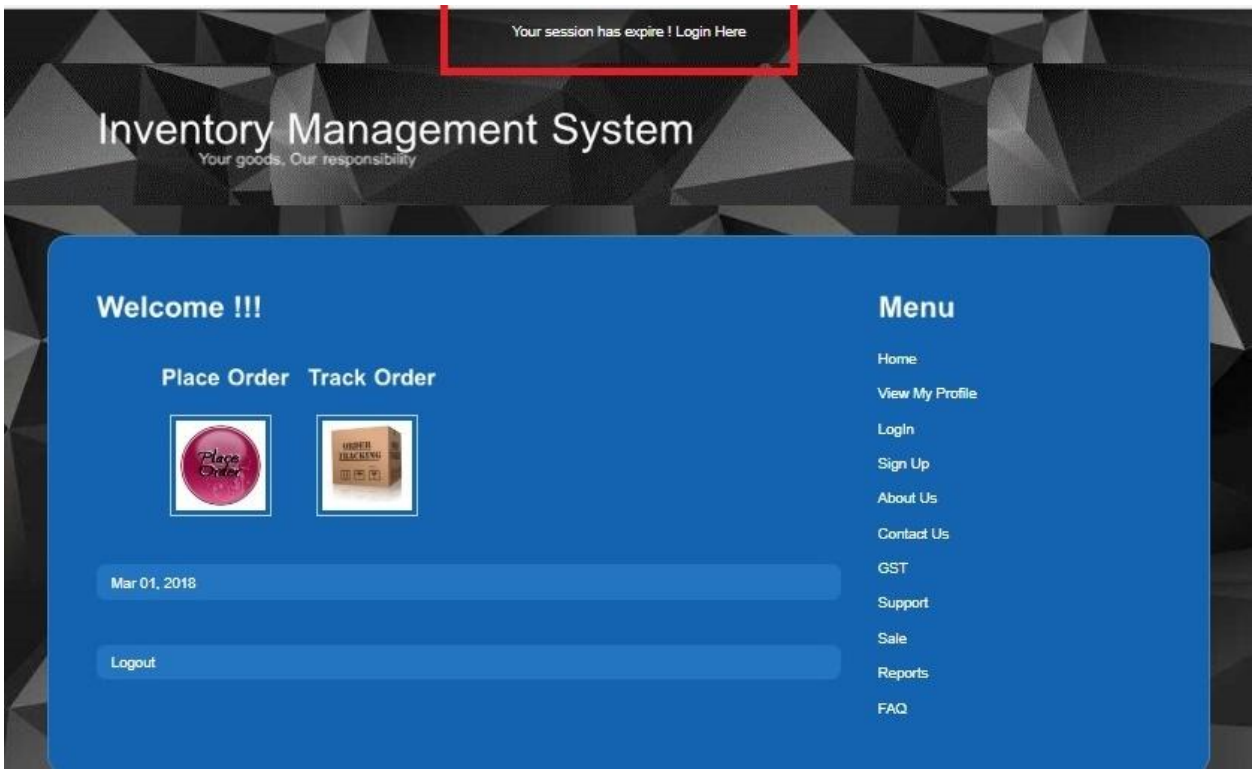
It is good habit to implement an application with session timeout feature.

If user is not using an application since certain amount of duration then application should destroy the session and user should not be able to work on it further until he/she logs in again.

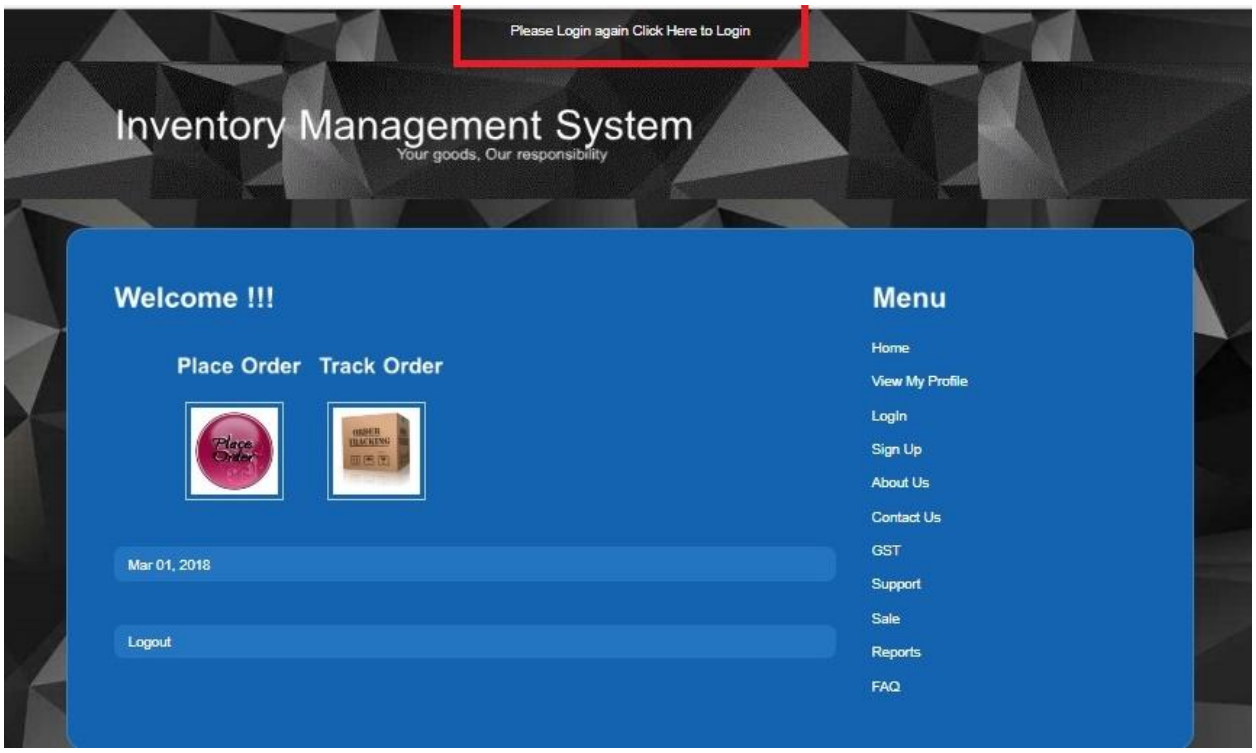
It is key feature to prevent attacks such as session hijacking, Man In Middle attacks, false authentication related crimes etc.

Idle timeout, where the user is away from the system leading to inactivity, e.g. if the page hasn't received an user activity, or the mouse hasn't triggered any on-mouse events. This timeout countdown will reset whenever the user interacts with the web page. Ensuring idle users are logged out quickly significantly reduces system exposure to data breech.

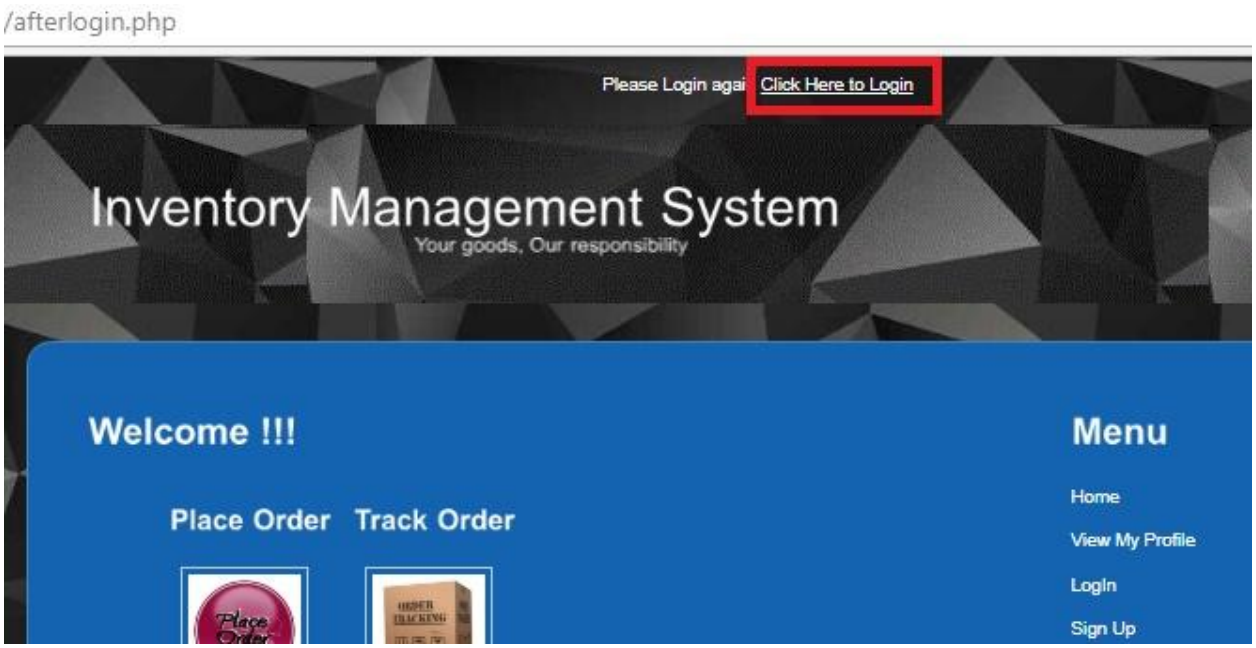
Demonstrating Session Timeout feature in IMS application. Please review screenshots Img 5.5.1, 5.5.2, 5.5.3



Img 5.5.1

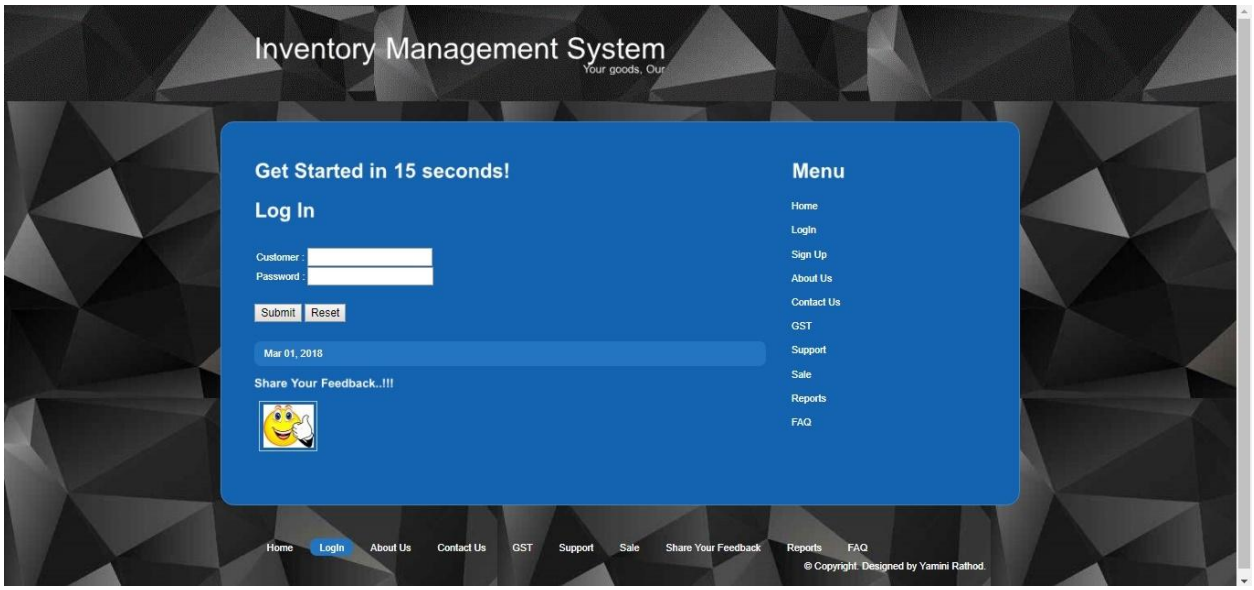


Img 5.5.2



Img 5.5.3

System will redirect user on Login page and will create new session ID, Cookie Id. It is good method to protect application from session hijacking and Man In Middle attacks. Img 5.5.4



Img 5.5.4

5.6 SQL Injection Prevention

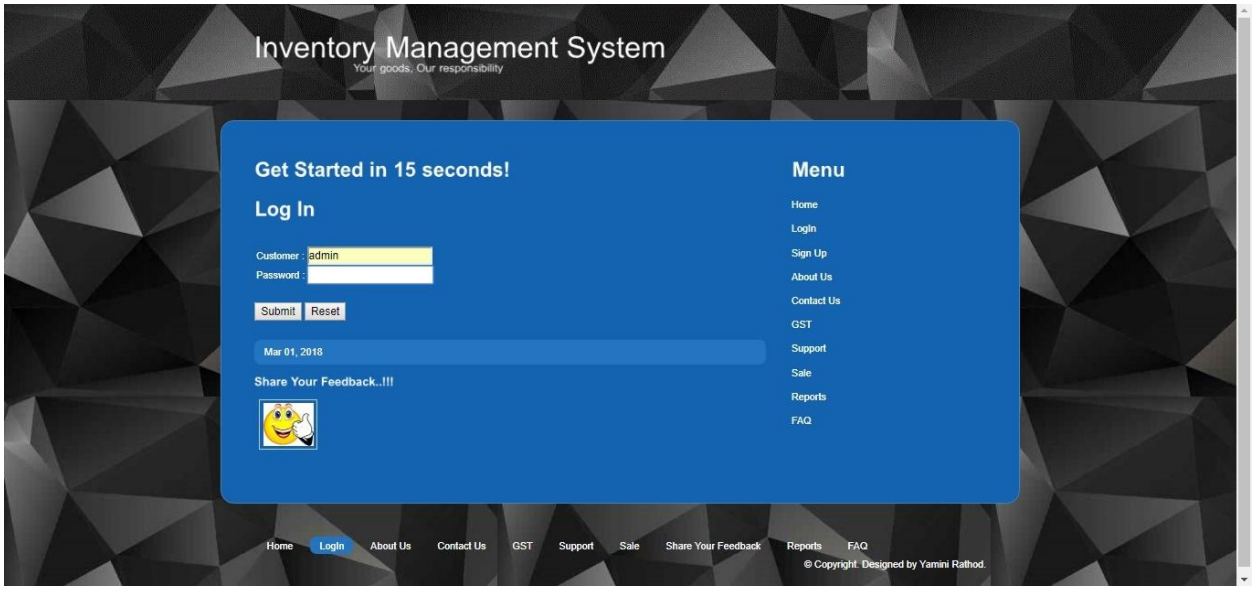
SQL Injection refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.

SQL Injection attacks and it's prevention techniques has been illustrated in detail in above chapters, hence on basis of that I have implemented SQL Injection Prevention techniques in IMS application.

SQL Injection Prevention can be implemented by including,

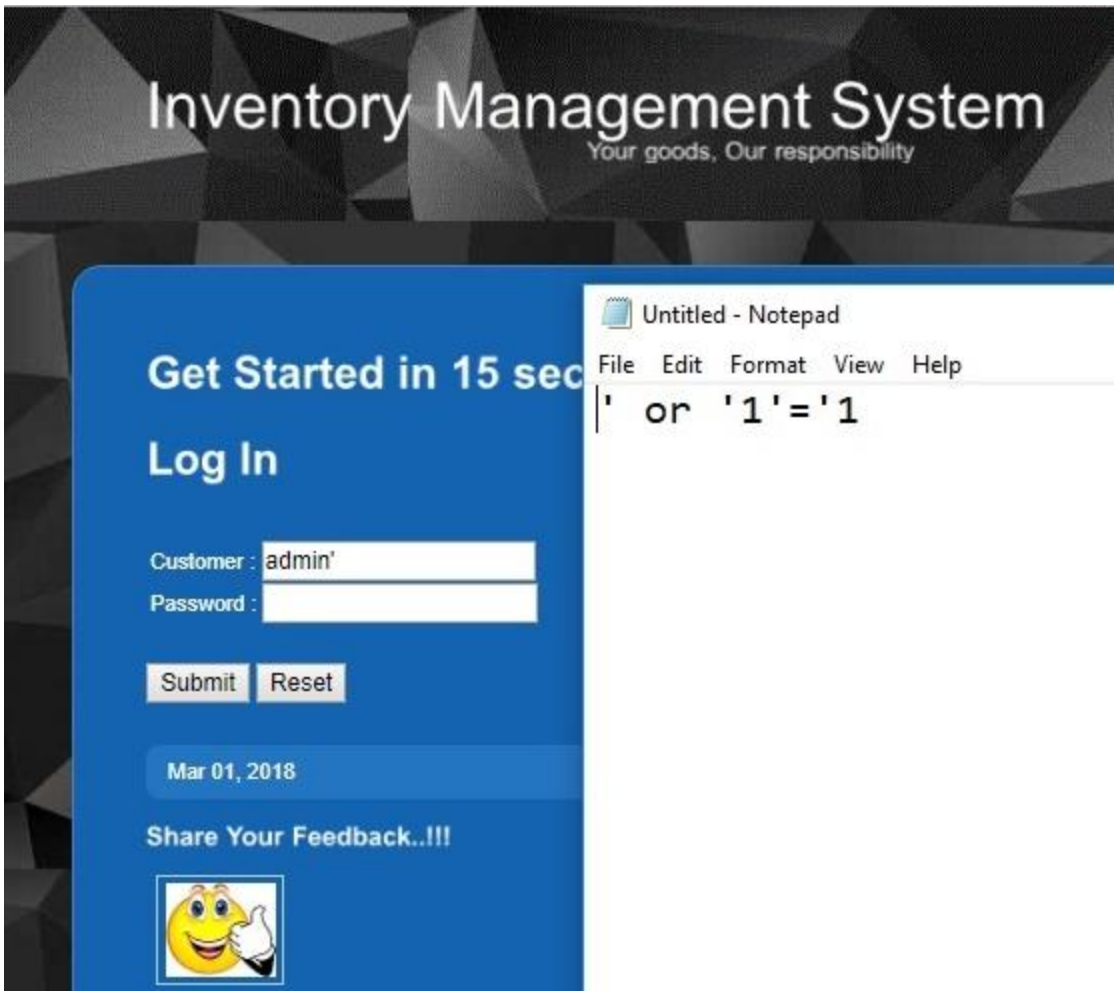
1. Encryption.
2. Implement SQL Injection Prevention code.
3. Implement the validation.
4. Implement SSL.
5. Use parameterized queries when dealing with SQL queries that contains user input.
6. A parameterized query allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
7. Do not display SQL errors to the user. If you need to show the user an error, use a generic error message that does not give away sensitive information.

Here, I am able to prevent the SQL Injection by including Encryption, Validation and changing query structure as parameterized query. Img 5.6.1



Img 5.6.1

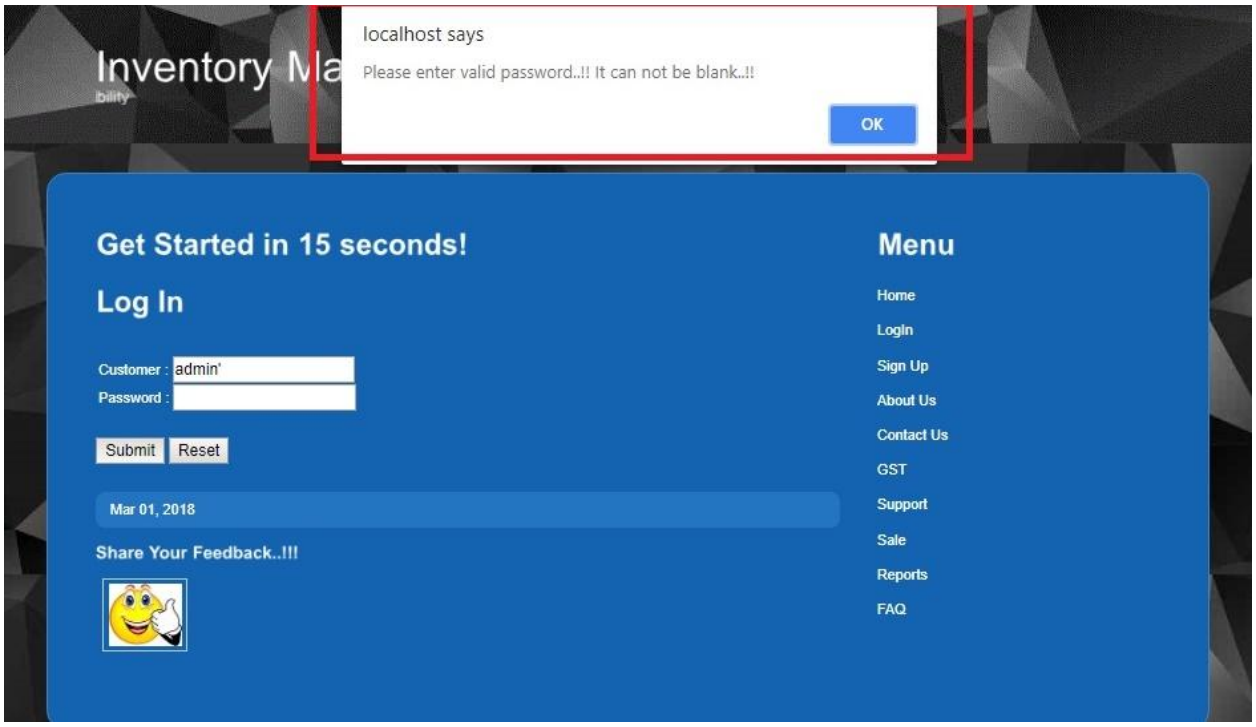
Now, let's try to enter malicious input such way that query gets executed with condition 1=1 which will be always true. Img 5.6.2



Img 5.6.2

Entered customer name as admin', basically ' is consider as Sanitization. In SQL Query it is a parameter. In case of inputs such as admin', system throws an error message with SQL data in it. It is dangerous way to display the error messages. Error message should not display errors in detail. Attackers do hack application using invalid entries such as data with special characters which SQL database cannot recognize.

Implementation of Validation with only limited error message will help to avoid Sanitization. Please refer screenshot Img 5.6.3

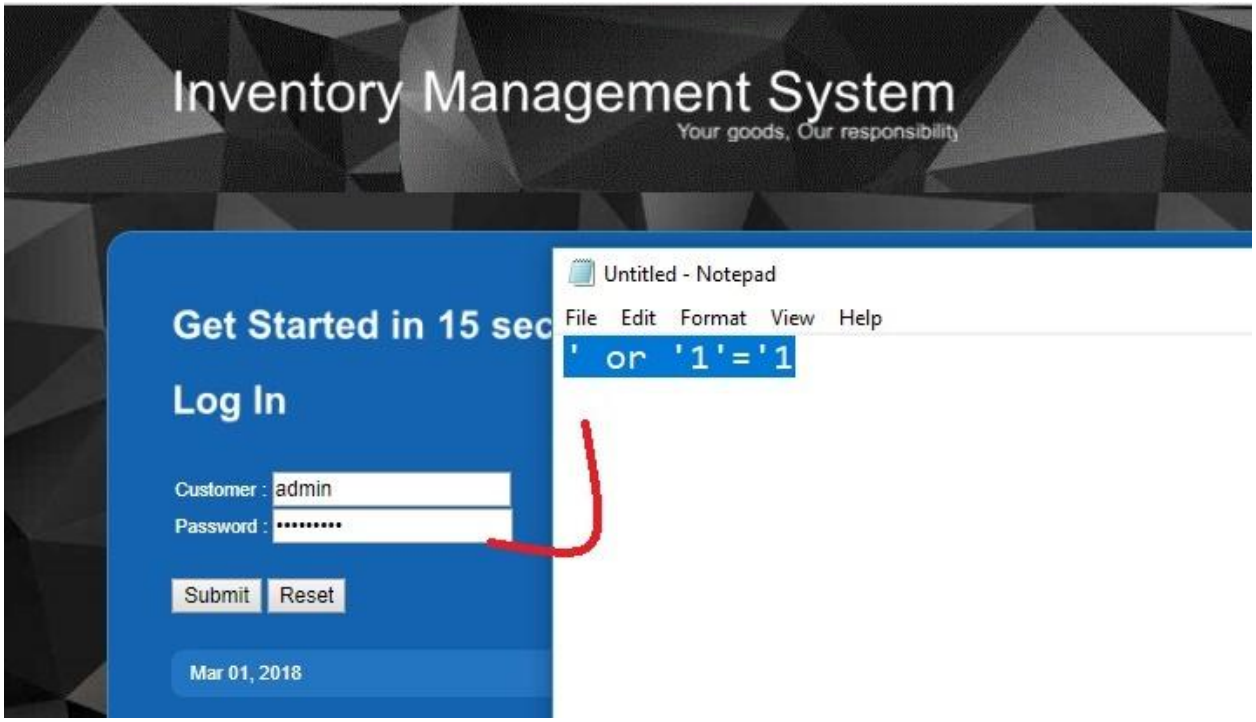


Img 5.6.3

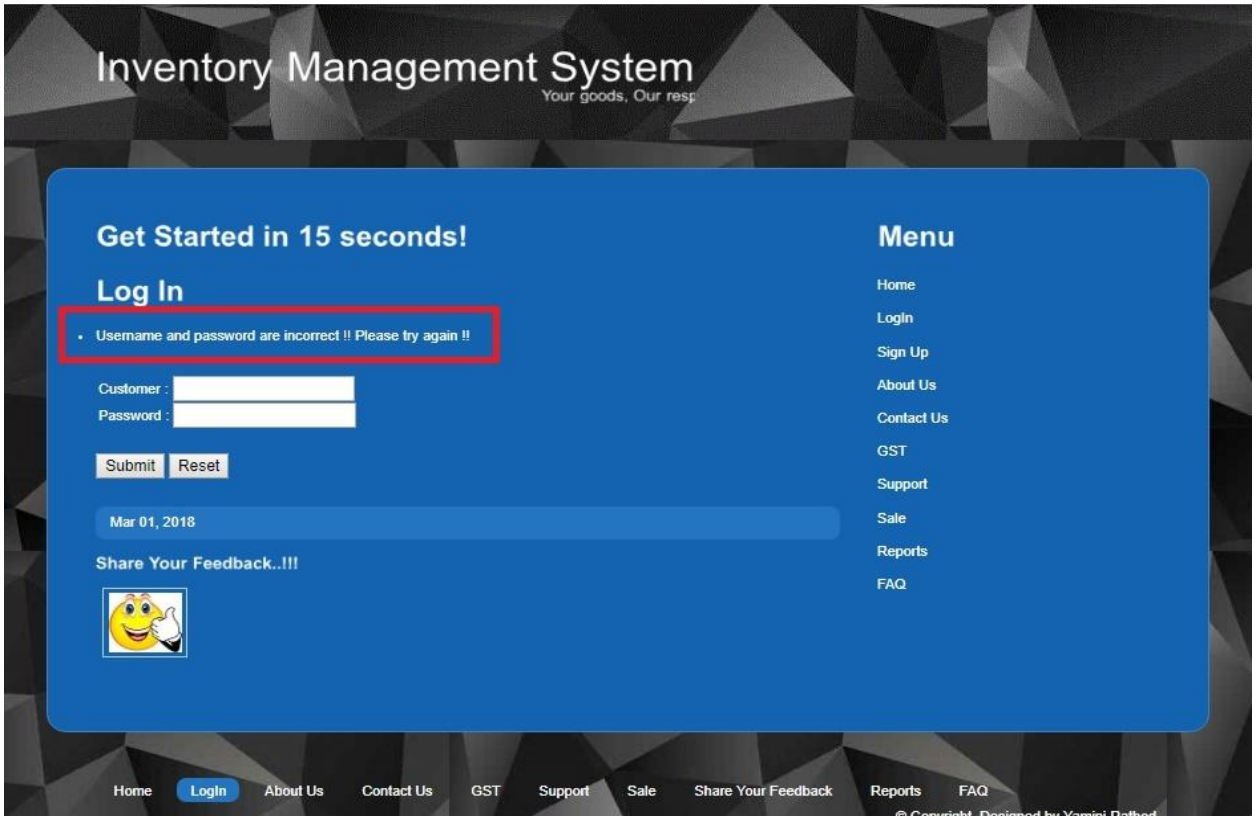
Let's try to enter one more malicious input which will execute SQL query in always true condition.

Entered password as : ' or '1'='1 Img 5.6.4

Here, implementation of parameterized query, Encryption, and Validation is preventing SQL Injection on all possible forms. Please observe outputs on Img 5.6.5



Img 5.6.4



Img 5.6.5

5.7 Cross Site Scripting

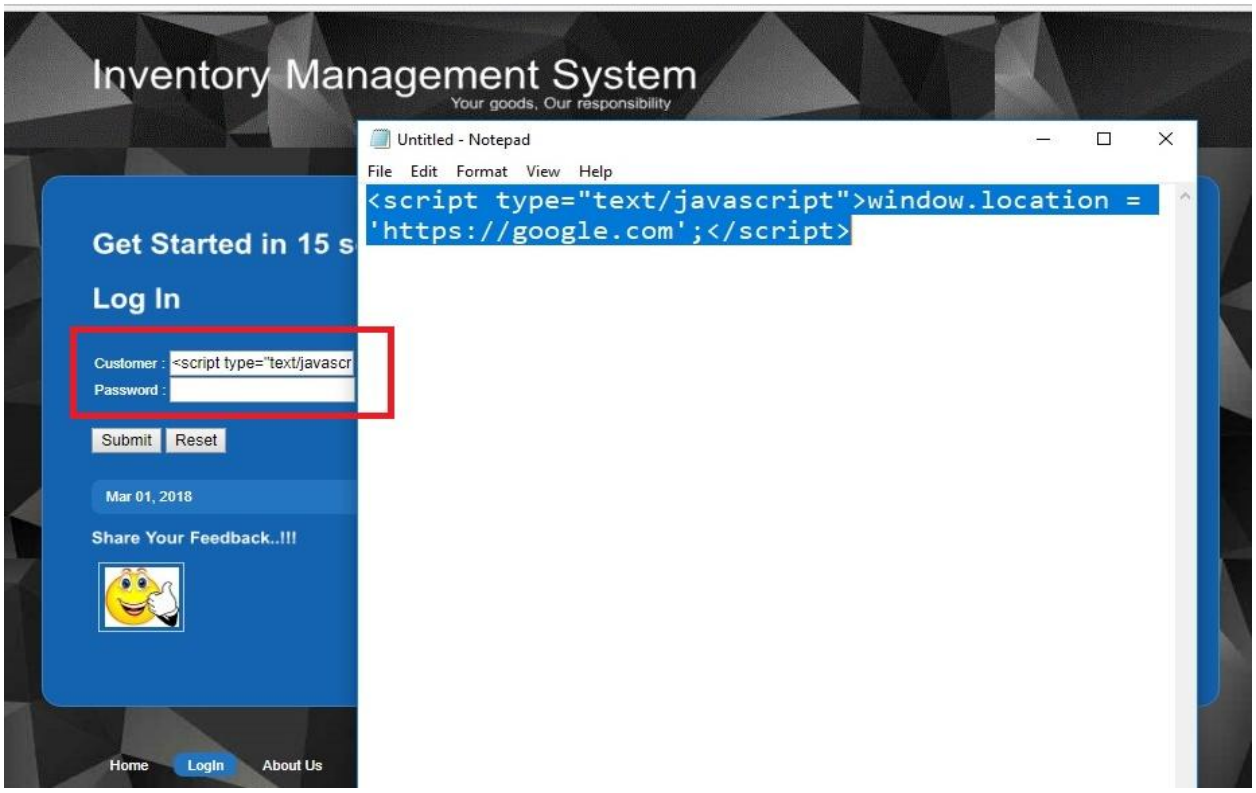
Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy.

Cross site scripting has been implemented as demo application.

Included below code with tag “htmlentities” to prevent the application from XSS.

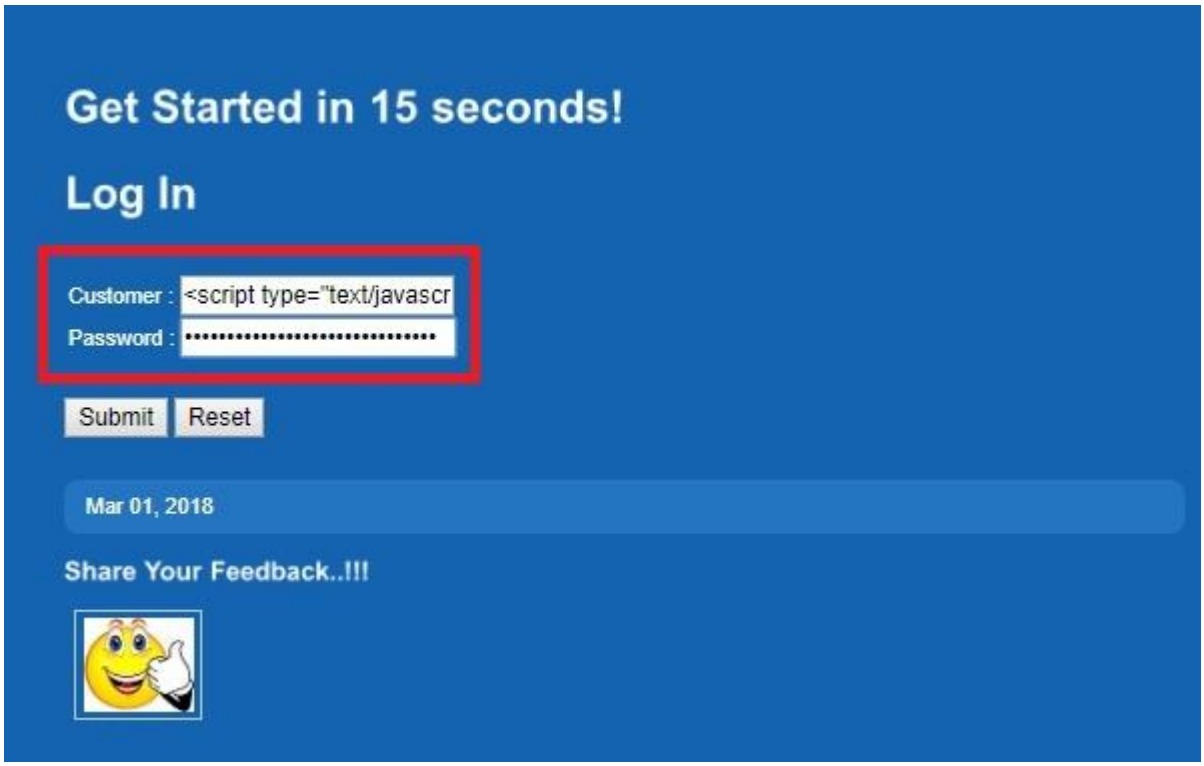
```
if(isset($_POST['input']))
        {
                $data = htmlentities($_POST['input']);
                echo "<p align='center'>$data</p>";
        }
```

Here, I am able to prevent this attack using Validation and Cross site scripting prevention code. Please review Img 5.7.1

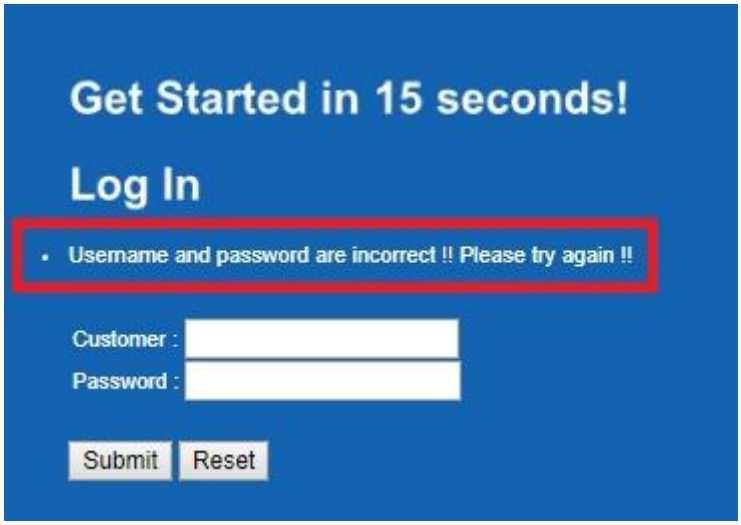


Img 5.7.1

Entered script tag in input field (As shown in notepad file), validation and prevention code is preventing an application from XSS attack. Img 5.7.2, 5.7.3



Img 5.7.2

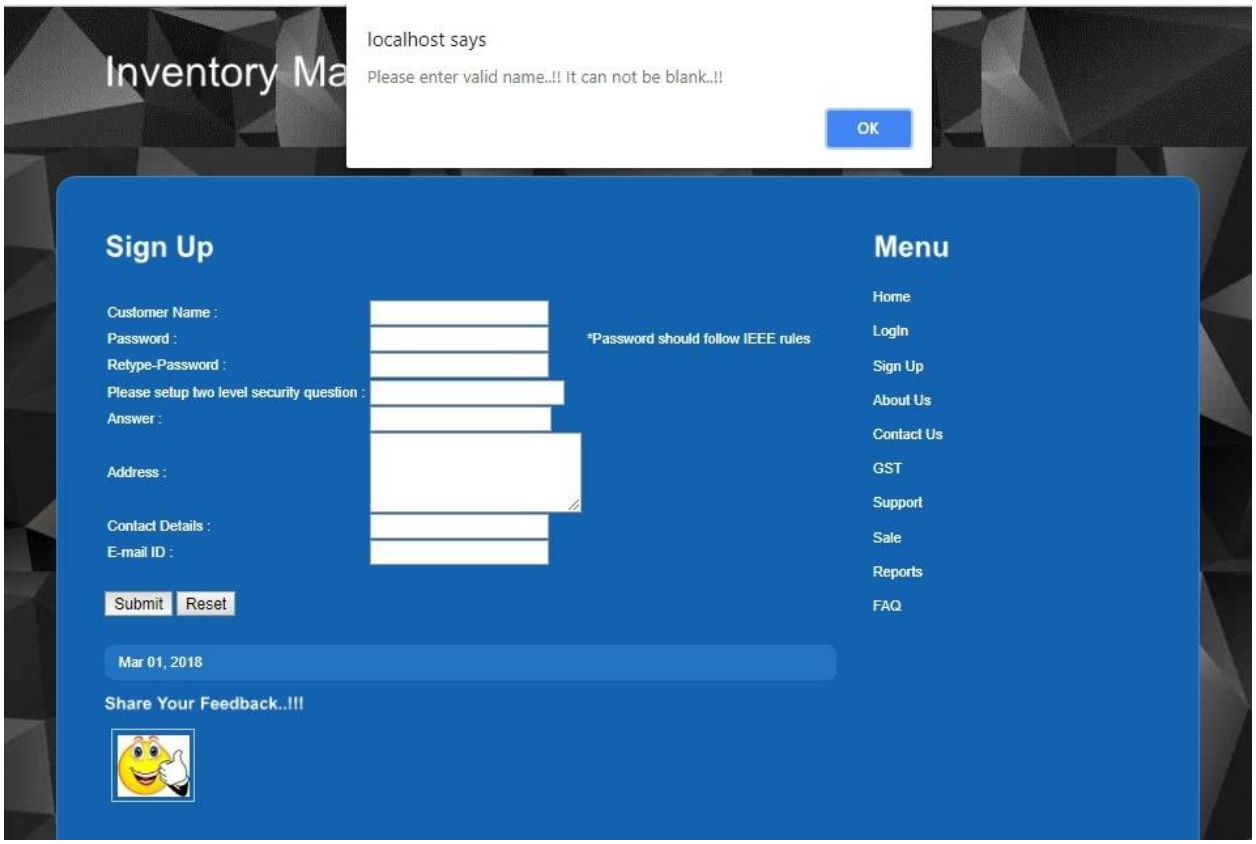


Img 5.7.3

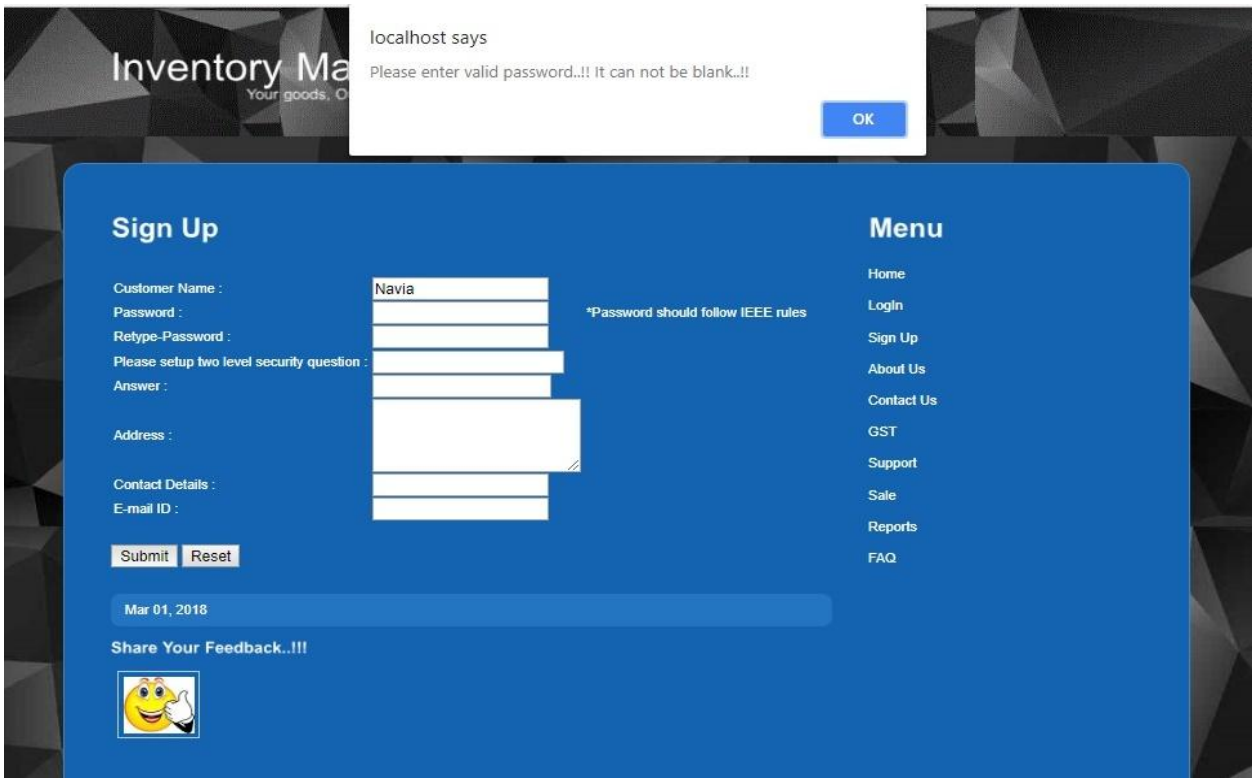
5.8 Customer Signup Form Validation

As discussed in above chapters, validation is an important key feature,

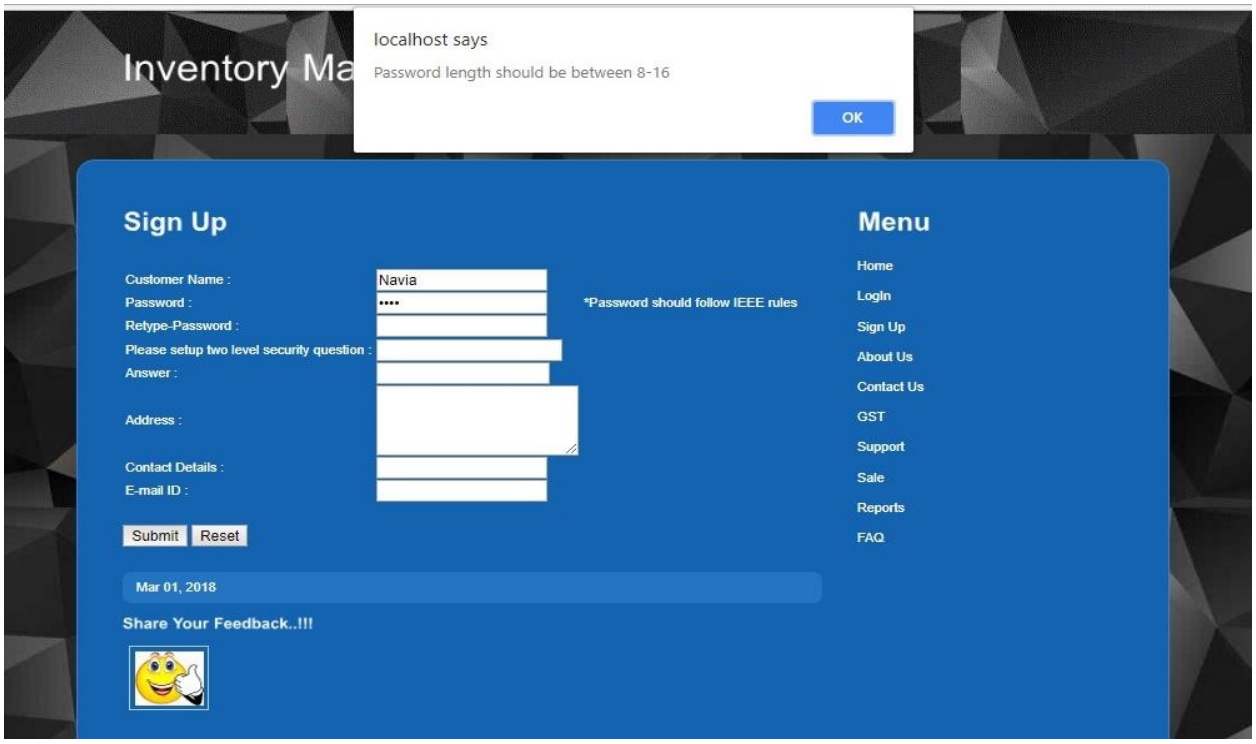
I have implemented validation of Null input check, character limits check, digits check, upper, case lower case character check, length check, syntax check etc. Please review Img 5.8.1, 5.8.2, 5.8.3, 5.8.4, 5.8.5, 5.8.6, 5.8.7, 5.8.8



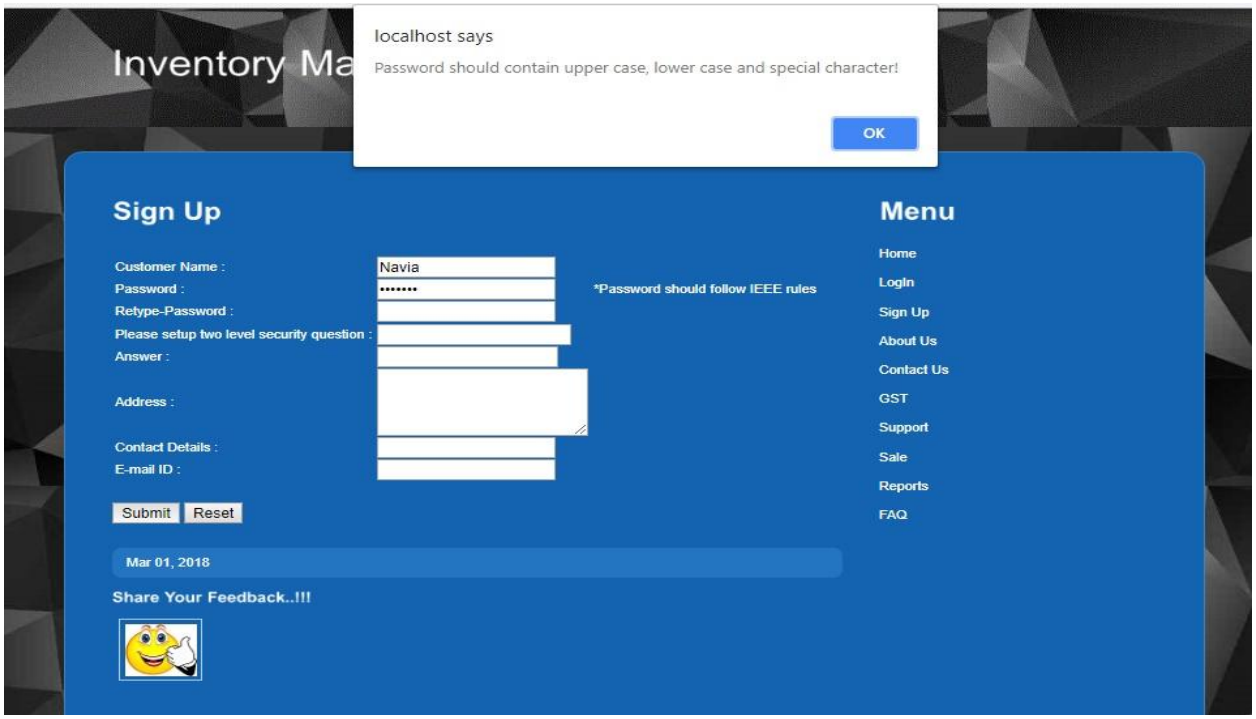
Img 5.8.1



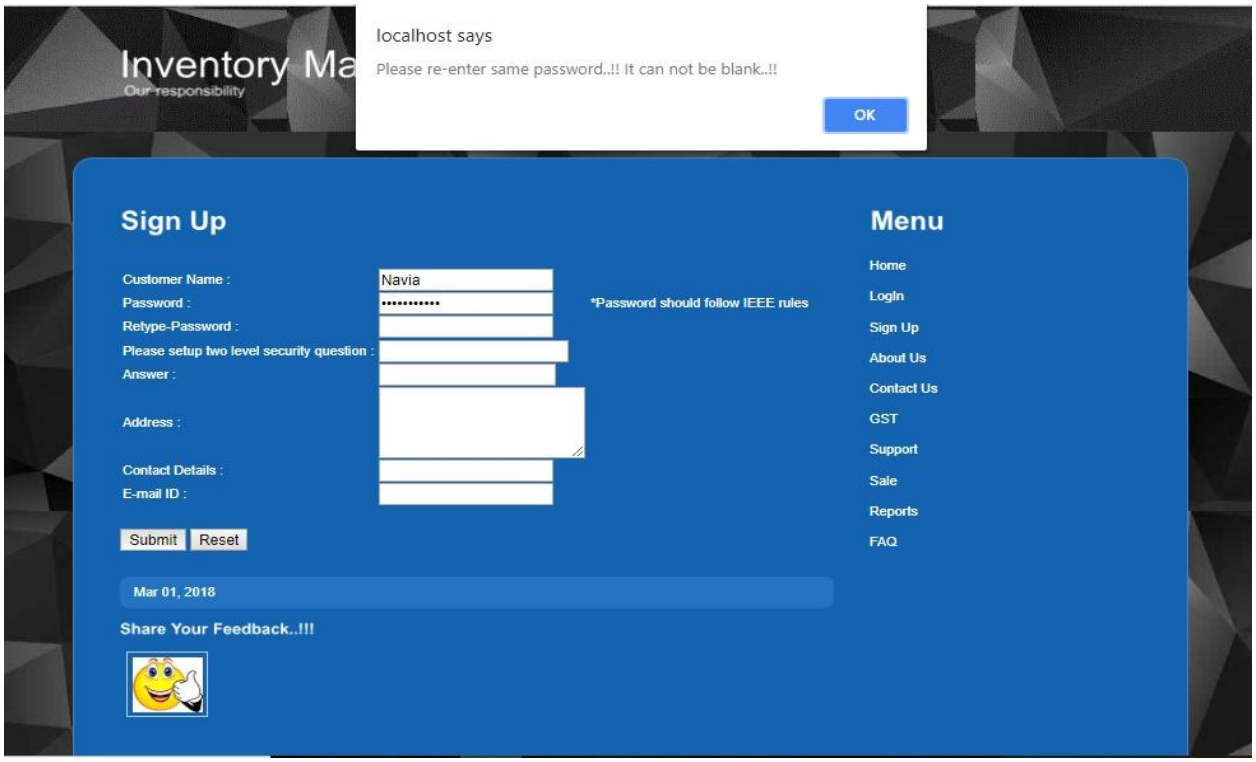
Img 5.8.2



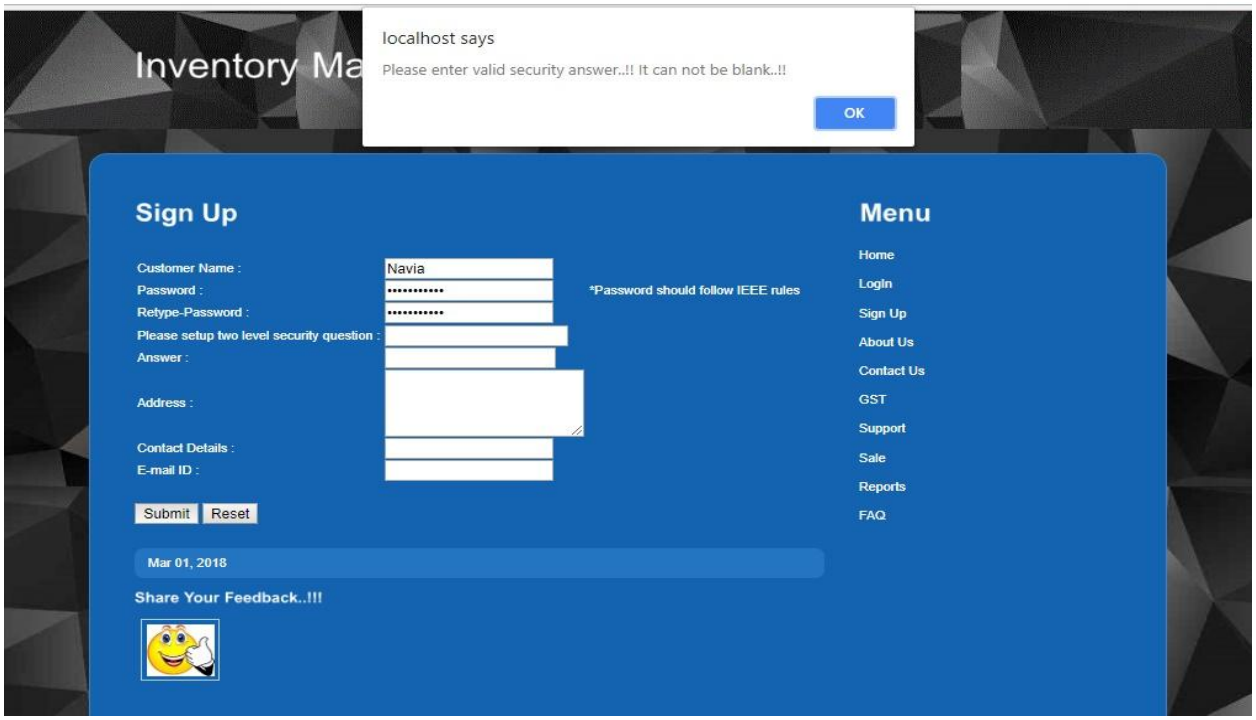
Img 5.8.3



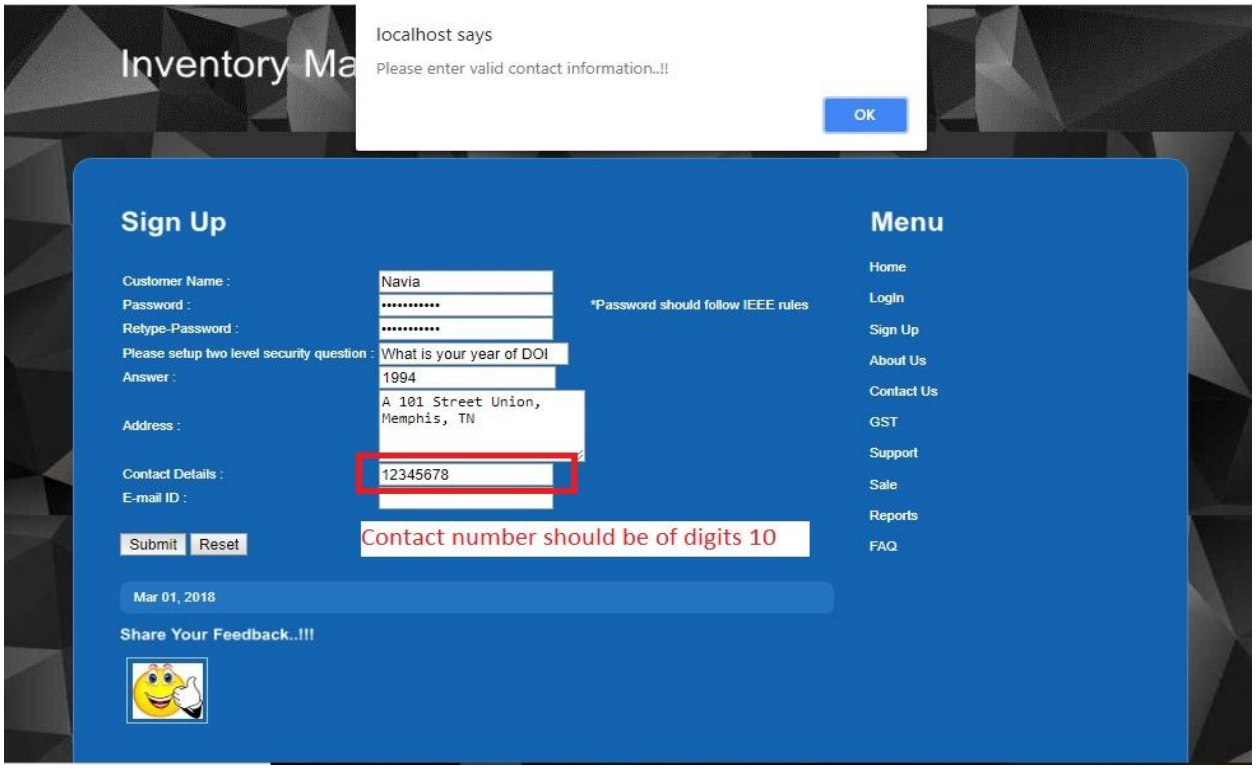
Img 5.8.4



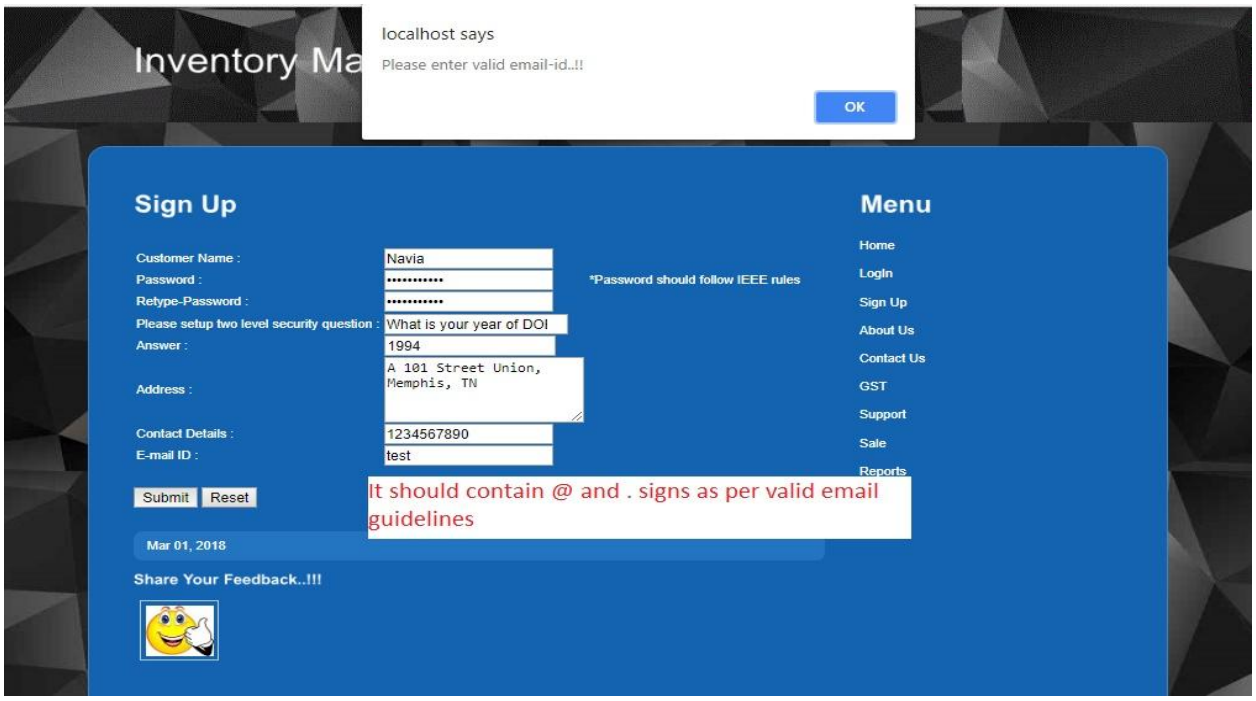
Img 5.8.5



Img 5.8.6



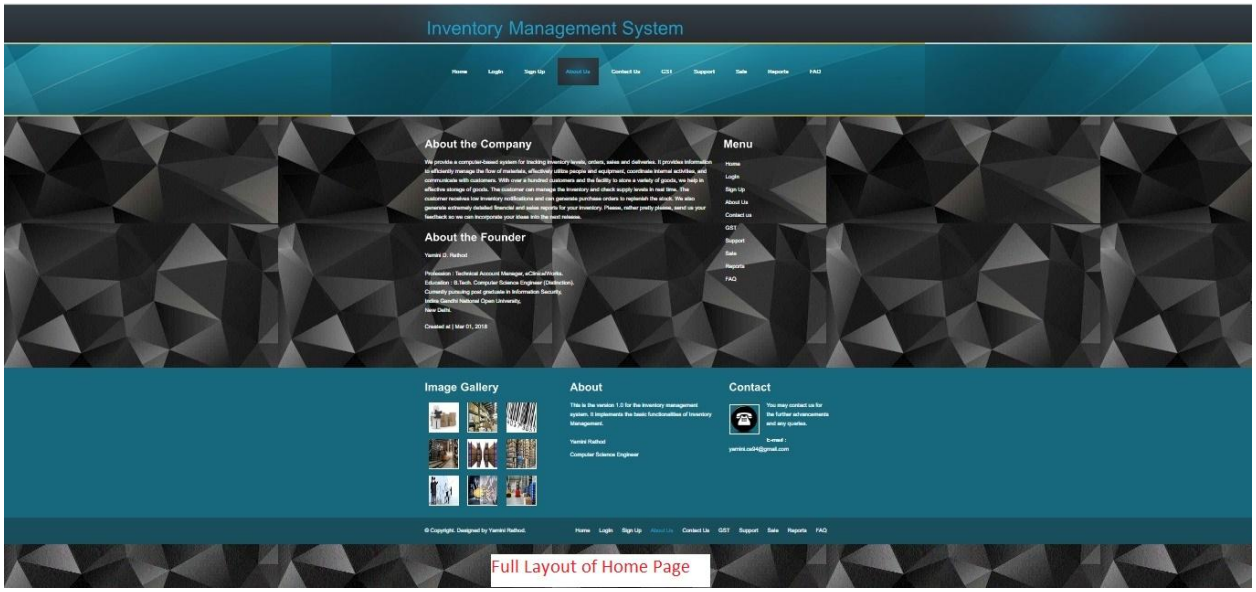
Img 5.8.7


Inventory Ma
localhost says
Please enter valid email-id..!!
OK
Sign Up
Customer Name :
Navia
Password :
Retype-Password :
*Password should follow IEEE rules
Please setup two level security question :
What is your year of DOI
Answer :
1994
Address :
A 101 Street Union, Memphis, TN
Contact Details :
1234567890
E-mail ID :
test
Submit
Reset
It should contain @ and . signs as per valid email guidelines
Mar 01, 2018
Share Your Feedback..!!!
Menu
Home
LogIn
Sign Up
About Us
Contact Us
GST
Support
Sale
Reports

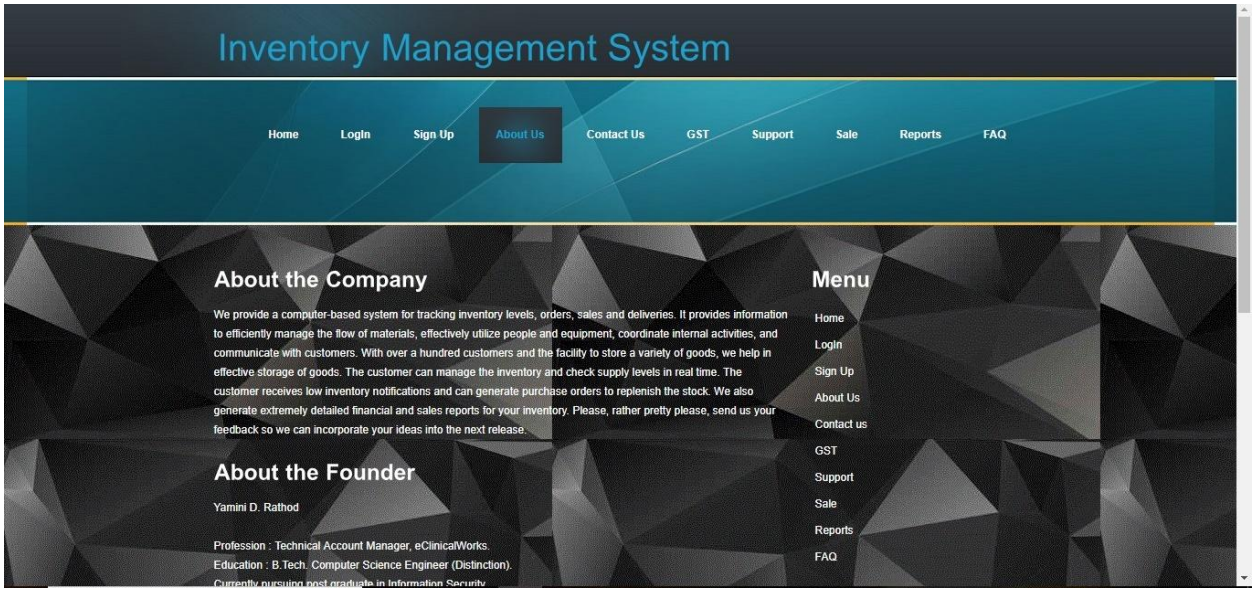
Img 5.8.8

5.9 Inventory Management System Feature

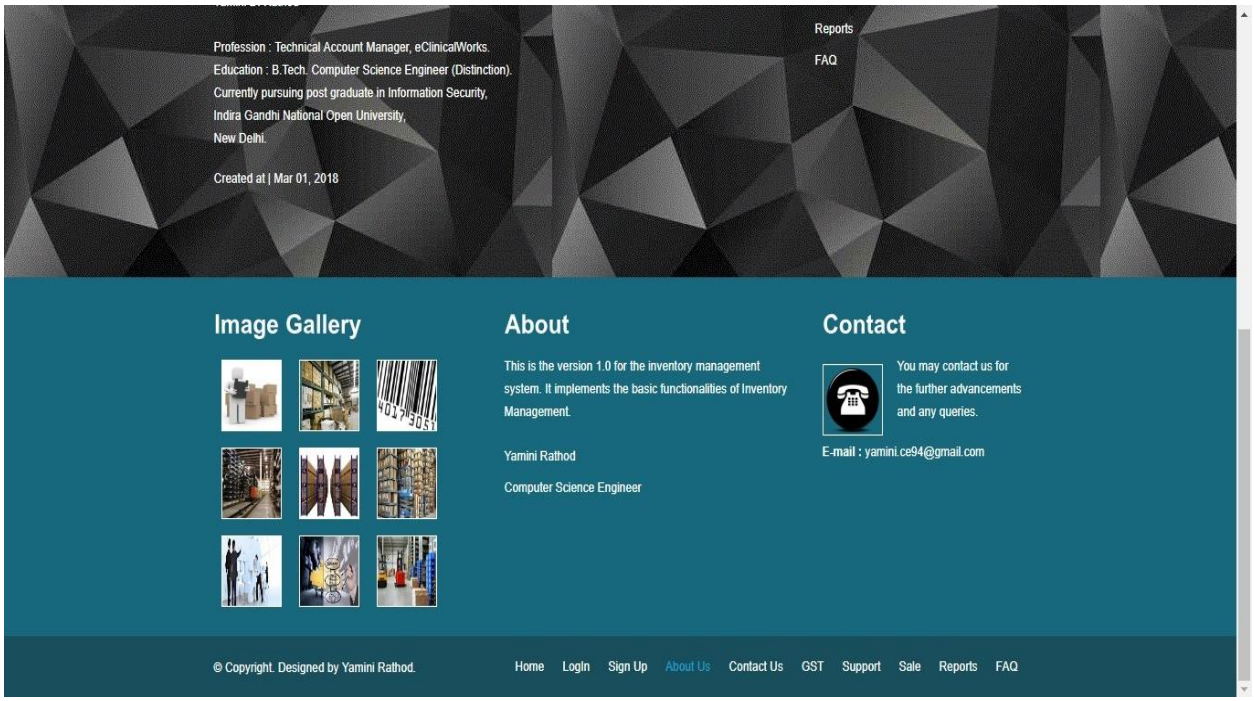
IMS application provides basic features such as they can contact our customer support team, can check FAQ Frequently asked questions, new customers may review the site owner details, customers may raise the questions, monitor GST values, current stoke/sales detail etc Admin get track the activity reports such as number of orders, customer details, growth etc. Please review Img 5.9.1, 5.9.2, 5.9.3, 5.9.4, 5.9.5, 5.9.6, 5.9.7, 5.9.8, 5.9.9, 5.9.10



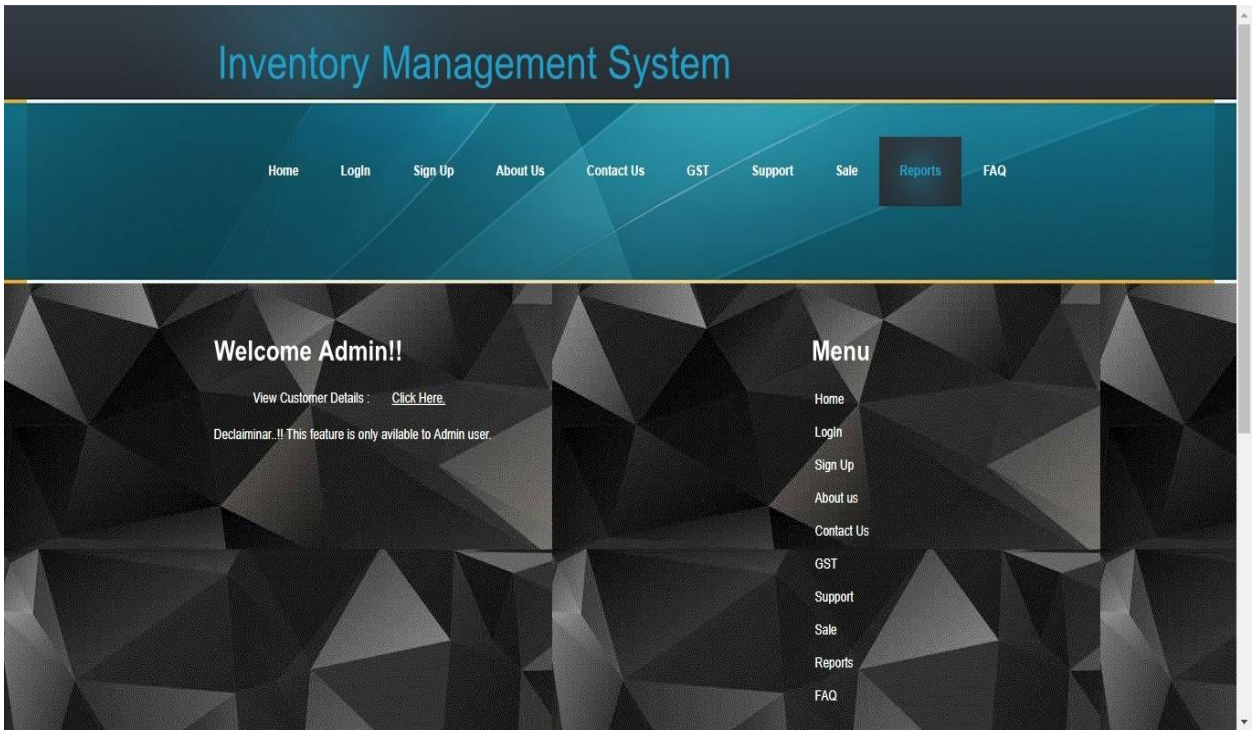
Img 5.9.1



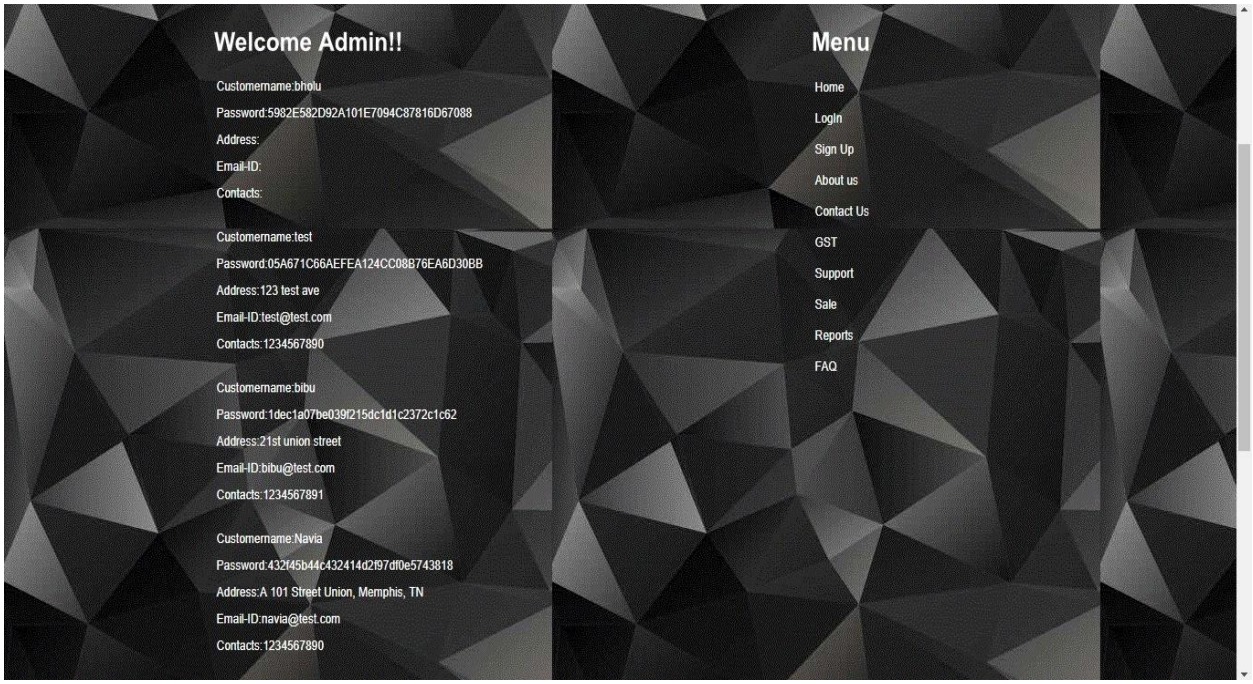
Img 5.9.2



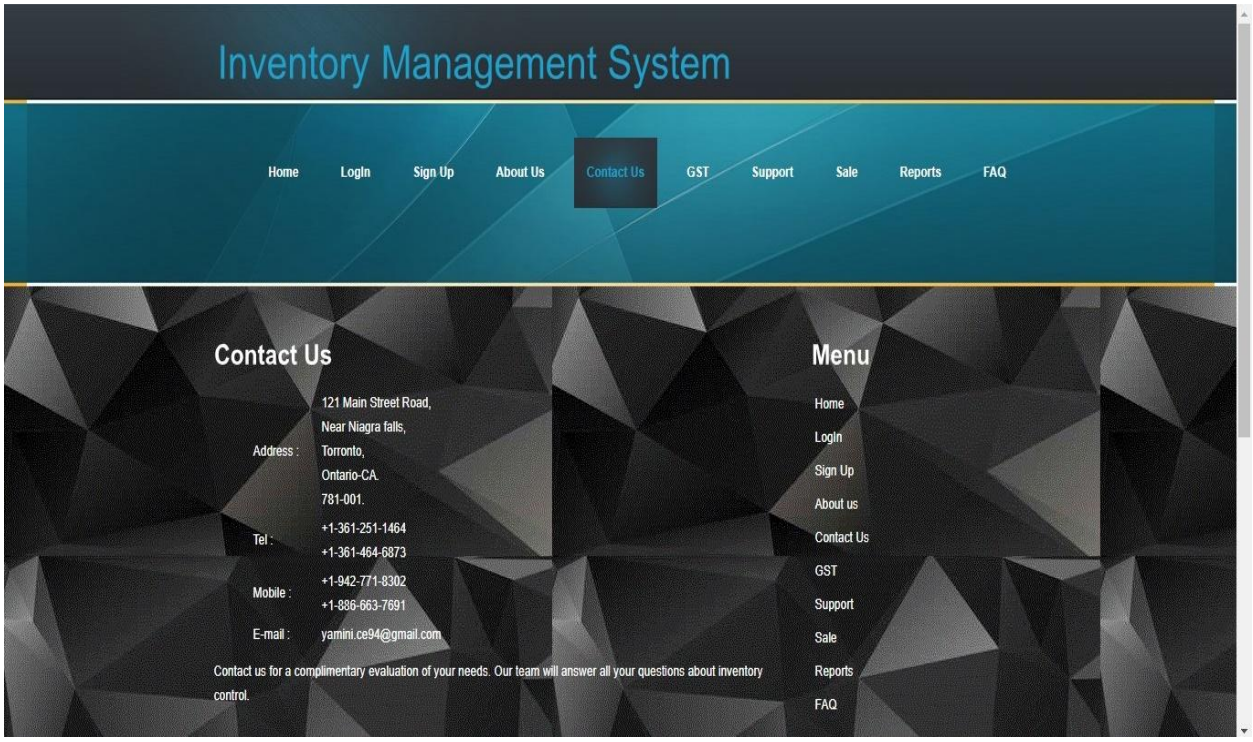
Img 5.9.3



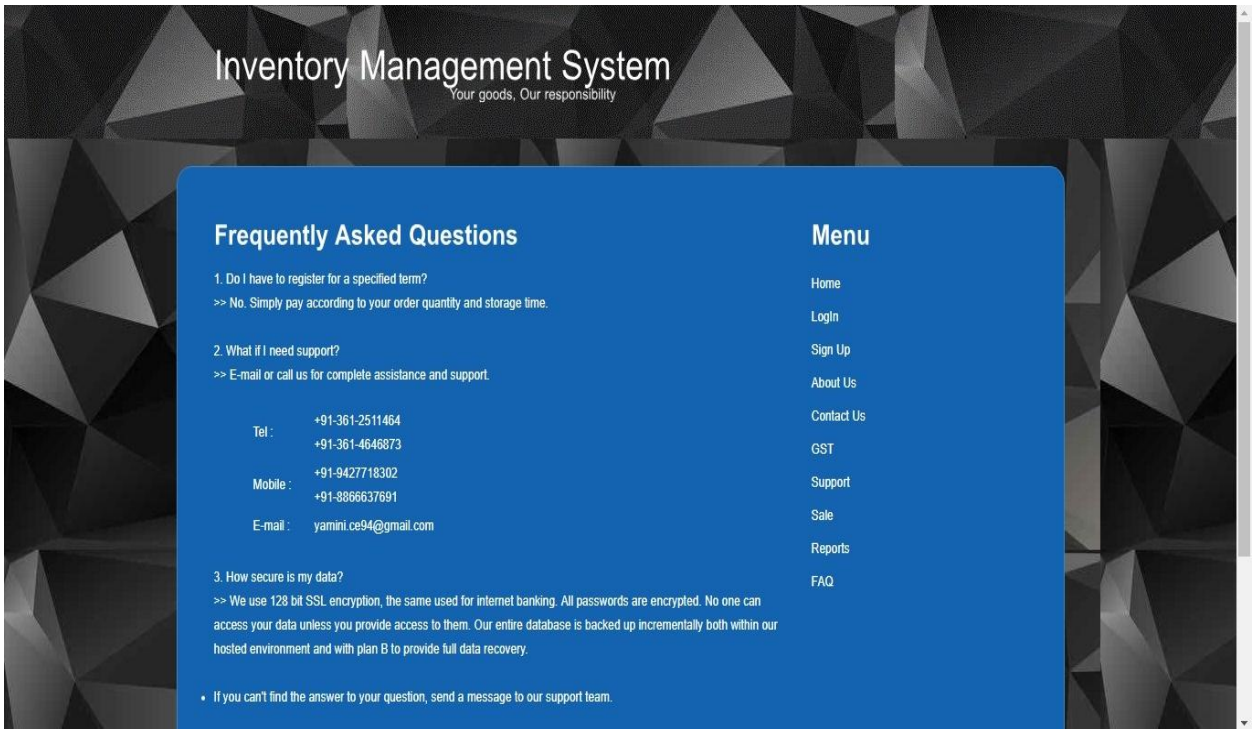
Img 5.9.4



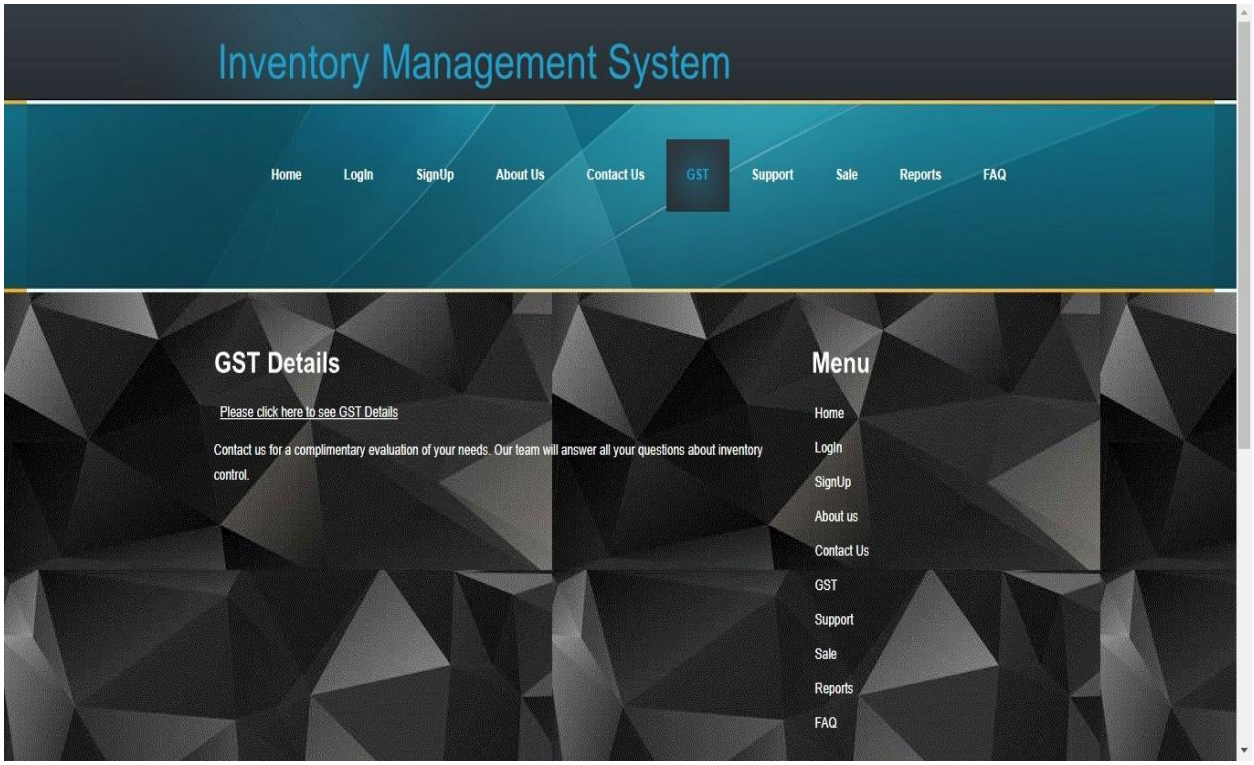
Img 5.9.5



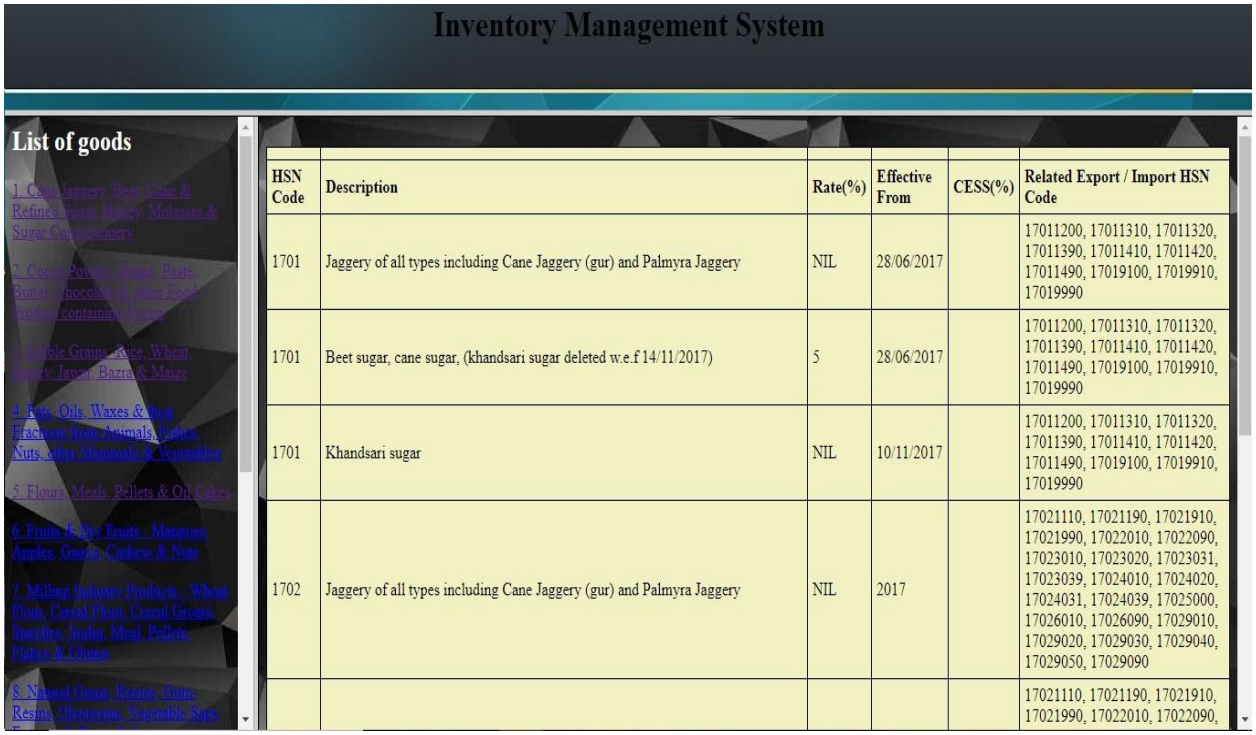
Img 5.9.6



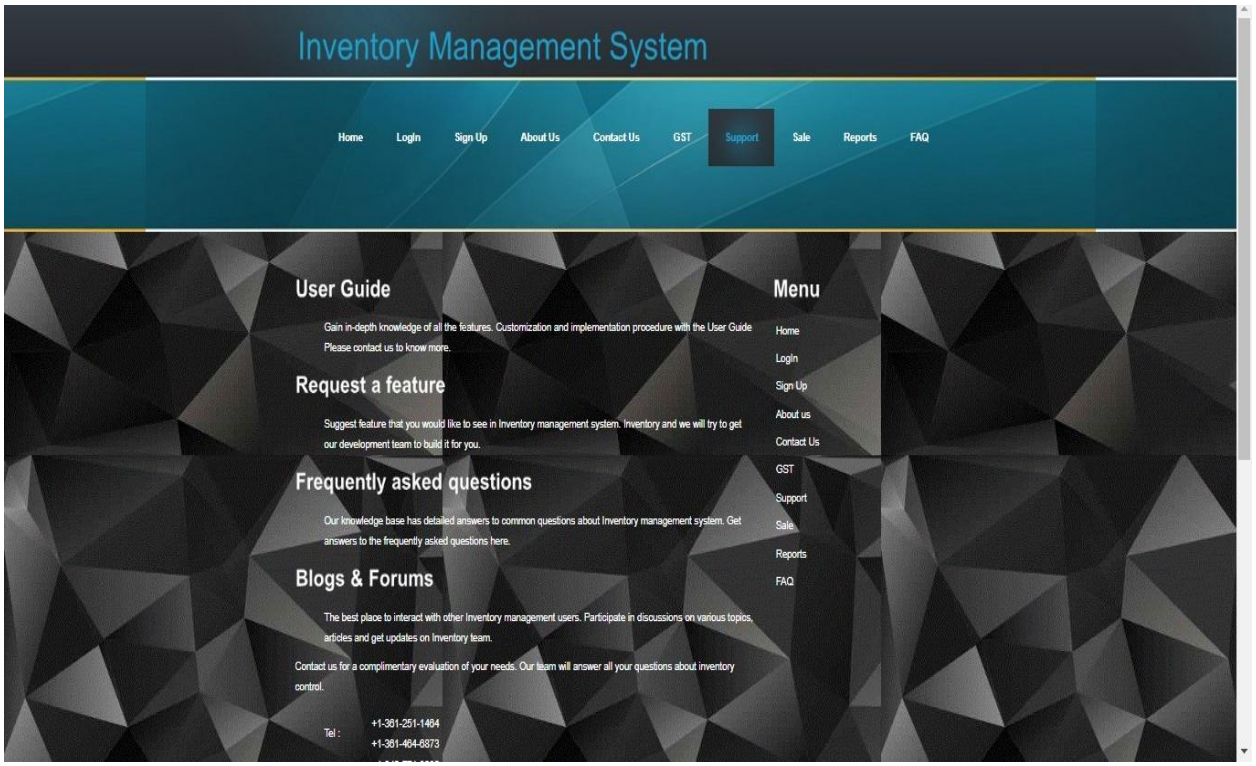
Img 5.9.7



Img 5.9.8



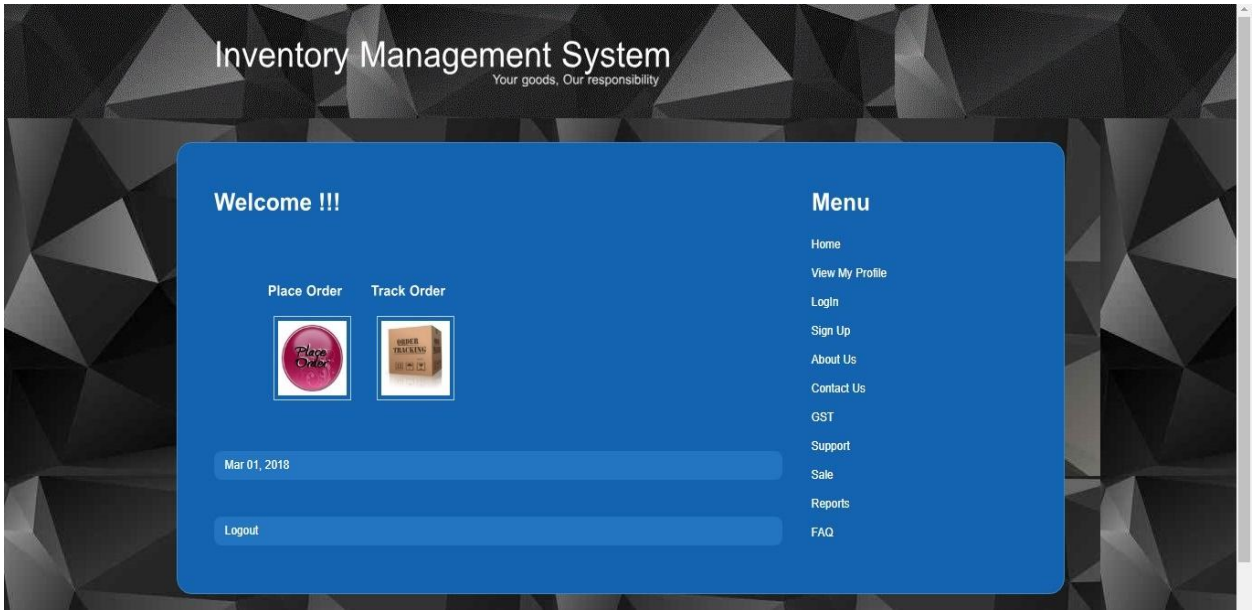
Img 5.9.9



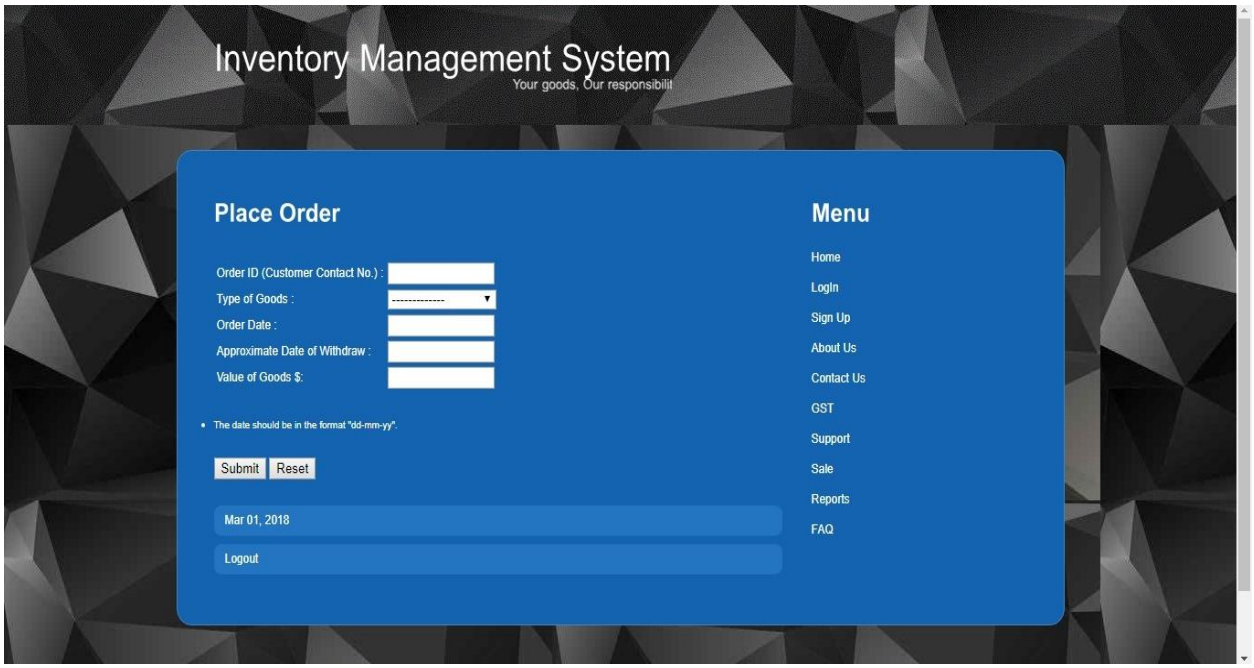
Img 5.9.10

5.10 Order Place and Track

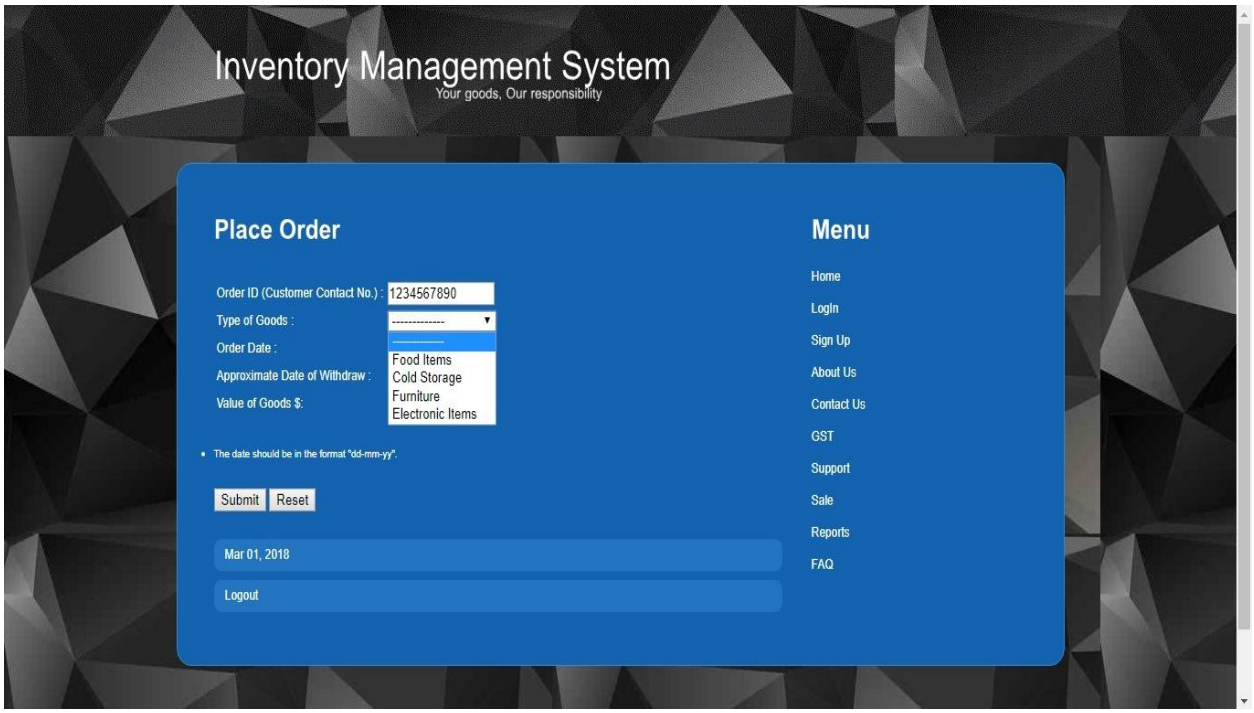
Customer may place the order and track the status of order using track order and place order feature post two level authentications. CustomerId is customer's contact number, I have taken this by considering unique identification parameter. Please review Img 5.10.1, 5.10.2, 5.10.3, 5.10.4, 5.10.5, 5.10.6, 5.10.7



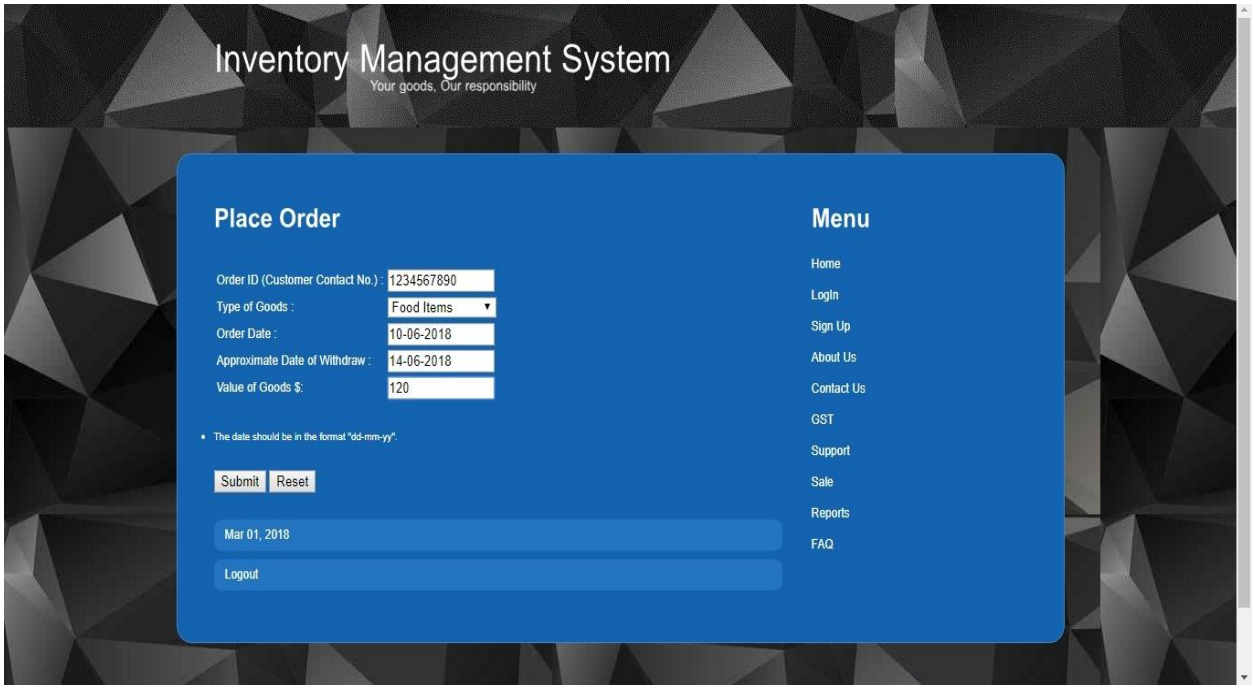
Img 5.10.1



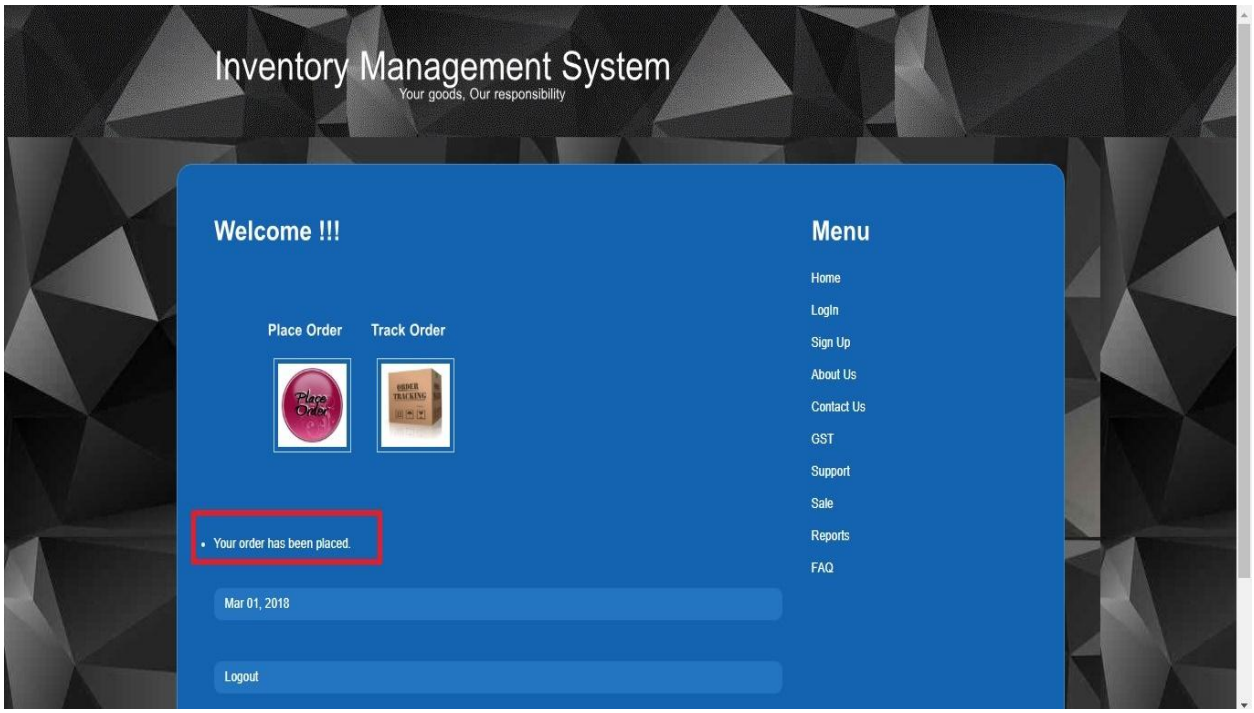
Img 5.10.2



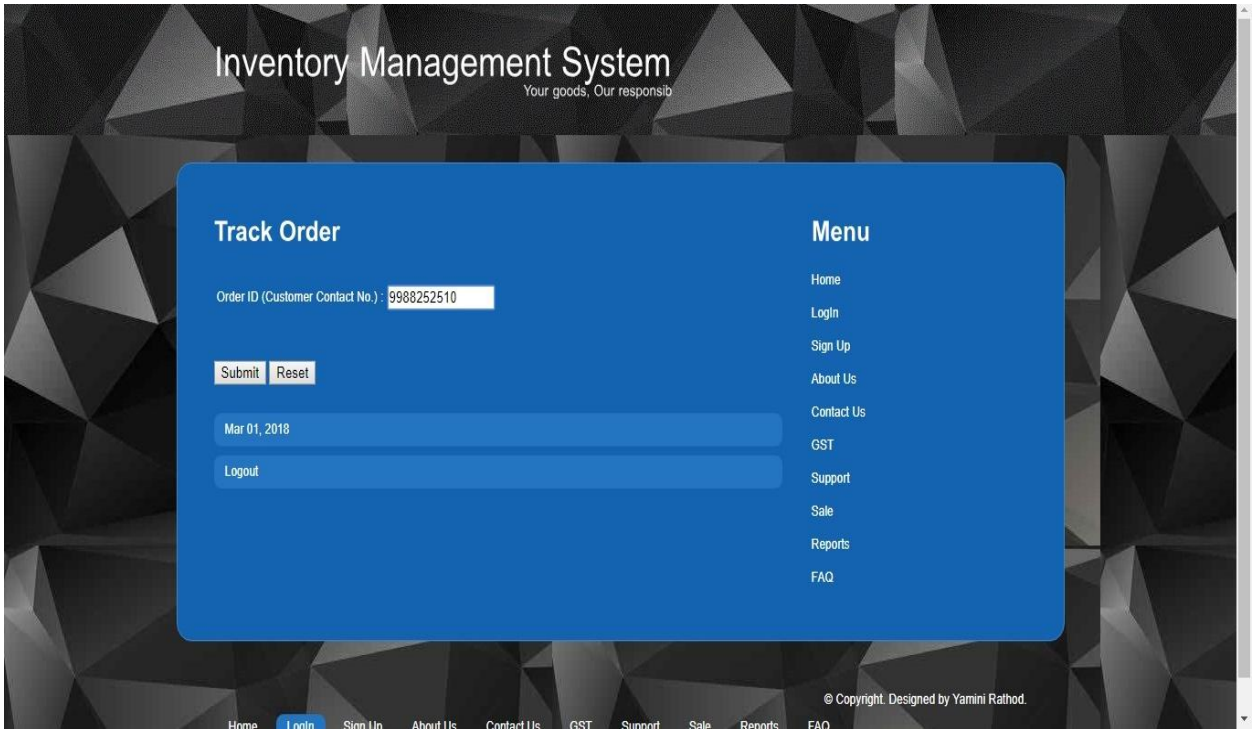
Img 5.10.3



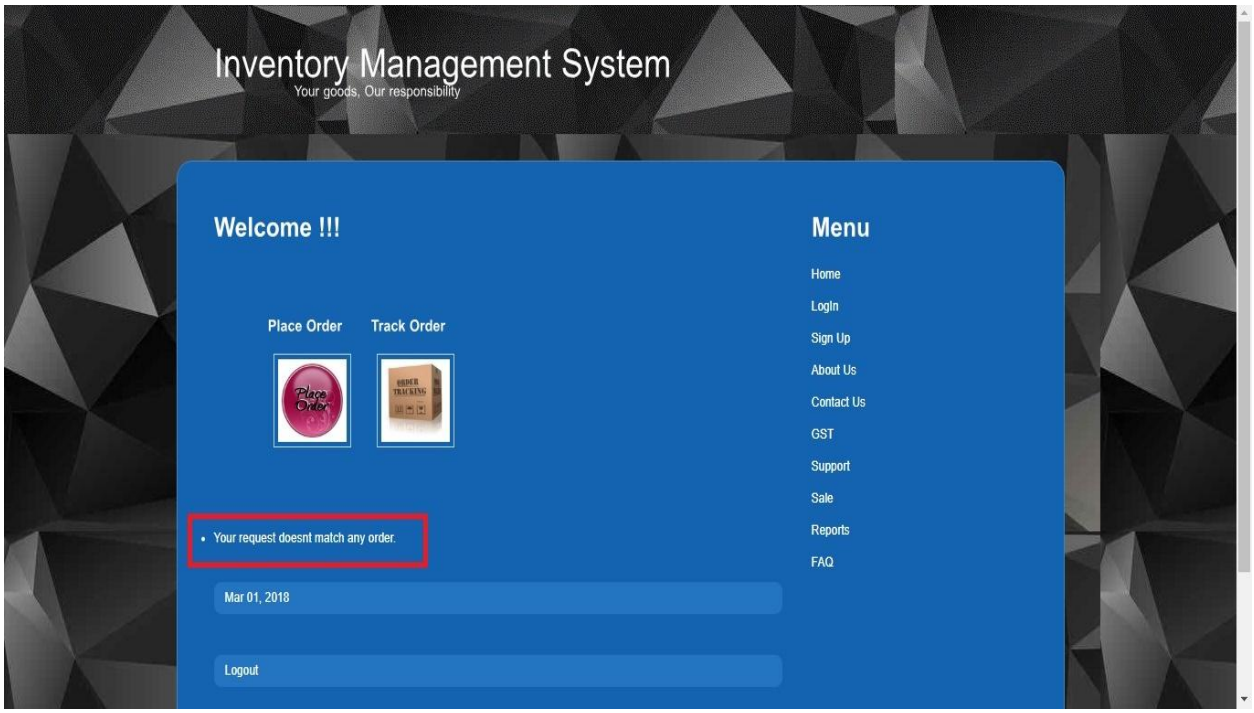
Img 5.10.4



Img 5.10.5



Img 5.10.6


Inventory Management System
Your goods, Our responsibility
Welcome !!!
Menu
Home
LogIn
Sign Up
About Us
Contact Us
GST
Support
Sale
Reports
FAQ
Place Order
Track Order
Your request doesnt match any order.
Mar 01, 2018
Logout

Img 5.10.7

Chapter 6

Conclusion and Suggestions

The agenda of this project is to build an application and implement the security by identifying the "security loop holes".

The main root cause of security loopholes is the faulted application. Coding should be done on basis of security standards.

However, in order to demonstrate the security loopholes we should have some application as platform. As per Indian Cyber laws we cannot perform hacking on online application unless we are part of organization Ethical hacker.

Hence I decided to design my own application to find out the current loopholes and implement same in my own application.

Case study on online application was limited scope. We cannot test the vulnerability attacks on online applications, it is Illegal.

Developers has to use the best features, methods, security tools, high level security identifications such as two level authentication, authorization, session management etc.

It is necessary to keep our resources safe and protected. I have mainly focused on best coding standards and how we can apply security features to keep our resources safe.

Below are the securities features have been implemented and illustrated in this project.

1. Validation

   Implemented Server side, client side, null values, script values, length check, keywords check, characters check, limits check validations.

2. Encryption

   Used this as digest authentication method in IMS application. Implemented all encryption algorithms as case study and compared its performance on basic of basic characteristics, as an outcome SHA-256 has been identified as best algorithm hence implemented same in IMS application.

3. SQL Injection Prevention.

   Implemented code using validation, encryption, sql queries as parameter and sanitization to prevent SQL injection attack in an application.

4. Session Management

   Implemented this feature to prevent attack such as MIM – Man In Middle attack. System will expire the session after certain amount of time if the user is not active in an application. That will redirect the user on main login page and system will regenerate the cookies. In MIM and session hijacking

attack, attacker uses the cookie value to get unauthorized access. It is always suggested to implement the session management in all online web applications.

5. Prevention of Cross Site Scripting

Implemented the prevention code in an application, it is preventing system from XSS attack. Basically in XSS attacker tried to enter the malicious script codes to hack the system, he will enter the script with url in such a way that when user logs in, they will be redirected to malicious webpage that was entered by attacker using script tags. There are rare scenarios where it actually happens, just to prevent this attack and reduce the vulnerability it is good habit to code well.

6. Prevention of Session Hijacking

Implemented two level authentication to prevent this attack. Attacker uses the value of cookie that is redirecting them to destination page without login, it by passes the authentications. However as per case study, in order to prevent this attack application must be SSL certified. It is suggested to secure an application with SSL certificate. This can be done once application goes on live.

7. Password complexity

Implemented this security feature as part of validation, I have tried to use the IEEE standard that forces user to make their password complex and system will not accept it if it is not followed by IEEE guidelines.

8. Two level authentication

Most of sensitive applications such as income text sites, online net banking sites has this feature. It is not limited to that, we may introduce the captcha validation too. Nowadays there are many attacks done by programs. Attacker create programs by implementing malicious code that is used to hack the login details on clients machine without any other interface. Use of captcha identifies between user/robot.

9. Authentication using email token

Implemented this feature in two level authentication screen. System generates the random token and sends it to registered client's email id. User has to get that token value from their email account and post validation they will be allowed to get into an application. This can be fully accomplished once an application goes live and become SSL certified.

10. Access permission rights management

As discussed before, authorization and access control level are not same. Certain customers will not have access on certain module. It will be based on the contract they sign in. They will get the features as per agreement. Admin has access to deactivate the users who are not active in system for more than 6months. When user will try to get it application, they should get the access deny message. This can be applying when the customer contract period has been complemented.

11. Error Handling

Smart coding is something that does not show the technical details in case error occurs somewhere in system. Hackers are highly technical profession people who use detailed errors to get the knowledge of an application, workflow, coding, design etc I have tried to minimize the errors as much as possible in this project.

12. Wireshark

This is widely used tool almost by all organization to keep track on network traffic. It is just limited to certain features. I.e. we won't be able t track the performance related issues. However this tool gives in depth information about all routing packets.

13. Fiddler

I have used this tool as testing tool to narrow down the performance of an application. It is used to track network traffic on both HTTP/HTTPS websites. We can also block a web page that seems to be malicious. Use of this tool would be an advantage to plus the security feature.

14. Customer Validation

In case of customer oriented organizations, organization sales team validates the customer nature, their requirements, business strategies etc. Post validating, the service contract will get signed. Application should be smart enough to validate the customer details, post registration admin will have to grant access to customer and then client will be able to use the application.

15. Prevention of DoS

It is suggested to an organization to build firewall on both client machine and on network, implement Intrusion Detection System, keep safe systems using antivirus, continuous monitor activities using tools such as wireshark, fiddler, block HTTP urls these are sufficient to prevent from DoS attacks.

Chapter 7

Future scope and further enhancement of the Project

This Supermarket Inventory System creates purchase orders once the inventory level reaches to a pre-defined level. Supermarkets and the vendor's warehouse use this system to create receipt and invoice. The accounting department uses this system to match invoice and receipt so that the payment can be recorded accurately.

By this project we are able to focus on both small and big retail stores in helping manage their Inventory of their store with security implementation.

The main goal as of now is to implement application by considering security loop holes. We have analyzed and implemented web security in this project followed by case study.

By security inputs, it decrease the vulnerability of application being from hacked and attacks. It provides the surety and trust to the customers that their resources are safe and secure. Customers trust should be an organizations job.

The information security in important in the organization because it can protect the confidential information, enables the organization function, also enables the safe operation of application implemented on the organization's Information Technology system, and information is an asset for an organization.

Future scope of this project is vast, as we are trying to implement security and decreasing loop holes that help to create security awareness.

It will be helpful and used in government organizations, private companies, non private sectors; researchers to make it much better and would be refer by institutions too.

Further enhancement of this project will really help to build own company that provides Inventory support with all security features that will be used by all store keepers, allow the supermarket staff to create, maintain and view the contents and value of its inventory of items in a categorized way with less vulnerability. This system is a tool for tracking asset levels, order management, safety stock, sales and deliveries. It would help to avoid product overstock and outages.

Chapter 8

Bibliography

Web References :

1. www.google.com
2. www.wikipedia.com
3. http://cyberlawsindia.net
4. http://www.omnisecu.com
5. https://timesofindia.indiatimes.com/india/one-cybercrime-in-india-every-10-minutes/articleshow/
6. https://searchsoftwarequality.techtarget.com
7. www.softwaretestingfundamentals.com
8. http://www.cyberralegalservices.com
9. https://www.tradegecko.com
10. https://www.cdse.edu/resources/case-studies/information-security.html
11. https://www.ibm.com/security/solutions/
12. https://developer.mozilla.org/en-US/docs/Web/Security

Articles:

1. Analysis of Cryptography Techniques by Yamini Rathod, Prof Shivani Desai IJCSC Volume 5 March Sept 2014.
2. Application Life Cycle Management by Yamini Rathod, IJSRD Volume 2, Issue 02, 2014.
3. Supply Chain Inventory Management and the Value of Shared Information by Gerard P. Cachon and Marshall Fisher.
4. Exploring the knowledge inventory in project-based organizations: a case study by Dirk Pieter van Donk and Jan Riezebos
5. Introduction to materials management by JRT Arnold, SN Chapman and CM Clive.
6. Operations management by WJ Stevenson and M Hojati.
7. Computer security: principles and practice by W Stallings, L Brown, MD Bauer and AK Bhattacharjee.
8. Integrating case study and survey research methods: an example in information systems by European journal of information systems.
9. ISRAM: information security risk analysis method by B Karabacak.
10. The positive outcomes of information security awareness training in companies–A case study by M Eminagaoglu, E Ucar and E Eren.
11. Network Security: A Case Study by SJ Lincke.
12. Software Engineering by Roger S. Pressman.