**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JNANA SANGAMA”, BELAGAVI- 590018**



**A MINI PROJECT REPORT**

**ON**

**ONLINE GIFT STORE MANAGEMENT SYSTEM**

Submitted in partial fulfilment of the requirements

For the award of degree of

**Bachelor of Engineering**

**In**

Computer Science and Engineering

**By**

KEERTHI.G.NAIDU

[1KS15CS041]

Under the guidance of

Mrs. Sougandhika Narayan Mr.K Venkata Rao Mr.Kumar K

Asst.Prof, Dept. of CSE Assoc.Prof,Dept. of CSE Asst.Prof, Dept. of CSE



**Department of Computer Science & Engineering**

**K.S. INSTITUTE OF TECHNOLOGY**

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109.

2016-2017

**K.S. INSTITUTE OF TECHNOLOGY**

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109.

**Department of Computer Science & Engineering**



**CERTIFICATE**

This is to certify that mini project work entitled “ONLINE GIFT STORE MANAGEMENT SYSTEM” carried out by **Ms. KEERTHI.G.NAIDU** bearing USN **1KS15CS041** bonafide student of **K.S. Institute of Technology** in the partial fulfilment for the award of the **Bachelor of Engineering in Computer Science & Engineering** of the **Visvesvaraya Technological University**, Belagavi, during the year 2017. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library . The mini project report has been approved as it satisfies the academic requirements in respect of mini Project work prescribed for the said degree.

**Dr. Rekha. Venkatapur Dr. T.V. Govindaraju**

Prof & HOD, CS&E Department Principal/Director, KSIT

**Mrs. Sougandhika Narayan Mr.K Venkata Rao Mr.Kumar K**

Asst.Prof, Dept. of CSE Assoc.Prof, Dept. of CSE Asst.Prof, Dept. of CSE

**Name of the Examiners Signature with date**

**1.**

**2.**

**3.**

**ACKNOWLEDGEMENT**

I take this opportunity to thank one and all involved in building this project. Firstly, I would like to thank the college for providing us an opportunity to work on the project.

First and foremost, I thank the management of **K. S. Institute of Technology** for providing all the resources required for the project.

I wish to acknowledge my sincere gratitude to our beloved **Principal, Dr. T.V. Govindaraju** for his encouragement and providing all facilities for the accomplishment of this project.

The project would not have been possible without the support of our beloved Prof & HOD, **Dr.Rekha.B.Venkatapur ,**  Dept of CS&E.

I am also highly grateful to my project guide, **Mrs.Sougandhika Narayan, Asst.Professor, Dept of CS&E, Mr.K Venkata Rao, Assoc.Prof, Dept. of CSE**, **Mr.Kumar K, Asst.Professor, Dept of CS&E,** who have been very generous in assisting and supporting, to do this Project “**Online Gift Store Management”,** which formally started as just a rough idea and now it is resulted in this.

I also would like to thank all the other teaching and nonteaching staff members who had extended their support and co-operation while bringing up this project.

**Keerthi.G.Naidu**

**(1KS15CS041)**

**ABSTRACT**

The system creates a web based manufacturing system that enables a manufacturing industry to schedule its manufacturing operations based on the daily update of sales from its dealers. Once the sales figures of items for the past week are entered by the dealers over the internet along with the orders for the next delivery, the schedule for the next week’s production will be drawn up. A report of the required raw materials or parts will be drawn up with the product requirements over the internet & asked to quote their rates.

Once the rates are quoted, the order will be placed with the required delivery schedules. Once the parts are supplied the stocks will be updated. Then a production plan will be drawn up taking the bill of materials into consideration. Once the production plan is approved, the stock will be updated when the material is issued. Once the finished products are available the delivery schedules will be drawn up based on the orders placed by the Dealers. The stocks with the dealers will also be maintained.

The Benefits of the Stores Management System is

* It is used as an internet Application.
* Providing High-Security.
* Easy Business Solutions.

**Modules used:**

* admin module:
* View Dealers and Customers.
* Approve New Dealers and Requests.
* Stock Maintenance
* secure customer info

* dealer module:
* New Dealer Signup.
* View items, Deals and Customers.
* Approve gift Requests.
* delivers items to admin.
* customer module:
* list of customers and their details
* Forward request for gifts to Dealer.
* Check Request status.
* items module:
* items present
* provides information about cost of the items
* brands

**CHAPTER 1**

**INTRODUCTION**

**1.1 Overview of the project**

This project named **“Online Gift Store Management”** is a database project used to choose online gifts easily.

While shopping online, the threat of fake and malicious gifts are always possible. This project mainly deals with this problem. The admin makes sure that his customer does not buy any unlicensed or malicious products. The user can choose any of the available gifts or he is given an exclusive privilege of adding his desired gifts. Admin will look into the availability and other aspects of the gift. After verifying everything the admin decides whether to provide the license for buying the gift or not. Once the license is provided the user can visit the respective site and buy the gift. If the admin doesn’t provide the license the product is detected to be malicious and the user cannot buy it.

This helps the user to have a systematic and secured shopping process avoiding chaos. The user is also given exposure to other new products in the market when he reaches the respective websites. This allows the user to also compare between the products and choose the best.

**1.2 Introduction to Database Management System**

Databases and database technology have had a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, social media, engineering, medicine, genetics, law, education and library science. The word **database** is so commonly used that we must begin by defining what a database is. Our initial definition is quite general.

A **database** is a collection of related data. By **data**, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers and addresses of people you know. Nowadays, this data is typically stored in mobile phones, which have their own simple database software. This can also be recorded in an indexed address book or stored on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database.

A **database management system (DBMS)** is a computerized system that enables users to create and maintain a database. ‘The DBMS is a general-purpose software system that facilitates the process of defining, constructing, manipulating and sharing databases among various users and applications.’ **Defining** a database involves specifying the datatypes, structures and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary, it is called **meta-data. Constructing** the database is the process of storing the dataon some storage medium that is controlled by the DBMS. **Manipulating** a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld and generating reports from the data. **Sharing** a database allows multiple users and programs to access the database simultaneously.

An **application program** access the database by sending queries or requests for data to the DBMS. A **query** typically causes some data to be retrieved, a **transaction** may cause some data to be read and some data to be written into the database.

Other important functions provided by the DBMS include protecting the database and maintaining it over a long period of time. **Protection** includes ‘system protection’ against hardware or software malfunction and ‘security protection’ against unauthorized or malicious access. A typical large database may have a life cycle of many years, so the DBMS must be able to **maintain** the database system by allowing the system to evolve as requirements change overtime. We will call the database and DBMS software together a **database system.**

**1.2 Introduction to HTML**

HTML is the standard markup language for creating Web pages.

* HTML stands for Hyper Text Markup Language.
* HTML describes the structure of Web pages using markup.
* HTML elements are the building blocks of HTML pages.
* HTML elements are represented by tags.
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on.
* Browsers do not display the HTML tags, but use them to render the content of the page.

An example is:

<!DOCTYPE html>  
<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>  
  
</body>  
</html>

All HTML documents must start with a document type declaration: **<!DOCTYPE html>**.The HTML document itself begins with **<html>** and ends with **</html>**.The visible part of the HTML document is between **<body>** and **</body>**.

**1.3 Introduction to CSS.**

* **CSS** stands for **C**ascading **S**tyle **S**heets
* CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
* CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
* External stylesheets are stored in **CSS files**
  1. **CSS Selectors**

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

**i)The element Selector -** The element selector selects elements based on the element name.

## **ii)The id Selector -** The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element should be unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

## **iii)The class Selector -** The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

**1.4 Introduction to JAVA**

## **1.41 What is Java?**

* Java is an object oriented language developed by Sun Microsystems and released in 1995.
* Java was originally developed by **James Gosling** at **Sun Microsystems** (which has since merge into Oracle Corporation).
* Java programs are platform independent which means they can be run on any operating system with any type of processor as long as the [Java interpreter](http://www.w3schools.in/java-tutorial/java-virtual-machine/) is available on that system.
* Java code that runs on one platform does not need to be recompiled to run on another platform, it’s called **write once, run anywhere(WORA)**.
* [Java Virtual Machine (JVM)](https://www.w3schools.in/java-tutorial/java-virtual-machine/) executes Java code, but is written in platform specific languages such as [C](https://www.w3schools.in/c/intro/)/[C++](https://www.w3schools.in/cplusplus/intro/)/ASM etc. JVM is not written in Java and hence **cannot be platform independent** and Java interpreter is actually a part of JVM.

## **1.42 Where is Java being Used?**

Earlier java was only used to design and program small computing devices but later adopted as one of the platform independent programming language and now according to Sun, 3 billion devices run java.

Java is one of the most important programming language in today’s IT industries.

* **JSP –**Java is used to create **web applications** like [PHP](https://www.w3schools.in/php/intro/) and ASP, JSP(Java Server Pages) used with normal HTML tags, which helps to create dynamic web pages.
* **Applets –**This is another type of Java program that used within a web page to add many new features to a web browser.
* **J2EE –**The software Java 2 Enterprise Edition are used by various companies to transfer data based on [XML](https://www.w3schools.in/xml/intro/) structured documents between one another.
* **JavaBeans –**This is something like Visual Basic, a reusable software component that can be easily assemble to create some new and advanced application.
* **Mobile** **–**Besides the above technology, Java is also used in mobile devices, many kind of games and services built-in Java. Today, all leading mobile service provider like Nokia, Siemens, Vodafone are using Java technology.

## **Types of Java Applications**

## **Web Application –** Java is used to create server-side web applications. Currently, servlet, jsp etc. technologies are used.

## **Standalone Application –** It is also known as desktop application or window-based application. An application that we need to install on every machine or server such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

## **Enterprise Application –** An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

## **Mobile Application –** Java is used to create application softwares for mobile devices. Currently Java ME is used for creating applications for small devices, and also Java is programming language for Google Android application development.

## **1.44 Facts about Java**

* **Object Oriented**– In java everything is an Object. Java can be easily expanded since it is based on the Object model.
* **Platform independent –**C and C++ are platform dependency languages hence the application programs written in one Operating system cannot run in any other Operating system, but in platform independence language like Java application programs written in one Operating system can able to run on any Operating system.
* **Simple –**Java is designed to be easy to learn. If you understand the basic concept of OOP java would be easy to master.
* **Secure –** With Java’s secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
* **Architectural-neutral –**Java compiler generates an architecture-neutral object file format which makes the compiled code to be executable on many processors, with the presence Java runtime system.
* **Portable –**being architectural neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler and Java is written in ANSI C with a clean portability boundary which is a POSIX subset.
* **Robust –**Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
* **Multi-threaded –**With Java’s multi-threaded feature it is possible to write programs that can do many tasks simultaneously. This design feature allows developers to construct smoothly running interactive applications.
* **Interpreted –**Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.
* **High Performance –**With the use of Just-In-Time compilers Java enables high performance.
* **Distributed –**Java is designed for the distributed environment of the internet.
* **Dynamic –**Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

**1.5 Introduction to Javascript**

JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and [Mobile](https://www.guru99.com/mobile-testing.html)application development.

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time. The language was initially called LiveScript and was later renamed JavaScript. There are many programmers who think that JavaScript and[Java](https://www.guru99.com/java-tutorial.html)are the same. In fact, **JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language**. The syntax of JavaScript is mostly influenced by the programming language C.

Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code**. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that **all modern web browsers support** JavaScript. So, you do not have to worry whether your site visitor uses Internet Explorer, Google Chrome, Firefox or any other browser. JavaScript will be supported. Also, JavaScript **runs on any operating system** including windows, linux or Mac. Thus, JavaScript overcomes the main disadvantages of[VBScript](https://www.guru99.com/vbscript-tutorials-for-beginners.html)which is limited to just IE and Windows.

**1.6 About Java Connectivity (JDBC Connections)**

**Java Database Connectivity(JDBC)** is an **Application Programming Interface(API)** used to connect Java application with Database. JDBC is used to interact with various type of Database such as Oracle, MS Access, My SQL and SQL Server. JDBC can also be defined as the platform-independent interface between a relational database and Java programming. It allows java program to execute SQL statement and retrieve result from database.

JDBC Driver is required to process SQL requests and generate result. The following are the different types of driver available in JDBC.

* **Type-1 Driver** or **JDBC-ODBC bridge**
* **Type-2 Driver** or **Native API Partly Java Driver**
* **Type-3 Driver** or **Network Protocol Driver**
* **Type-4 Driver** or **Thin Driver**

**CHAPTER 2**

**REQUIREMENTS ANALYSIS**

The requirement analysis specifies the requirements needed to develop a DBMS project. In this phase, we collect the requirements needed for building the project. The requirements collected are then analysed and carried to next phase.

**2.1 Software Requirements**

This document will outline the software design and specification for our application. The application is run on MicrosoftOperatingSystem.

Characteristics required for the interface between the software and each of the hardware components are:

* Any mouse and keyboard that is compatible with the OS.

Software required are:

* MySQL Command Line Client(Version 5.7)/ MySQLYOG(Version 5.02) – For creating database, tables, storage.
* Net beansIDE(Version 8.2) – for writing required codes, connectivity, designing, deploying and executing.

The application is very self-contained. Robust error handling will be implemented and code will be procedure oriented to allow for easier maintenance and feature additions.

**2.2 Hardware requirements**

There are no rigorous restrictions on the machine configuration. The model was made to work on Microsoft Operating System with following specifications:

* Processor: Intel® CORE™ i5-7200U CPU @ 2.50GHz 2.71GHz
* RAM: 8.00GB(7.89GB usable)
* System type: 64-bit operating system, x64 – based processor
* Keyboard: Standard.

**CHAPTER 3**

**ER DIAGRAM AND SCHEME DIAGRAM**

Providing

license

App

Admin

1 1

detection

Fig 3.1: ER Diagram

n

n

1

1

n

m

Gift verfication

ver

register

verification

Malicious

Dependent

proapp

Admin

Username password

App

Username appid appname license no appurl status appicon aid

Malicious

Mid malicious

Proapp

aprid username appid license no appurl app icon

Register

Uid username password email gender country phone no file

Fig 3.2: Schema Diagram

**CHAPTER 4**

**DESCRIPTION OF TABLES**

There are various tables included in the database to perform the required operations.

**4.1 Admin Table.**

This table contains the details of the admin.

The columns are:

* Username Varchar(10) – holds the name of the admin.
* Password Varchar(20) – holds the password with which the admin logs in.

**4.2 App Table**

This table contains details of the user, products, respective links, IDs.

The columns are:

* Username Varchar(200) – holds the name of the registered user.
* Appid Varchar(200) – holds the unique id of the products.
* AppName Varchar(200) – holds the name of the product.
* LicenseNo Varchar(200) – holds the license number generated after the admin provides license.
* AppUrl Varchar(200) – holds the urls of the respective websites.
* Status Varchar(200) – holds the status of the product, that is, whether licensed or unlicensed.
* AppIcon Varchar(200) – holds the image of the product.
* aid int(200) – holds the id number.

**4.3 Malicious Table**

This table contains the details of malicious websites and products.

The columns are:

* mid int(200) – holds the malicious id.
* Malicious varchar(200) – holds the malicious websites.x

**4.4 Proapp Table**

This table contains the updates after the user logs in to the database and views and adds the gifts.

The columns are:

* aprid int(200) – holds the general serial number generated when the user logs in. same user can have several aprids. Everytime an user logs in an apprid is generated.
* username varchar(200) – holds the name of the logged in user.
* appid varchar(200) – holds the id of the selected or added gift.
* appname varchar(200) - holds the name of the selected or added gift.
* licenseno varchar(200) - holds the status of the selected or added gift, that is, whether licensed or unlicensed.
* appurl varchar(200) - holds the url of the selected or added gift.
* appicon varchar(200) - holds the image of the selected or added gift.

**4.5 Register Table**

This table contains the details of the users given during their registration.

The columns are:

* uid int(10)- id generated when an user logs in. Its unique for each user.
* **username varchar(45) – holds the name of the registered user.**
* password varchar(45) – holds the password given by the user during registration. User will be further using this software to login.
* email varchar(45) - holds the email id of the registered user.
* gender varchar(45) - holds the gender of the registered user.
* country varchar(45) - holds the country of the registered user.
* phoneno varchar(45) - holds the phone number of the registered user.
* file varchar(300) - holds the profile picture of the registered user.

**4.6 Friendlist Table**

This table is used for further implementations. This table is updated when the user refers the website to his/her friends.

The columns are:

* fid int(200) – holds the friend id.
* rfrom varchar(200) – holds the name of the user who refers to the website to his friend.
* rto varchar(200)- holds the name of the friend to whom the user refers the website.
* status varchar(200) – holds the status of the request sent by the user, that whether accepted or rejected by the user.

**4.7 Message Table**

This table is also used for further implementation. It provides the chat option on the website.

The columns are:

* mid int(200) – holds the message id.
* msgfrom varchar(200) – holds the name of the sender of the message.
* msgto varchar(200) - holds the name of the receiver of the message.
* msg varchar(200) – holds the message.
* file varchar(200) – holds the images being shared.

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 MODULES**

**Modules used:**

* admin module:
* View Dealers and Customers.
* Approve New Dealers and Requests.
* Stock Maintenance
* secure customer info

* dealer module:
* New Dealer Signup.
* View items, Deals and Customers.
* Approve gift Requests.
* delivers items to admin.
* customer module:
* list of customers and their details
* Forward request for gifts to Dealer.
* Check Request status.
* items module:
* items present
* provides information about cost of the items
* brands

**5.2 Snippets with Description**

**i)Home Page**

index.jsp page contains the snippets for the home page.

* <div id="loginContainer"><a href="#" id="loginButton"><span> User Login</span></a> = it contains the code to insert and activate the login button.
* <label for="user">Username</label>

<input type="text" name="username" >

<label for="password">Password</label>

<input type="password" name="password" id="password">

It inserts the username and password given by the user during registration or during logging in.

* <input type="text" value="Enter email" onfocus="this.value='';" onblur="if (this.value == '') {this.value ='Enter email';}">

<input type="submit" value="SUBSCRIBE" >

It inserts the email id of the user for further utilization. It inserts the submit button after the entire registration.

**ii)Register Page**

register.jsp holds the the snippets for the register page.

* <label for="username">User Name</label>

<input type="text" name="username" required="" id="email">

Contains the user name label and user enters his/her name here during registering which they later use to login.

* <label for="password">Password</label>

<input type="password" name="password" required="" id="password>

Contains the password label and user enters his/her password here during registering which they later use to login.

* <label for="email">Email Address</label>

<input type="text" name="email" id="email”>

Contains the email label and user enters his/her email here during registering which is used for future reference.

* <label for="sex">Gender</label>

<select name="sex">

<option>

Male

</option>

<option>

Female

</option>

</select>

Contains the gender label and user enters his/her gender here during registering.

* <label for="country">Country</label>

<input type="text" name="country" id="country">

Contains the country label and user enters his/her country here during registering.

* <label for="phoneno">Mobile</label>

<input type="text" name="phoneno" id="phoneno">

Contains the phone no label and user enters his/her phone no here during registering.

* <label for="pic">Profile Pic</label>

<input type="file" name="file" id="file">

Contains the profile picture label and user enters his/her profile pic here during registering.

**iii)Userhome Page**

It shows the details filled by the user during registration.

* <li><a href="userhome.jsp" class="active hvr-sweep-to-bottom">Home</a></li>
* <li><a class="hvr-sweep-to-bottom" href="addapp.jsp">Add Gift</a></li>
* <li><a class="hvr-sweep-to-bottom" href="index.jsp">Log out</a></li>

It refers to other related pages to perform the required operations.

**iv)Add Gift Page**

addapp.jsp holds the the snippets for the add gift page.

* <button type="button" class="btn btn-1 btn-success"> <a href="addapplication.jsp">Add Gift to site</a></button></li>
* <button type="button" class="btn btn-1 btn-success"> <a href="viewapp.jsp">View Gifts</a></button></li>
* <button type="button" class="btn btn-1 btn-success"> <a href="myapp.jsp">View my Added Gifts</a></button>
* </<button type="button" class="btn btn-1 btn-success"> <a href="myproapp.jsp">View my profile Gifts</a></button></li>

It contains buttons that refer to the respective pages to perform the functionalities.

**a)Add Gift To Site** button: it refers to addapplication.jsp page. The user desired gifts can be added.

* <label for="Application ID">Gift ID</label> <input type="text" name="appid" id="appid">

The gift id is inserted.

* <label for="Application Name">Gift Name</label>

<input type="text" name="appname" id="appname">

The gift name is inserted.

* <label for="Application URL">Gift URL</label>

<input type="text" name="appurl" id="appurl">

The respective url of the gift is inserted.

* <label for="Application Icon">Gift Icon</label>

<input type="file" name="appicon" id="appicon">

The gift image is inserted.

**b)View Gifts** button: it refers to viewapp.jsp page. All the gifts present on the website can be viewed.

* String Request=rs2.getString("status");
* String appname=rs2.getString("appname");
* String licenseno=rs2.getString("licenseno");
* String appurl=rs2.getString("appurl");
* String appicon=rs2.getString("appicon");

The user can add any of his selected gifts to his profile.

**c)View my Added Gifts** button: it refers to myapp.jsp page. The list of the gifts added in user profile are shown.

<a href="apps/<%=appicon%>" class="b-link-stripe b-animate-go swipebox" width="100" height="100" title="Image Title"><img src="apps/<%=appicon%>" alt="" width="100" height="100" class="img-responsive">

**d)View my Profile gifts** button: it refers to myproapp.jsp page. It contains the the gift added by user to his profile. The user can visit the related site and buy the product.

<a href="apps/<%=appicon%>" class="b-link-stripe b-animate-go swipebox" width="200" height="200" title="Image Title"></a><img src="apps/<%=appicon%>

**v)Admin Page:** it refers to admin page. This page helps the admin to look into all the details of the website and provide required authentications.

* <li><a href="adminhome.jsp" class="active hvr-sweep-to-bottom">Admin Home</a></li>
* <li><a class="hvr-sweep-to-bottom" href="frappe.jsp">Gift Verification</a><li>
* <li><a class="hvr-sweep-to-bottom" href="users.jsp">View Users</a></li>
* <li><a class="hvr-sweep-to-bottom" href="apps.jsp">View All Gifts</a></li>
* **i)Gift Verification** button: it refers to frappe.jsp page. Shows the list of gifts available on the website. Admin views the gift details and decides whether to provide license or not.
* String appname=rs.getString(3);
* String appid=rs.getString(2);
* String appicon=rs.getString(7);

**ii)View Users** button: it refers to users.jsp page. Admin can view all the details entered by the user during registration.

* <td align="center"><%=rs.getString("uid")%> </td>
* <td align="center"><%=rs.getString("username")%> </td>
* <td align="center"><%=rs.getString("password")%></td>
* <td align="center"><%=rs.getString("email")%></td>
* <td align="center"><%=rs.getString("gender")%> </td>
* <td align="center"><%=rs.getString("country")%> </td>
* <td align="center"><%=rs.getString("phoneno")%> </td>

**iii)View All Gifts** button: it refers to apps.jsp page. It contains all the details about the gift and the users who have chosen it.

* <td align="center"><%=rs.getString("aid")%> </td>
* <td align="center"><%=rs.getString("username")%> </td>
* <td align="center"><%=rs.getString("appid")%></td>
* <td align="center"><%=rs.getString("appname")%></td>
* <td align="center"><%=rs.getString("licenseno")%> </td>
* <td align="center"><%=rs.getString("appurl")%> </td>
* <td align="center"><b><%=rs.getString("status")%></b> </td>
* <td> <img src="apps/<%=rs.getString("appicon")%>" width="100" height="100"/></td>

**vi)JDBC Connection:** the snippet is present in the page dbconnection.jsp. It is the connectivity between the front end and the backend.

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (Exception e) {

out.println(e);

}

try {

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fb", "root", "root");

}

catch (Exception ex) {

out.println(ex);

}

**CHAPTER 6**

**TESTING**

**Testing** is an investigation conducted to provide stakeholders with information about the quality of the **software** product or service under test.Testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding **software bugs** (errors or other defects), and verifying that the software product is fit for use.

Testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time,
* is sufficiently usable,
* can be installed and run in its intended environments, and
* achieves the general result its stakeholders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding **software bugs** (errors or other defects). The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones.

Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The  overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an agile approach, requirements, programming, and testing are often done concurrently.

**6.1 MODLUE TESTING**

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program.

Module testing is largely a white box oriented. The objective of doing Module, testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module.

Module testing allows to implement parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

Module Testing is recommended because

* Probability of identifying errors or bugs on smaller chunks of program becomes higher.
* Multiple modules can be tested simultaneously and hence supports parallel testing.
* Complexity of testing can be easily managed.

Strategies for module testing are:

* First, it combines all modules and then test the whole program
* Incremental method- each module is tested first and then gradually incremented to the tested collection. It does a step wise retesting
* Incremental Testing, there are two approaches – Top down and Bottom Up testing
* To execute the module with the selected data, it requires a **driver**For Module Testing, designing a Test Case is an important segment. While designing test cases for a module test, a tester has to take two things into consideration.
  + Specification for the module
  + The module's source code
* Analyze the module's logic by using one or more of the white box methods, and then supplement these test cases by applying black box methods to the modules specification
* Once the test case is designed, the next step is to combine the module for testing. For this, the method used is either an **Incremental or non-Incremental method**.
* Non-incremental method- all modules are tested independently for supplying the test data, monitoring the execution and capturing the results.

**6.2 INTEGRATION TESTING**

In integration Testing, individual software modules are integrated logically and tested as a group.

A typical software project consists of multiple software modules, coded by different programmers.

 integration Testing focuses on checking data communication amongst these modules.

Hence it is also termed as **'I & T'** (Integration and Testing), **'String Testing'** and sometimes 'Thread Testing'.

Reasons to choose Integration Testing are:

* A Module in general is designed by an individual software developer whose understanding and programming logic may differ from other programmers. integration Testing becomes necessary to verify the software modules work in unity.
* At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary.
* Interfaces of the software modules with the database could be erroneous.
* External Hardware interfaces, if any, could be erroneous
* Inadequate exception handling could cause issues.

Integration test case differs from other test cases in the sense it**focuses mainly on the interfaces & flow of data/information between the modules**. Here priority is to be given for the **integrating links** rather than the unit functions which are already tested.

## Approaches/Methodologies/Strategies of Integration Testing:

* Big Bang Approach .
* Incremental Approach: which is further divided into following
  + Top Down Approach.
  + Bottom Up Approach.
  + Sandwich Approach - Combination of Top Down and Bottom Up.

**i)Big Bang Approach**

Here all component are integrated together at **once**, and then tested.

**Advantages:**

* Convenient for small systems.

**Disadvantages:**

* Fault Localization is difficult.
* Given the sheer number of interfaces that need to be tested in this approach, some interfaces links to be tested could be missed easily.
* Since the integration testing can commence only after "all" the modules are designed, testing team will have less time for execution in the testing phase.
* Since all modules are tested at once, high risk critical modules are not isolated and tested on priority. Peripheral modules which deal with user interfaces are also not isolated and tested on priority.

**ii)Incremental Approach**

In this approach, testing is done by joining two or more modules that are ***logically related***. Then the other related modules are added and tested for the proper functioning. Process continues until all of the modules are joined and tested successfully.

This process is carried out by using dummy programs called **Stubs and Drivers**. Stubs and Drivers do not implement the entire programming logic of the software module but just simulate data communication with the calling module.

**Stub**: Is called by the Module under Test.

**Driver**: Calls the Module to be tested.

Incremental Approach in turn is carried out by two different Methods:

**a)Bottom Up**

**b)Top Down**

**a)Bottom Up Integration**

In the bottom up strategy, each module at lower levels is tested with higher modules until all modules are tested. It takes help of Drivers for testing.

**Advantages:**

* Fault localization is easier.
* No time  is wasted waiting for all modules to be developed unlike Big-bang approach.

**Disadvantages:**

* Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects.
* Early prototype is not possible.

**b)Top Down Integration**

In Top to down approach, testing takes place from top to down following the control flow of the software system. Takes help of stubs for testing.

**Advantages:**

* Fault Localization is easier.
* Possibility to obtain an early prototype.
* Critical Modules are tested on priority; major design flaws could be found and fixed first.

**Disadvantages:**

* Needs many Stubs.
* Modules at lower level are tested inadequately.

Integration Testing procedure are:

1. Prepare the Integration Tests Plan
2. Design the Test Scenarios, Cases, and Scripts.
3. Executing the test Cases followed by reporting the defects.
4. Tracking & re-testing the defects.
5. Steps 3 and 4 are repeated until the completion of Integration is successfully.

**CHAPTER 7**

**SNAPSHOTS**

In computer file systems, a snapshot is a copy of a set of files and directories as they were at a particular point in the past. The term was coined as an analogy to that in photography.

One approach to safely backing up live data is to temporarily disable write access to data during the backup, either by stopping the accessing applications or by using the locking API provided by the operating system to enforce exclusive read access. This is tolerable for low-availability systems (on desktop computers and small workgroup servers, on which regular downtime is acceptable). High-availability 24/7systems, however, cannot bear service stoppages.

To avoid downtime, high-availability systems may instead perform the backup on a snapshot—read-only copies of the data set frozen at a point in time—and allow applications to continue writing to their data. Most snapshot implementations are efficient and can create snapshots in O (1). In other words, the time and I/O needed to create the snapshot does not increase with the size of the data set, whereas the same for a direct backup is proportional to the size of the data set. Snapshots are the pictures representing different phases of the program execution.

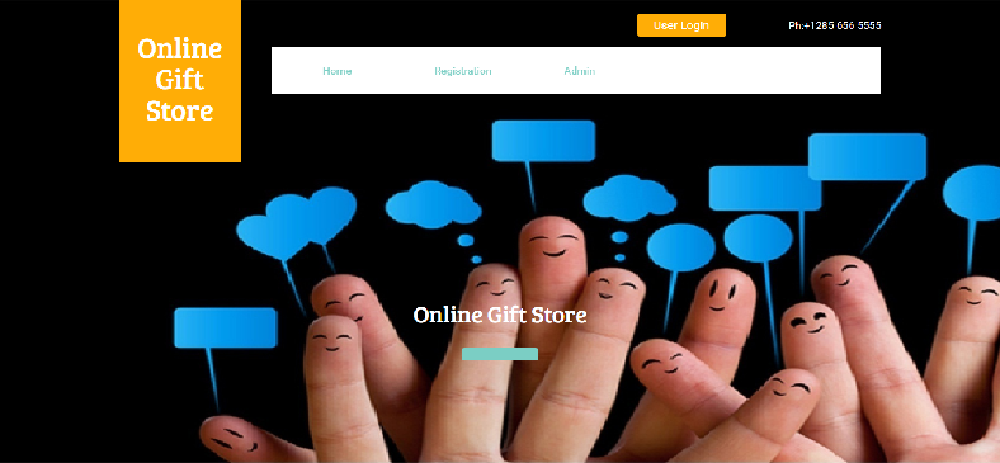


Fig 7.1: Home Page

The above figure shows the home page of the website. This the page that appears to the user when he/she logs in. This is the first/welcome/introduction page of the website.

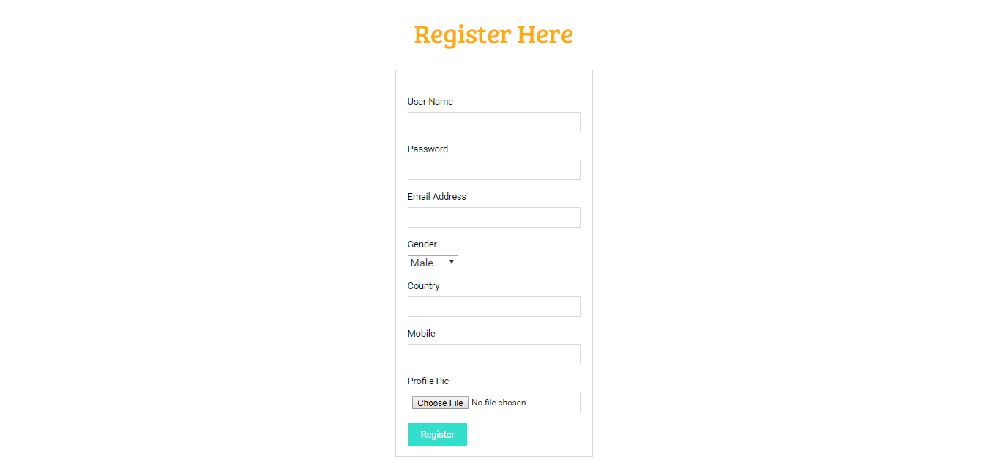


Fig 7.2:Registration Form

The above figure shows the registration form. Here the user fills the user details to register to the website.

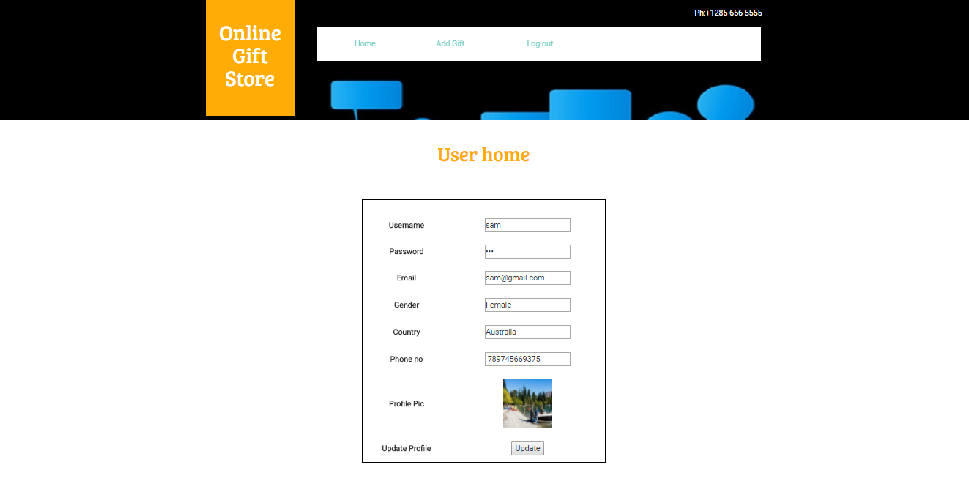


Fig 7.3:User Home Page

When user logs in to website using the user name and password, the following page appears. It first displays the registration form filled by the user. It further contains the sections that the user can select to look into the products.

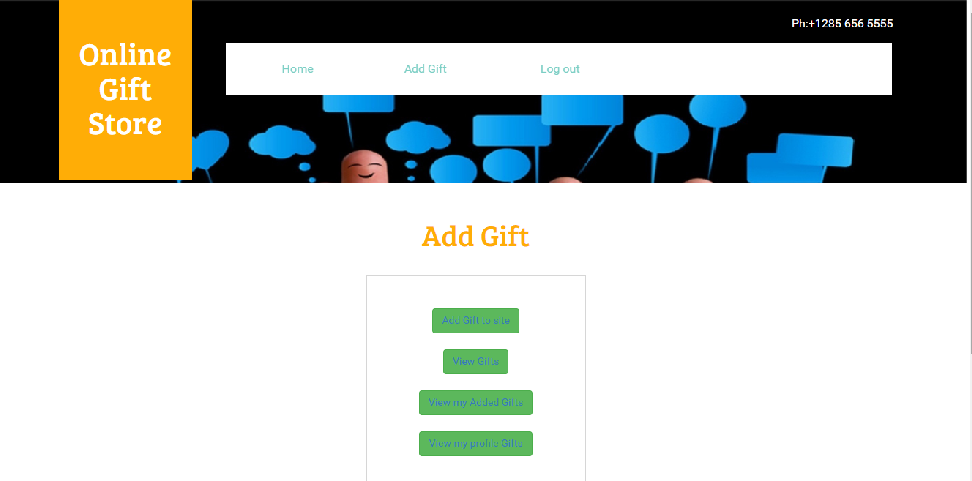


Fig 7.4: Add Gift Page

When the user clicks the add gift button, the following page appears. Here the user can again select the required buttons to view and select the products and purchase them.

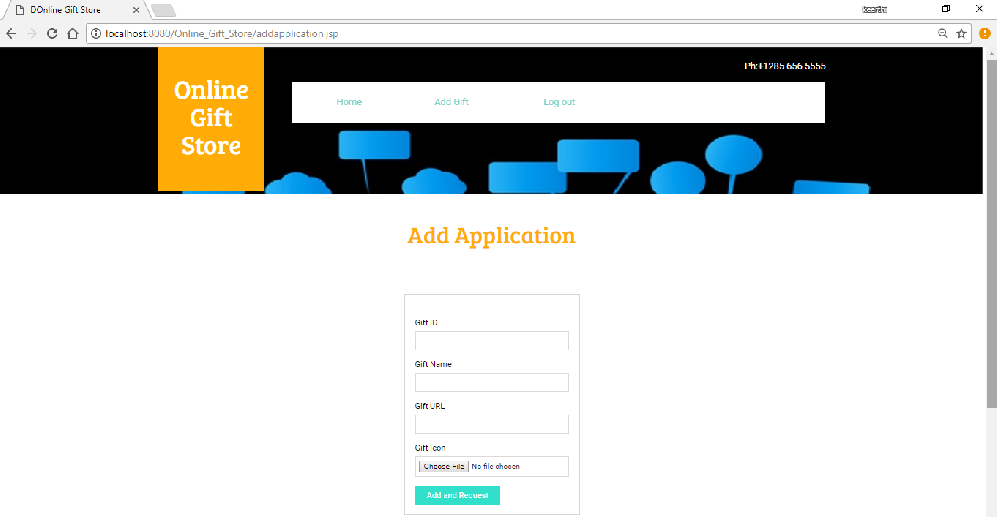


Fig:7.5: Add Aplication Page

When the user clicks the add gift to site button, the user enters to the following page. Here the user can added his desired gift to site by filling the related details of the product.

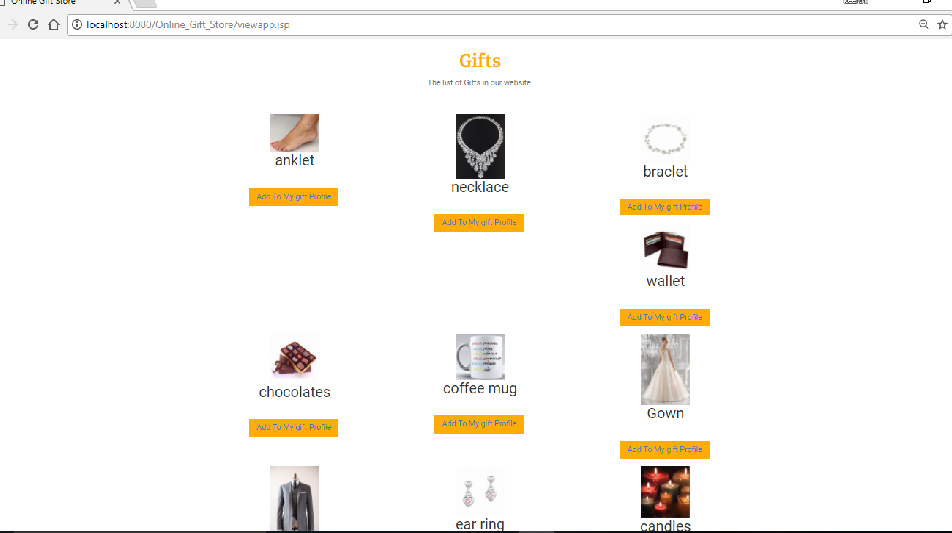


Fig 7.6: Gifts Page

When the user clicks the view gifts button the user enters, to the following page. Here the user can view all the gifts present on the website and choose any desired gift and add the desired gift to the user profile.

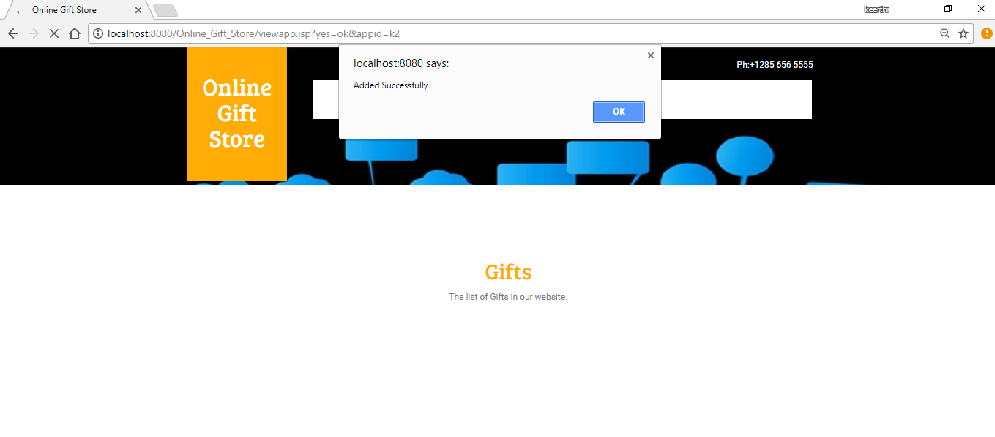


Fig 7.6.1: Gift Addition

User’s gift is added successfully, since the gift is verified.

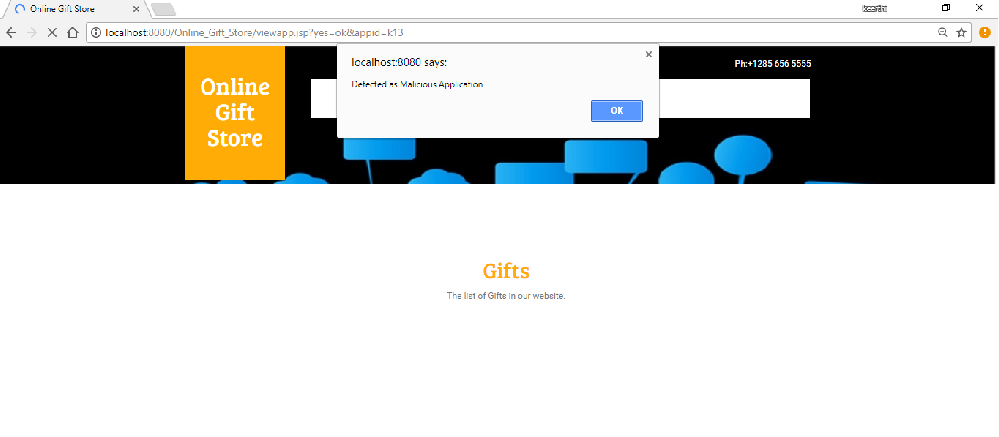


Fig 7.6.2 Gift Not Added

In the above figure, the user is not allowed to add the desired gift as the admin has given license to the gift.

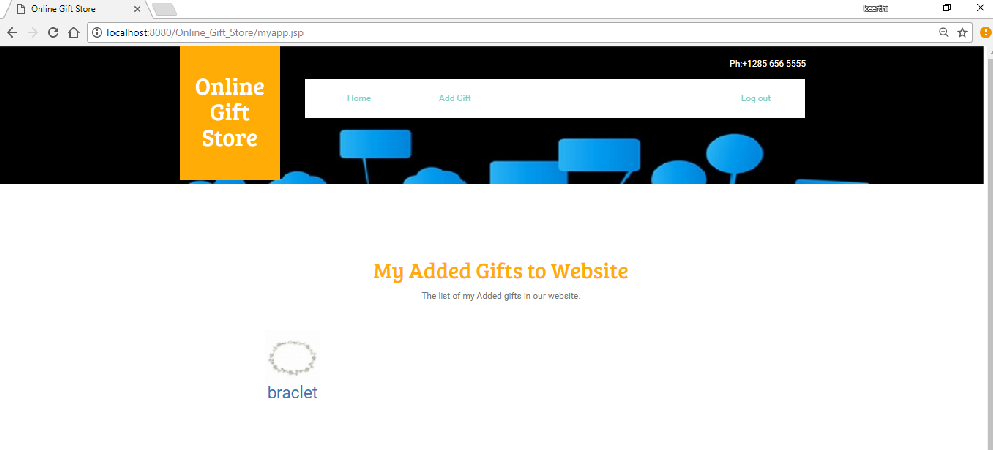


Fig 7.7: Added Gifts

When the user clicks the view my added gifts button, the user will enter to the following page. Here the gift added by the user is displayed.

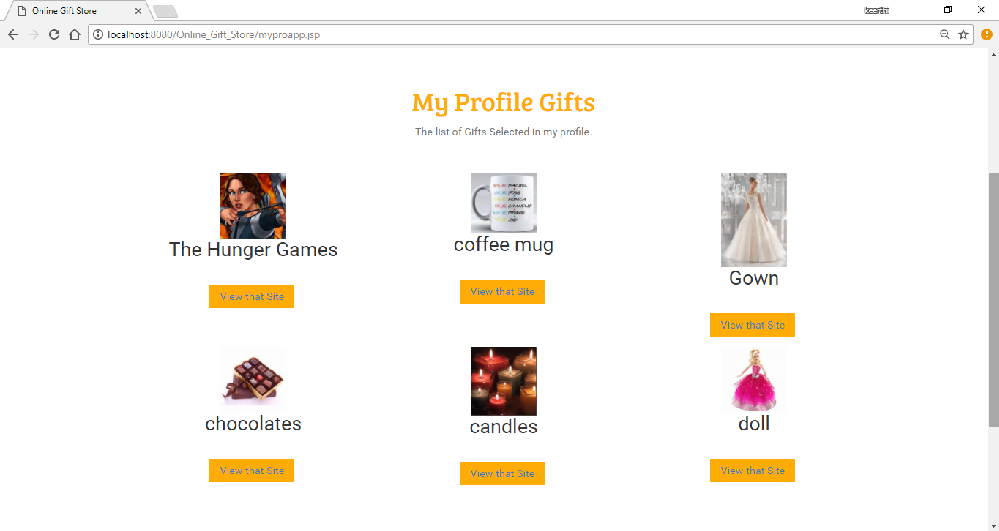


Fig:7.8 Profile Gifts

All the gifts chosen and added by the user are displayed in this page. The above figure shows some of the added gifts by the user. Later the user can visit the related site and buy the gift.

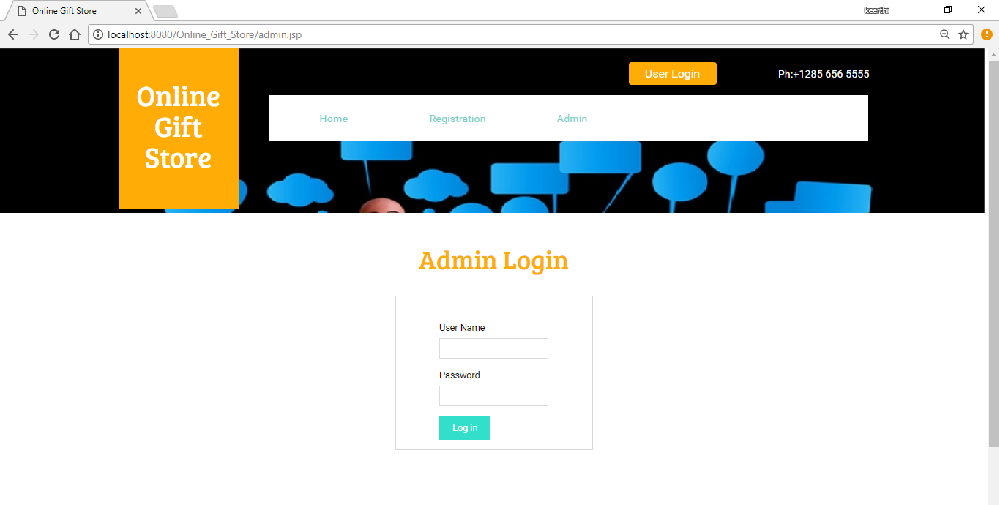


Fig:7.9 Admin page

The above figure shows the admin login page, where the admin enters his username and password and logs in.

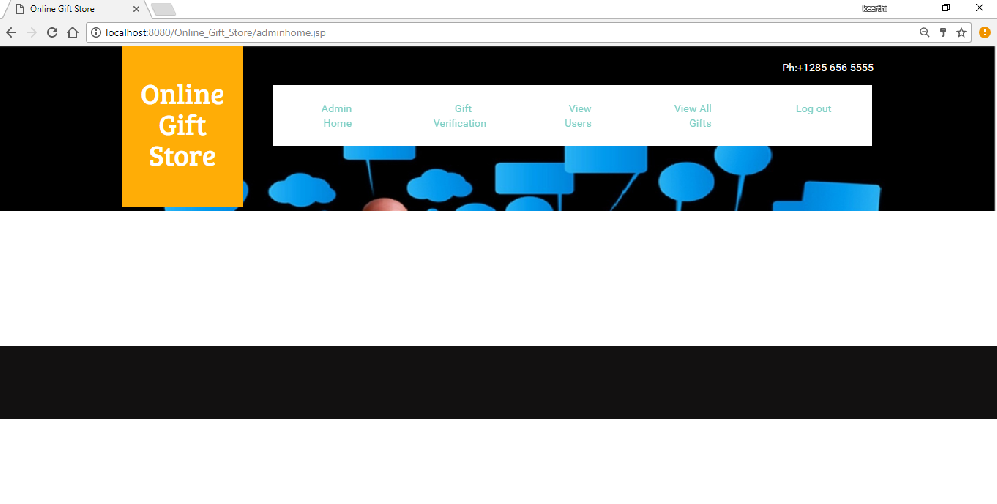


Fig 7.10: Admin Page

The above figure shows the admin page after the admin logs in. it contains various sections for the admin to perform various operations.

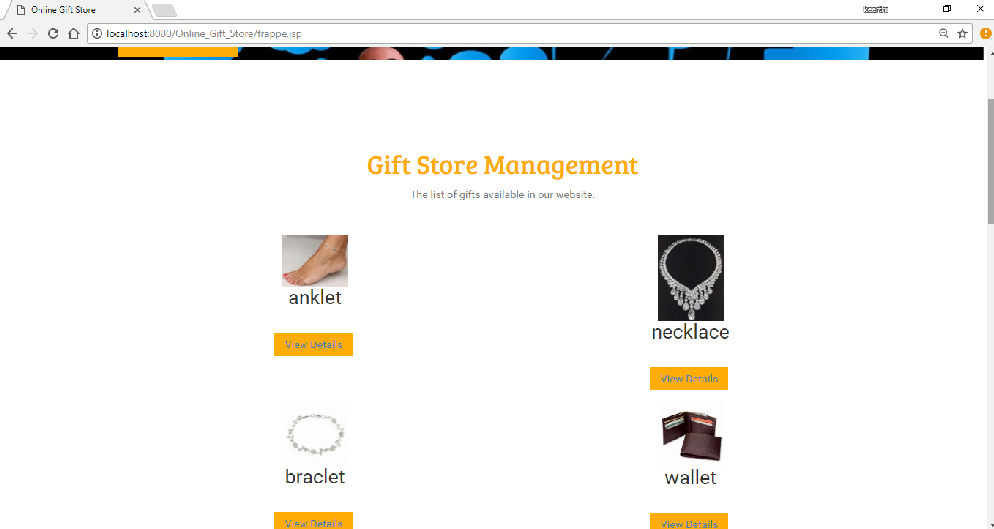


Fig 7.11:Gift Verification Page

Here the admin looks into the details of the products and considers the requests sent by the users for authentication. The admin decides whether to give license or not, on checking all the criterias.

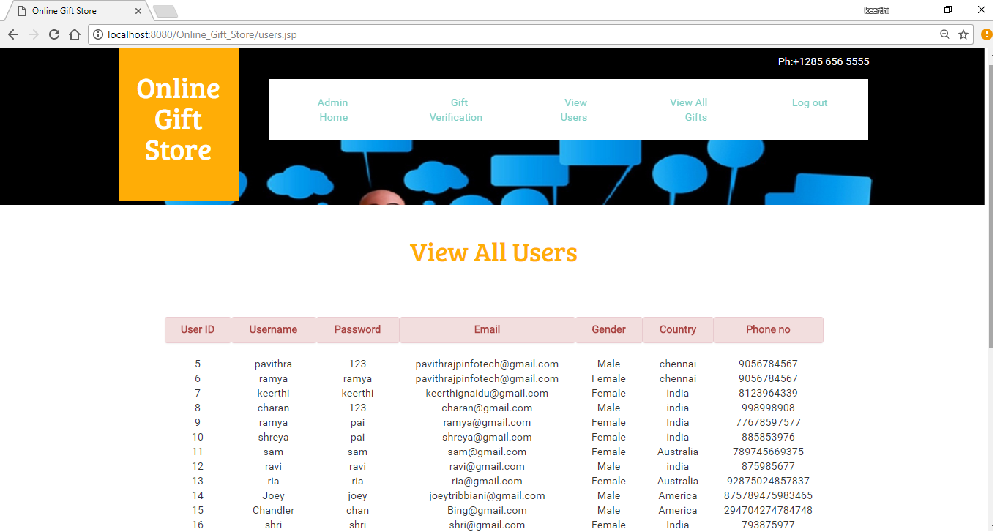


Fig 7.12:View Users Page

When the admin clicks the view users button, the admin enters to the following page. Here the admin can view all the details of all the users entered while they register.

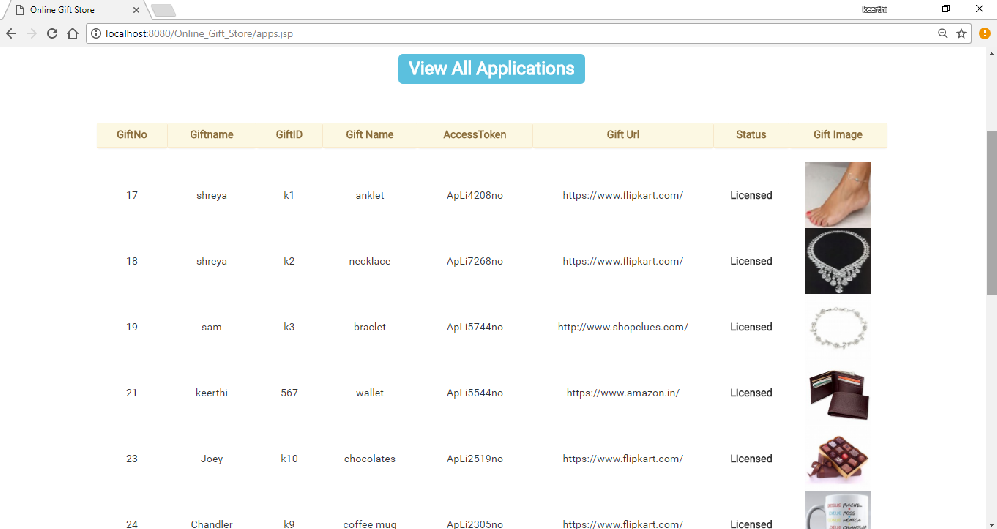


Fig 7.13: View applications.

When the admin clicks the view all gifts button, the admin enters to the following page. Here the admin can check the status of the products, user choices,license ids.

**CONCLUSION**

This DBMS project “ONLINE GIFT STORE MANAGEMENT SYSTEM” contains the database for an online gift store. This website helps the users to shop more easily and efficiently. While selecting a gift everybody would want to choose the best. Gifts are expected to be unique and adorable. This indeed means comparision with other gifts. While comparing many aspects come into picture like quility, cost, availability, originality etc. In this website the admin helps the user do most of the job.

Malicious products are a great problem in online shopping. Damaged and malicious products are always a threat. The admin makes sure that the users donot get acces to any such products. Hence perfect products are a gurentee that the admin provides through this website. The user is also given exposure to other products and newer websites for shopping other things, it needed be like that user needs to focus on only his gifts, he can look into others also and buy those too if he is intrested.

**FUTURE ENHANCEMENTS**

The currently developed project can improve in many areas. Time constraint and deficient knowledge has been a major factor in limiting the features offered by this display model.

The following features were thought and will be implemented to enhance the project.

* Two extra tables have been added:
* **Friendlist table :**

The user can suggest the website to his other friends. This makes a better profit to the website and could also be one of the marketing strategies to make the website popular.On the other hand many other people also will be benifited by using the website.

* **Message table:**

Online chats could be another efficient enhancement. It could be the means of communication between the user and admin, the user and admin can connect well and choosing becomes easier. One user can also chat with another.

* More details of the product can be provided, rather than just providing the related link.

**REFERENCES**

1. Fundamental of DATABASE SYSTEMS(7th edition) by Ramez Elmasri and Shamkant. B. Navathe.
2. Software Engineering (9th Edition) by Ian Sommervale.
3. Wikipedia.
4. W3 Schools.
5. Google.
6. Youtube.