

Mini Project Report

On

**ONLINE SHOPPING  
SYSTEM**

*Submitted By:*

**A NOORJAHAN BEGUM(R170573)  
T MEENA(R170572)  
T JYOTHSNA(R170829)  
M BINDU MANASA(R170496)**

*In partial fulfillment for the award of the degree of*

**B. TECH DEGREE**

*In*

**COMPUTER SCIENCE & ENGINEERING  
RGUKT  
RKVALLEY**

## **Division of Computer Science**

**RGUKT**

**RKVALLEY**

### **CERTIFICATE**

*Certified that this is a bonafide record of the project work titled*

***ONLINE SHOPPING SYSTEM***

*Done by*

**A NOORJAHAN BEGUM(R170573)**

**T MEENA(R170572)**

**T JYOTHSNA(R170829)**

**M BINDU MANASA(R170496)**

*of II semester Computer Science & Engineering in the year 2022 in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science & Engineering of RGUKT, RKVALLEY.*

***B LINGAM MURTHY***

*Project Guide*

***P HARINADH***

*Head of the Division*

## **ABSTRACT**

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipcart or ebay. Since the application is available in the Smartphone it is easily accessible and always available.

## LIST OF FIGURES

1.1 ADMIN MODULE	5
1.2 MANAGE MODERATORS	6
1.3 MANAGE PRODUCTS	8
1.4 MANAGE USERS	9
1.5 MANAGE ORDERS	10
1.5 MODERATOR MODULE	11
1.7 USER MODULE	12
<b>E-R DIAGRAMS &amp; DFD</b>	
3.5 LOGIN	30
3.6 USER DETAILS	30
3.7 PRODUCT DETAILS	31
3.8 PRODUCT ORDERS	31
3.9 COMPLETE DIAGRAM	32

3.10 LOGIN DFD	35
3.11 REGISTRATION DFD	36
3.12 ADMIN DFD	37
3.13 MODERATOR DFD	38

## LIST OF TABLES

3.1 LOGIN TABLE	28
3.2 USER DETAILS	28
3.3 PRODUCT DETAILS	29
3.4 PRODUCT ORDERS	29

## ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, Mrs. **Lingam Murthy**, for providing me with the right guidance and advice at the crucial junctures and for showing me the right way. I extend my sincere thanks to our respected **Head of the division Mr. Harinadh**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

A NOORJAHAN BEGUM(R170573)  
T MEENA(R170572)  
T JYOTHSNA(R170829)  
M BINDU MANASA(R170496)

## TABLE OF CONTENTS

ABSTRACT	i
LIST OF FIGURE	ii
LIST OF TABLES	iv
ACKNOWLEDGEMENT	v
<b>1. INTRODUCTION</b>	<b>PAGE</b>
1.1 PROJECT OBJECTIVES	1
1.2 PROJECT OVER VIEW	2
1.3 PROJECT SCOPE	2
1.4 STUDY OF SYSTEMS	3
1.4.1 MODULES	3
1.4.1.1 ADMIN	3
1.4.1.2 MODERATOR	11
1.4.1.3 USER	12
<b>2. SYSTEM ANALYSIS</b>	<b>15</b>
2.1 EXISTING SYSTEM	15
2.2 PROPOSED SYSTEM	16
2.3 SYSTEM REQUIREMENT SPECIFICATION	16
2.3.1 GENERAL DESCRIPTION	16



2.3.2 SYSTEM OBJECTIVES	17
2.3.3 SYSTEM REQUIREMENTS	17
2.3.3.1 NON FUNCTIONAL REQUIREMENT	18
2.3.3.2 FUNCTIONAL REQUIREMENT	19
<b>3. SYSTEM DESIGN</b>	<b>23</b>
3.1 INPUT AND OUTPUT DESIGN	24
3.1.1 INPUT DESIGN	24
3.1.2 OUTPUT DESIGN	24
3.2 DATABASE	25
3.3 SYSTEM TOOLS	25
3.3.1 FRONT END	26
3.3.2 BACK END	27
3.4 TABLES	28
3.5 E-R DIAGRAMS	30
3.6 DATA FLOW DIAGRAMS (DFD)	32
3.7 SCREEN SHOTS	39

3.8 SAMPLE CODE	47
-----------------	----

4. CONCLUSION	67
---------------	----

REFERENCES	69
------------	----

## **CHAPTER 1**

# **INTRODUCTION**

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform.

Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop

### **1.1 PROJECT OBJECTIVE:**

The objective of the project is to make an application in android platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

## **1.2 PROJECT OVER VIEW:**

The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. The information pertaining to the products are stores on an RDBMS at the server side (store).

The Server process the customers and the items are shipped to the address submitted by them. The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information pertaining to the articles and those of the customers. The end user of this product is a departmental store where the application is hosted on the web and the administrator maintains the database. The application which is deployed at the customer database, the details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various levels of users. Once the authorized personnel feed the relevant data into the system, several reports could be generated as per the security.

## **1.3 PROJECT SCOPE:**

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24\*7 and a home delivery system which can make customers happy.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipcart or ebay. Since the application is available in the Smartphone it is easily accessible and always available.

## **1.4 STUDY OF THE SYSTEM**

### *1.4.1 MODULES:*

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

- Administrator
- Moderators
- Users

#### **1.4.1.1 ADMINISTRATOR:**

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the shop. The administrator has all the information about all the users and about all products.

This module is divided into different sub-modules.

1. Manage Moderators
2. Manage Products
3. Manage Users
4. Manage Orders

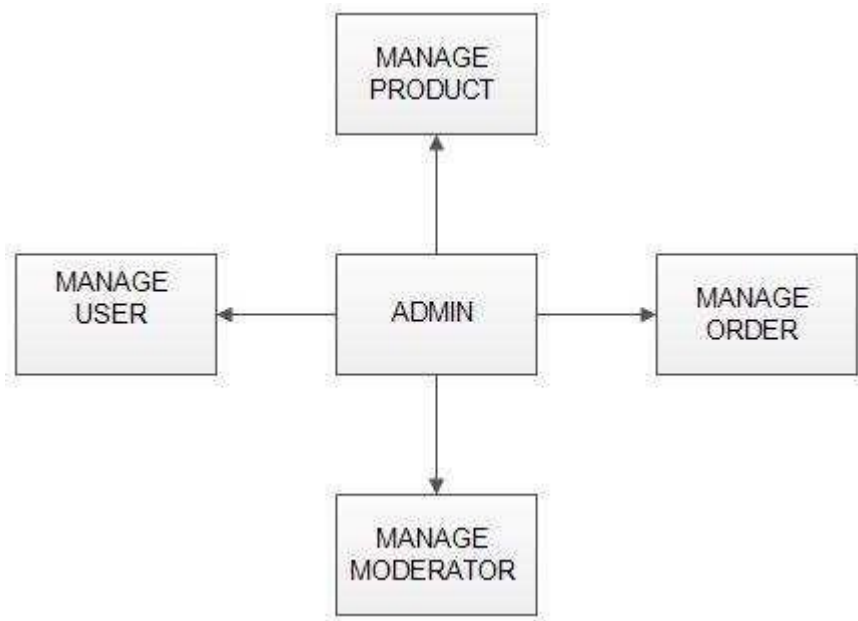


Fig 1.1: Admin module

## **MANAGE MODERATOR**

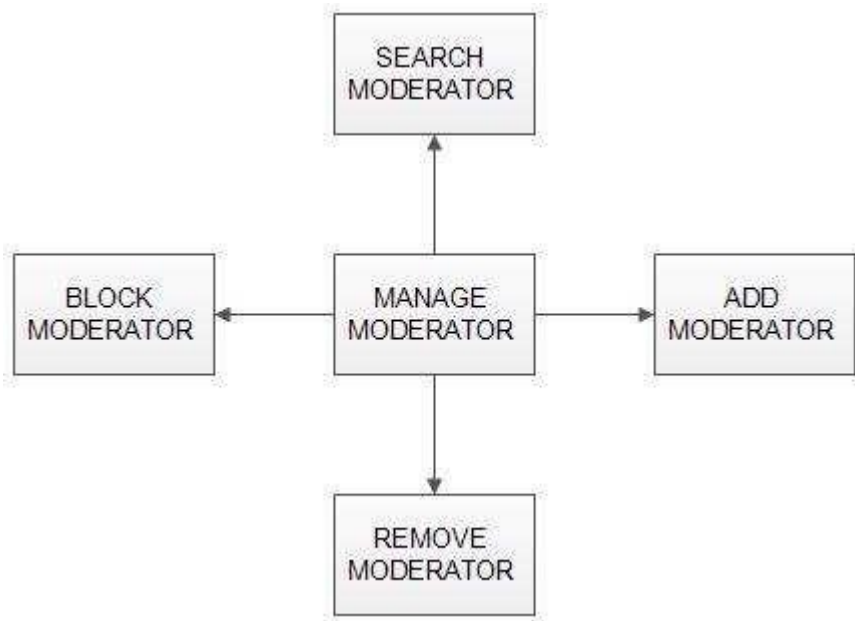


Fig 1.2: Manage Moderator

➤ Add Moderator

Only admin is having the privilege to add a moderator. A moderator can be considered as a staff who manages the orders or owner of a group of products.

➤ Block moderator



Admin can restrict a moderator from managing the orders by blocking them. Admin can unblock a blocked user if needed.

➤ Remove Moderator

Admin has privilege to delete a moderator who was added.

➤ Search moderator:

All existing moderators can be viewed by the administrator as a list. If there is number of moderators and admin need to find one of them, the admin can search for a moderator by name.

## **MANAGE PRODUCTS**

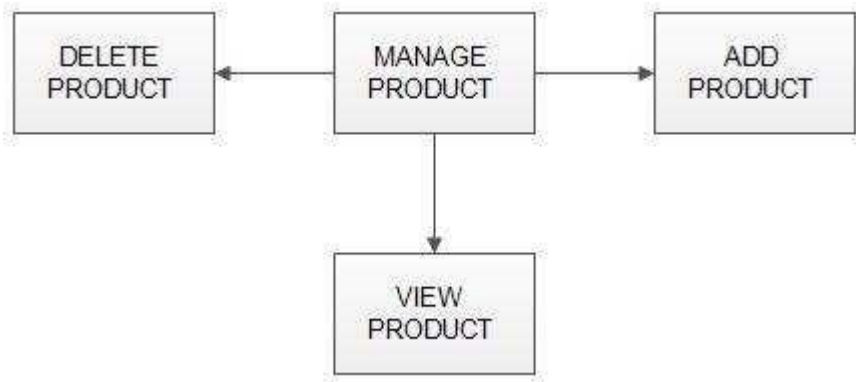


Fig 1.3: Manage Products

➤ Add Products

The shopping cart project contains different kind of products. The products can be classified into different categories by name. Admin can add new products into the existing system with all its details including an image.

➤ Delete Products

Administrator can delete the products based on the stock of that particular product.

➤ Search products

Admin will have a list view of all the existing products. He can also search for a particular product by name.

### **MANAGE USER**

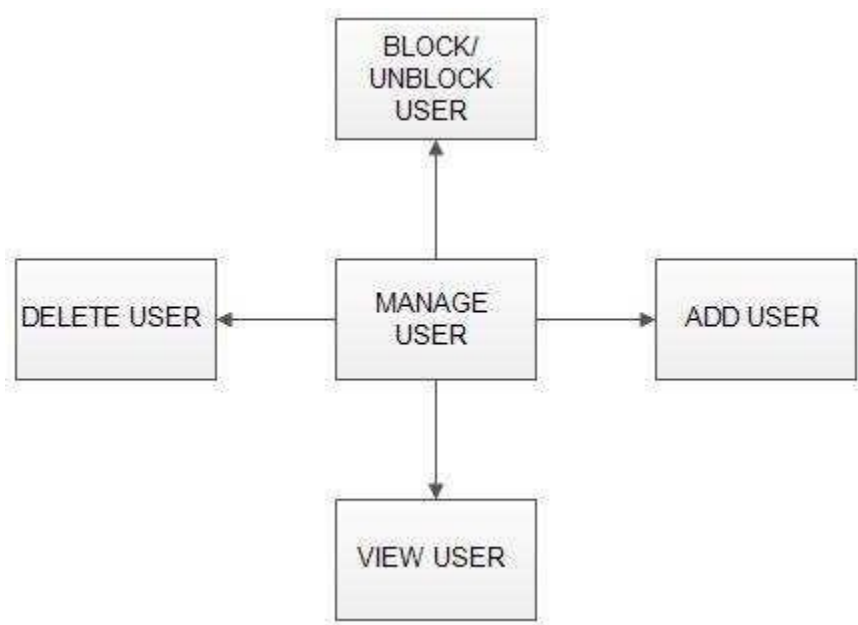


Fig 1.4: Manage User

➤ View Users

The admin will have a list view of all the users registered in the system. Admin can view all the details of each user in the list except password.

➤ Add Users

Admin has privileges to add a user directly by providing the details.

➤ Delete &Block Users

Administrator has a right to delete or block a user. The default status of a new user registered is set as blocked. The admin must accept the new user by unblocking him.

## **MANAGE ORDERS**



Fig 1.5: Manage Orders

➤ View Order

Administrator can view the Orders which is generated by the users. He can verify the details of the purchase.

➤ Delete order

Admin can delete order from the orders list when the product is taken for delivery.

#### 1.4.1.2 MODERATORS

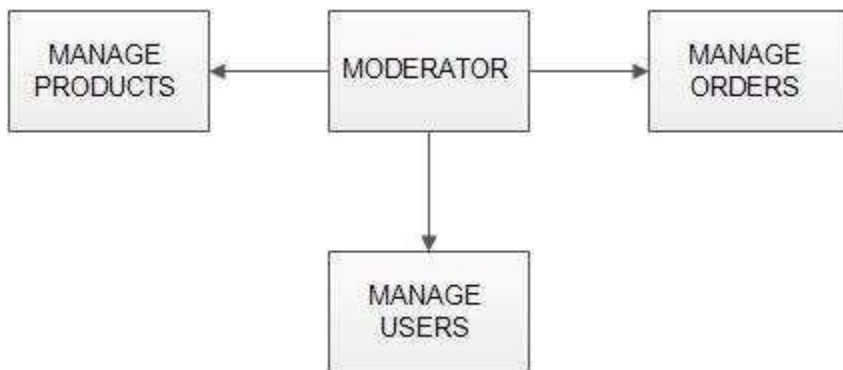


Fig 1.6: Moderator Module

A moderator is considered as a staff who can manage orders for the time being. As a future update moderator may give facility to add and manage his own products . Moderators can reduce the work load of admin. Now moderator has all the privilege an admin having except managing other

moderators. He can add products and users. He can also check the orders and edit his profile.

- Manage products
- Manage users
- Manage orders

#### 1.4.1.3 USERS

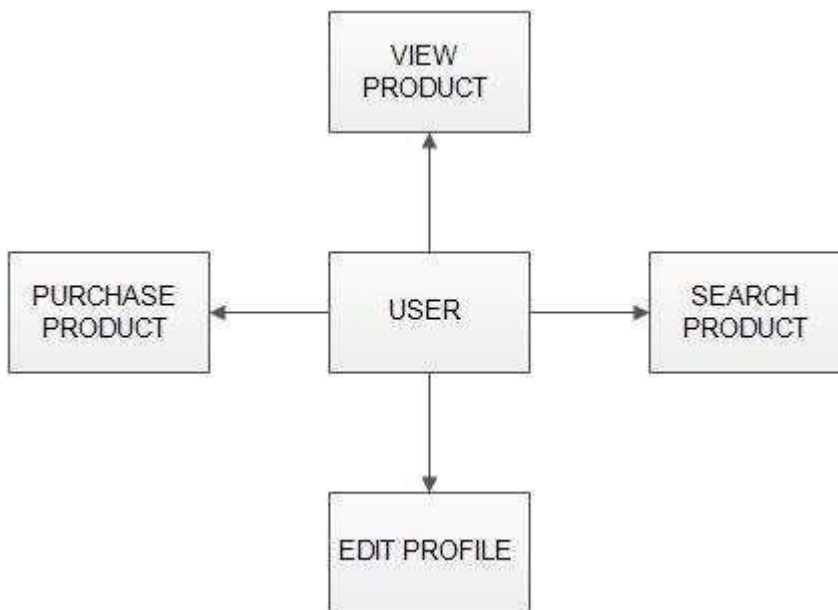


Fig 1.7: User Module

- Registration

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

➤ Login

A user must login with his user name and password to the system after registration.

➤ View Products

User can view the list of products based on their names after successful login. A detailed description of a particular product with product name, products details, product image, price can be viewed by users.

➤ Search Product

Users can search for a particular product in the list by name.

➤ Add to cart:

The user can add the desired product into his cart by clicking add to cart option on the product.

He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.

➤ Submit Cart:

After confirming the items in the cart the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

➤ History

In the history the user will have a view of pending orders.

➤ Edit Profile

The user can view and edit the profile.



## **CHAPTER 2**

# **SYSTEM ANALYSIS**

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

### **2.1 EXISTING SYSTEM**

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item .

1. It is less user-friendly.

2. User must go to shop and select products.
3. It is difficult to identify the required product.
4. Description of the product limited.
5. It is a time consuming process
6. Not in reach of distant users.

## **2.2 PROPOSED SYSTEM**

In the proposed system customer need not go to the shop for buying the products. He can order the product he wish to buy through the application in his Smartphone. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products.

## **2.3 SYSTEM REQUIREMENT SPECIFICATION**

### ***2.3.1 GENERAL DESCRIPTION***

#### **Product Description:**

The system consists of two parts .A web application which can provide the online shopping service and an android application for the customer to

access the web service from his Smartphone. Web application should be able to help the customer for selecting his item and to help the owner in managing the orders from the customers.

### Problem Statement:

As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. For competing with those online brands, If shops are providing an online portal where their customers can shop through internet and get the products at their doors it will increase the number of customers.

### *2.3.2 SYSTEM OBJECTIVES*

- To provide an android application for online shopping of products in an existing shop.
- To provide a online shopping web site for the same shop.

### *2.3.3 SYSTEM REQUIREMENTS*

### 2.3.3.1 NON FUNCTIONAL REQUIREMENTS

#### i. EFFICIENCY REQUIREMENT

When an online shopping cart android application implemented customer can purchase product in an efficient manner.

#### ii. RELIABILITY REQUIREMENT

The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.

#### iii. USABILITY REQUIREMENT

The android application is designed for user friendly environment and ease of use.

#### iv. IMPLEMENTATION REQUIREMENT

Implementation of the system using css and html in front end with jsp as back end and it will be used for database connectivity. And the database part is developed by mysql. Responsive web designing is used for making the website compatible for any type of screen.

#### v. DELIVERY REQUIREMENT

The whole system is expected to be delivered in four months of time with

a weekly evaluation by the project guide.

### 2.3.3.2 FUNCTIONAL REQUIREMENTS

#### **USER**

##### ➤ USER LOGIN

##### Description of feature

This feature used by the user to login into system. A user must login with his user name and password to the system after registration. If they are invalid, the user not allowed to enter the system.

##### Functional requirement

- Username and password will be provided after user registration is confirmed.
- Password should be hidden from others while typing it in the field

##### ➤ REGISTER NEW USER

##### Description of feature

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

#### Functional requirement

- System must be able to verify and validate information.
- The system must encrypt the password of the customer to provide security.
- PURCHASING AN ITEM

#### Description of feature

The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove. After confirming the items in the cart the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

#### Functional requirement

- System must ensure that, only a registered customer can purchase items.

## **ADMIN**

### ➤ MANAGE USER

#### Description of feature

The administrator can add user, delete user, view user and block user.

### ➤ MANAGE MODERATOR

#### Description of feature

The administrator can add moderator, delete moderator, block moderator and search for a moderator.

### ➤ MANAGE PRODUCTS

#### Description of feature

The administrator can add product, delete product and view product.

### ➤ MANAGE ORDERS

#### Description of feature

The administrator can view orders and delete orders.

#### Functional requirements

-The system must identify the login of the admin.

-Admin account should be secured so that only owner of the shop can access that account

## **MODERATOR**

### Description of features

A moderator is considered as a staff who can manage orders for the time being. As a future update moderator may give facility to add and manage his own products. Moderators can reduce the work load of admin. Now moderator has all the privilege of an admin having except managing other moderators. He can manage users and manage products. He can also check the orders and edit his profile.

### Functional requirement

-The system must identify the login of a moderator.



## **CHAPTER 3**

# **SYSTEM DESIGN**

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. It emphasizes on translating design specifications to performance specification. System design has two phases of development

- Logical design
- Physical design

During logical design phase the analyst describes inputs (sources), outputs (destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify

exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform

necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

### **3.1 INPUT AND OUTPUT DESIGN**

#### **3.1.1 INPUT DESIGN:**

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

#### **3.1.2 OUTPUT DESIGN:**

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

### **3.2 DATABASE**

#### **DATABASE DESIGN:**

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- ☐ Primary Key- the field that is unique for all the record occurrences.
- ☐ Foreign Key-the field used to set relation between tables.

Normalization is a technique to avoid redundancy in the tables.

### **3.3 SYSTEM TOOLS**

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

### 3.3.1.FRONT END:

JSP, HTML, CSS, JAVA SCRIPT, ANDROID are utilized to implement the frontend.

#### Java Server Pages (JSP)

Different pages in the applications are designed using jsp. A Java Server Pages component is a type of Java servlet that is designed to fulfil the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands. Using JSP, one can collect input from users through web page.

#### HTML (Hyper Text Markup Language)

HTML is a syntax used to format a text document on the web.

#### CSS (Cascading Style Sheets)

CSS is a style sheet language used for describing the look and formatting of a document written in a markup language.

#### Java Script

JS is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side

scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

Java Script is used to create pop up windows displaying different alerts in the system like “User registered successfully”, ”Product added to cart” etc.

### Android

The application is delivered to customer through an android application. So android platform is used to develop the user application.

### 3.3.2 BACK END

The back end is implemented using MySQL which is used to design the databases.

### MySQL

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language.

An application software called Navicat was used to design the tables in MySQL.

### 3.4 TABLES

#### ➤ LOGIN

login					
Field Name	Field Type	Default	AllowNull	PriKey	Extra
uid	int(11) FIRST		NO	YES	auto_increment
uname	varchar(50) AFTER `uid`		NO	NO	
upass	varchar(50) AFTER `uname`		NO	NO	
utype	varchar(20) AFTER `upass`		NO	NO	
enabled	varchar(10) AFTER `utype`		NO	NO	

Fig 3.1: Login Table

#### ➤ USER DETAILS

userdetails					
Field Name	Field Type	Default	AllowNull	PriKey	Extra
uid	int(11) FIRST		NO	YES	
uloginname	varchar(50) AFTER `uid`		NO	NO	
uemailid	varchar(100) AFTER `uloginname`		NO	NO	
umobno	varchar(30) AFTER `uemailid`		NO	NO	

Fig 3.2: User Details Table

➤ PRODUCT DETAILS

product_details					
Field Name	Field Type	Default	AllowNull	PriKey	Extra
id	int(11) FIRST		NO	YES	auto_increment
name	varchar(100) AFTER `id`		NO	NO	
description	varchar(500) AFTER `name`		YES	NO	
price	varchar(50) AFTER `description`		YES	NO	
photo	varchar(100) AFTER `price`		YES	NO	

Fig 3.3: Product Details Table

➤ PRODUCT ORDERS

product_orders					
Field Name	Field Type	Default	AllowNull	PriKey	Extra
id	int(11) FIRST		NO	YES	auto_increment
product_id	int(11) AFTER `id`		NO	NO	
user_id	int(11) AFTER `product_id`		NO	NO	
deliver_address	varchar(500) AFTER `user_id`		YES	NO	

Fig 3.4: Product Orders table

### 3.5 E-R DIAGRAMS

#### ➤ LOGIN

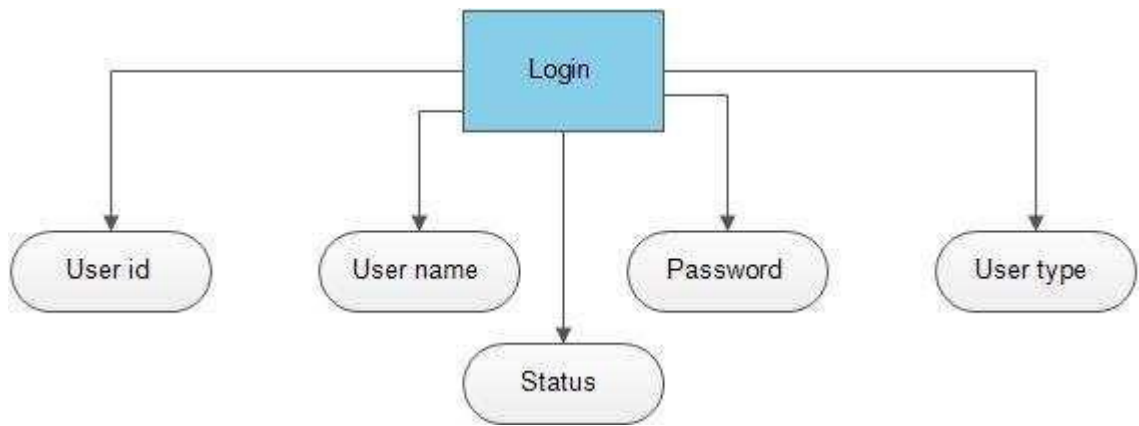


Fig 3.5: Login

#### ➤ USER DETAILS

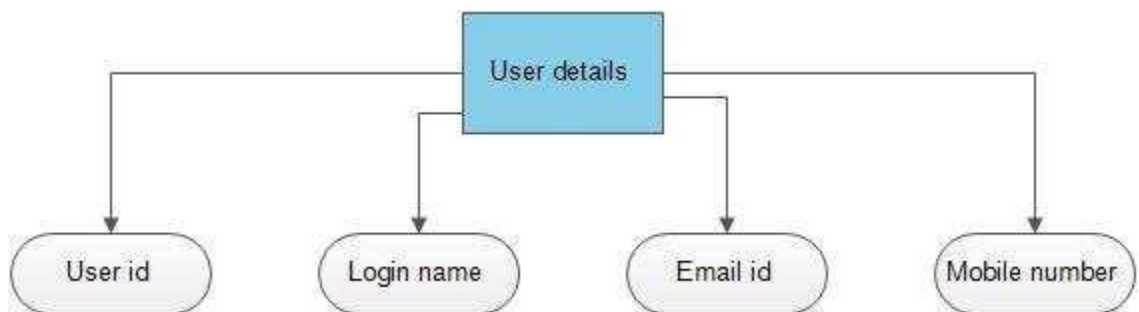


Fig 3.6: User Details



➤ PRODUCT DETAILS

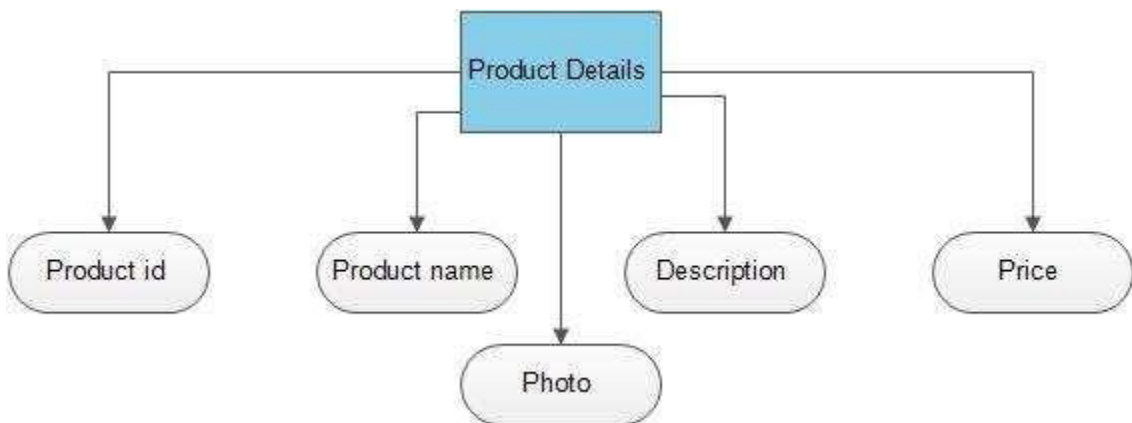


Fig 3.7: Product Details

➤ PRODUCT ORDERS

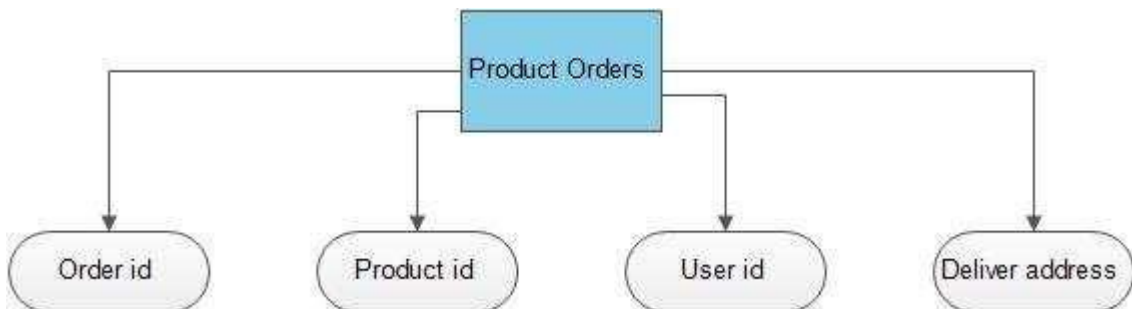


Fig 3.8: Product Orders

➤ COMPLETE DIAGRAM

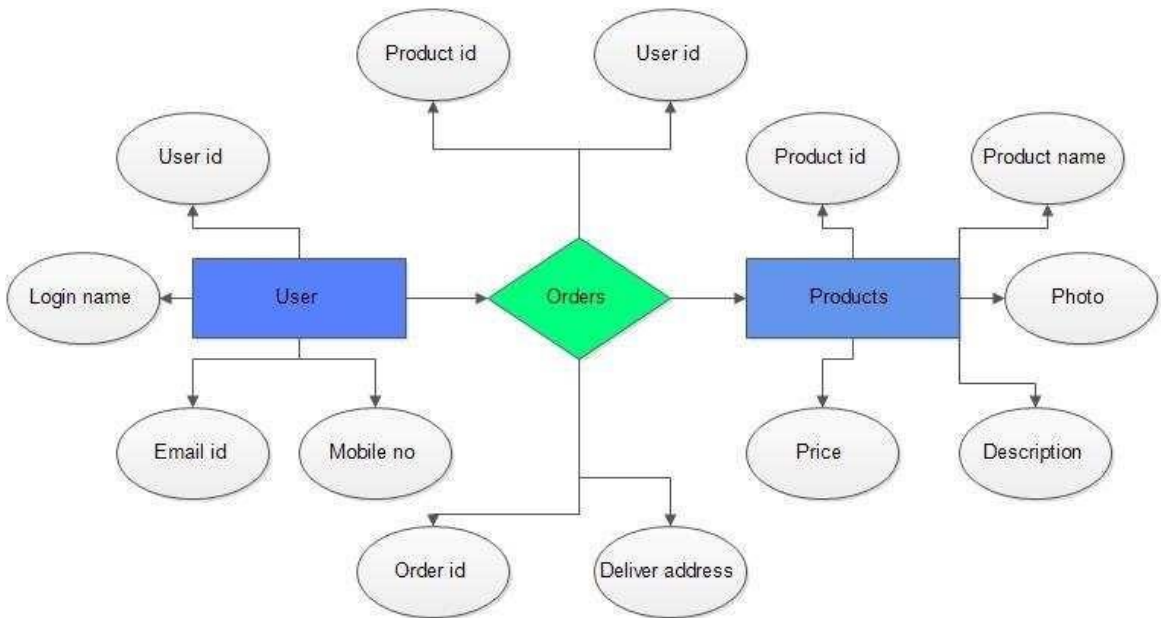


Fig 3.9: Complete Diagram

### 3.6 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system

requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons DFD's are often referred to as logical data flow diagrams.

## EXTERNAL ENTITY

An external entity is a source or destination of a data flow. Only those entities which originate or receive data are represented on a data flow diagram. The symbol used is a rectangular box.

## PROCESS

A process shows a transformation or manipulation of data flow within the system. The symbol used is an oval shape.

## DATAFLOW

The data flow shows the flow of information from a source to its destination. Data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents.

## DATA STORE

A data store is a holding place for information within the system: It is represented by an open ended narrow rectangle. Data stores may belong-term files such as sales ledgers, or may be short-term accumulations: for example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.

### ➤ LOGIN DFD

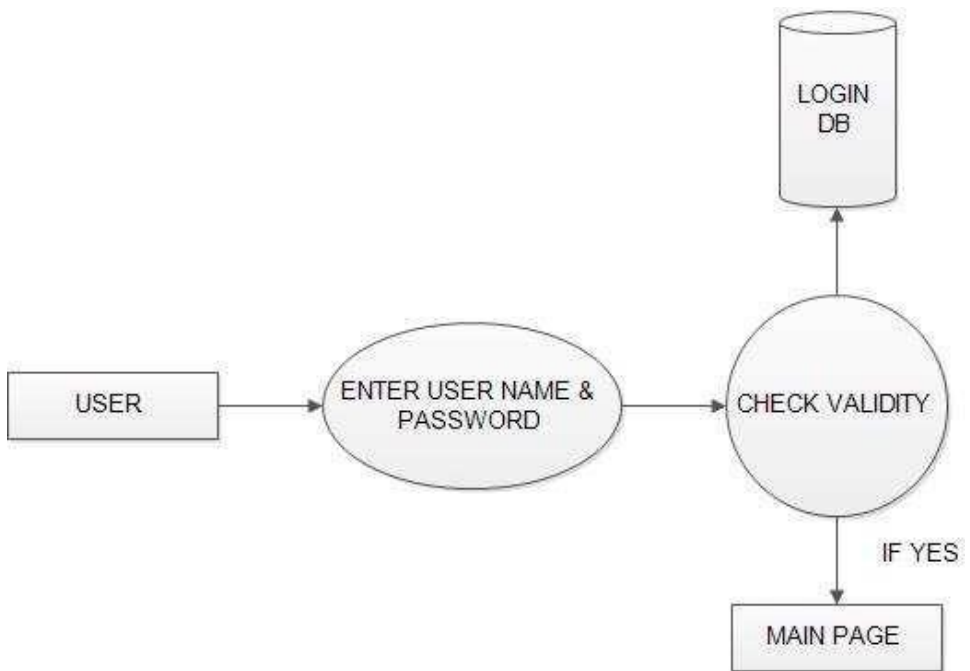


Fig 3.10: Login DFD

➤ REGISTRATION DFD

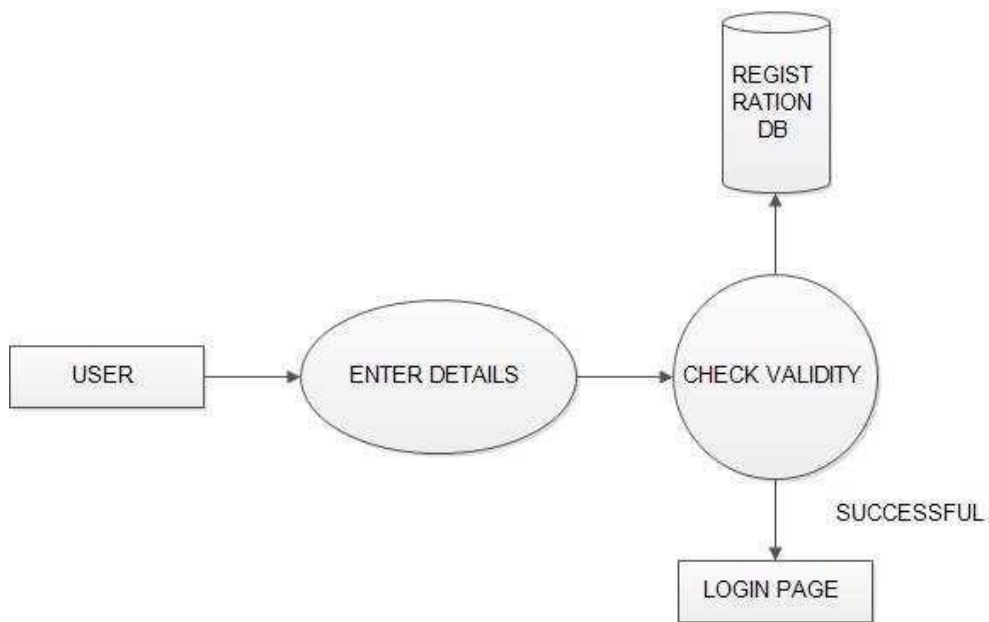


Fig 3.11: Registration DFD

➤ ADMIN DFD

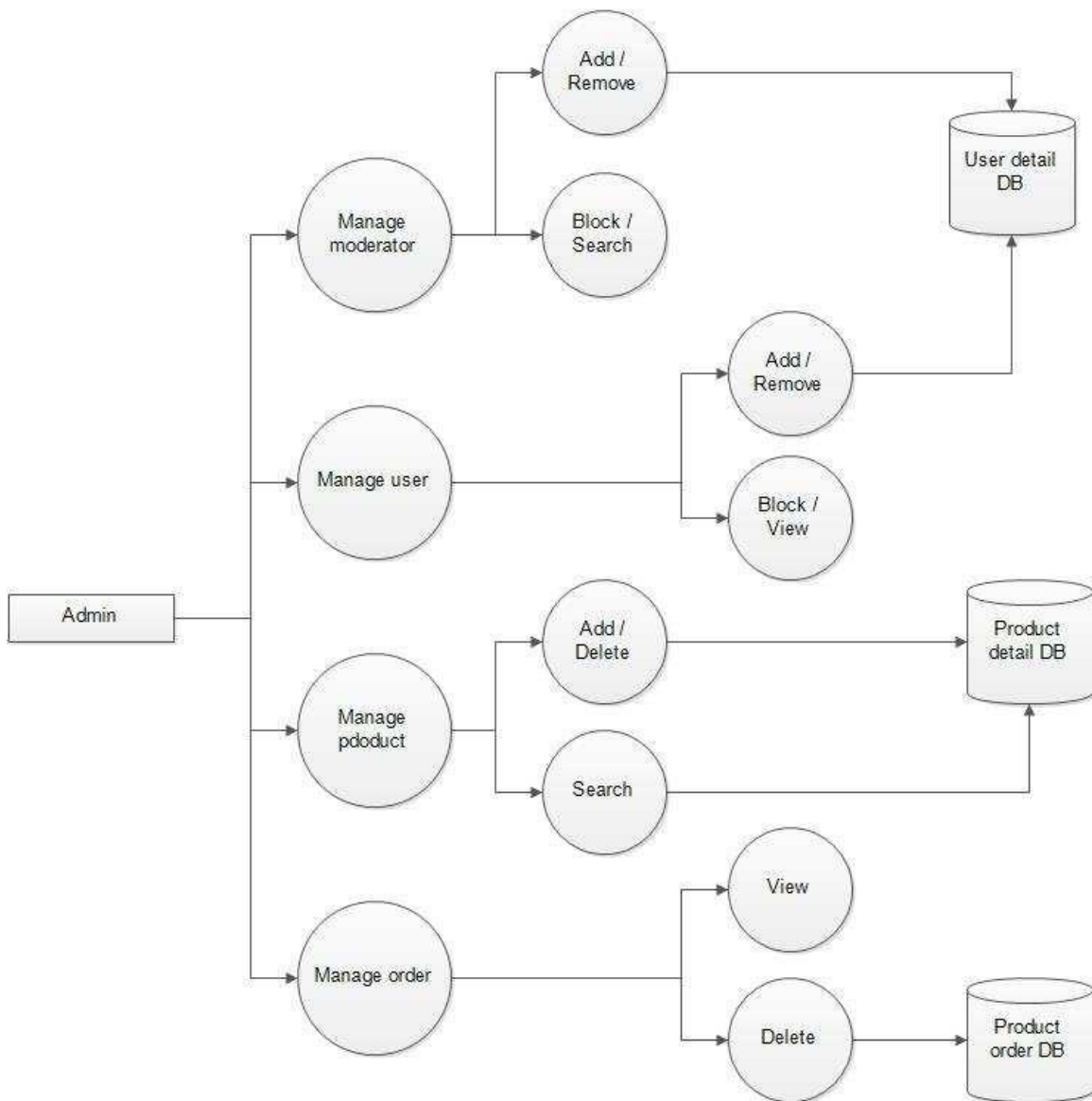


Fig 3.12: Admin DFD

➤ MODERATOR DFD

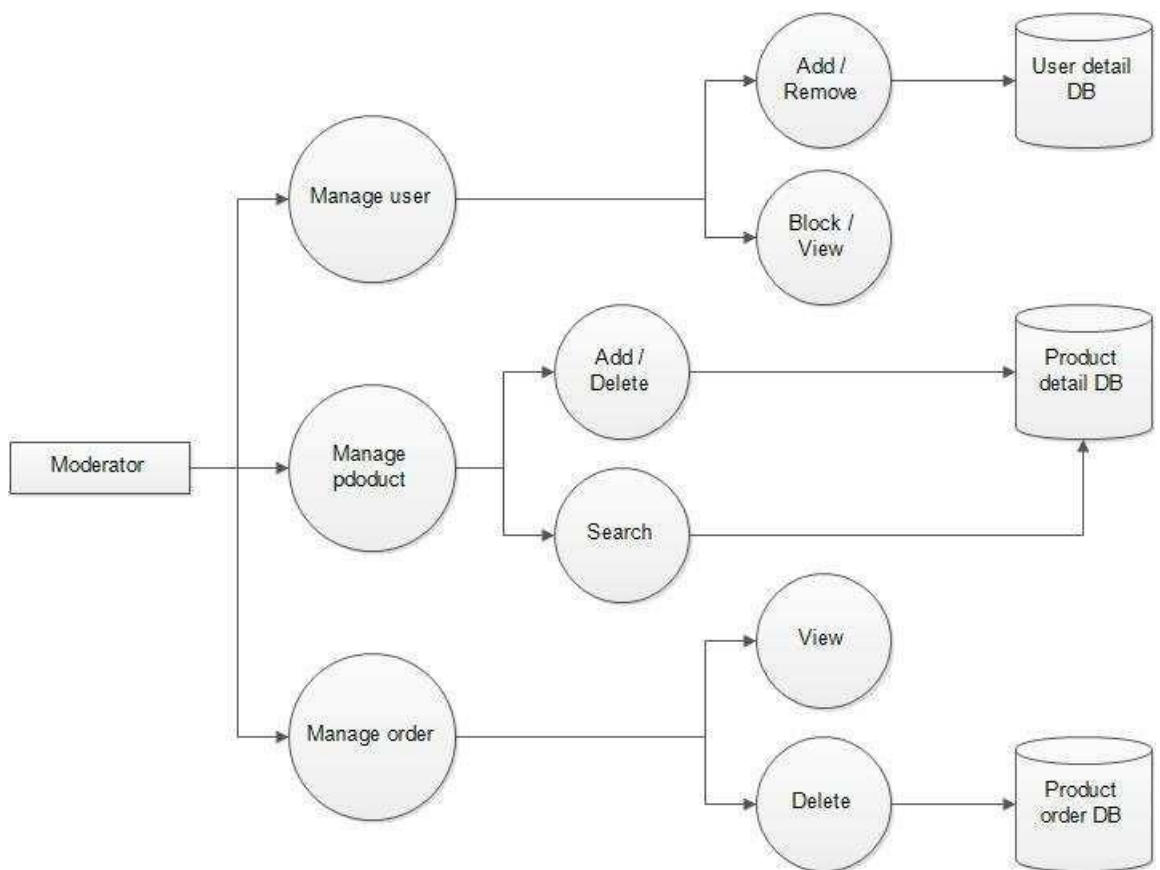
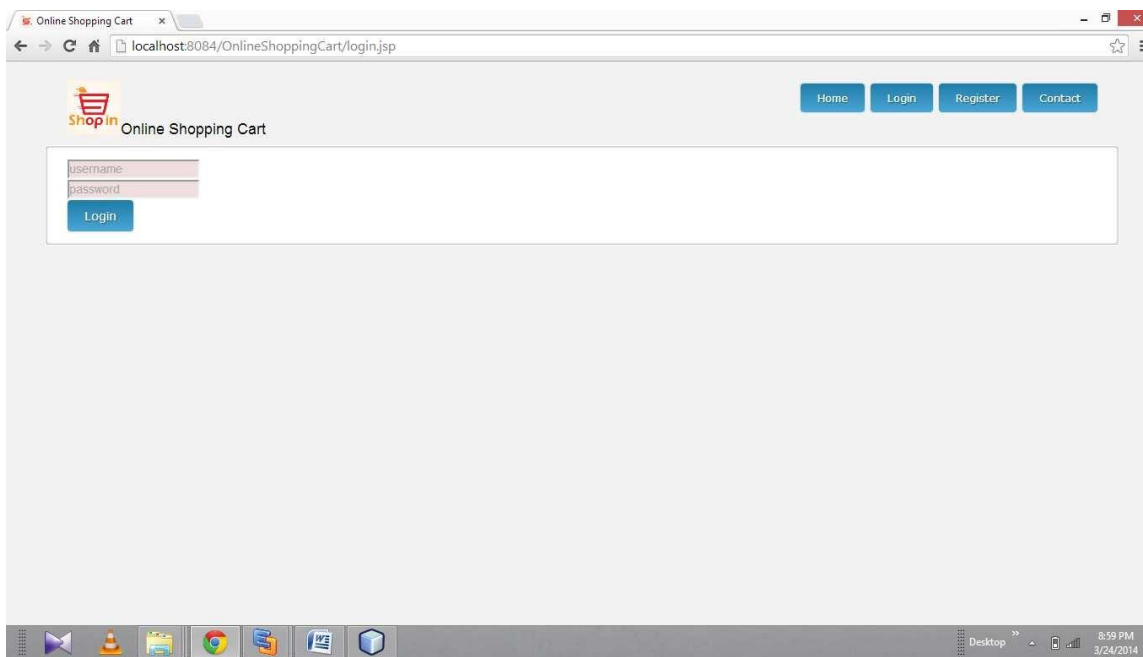


Fig 3.13: Moderator DFD



## 3.7 SCREEN SHOTS

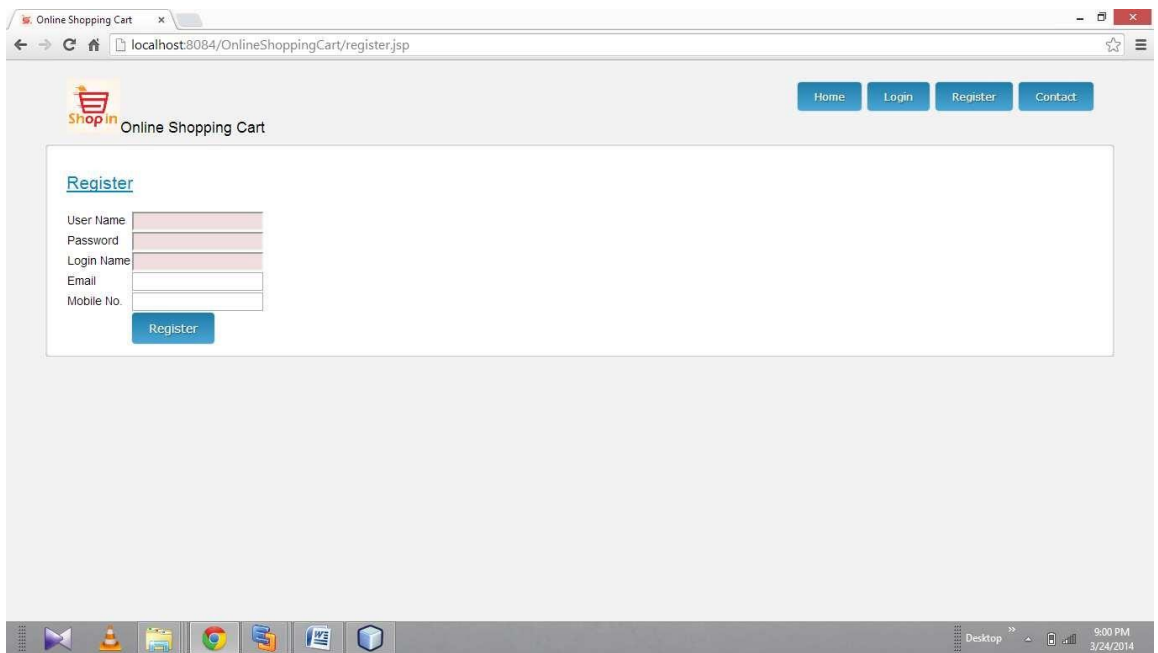
### ➤ LOGIN



### ➤ REGISTRATION

## Online Shopping System

---

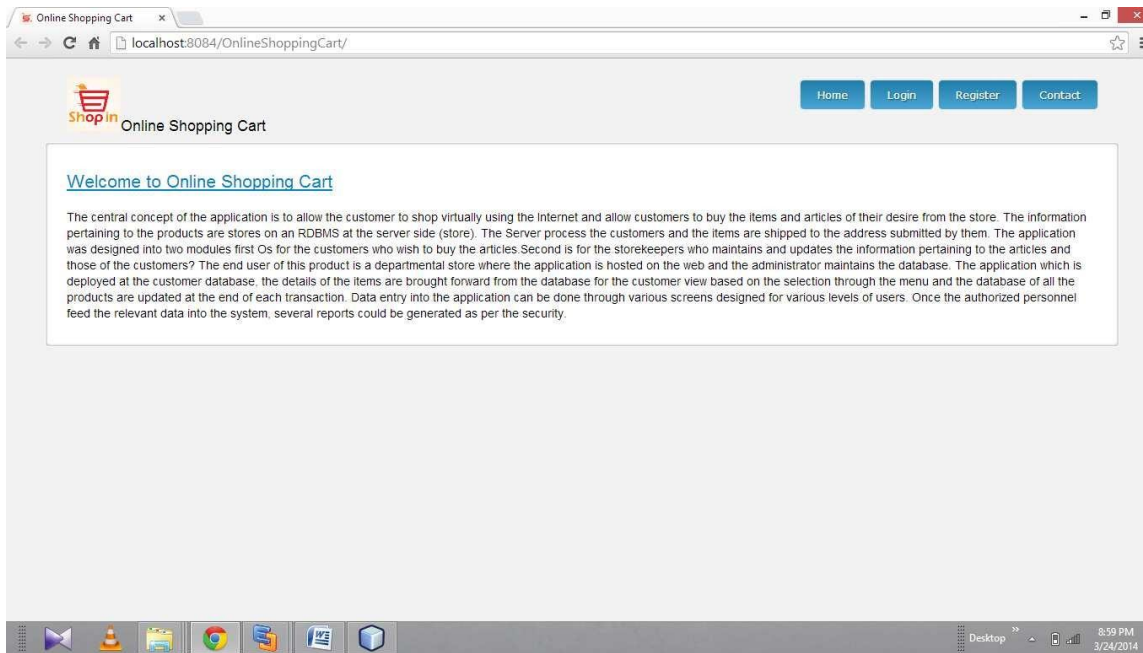


The screenshot shows a web browser window with the title "Online Shopping Cart" and the address bar displaying "localhost:8084/OnlineShoppingCart/register.jsp". The page features a header with a "Shop in" logo and the text "Online Shopping Cart". Navigation buttons for "Home", "Login", "Register", and "Contact" are located in the top right. The main content area is titled "Register" and contains a registration form with the following fields: "User Name", "Password", "Login Name", "Email", and "Mobile No.". A "Register" button is positioned below the "Mobile No." field. The browser's taskbar at the bottom shows various application icons and the system clock indicating 9:00 PM on 3/24/2014.

➤ HOME

## Online Shopping System

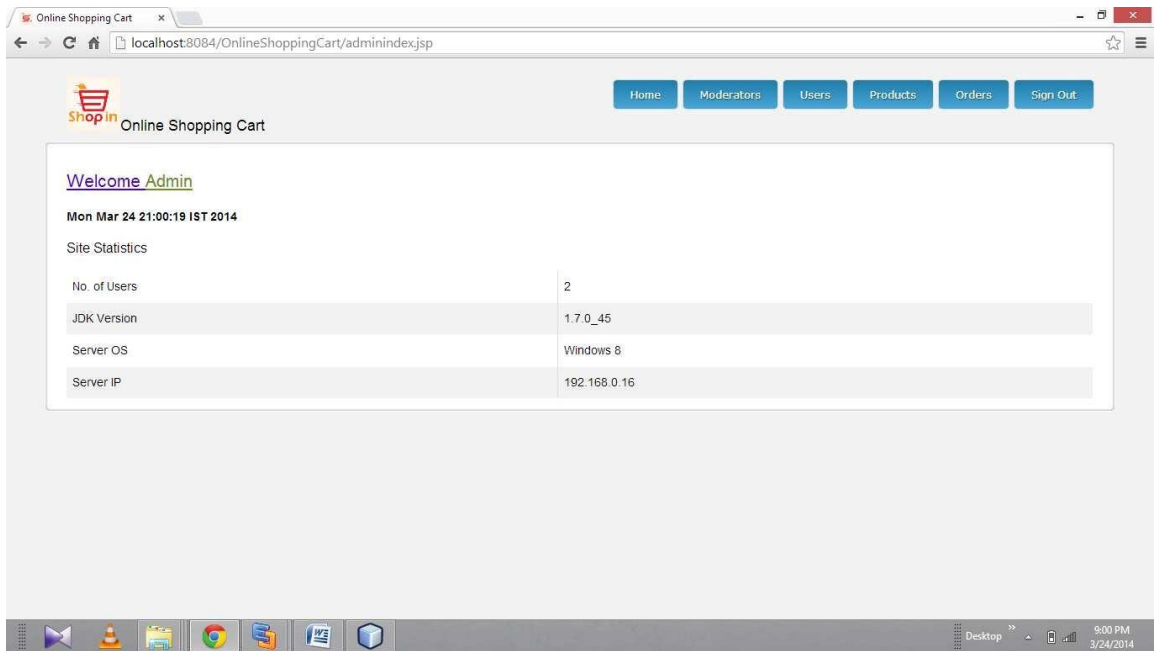
---



### ➤ ADMIN HOME

## Online Shopping System

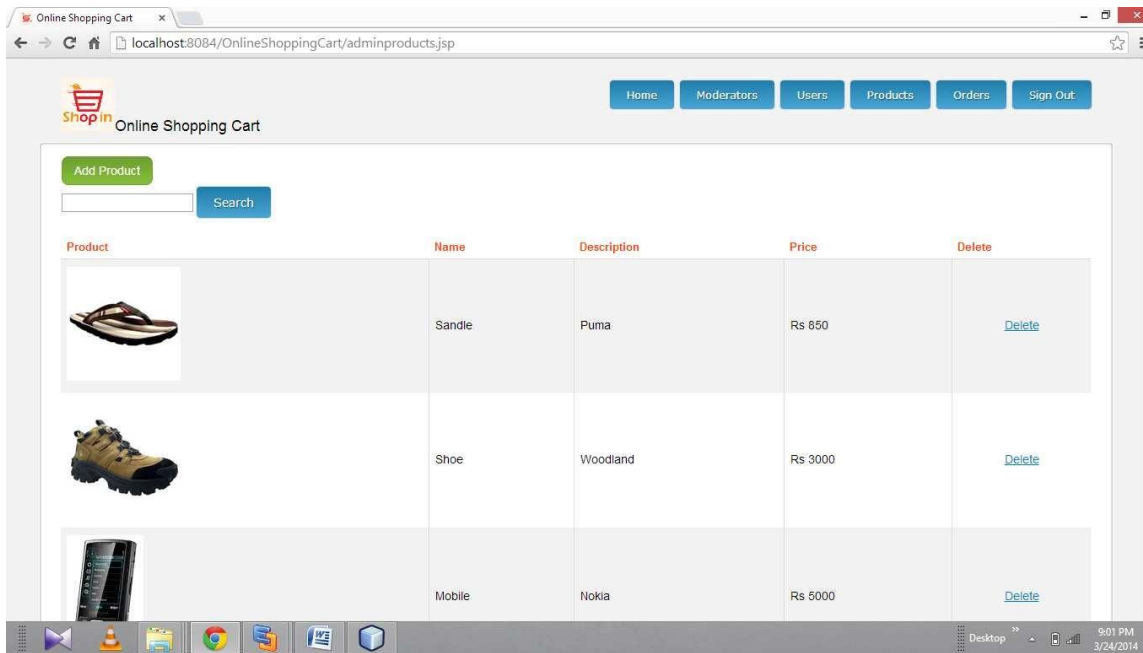
---



### ➤ ADMIN PRODUCT

## Online Shopping System

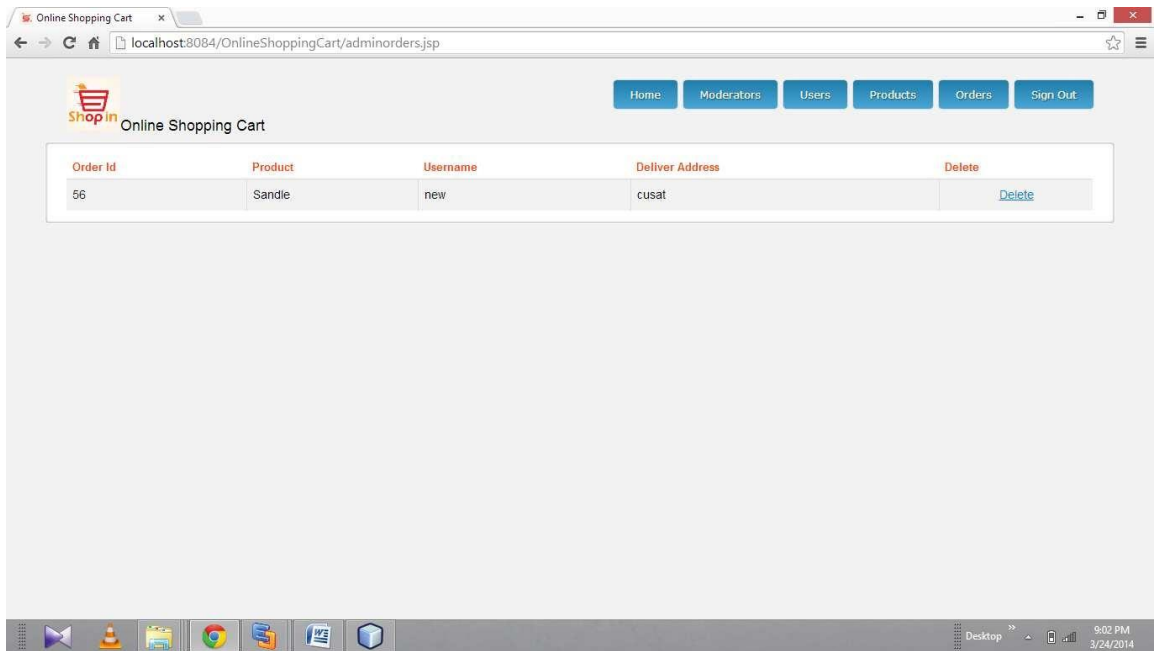
---



### ➤ ADMIN ORDER

## Online Shopping System

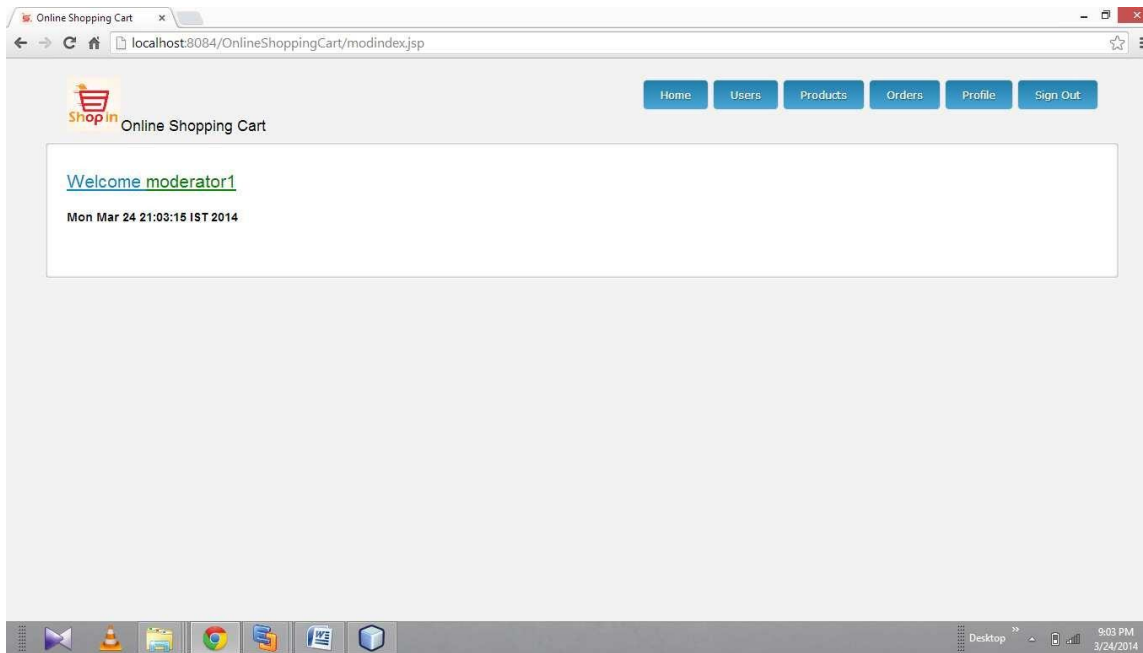
---



### ➤ MODERATOR HOME

## Online Shopping System

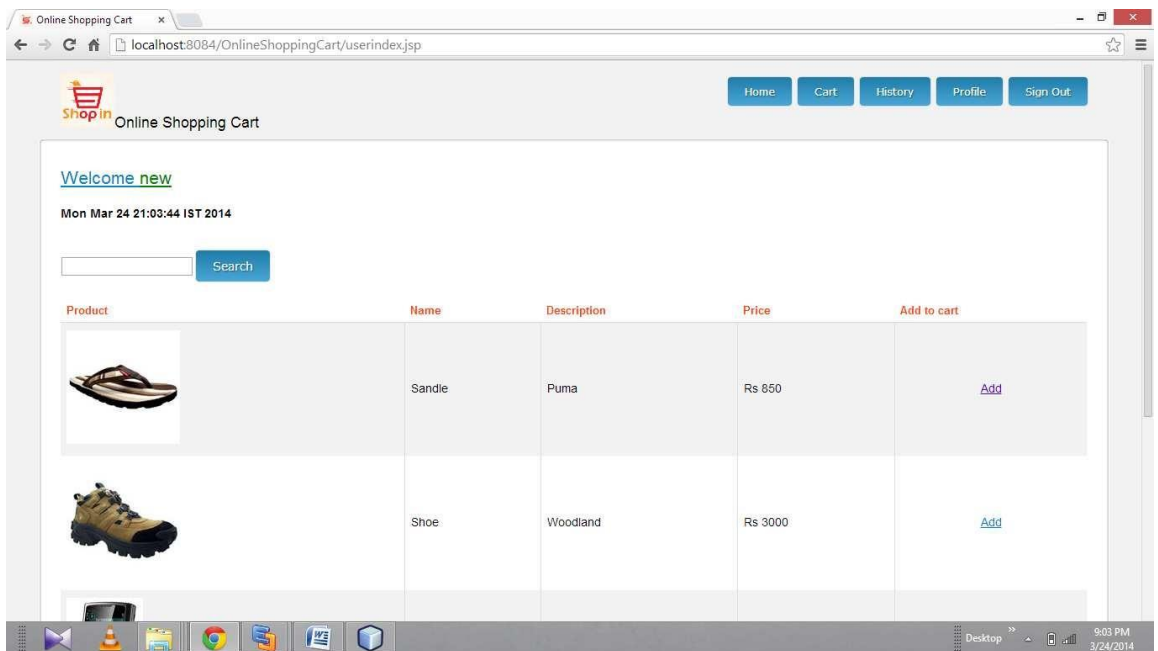
---



### ➤ USER HOME

## Online Shopping System

---

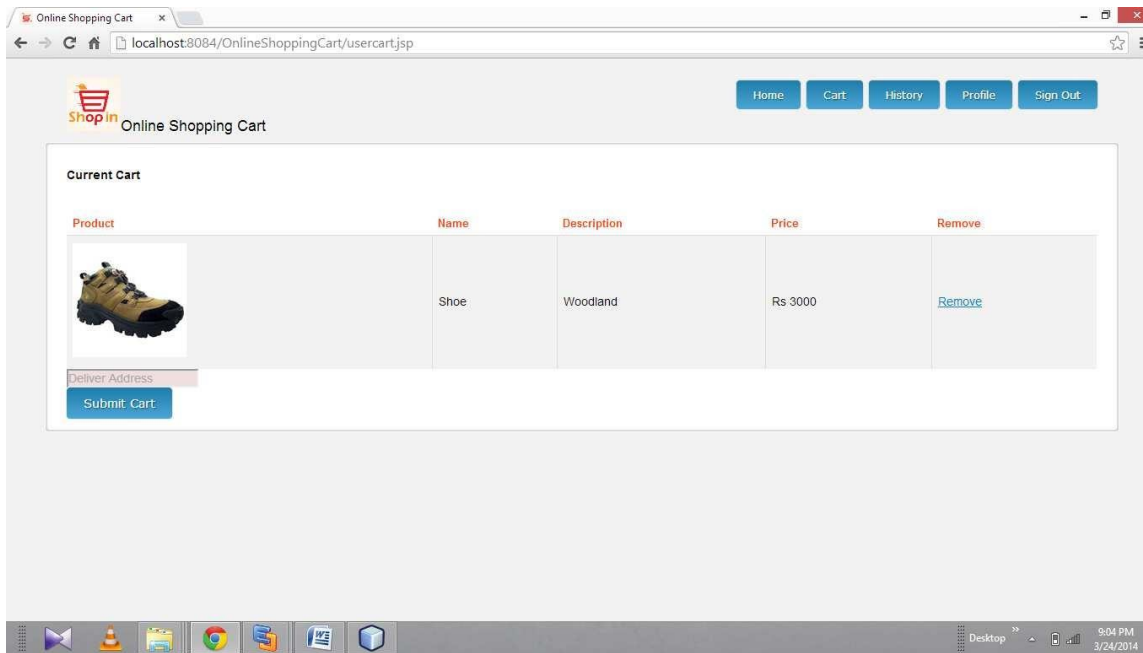


### ➤ USER CART



## Online Shopping System

---



### 3.8 SAMPLE CODE

#### ➤ LOGIN PAGE

```
<%@include file="template/top.jsp" %>
```

```
<%@include file="template/menu_public.jsp" %>
```

```
<%@include file="template/body_start.jsp" %>
```

```
<form id="form" action="loginbaseselect.jsp">
```

```
<table>

  <tr>

    <td><input type="text" name="uname" id="uname"
placeholder="username" required=""/>  </td>

  </tr>

  <tr>

    <td><input type="password" name="upass" id="upass"
placeholder="password" required=""/></td>

  </tr>

  <tr>

    <td><button class="button" type="submit" id="search-
submit">Login</button>  </td>

    <td><div id="result"></div></td>

  </tr>

</table>

</form>

<script type="text/javascript">

  <%
```

```
        if (request.getParameter("login") != null &&  
request.getParameter("login").contentEquals("failed")) {
```

```
    %>
```

```
    alert("Login Failed !!!");
```

```
    <%                                }
```

```
    %>
```

```
    <%
```

```
        if (request.getParameter("login") != null &&  
request.getParameter("login").contentEquals("dissabled")) {
```

```
    %>
```

```
    alert("User Blocked !!!");
```

```
    <%                                }
```

```
    %>
```

```
    <%
```

```
        if (request.getParameter("login") != null &&  
request.getParameter("login").contentEquals("unknown")) {
```

```
    %>
```

```
    alert("Unknown user !!!");
```

```
    <%                                }
```

```
%>  
</script>
```

```
%@include file="template/body\_end.jsp" %
```

### ➤ REGISTRATION

```
<%@page import="DataBase.DB"%>  
<%@page import="java.sql.ResultSet"%>  
<%@include file="template/top.jsp" %>  
<%@include file="template/menu_public.jsp" %>  
<%@include file="template/body_start.jsp" %>  
<h2 class="title"><a href="#">Register </a></h2>  
<div class="entry">  
    <form action="dbregister.jsp" onsubmit="return check()">  
        <table>  
            <tr>  
                <td>User Name  
                </td>
```

```
        <td><input type="text" required="" name="uname" id="uname"
/>

        </td>

    </tr>

    <tr>

        <td>Password

        </td>

        <td><input type="password" required="" name="upass"
id="upass" />

        </td>

    </tr>

    <tr>

        <td>Login Name

        </td>

        <td><input type="text" required="" name="lname" id="lname" />

        </td>

    </tr>
```

```
<tr>
    <td>Email
    </td>
    <td><input type="email" name="emailid" id="emailid" />
    </td>
</tr>
```

```
<tr>
    <td>Mobile No.
    </td>
    <td><input type="text" name="mobno" id="mobno" />
    </td>
</tr>
```

```
<tr>
    <td>
    </td>
    </td>
```

```
<td><button class="button" type="submit">Register</button>
</td>
</tr>

</table>
</form>
</div>

<script type="text/javascript">

    <%
        if (request.getParameter("reg") != null &&
request.getParameter("reg").equals("failed")) {
    %>

        alert("Registration Failed !!!");
    <%                                }
    %>

    <%
```

```
        if (request.getParameter("reg") != null &&
request.getParameter("reg").equals("success")) {

    %>

        alert("Registration Completed")

    <%                                }

    %>

</script>
```

[%@include file="template/body\\_end.jsp" %](#)

#### ➤ DB REGISTRATION

```
<%@page import="config.Config"%>

<%@page import="java.io.File"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="DataBase.DB"%>

<%

    if (DB.executeUpdate("insert into login values("

        + "null" + ",""
```



```
+ request.getParameter("uname") + "',SHA1(""  
+ request.getParameter("upass") + "'),"  
+ "user" + "','"  
+ "false" + "')")) {  
ResultSet r = DB.executeQuery("select max(uid) from login");  
r.next();  
String uid = r.getString(1);  
if (DB.executeUpdate("insert into userdetails values("  
+ uid + ",'" + request.getParameter("lname") + ",'" + request.getParameter("emailid") + ",'" + request.getParameter("mobno") + "')")) {  
    // this code generates the qr code inside the QR_Code folder  
    in the c drive  
    response.sendRedirect("index.jsp?reg=success");  
}  
} else {  
    response.sendRedirect("register.jsp?reg=failed");  
}
```

%>

➤ LOGIN BASE SELECTION

```
<%@page import="DataBase.DB"%>
```

```
<%@page import="java.sql.ResultSet"%>
```

```
<%
```

```
    String uname = request.getParameter("uname");
```

```
    String upass = request.getParameter("upass");
```

```
    ResultSet r = DB.executeQuery("select * from login where  
uname='" + uname + "' and upass=SHA1('" + upass + "')");
```

```
    if (r != null && r.next()) {
```

```
        if (r.getString("enabled") != null &&  
r.getString("enabled").contentEquals("true")) {
```

```
            session.setAttribute("uname", uname);
```

```
            session.setAttribute("uid", r.getString("uid"));
```

```
            session.setAttribute("utype",r.getString("utype"));
```

```
        if (r.getString("utype").contentEquals("admin")) {
            response.sendRedirect("adminindex.jsp");
        } else if (r.getString("utype").contentEquals("mod")) {
            response.sendRedirect("modindex.jsp");
        } else if (r.getString("utype").contentEquals("user")) {
            response.sendRedirect("userindex.jsp");
        } else {
            session.setAttribute("uname", null);
            session.setAttribute("uid", null);
            session.setAttribute("utype", null);
            response.sendRedirect("login.jsp?login=unknown");
        }

    } else {
        response.sendRedirect("login.jsp?login=disabled");
    }

} else {
```

```
        response.sendRedirect("login.jsp?login=failed");  
    }  
%>
```

### ➤ DATABASE CONNECTION

```
package DataBase;
```

```
import java.sql.*;  
public class DB {
```

```
    static Connection con = null;  
    static Statement stmt = null;  
    static ResultSet rs = null;
```

```
    public static Connection getConnection() {  
        try {  
            Class.forName("com.mysql.jdbc.Driver");
```

```
        con =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/online_shoppi  
ng", "root", "root");  
  
        } catch (Exception e) {  
  
        System.out.println("Exception" + e);  
  
        }  
  
        return con;  
  
    }
```

```
public static boolean executeUpdate(String query) {  
  
    boolean f = false;  
  
    try {  
  
  
  
  
        con = getConnection();  
  
        stmt = con.createStatement();  
  
        int b = stmt.executeUpdate(query);  
  
        if (b > 0) {  
  
            f = true;  
  
        } else {
```

```
        f = false;
    }

    } catch (Exception e) {
        System.out.println("Exception" + e);
    }
    return f;
}

public static ResultSet executeQuery(String sql) {
    try {
        con = getConnection();
        stmt = con.createStatement();
        rs = stmt.executeQuery(sql);
        return rs;
    } catch (Exception e) {
        System.out.println("Exception" + e);
        return null;
    }
}
```

```
    }  
  
    }  
  
    public static void close() {  
        try {  
            con.close();  
            stmt.close();  
            rs.close();  
        } catch (Exception e) {  
            System.out.println("Exception" + e);  
        }  
    }  
}
```

➤ ANDROID WEB-VIEW ACTIVITY

```
package com.onlineshopping.activity;
```

```
import com.onlineshopping.activity.R;

import com.onlineshopping.activity.custom.CustomWebChromeClient;

import com.onlineshopping.activity.custom.CustomWebViewClient;


import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.res.Configuration;
import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebSettings.PluginState;
import android.webkit.WebView;


@SuppressLint("SetJavaScriptEnabled")
public class WebViewActivity extends Activity {

    // the web view refence object
    private WebView webView;

    // the site url link
```



```
public static final String SITE_URL_START = "http://";  
  
public static String SITE_URL = "";  
  
public static final String SITE_URL_END =  
":8084/OnlineShoppingCart/";
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_web_view);  
  
    // get the reference of the web view  
    webView = (WebView) findViewById(R.id.siteWebView);  
  
    // set the web view chrome client and set for showing a  
    progress bar  
    this.webView.setWebChromeClient(new  
    CustomWebChromeClient(this));  
  
    this.webView.setWebViewClient(new  
    CustomWebViewClient(  

```

```
        getApplicationContext()));

        // on the ui thread do the processing
        runOnUiThread(new Runnable() {
            @SuppressWarnings("deprecation")
            public void run() {
                // set the javascript on

                webView.getSettings().setJavaScriptEnabled(true);

                // enable all the plugins

                webView.getSettings().setPluginState(PluginState.ON);

                // load the url

                webView.loadUrl(SITE_URL_START +
SITE_URL + SITE_URL_END);
            }
        });
    }
```

// To handle "Back" key press event for WebView to go back to previous

// screen.

@Override

```
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    if ((keyCode == KeyEvent.KEYCODE_BACK) &&  
webView.canGoBack()) {  
        webView.goBack();  
        return true;  
    }  
    return super.onKeyDown(keyCode, event);  
}
```

// flipscreen not loading again

@Override

```
public void onConfigurationChanged(Configuration newConfig) {  
    super.onConfigurationChanged(newConfig);  
}
```

}

## **CHAPTER 4**

# **CONCLUSION**

The project entitled **Online shopping system** was completed successfully.

The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application and an android application for purchasing items from a shop.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using html & css, usage of responsive templates, designing of android applications, and management of database using mysql . The entire system is secured. Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications.

There is a scope for further development in our project to a great extend. A number of features can be added to this system in future like providing

moderator more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of history of purchases of each customer and provide suggestions based on their history. These features could have implemented unless the time did not limited us.

## REFERENCES

[1] JavaScript Enlightenment, Cody Lindley-First Edition, based on JavaScript 1.5, ECMA-262, Edition

[2] McGraw-Hill's, Java : The complete reference 7th Edition, Herbert Schildt

[3] Complete CSS Guide, Maxine Sherrin and John Allsopp-O'Reilly Media; September 2012

[4] <http://www.w3schools.com/html/default.asp>,  
<http://www.w3schools.com/css/default.asp>,  
<http://www.w3schools.com/js/default.asp>