STELLER

(E-commerce)

S. R. S. Report – I

**Team Members**

**Shubhika Gulati**

**(161500554)**

**Shubham Kushwah**

**(161500542)**

**Shivangni Sharma**

**(161500526)**

**Utkarsh Singh**

**(161500601)**

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.2.2 <Functional Requirement or Feature #2> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.4.2 <Class / Object #2> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

4. Analysis Models 4

4.1 Sequence Diagrams 5

4. 2Data Flow Diagrams (DFD) 5

4.3 ENTITY RELATIONSHIP DIAGRAM 5

A. Appendices 5

A.1 Appendix 1 5

A.2 Appendix 2 5

# 1. Introduction

The Software Requirements Specification is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested. Its primary purpose is to provide a clear and descriptive “statement of user requirements” that can be used as a reference in further development of the software system. This document is broken into a number of sections used to logically separate the software requirements into easily referenced parts. This Software Requirements Specification aims to describe the Functionality, External Interfaces, Attributes and Design **Constraints imposed** on Implementation of the software system described throughout the rest of the document. Throughout the description of the software system, the language and terminology used should unambiguous and consistent throughout the document.

## 1.1 Purpose

Defining and describing the functions and specifications of the E-commerce website (Steller) is the primary purpose of this Software Requirements Specification (SRS). This Software Requirements Specification illustrates, in clear terms, the system’s primary uses and required functionality as specified by our team according to current need of customers who used it in future.

## 1.2 Scope

The software system being produced is called E-Commerce Website or Steller. It is being produced for a customer interested in buying Cloths via Internet. This system is designed to “provide automation support” for the process of placing cloths for sale on the Internet and facilitating the actual sale.

The system will be run on a central server with each user having a remote user interface through a web browser to interact with it.

The Book E-Commerce System will allow any user to create an account to become a customer. It is helpful for local merchants and designers to find customers easily and provide platform to show their product effectively.

There is no more need to go different types of shops to buy different cloths every variety is available on a single site.

## 1.3 Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

## 1.4 References

IEEE Recommended Practice for Software Requirements Specification-

IEEE Std830-1993.

## 1.5 Overview

This Software Requirements Specification document is divided in to multiple subsections. The first section includes explanations of the Purpose, Scope and Organization of the document. The first section also handles the description of project specific words, acronyms and abbreviations that will be used in the document. The second section of the document is separated into the following five different sections, each detailing specific details of system uses and their corresponding actions: Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies, Apportioning of Requirements. The third section is an enumerated listing of all of the requirements described for this system. The fourth section encompasses all of the Use-case, Sequence, State and Class diagrams that model the system. In the fifth section there exists a Prototype of the system along with a sample scenario that graphically describes the use of the system. The sixth section contains a listing of all related reference materials used in this document. The seventh and final subsection is dedicated to providing a point of contact for any viewer of this document.

# 2. General Description

This section includes details about what is and is not expected of the Steller system in

addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the Steller system.

## 2.1 Product Perspective

(i) User interfaces the Website will have a user friendly and menu based interface.

Following screens will be provided:

(ii) A login screen for entering the username, password will be provided. Access to different screens will be based upon the user.

(iii) There is a screen for displaying information regarding products customer brought .

(iv) There is a screen for displaying information regarding the product user want to buy (Cart ).

(v) There is a screen for displaying information menu regarding what kind of cloths you want to buy(men ,women , kids).

(vi) There is a screen for displaying cloths details like color, size, brand of product.

(vii) There is a screen for transaction.

(viii) Hardware interfaces.

(a) Support for printer for printing payment receipt then and there.

(b) Screen resolution of at least 800X600 is required for proper and

Complete viewing of screens. Higher resolution will be accepted.

(ix) Software interfaces

(a) Any windows based operating system.

(b) MS Access 2000 as the DBMS-for database.

(c) IDE (Anaconda) for developing code.

(x) Communications interface: None

(xi) Site Adaptation Requirements: Web browser with cookies enabled.

## 2.2 Product Functions

STELLER will provide a number of functions; each is listed below.

* Maintain data associated with the inventory (a collection of products)

(a) A product has its brandname, type and price.

(b) The inventory also keep track of the stock/quantity of each product.

* Maintain records for many customers

1. A customer has a username (unique across all users), password (no restrictions), email address (no restrictions), and postal address (unverified).
2. Anyone may sign up for a customer account.

* Show a listing of available products.
* Products are to be displayed in different divisions according to type.
* Allow customers and managers to log in and out of the system.
* Users (both customers and the manager) will be logged out if inactive for 30 minutes.
* Shopping cart

(a) Anyone is able to add one or more books to the shopping cart.

(b) The shopping cart does not need to allow multiple copies of any book.

* Checkout

(a)Checkout is only available to logged-in customers. A user that is not logged in as a

customer is given a chance to log in.

* The promotion is a fixed percentage discount that is to be applied to an entire order.
* The discount is specified by the manager at the time of the promotion’s creation or

most recent update/edit.

* Only cash on delivery is avialble.
* Log/record the orders placed by a user.
* Allow manager to update stock quantities

1. Allow manager to change any book's price

Allow manager to view transaction logs

## 2.3 User Characteristics

1. Educational level: Users should be comfortable with the English language.
2. Experience: Users should have prior information regarding what type of product they want.
3. Skills: Users should have basic knowledge and should be comfortable using general purpose websites on computers.

## 2.4 General Constraints

As stated by the customer, security is not a concern for this system because payment made through cash on delivery. The database may store passwords in plain text and there doesn't need to be a password recovery feature nor lockout after numerous invalid login attempts. As such, the system may not work correctly in cases when security is a concern. These cases include those listed above in addition to lack of an encrypted connection when sending credit card information and forcing users to use “strong” passwords. A strong password is a password that meets a number of conditions that are set in place so that user's passwords cannot be easily guessed by an attacker. Generally, these rules include ensuring that the password contains a sufficient number of characters and contains not only lowercase letters but also capitals, numbers, and in some cases, symbols.

## 2.5 Assumptions and Dependencies

# Proper working of this website is dependent on the internet connectivity of the users’ computer. Assumptions and dependencies:

# 1. It is assumed that the user has basic knowledge of the system (i.e. he/she is not a first time user).

# 2. It is assumed that the data entered by the user while loging is true

# 3. Specific Requirements

This section provides software requirements to a Level of detail sufficient to enable designers to design the system and Testers to test the system.

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

**\*User Registration Screen:** Various fields available on this screen will be:

\* User Name

\* Password

\* Mobile

\* Email

**\*User Login Screen:** Fields available on this screen are:

\* User Name

\* Password

**Product Details Screen:** Various Fields are:

\* Size of clothes

\* colour

\* design

### 3.1.3 Software Interfaces

\* Any windows based operating system.

\* WAMPP Platform

\* Mysql Database.

### 3.1.4 Communications Interfaces

## 3.2 Functional Requirements

It deals with the functionalities required from the system which are as follows:

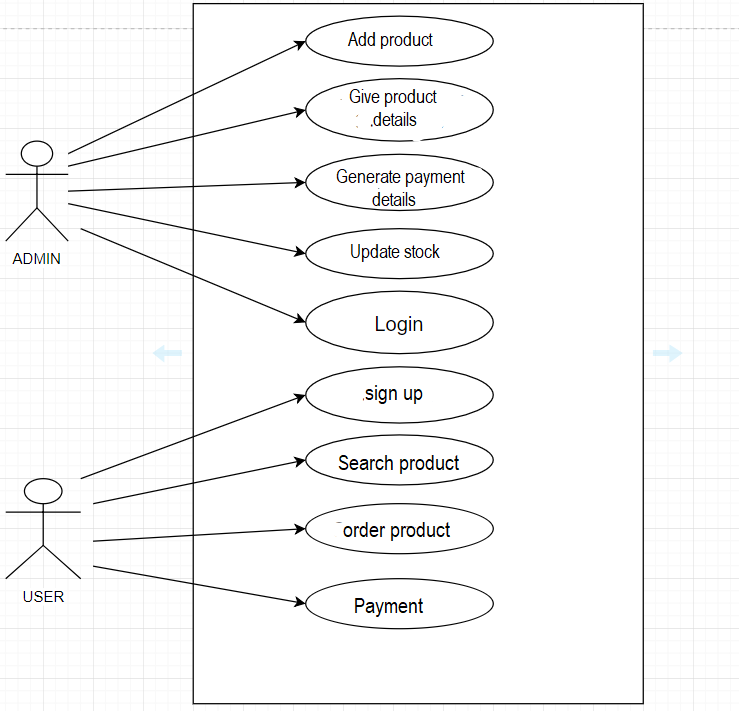
* The website will help the people to buy clothes online.
* Only authorized person can access related details.
* Organizations can change their information regarding themselves.
* The user can login through USER-ID and PASSWORD
* Sellers will be responsible for updating the app

**The technologies used to develop this app are:-**

* PHP :serverside scripting language.
* HTML: Hypertext markup language
* CSS: cascading style sheet
* SQL (STANDARD QUERY LANGUAGE):- SQL is a language which use to Communicate with databases.

## 3.3 Use Cases

### 3.3.1 Use Case #1



…

## 3.4 Classes / Objects



### 3.4.1 <Class / Object #1>

3.4.1.1 Attributes

Username and Password

3.4.1.2 Functions

Update, edit and delete product details

### 3.4.2 <Class / Object #2>

…

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

No. of terminals to be supported is dependent on the server that we will use at the time of deployment. The application server used should provide good performance and ability to manage performance with techniques. After completing the order, the entire cost of product will be calculated as per the rules.

### 3.5.2 Reliability

It means the extent to which program performs with required precision.

The app developed should be extremely reliable and secure so that information about any user is not leaked.

### 3.5.3 Availability

This website will have 24\*7 availability. It can be accessed for 24 hours a day.

However anybody can open this website anytime to access other information.

### 3.5.4 Security

Some of the factor that are identified to protect the software from accidental or malicious access, use, modification below. Keep specific log or history data sets

Assign certain functions to different modules

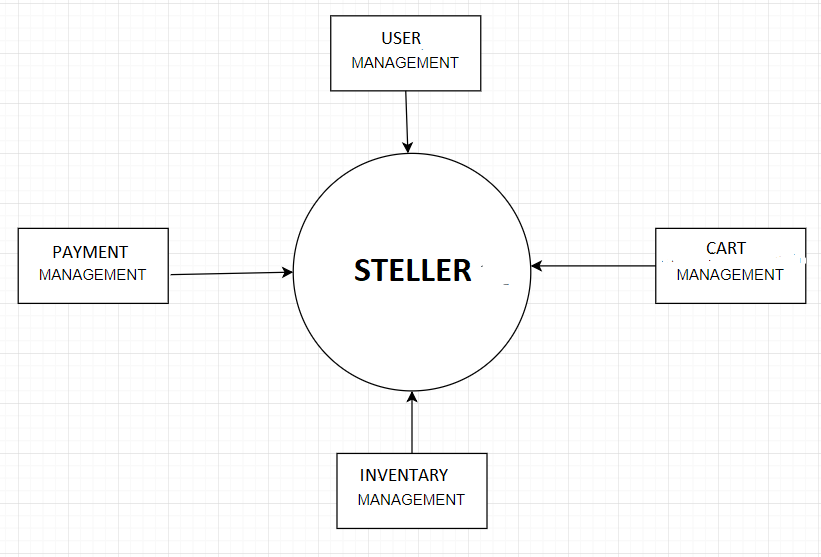
### 3.5.5 Maintainability

The app can be maintained in present or future. It will be easy to incorporate new requirements in the individual modules.

### 3.5.6 Portability

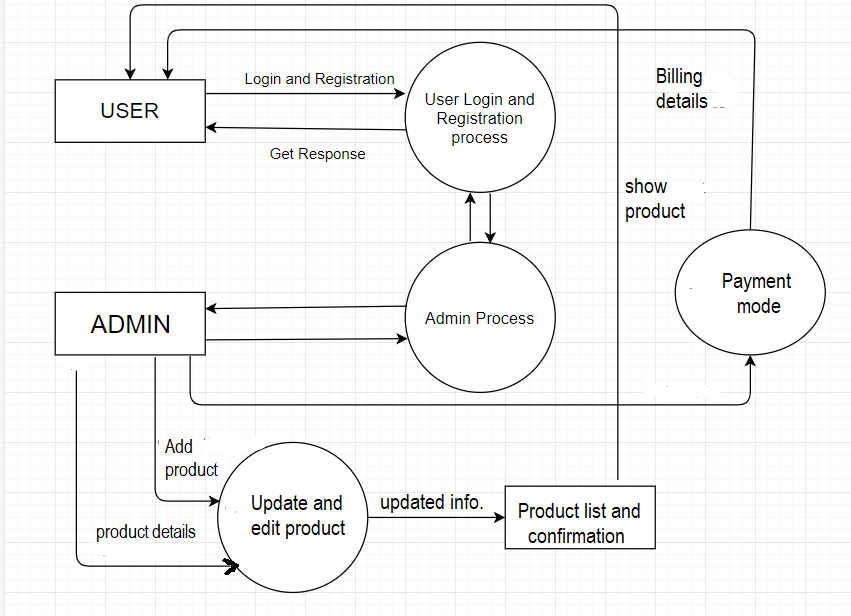
The app should be user friendly and should require least effort to operate. So that people can use it freely and happily.

## 4.1 Sequence Diagrams

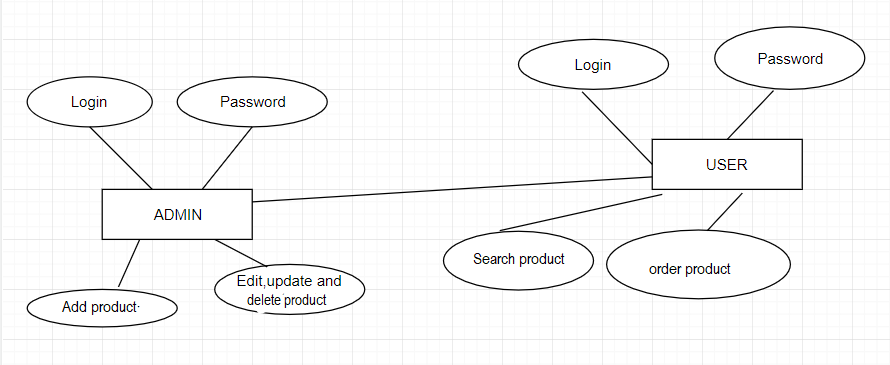




## 4.2 Data Flow Diagrams (DFD)



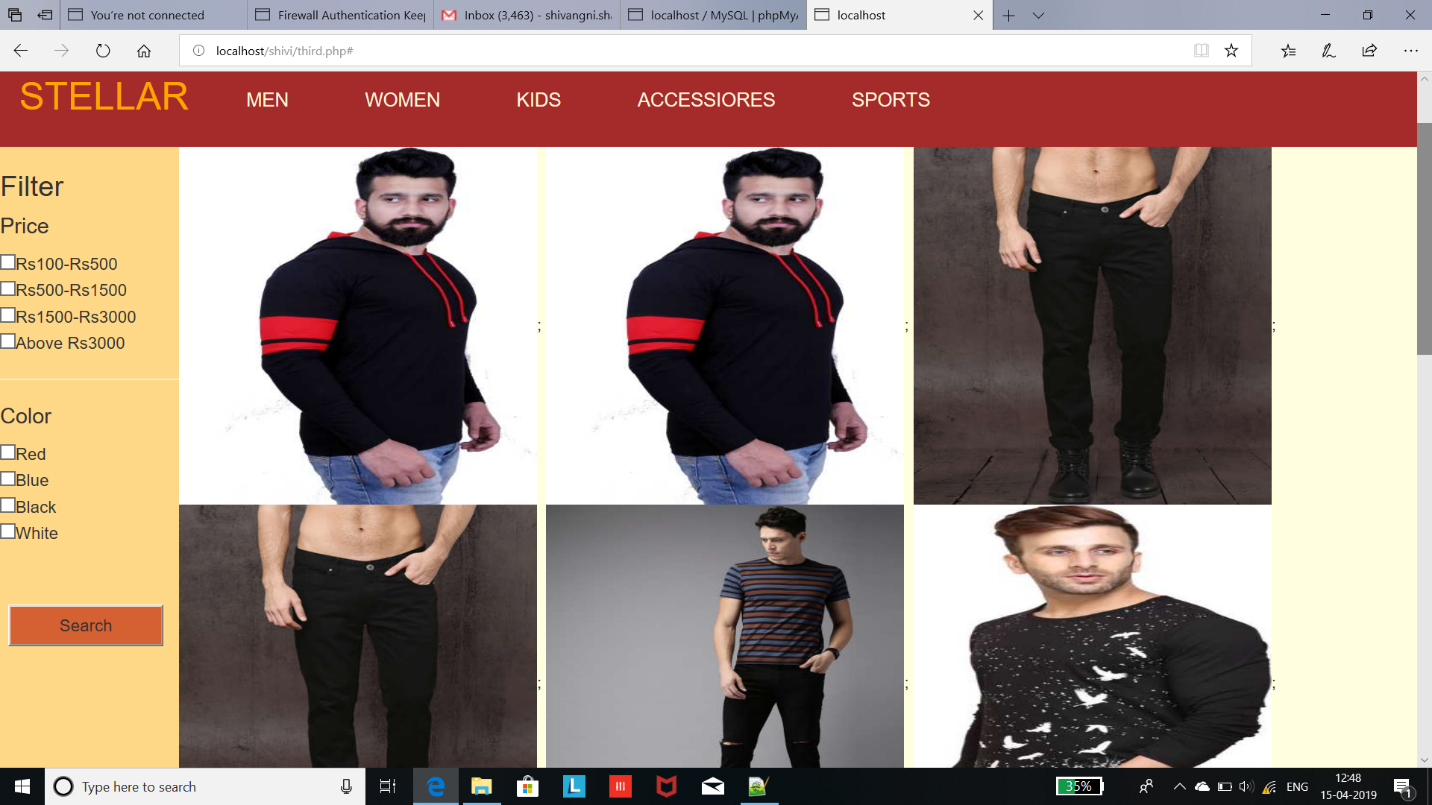
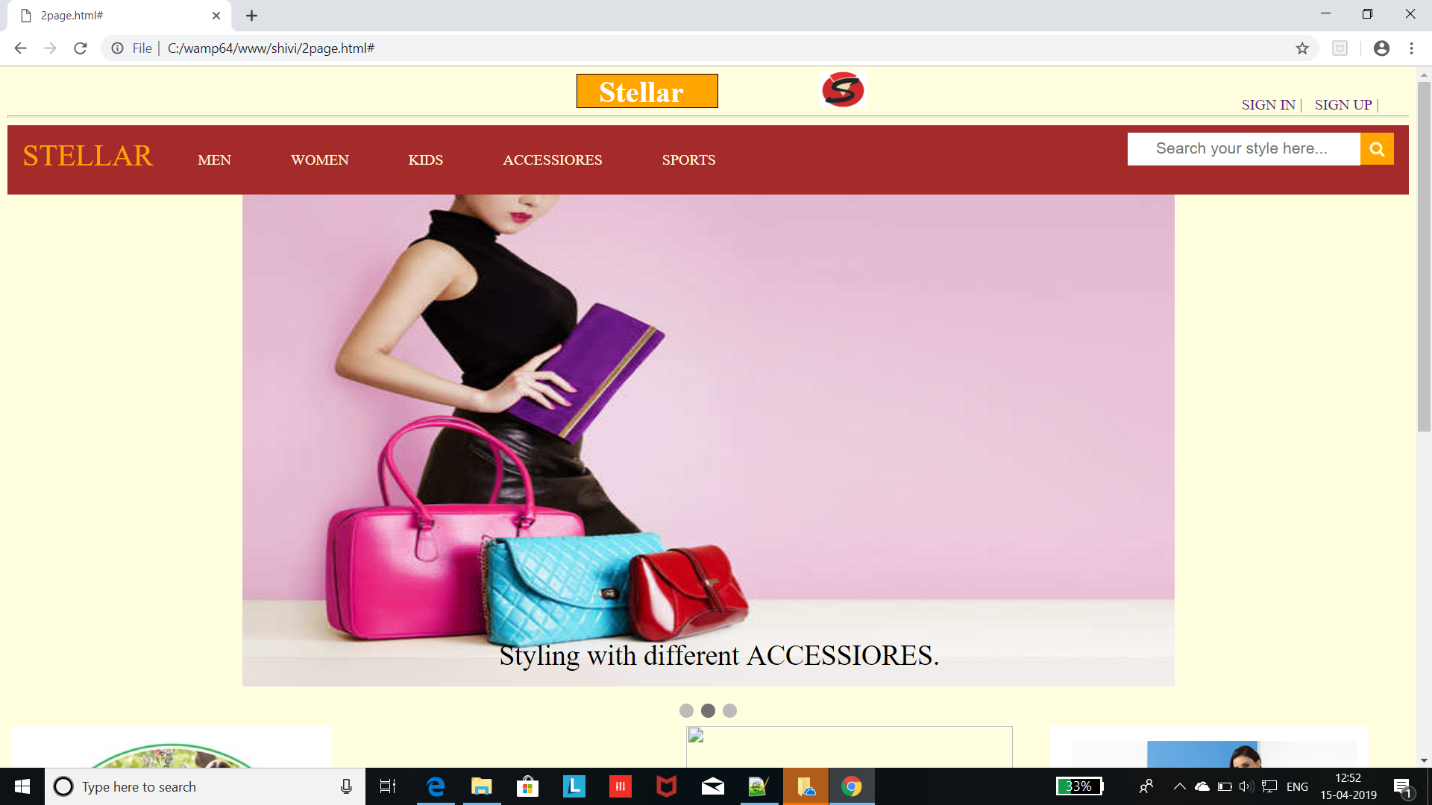
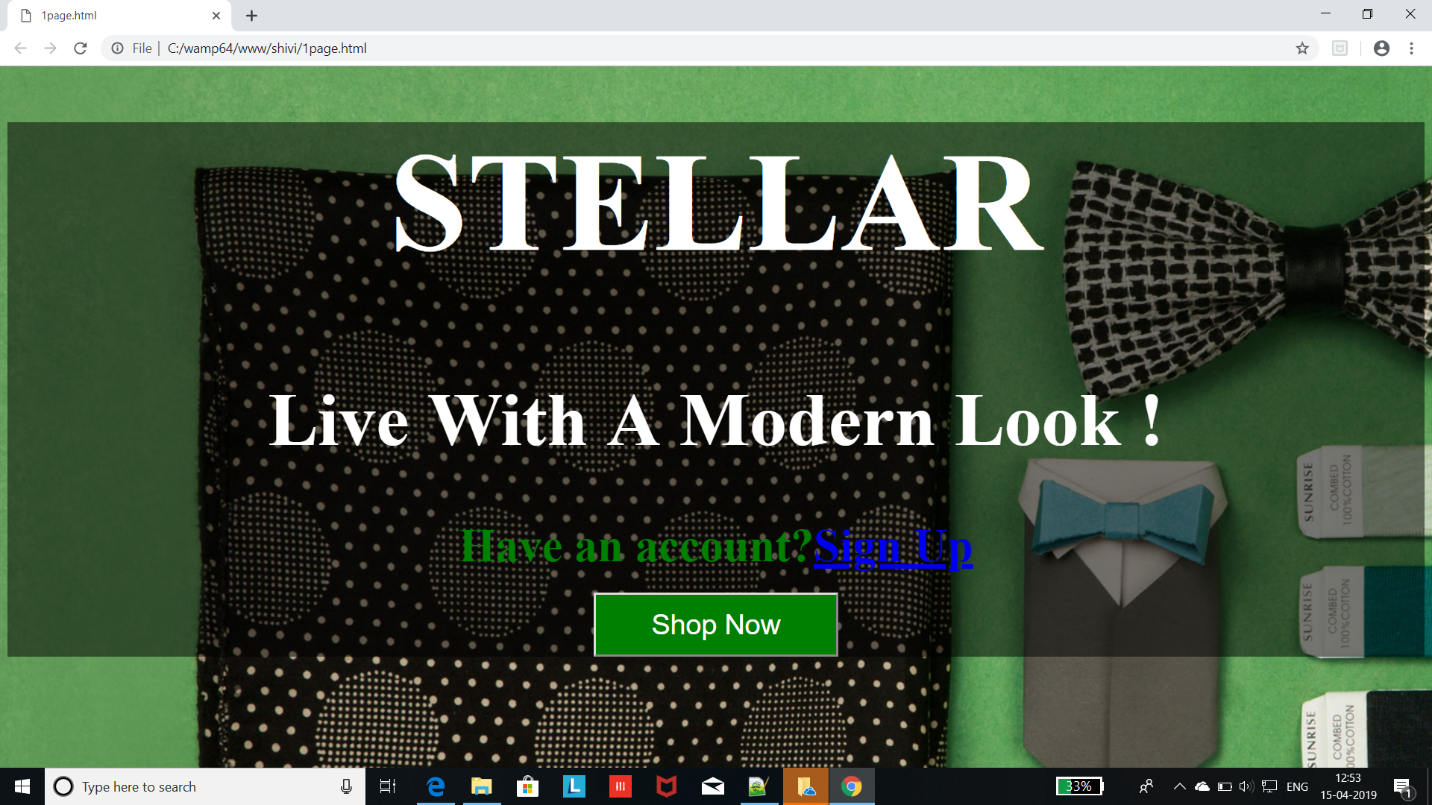
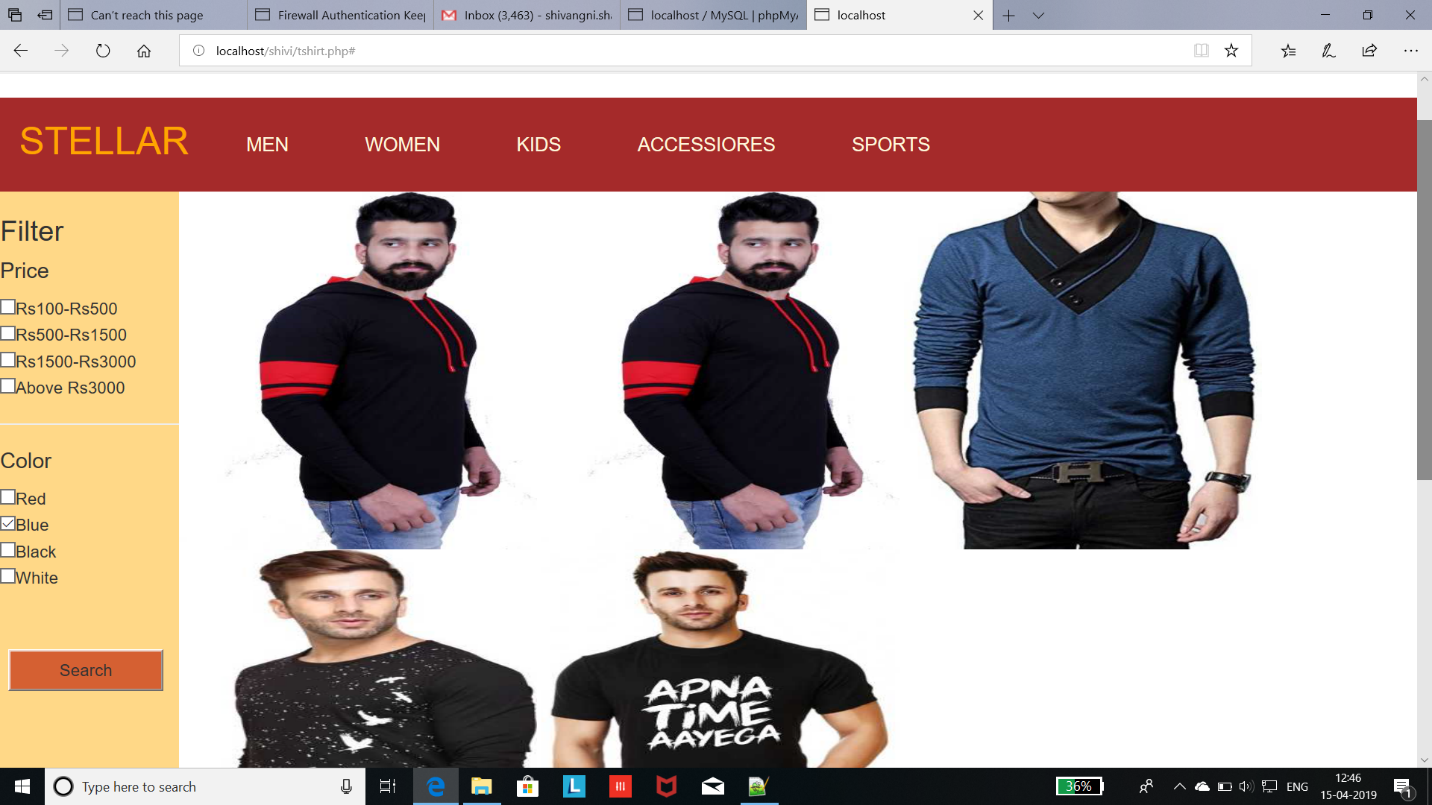
## 4.3 Entity Relationship Diagrams (ERD)



# A. Appendices

*.*

## A.1 Appendix 1



## A.2 Appendix 2

**1.Registration Code**

<?php

session\_start();

$con=mysqli\_connect('localhost','root');

if($con){

echo "connection successful";

}

else{

echo "connection unsuccessful";

}

mysqli\_select\_db($con,'shubh2');

$name= $\_POST['username'];

$pass=$\_POST['password'];

$q="select \* from user where name='$name' && password='$pass'";

$result=mysqli\_query($con,$q);

echo $result ;

$num=mysqli\_num\_rows($result);

if($num==1){

echo "existing user";

}

else{

$qy="insert into shubh2(name,password) values ('$name','$pass')";

mysqli\_query($con,$qy);

}

?>

**2. Login Code**

<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="UTF-8">

<title>login\_page</title>

<link rel='stylesheet prefetch' href='https://fonts.googleapis.com/css?family=Open+Sans:600'>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<h1><center>STELLER</center></h1>

<div class="login-wrap">

<div class="login-html">

<input id="tab-1" type="radio" name="tab" class="sign-in" checked><label for="tab-1" class="tab">Sign In</label>

<input id="tab-2" type="radio" name="tab" class="sign-up"><label for="tab-2" class="tab">Sign Up</label>

<div class="login-form">

<form class="sign-in-htm" action="validation.php" method="POST">

<div class="group">

<label for="user" class="label">Username</label>

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

</div>

<div class="group">

<label for="pass" class="label">Password</label>

<input id="password" name="password" type="password" class="input" data-type="password">

</div>

<div class="group">

<input id="check" type="checkbox" class="check" checked>

<label for="check"><span class="icon"></span> Keep me Signed in</label>

</div>

<div class="group">

<input type="submit" class="button" value="Sign In">

</div>

<div class="hr"></div>

<div class="foot-lnk">

<a href="#forgot">Forgot Password?</a>

</div>

</form>

<form class="sign-up-htm" action="registration.php" method="POST">

<div class="group">

<label for="user" class="label">Username</label>

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

</div>

<div class="group">

<label for="pass" class="label">Password</label>

<input id="password" name="password" type="password" class="input" data-type="password">

</div>

<div class="group">

<input type="submit" class="button" value="Sign Up">

</div>

<div class="hr"></div>

<div class="foot-lnk">

<label for="tab-1">Already Member?</a>

</div>

</form>

</div>

</div>

</div>

</body>

</html>

**3. Main website Code**

<?php

session\_start();

$conn1=mysqli\_connect("localhost","root","","e-commerce");

if(!isset($\_SESSION['username'])){

header('location:login.php')

}

if(isset($\_POST["add\_to\_cart"]))

{

if(isset($\_SESSION["shopping\_cart"]))

{

$item\_array\_id=array\_column($\_SESSION["shopping\_cart"],"item\_id");

if(!in\_array($\_GET["id"],$item\_array\_id))

{

$count=count($\_SESSION["shopping\_cart"]);

$item\_array = array(

'item\_name' => $\_POST["hidden\_name"],

'item\_price' => $\_POST["hidden\_price"],

'item\_quantity' => $\_POST["quantity"],

);

$\_SESSION["shopping\_cart"][$count]=$item\_array;

}

else

{

echo '<script>alert("Item Already added")</script>';

echo '<script>window.location="third.php"</script>';

}

}

else

{

$item\_array = array(

'item\_name' => $\_POST["hidden\_name"],

'item\_price' => $\_POST["hidden\_price"],

'item\_quantity' => $\_POST["quantity"]

);

$\_SESSION["shopping\_cart"][0]=$item\_array;

}

}

?>

<html>

<head>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"></link>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h2 class="text-center text-success">WELCOME <?php echo "$\_SESSION['username']; ?>"</h2>

<a href="login.php">LOGOUT</a>

</div>

<div class="cc" >

<div>

<div class="box"><h1>Stellar</h1></div>

<img class="c" src="p2logo.png">

<a href="" class="a" >SIGN IN |</a>

<a href="" class="b" >SIGN UP |</a>

</div>

<hr>

<div class="topnav">

<a href="home" style="color:orange";><font size="6">STELLAR</font></a>

<div class="men1">

<button class="men2"> <a href="home.html">MEN</a></button>

<div class="men">

<a href="tshirt.php">T-shirts</a>

<a href="#">Trousers</a>

<a href="#">Jeans</a>

<a href="#">Jackets</a>

<a href="#">Shoes</a>

<a href="#">Shirts</a>

</div></div>

<div class="men1">

<button class="men2"><a href="admin.html">WOMEN</a></button>

<div class="men">

<a href="#">Tops</a>

<a href="#">saree</a>

<a href="#">Dresses</a>

<a href="#">Suits</a>

<a href="#">Footwear</a>

<a href="#">Gowns</a>

</div></div>

<div class="men1">

<button class="men2"><a href="owner.html">KIDS</a></button>

<div class="men">

<a href="#">Shirt</a>

<a href="#">Jeans</a>

<a href="#">Dresses</a>

<a href="#">Frock</a>

<a href="#">Footwear</a>

<a href="#">Bags</a>

</div></div>

<div class="men1">

<button class="men2"><a href="customer.html">ACCESSIORES</a></button>

<div class="men">

<a href="#">Bags</a>

<a href="#">Jwellery</a>

<a href="#">Watches</a>

<a href="#">Bracelet</a>

<a href="#">Sunglasses</a>

<a href="#">Wallet</a>

</div></div>

<div class="men1">

<button class="men2"> <a href="About.html">SPORTS</a></button>

<div class="men">

<a href="#">FootBall</a>

<a href="#">Sport shoes</a>

<a href="#">Shorts</a>

<a href="#">Track suit</a>

<a href="#">Badminton</a>

</div></div>

</div>

<form action="#" method="post">

<div class="filter">

<h3>Filter</h3>

<h4>Price</h4>

<input type="checkbox" name="c1">Rs100-Rs500<br>

<input type="checkbox" name="c2">Rs500-Rs1500<br>

<input type="checkbox" name="c3">Rs1500-Rs3000<br>

<input type="checkbox" name="c4">Above Rs3000

<hr>

<h4>Color</h4>

<input type="checkbox" name="c5">Red<br>

<input type="checkbox" name="c6">Blue<br>

<input type="checkbox" name="c7">Black<br>

<input type="checkbox" name="c8">White<br>

<input type="submit" name="submit" class="butn" value="Search">

</div>

</div>

</form>

<?php { ?>

<?php

$servername = "localhost";

$username = "root";

$password = "";

$conn1=mysqli\_connect($servername,$username,$password,"e-commerce");

$img="select img from men";

$sql1="select \* from men where color='blue'";

$sql2="select img from men where price between 100 and 500";

$sql3="select img from men where color ='red'";

$sql4="select img from men where price between 500 and 1500";

$sql5="select img from men where color ='black'";

$sql6="select img from men where price between 1500 and 3000";

$sql7="select img from men where color ='white'";

$sql8="select img ,price,details from men where price greator than 3000";

$result=mysqli\_query($conn1,$img);

$res1=mysqli\_query($conn1,$sql1);

$res2=mysqli\_query($conn1,$sql2);

$res3=mysqli\_query($conn1,$sql3);

$res4=mysqli\_query($conn1,$sql4);

$res5=mysqli\_query($conn1,$sql5);

$res6=mysqli\_query($conn1,$sql6);

$res7=mysqli\_query($conn1,$sql7);

$res8=mysqli\_query($conn1,$sql8);

$get\_img="select img from men";

while($rows1 = mysqli\_fetch\_array($res1) ) {

if(isset($\_POST['submit'])){

if(!empty($\_POST['c6'])){?>

<div class="col-md-4">

<form method="post" action="third.php?action=add>

<div style="border:1px solid #333; background-color:#f1f1f1; border-radius:5px; padding:16px">

<img src="<?php echo $rows1["img"]; ?>" class="img-responsive"width="300" alt="shivi" height="300"/><br/>

<h4 class="text-info"><?php echo $rows1["name"]; ?></h4>

<h4 class="text-danger">$ <?php echo $rows1["price"]; ?></h4>

<input type="text" name="quantity" class="form-control" value="1" />

<input type="hidden" name="hidden\_name" value="<?php echo $rows1["name"]; ?>" />

<input type ="hidden" name="hidden\_price" value="<?php echo $rows1["price"]; ?>" />

<input type ="submit" name="add\_to\_cart" style="margin-top:5px;" class="btn btn-success" value="add to cart" />

<input type ="submit" name="buy now" style="margin-top:5px;" class="btn btn-success" value="buy now" onclick="1page.html"/>

</div>

</form>

</div>

<?php } ?>

<?php } ?><?php } ?>

<?php

while($rows2 = mysqli\_fetch\_array($res2) ) {

if(isset($\_POST['submit'])){

if(!empty($\_POST['c1'])){?>

<img src="<?php echo $rows2["img"]; ?>" class="image" width="300" alt="shivi" height="300">;

<?php } ?>

<?php } ?>

<?php } ?>

<?php

while($rows3 = mysqli\_fetch\_array($res3) ) {

if(isset($\_POST['submit'])){//to run PHP script on submit

if(!empty($\_POST['c5'])){?>

<img src="<?php echo $rows3["img"]; ?>" class="image" width="300" alt="shivi" height="300">;

<?php } ?>

<?php } ?>

<?php } ?>

<?php

while($rows4 = mysqli\_fetch\_array($res4) ) {

if(isset($\_POST['submit'])){//to run PHP script on submit

if(!empty($\_POST['c2'])){?>

<img src="<?php echo $rows4["img"]; ?>" class="image" width="300" alt="shivi" height="300">;

<?php } ?>

<?php } ?>

<?php } ?>

<?php

while($rows5 = mysqli\_fetch\_array($res5) ) {

if(isset($\_POST['submit'])){//to run PHP script on submit

if(!empty($\_POST['c7'])){?>

<img src="<?php echo $rows5["img"]; ?>" class="image" width="300" alt="shivi" height="300">;

<?php } ?>

<?php } ?>

<?php } ?>

<?php

while($rows6 = mysqli\_fetch\_array($res6) ) {

if(isset($\_POST['submit'])){//to run PHP script on submit

if(!empty($\_POST['c3'])){?>

<img src="<?php echo $rows6["img"]; ?>" class="image" width="300" alt="shivi" height="300">;

<?php } ?>

<?php } ?>

<?php } ?>

<?php

while($rows7 = mysqli\_fetch\_array($res7) ) {

if(isset($\_POST['submit'])){//to run PHP script on submit

if(!empty($\_POST['c8'])){?>

<a href="a.php?$a=$rows7['img'];"><img src="<?php echo $rows7["img"];?>" class="image" width="300" alt="shivi" height="300">;</a>

<?php } ?>

<?php } ?>

<?php } ?>

<?php } ?>

<div style="clear:both"></div>

<br/>

<h3>Order Details</h3>

<div class=table-responsive></div>

<table class="table table-bordered">

<tr>

<th width="10%">Item Name </th>

<th width="10%">quantity </th>

<th width="20%">price </th>

<th width="15%">total </th>

<th width="5%">Action </th>

</tr>

<?php

if(!empty($\_SESSION["shopping\_cart"]))

{

$total=0;

foreach($\_SESSION["shopping\_cart"] as $keys => $values)

{

?>

<tr>

<td><?php echo $values["item\_name"]; ?></td>

<td><?php echo $values["item\_quantity"]; ?></td>

<td><?php echo $values["item\_price"]; ?></td>

<td><?php echo (int)$values["item\_quantity"] \*(int) $values["item\_price"] ;?></td>

<td><a href="third.php?action=delete&id=<?php echo $values["item\_id"]; ?>"><span class="text-danger">Remove</span></a></td>

</tr>

<?php

$total=(int)$total+((int)$values["item\_quantity"] \* (int)$values["item\_price"]);

}

}

?>

<tr>

<td colspan="3" align="right">Total</td>

<td align="right"> <?php echo number\_format($total,2); ?></td>

<td></td>

</tr>