

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018



DBMS MINI PROJECT

REPORT ON

“MOBILE STORE MANAGEMENT SYSTEM”

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

VAISHNAVI U KULKARNI(1KG21CS115)

YASHWIN KUMAR R(1KG21CS125)

Under the Guidance of

Mrs.SOUGANDHIKA NARAYAN

Assistant Professor
Department of CSE
K.S.S.E.M

Mrs.DEEPASHREE N

Assistant Professor
Department of CSE
K.S.S.E.M



KSSEM
K'S SCHOOL OF ENGINEERING AND MANAGEMENT

Department of Computer Science & Engineering
K. S. SCHOOL OF ENGINEERING AND MANAGEMENT

#15, Mallasandra, off. Kanakapura Road, Bengaluru – 560109

2023-2024

K. S. SCHOOL OF ENGINEERING AND MANAGEMENT BENGALURU - 560109

Department of Computer Science & Engineering



CERTIFICATE

This is to certify that the **DBMS MINI PROJECT** entitled “**MOBILE STORE MANAGEMENT SYSTEM**” presented by Miss **VAISHNAVI U KULKARNI**, USN:1KG21CS115, Mr **YASHWIN KUMAR R** USN:1KG21CS125 of **V SEMESTER** in partial fulfillment of the award of **BACHELOR OF ENGINEERING** in **COMPUTER SCIENCE & ENGINEERING** in **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**, Belagavi during the academic year **2023-2024**. The **DBMS MINI PROJECT** has been approved as it satisfies the academic requirements in respect of **DBMS LABORATORY & MINI PROJECT(21CSL55)** prescribed for the Bachelor of Engineering degree.

Mrs.Sougandhika Narayan
& Mrs.Deepashree N
Assistant Professor, CSE
K.S.S.E.M., Bengaluru

Dr. K.Venkata Rao
Prof. & Head, CSE
K.S.S.E.M., Bengaluru

Dr. K.RAMANARASIMHA
Principal / Director
K.S.S.E.M., Bengaluru

Name of the Student:	VAISHNAVI U KULKARNI	YASHWIN KUMAR R
USN:	1KG21CS115	1KG21CS125

Signature of the Student:

	Name of the examiners	Signature with date
1.	_____	_____
2.	_____	_____

ACKNOWLEDGEMENT

The successful presentation of the **DBMS MINI PROJECT** would be incomplete without the mention of the people who made it possible and whose constant guidance crowned my effort with success.

I would like to extend my gratitude to the **MANAGEMENT, KAMMAVARI SANGHAM**, Bengaluru, for providing all the facilities to present the Data Base Application Mini Project.

I would like to extend my gratitude to **Dr. K. RAMA NARASIMHA**, Principal / Director, K. S. School of Engineering and Management, Bengaluru, for facilitating me to present the Data Base Application Mini Project.

I thank **Dr. K Venkata Rao**, Professor and Head, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, for his encouragement.

I would like to thank our Project Guide, **Mrs.Sougandhika Narayan**, and **Mrs.Deepashree N** An Assistant Professor, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, for their constant guidance and inputs.

I would like to thank all the **Teaching** Staff and **Non-Teaching** Staff of the college for their co-operation.

Finally, I extend my heart-felt gratitude to my **family** for their encouragement and support without which I wouldn't have come so far. Moreover, I thank all my **friends** for their invaluable support and cooperation.

Vaishnavi U Kulkarni(1KG21CS115)

Yashwin Kumar R(1KG21CS125)

ABSTRACT

Information Technology has been growing rapidly, with the course of time, as technology and human knowledge converge to a twilight, need for betterment of their own sustaining life also increases. Technology has given many gifts to the human race to lead a sophisticated life. One of the fabulous gifts was Mobile Phone which was invented by Joel Engel and Richard Frenkiel in 1983. The Mobile Store Management System is developed for desktop systems to facilitate mobile shop owners' management of customer details and inventory data, which will include mobile phones and accessories. It can be used efficiently for physically separated shops in different locations. This software will provide in a simple and easy to operate user interface, which can be managed by any user without having prior in-depth knowledge of the computer system. One can use this software to get a sales report. Administrators can pull data, from any location from the server. This software is a complete package for small organizations which will allow them to keep track of their sales and inventory, and provide a computerized billing system. There are various applications with more complex implementation and features available in the market, but they are generally very expensive. Therefore, creating an application with the basic requirement of low cost is essential for small organizations. This application will allow stores to manage customer details, keep inventory of all products and purchase information, in a very simple way, using a state-of-the-art software application. It will automatically generate invoices and update inventory.

TABLE OF CONTENTS

Contents	Page No.
Acknowledgement	I
Abstract	II
Table of Contents	III
List of Figures	IV
List of Tables	

TABLE OF CONTENTS

Chapter 1 INTRODUCTION	1
1.1 OVERVIEW	1
1.2 PROBLEM STATEMENT	1
1.3 DATABASE MANAGEMENT SYSTEM	2
1.4 SQL	2
1.5 HTML / JAVASCRIPT	3
Chapter 2 REQUIREMENTS SPECIFICATION	4
2.1 OVERALL DESCRIPTION	4
2.2 SPECIFIC REQUIREMENTS	4
2.3 SOFTWARE REQUIREMENTS	4
2.5 HARDWARE REQUIREMENTS	4
2.6 TECHNOLOGY	5
Chapter 3 DETAILED DESIGN	6
3.1 SYSTEM DESIGN	6
3.2 ENTITY RELATIONSHIP DIAGRAM	7
3.3 RELATIONAL SCHEMA	9
3.4 DESCRIPTION OF TABLES	10
Chapter 4 TESTING	12
4.1 SOFTWARE TESTING	12
4.2 MODULE TESTING AND INTEGRATION	12
4.3 LIMITATIONS	12

Chapter 5	IMPLEMENTATION	13
5.1	HOME	13
5.2	LOGIN	19
5.3	PRODUCTS	21
Chapter 6	SNAP SHOTS	23
6.1	LOGIN PAGE	23
6.2	REGISTRATION PAGE	23
6.3	HOME PAGE	24
6.4	SMARTPHONES	24
6.5	ACCESSORIES	25
6.6	PAYMENTS	25
6.7	CART	26
6.8	ABOUT US	27
Chapter 8	CONCLUSION	28
Chapter 9	FUTURE ENHANCEMENTS	29
	REFERENCES	30

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	JSP ARCHITECTURE	6
3.2	ENHANCED ER DIAGRAM OF MOBILE STORE MANAGEMENT SYSTEM	8
3.3	ER DIAGRAM OF MOBILE STORE MANAGEMENT SYSTEM	8
3.4	SCHEMA DIAGRAM	9

LIST OF TABLES

Table No.	Table Name	Page No.
5.1	BRAND	10
5.2	USER	10
5.3	CART	10
5.4	PAYMENT	10

Chapter 1

INTRODUCTION

OVERVIEW

The Mobile Store Management System is developed for desktop systems to facilitate mobile shop owners' management of customer details and inventory data, which will include mobile phones and accessories. It can be used efficiently for physically separated shops in different locations. This software will provide in a simple and easy to operate user interface, which can be managed by any user without having prior in- depth knowledge of the computer system. One can use this software to get a sales report. Administrators can pull data, from any location from the server. This software is a complete package for small organizations which will allow them to keep track of their sales and inventory, and provide a computerized billing system. There are various applications with more complex implementation and features available in the market, but they are generally very expensive. Therefore, creating an application with the basic requirement of low cost is essential for small organizations. This application will allow stores to manage customer details, keep inventory of all products and purchase information, in a very simple way, using a state-of the-art software application. It will automatically generate invoices and update inventory.

PROBLEM STATEMENT

The main aim of “Mobile store management system” is to make an easy interface for users, customers to use the provided interface effectively to purchase the required accessories.

DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

SQL

SQL is a standard language for storing, manipulating and retrieving data in databases.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

HTML /JavaScript

HTML is a markup language used for structuring and presenting content on the web and the fifth and current major version of the HTML standard. HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and

introduces markup and application programming interfaces (APIs) for complex web applications.

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

JAVA CONNECTIONS

To connect the database with the front end we use a java connector JDBC (Java Database Connectivity). JDBC is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation.

To achieve connectivity, we use JSPs (Java Server Pages) in this project. Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. JSP is similar to PHP and ASP, but it uses the Java programming language.

Chapter 2

REQUIREMENTS SPECIFICATION

A computerized way of handling information about property and users details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

SPECIFIC REQUIREMENTS

The specific requirements of the Mobile Store System are stated as follows:

SOFTWARE REQUIREMENTS

- ☐ IDE - NetBeans8.2
- ☐ Web Browser – Firefox 50 or later, Google Chrome – 60 or later
- ☐ Database support - MySQL5.7
 - ☐ MySQL Server5.7
 - ☐ MySQL Shell1.0.10
 - ☐ MySQL Workbench
- ☐ Operating system – Windows 7 / Ubuntu16.04
- ☐ JDK 1.8
- ☐ Server deployment - Tomcat server / Glassfish Server

HARDWARE REQUIREMENTS

- ☐ Processor – Pentium IV or above
- ☐ RAM – 2 GB or more
- ☐ Hard disk – 3 GB or more
- ☐ Monitor – VGA of 1024x768 screen resolution
- ☐ Keyboard and Mouse

TECHNOLOGY

- ❑ HTML is used for the front-end design. It provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
- ❑ CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used set the visual style of web page sand user interfaces written in HTML and XHTML,the language can be applied to any XML document.
- ❑ SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.
- ❑ Java Server pages is a simple yet powerful technology for creating and maintaining dynamic-content web pages. It is based on the Java programming language. It can be thought of as an extension to servlet because it provides more functionality than servlet A JSP page consists of HTML tags and JSP tags. The jsp pages are easier to maintain than servlet because we can separate designing and development.
- ❑ We require a JDBC connection between the front end and back end components to write to the database and fetch required data.

Chapter 3

DETAILED DESIGN

SYSTEM DESIGN

The web server needs a JSP engine, i.e., a container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages. A JSP container works with the Web server to provide the runtime environment and other services a JSP needs. It knows how to understand the special elements that are part of JSPs. This server will act as a mediator between the client browser and a database.

The following diagram shows the JSP architecture.

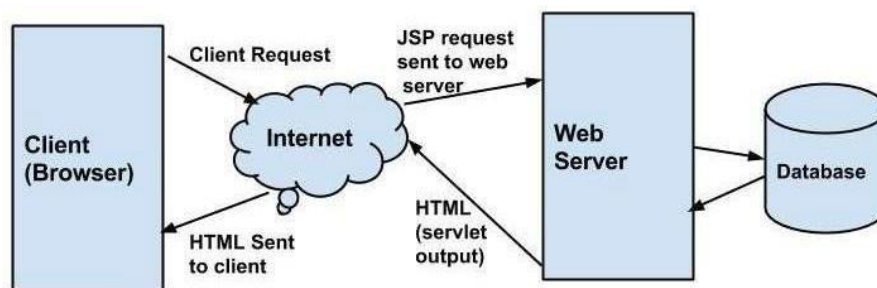


Fig. 3.1: JSP Architecture

Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic (constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database architects may use one of several languages to create databases, such as structured query language.

ENTITY RELATIONSHIP DIAGRAM

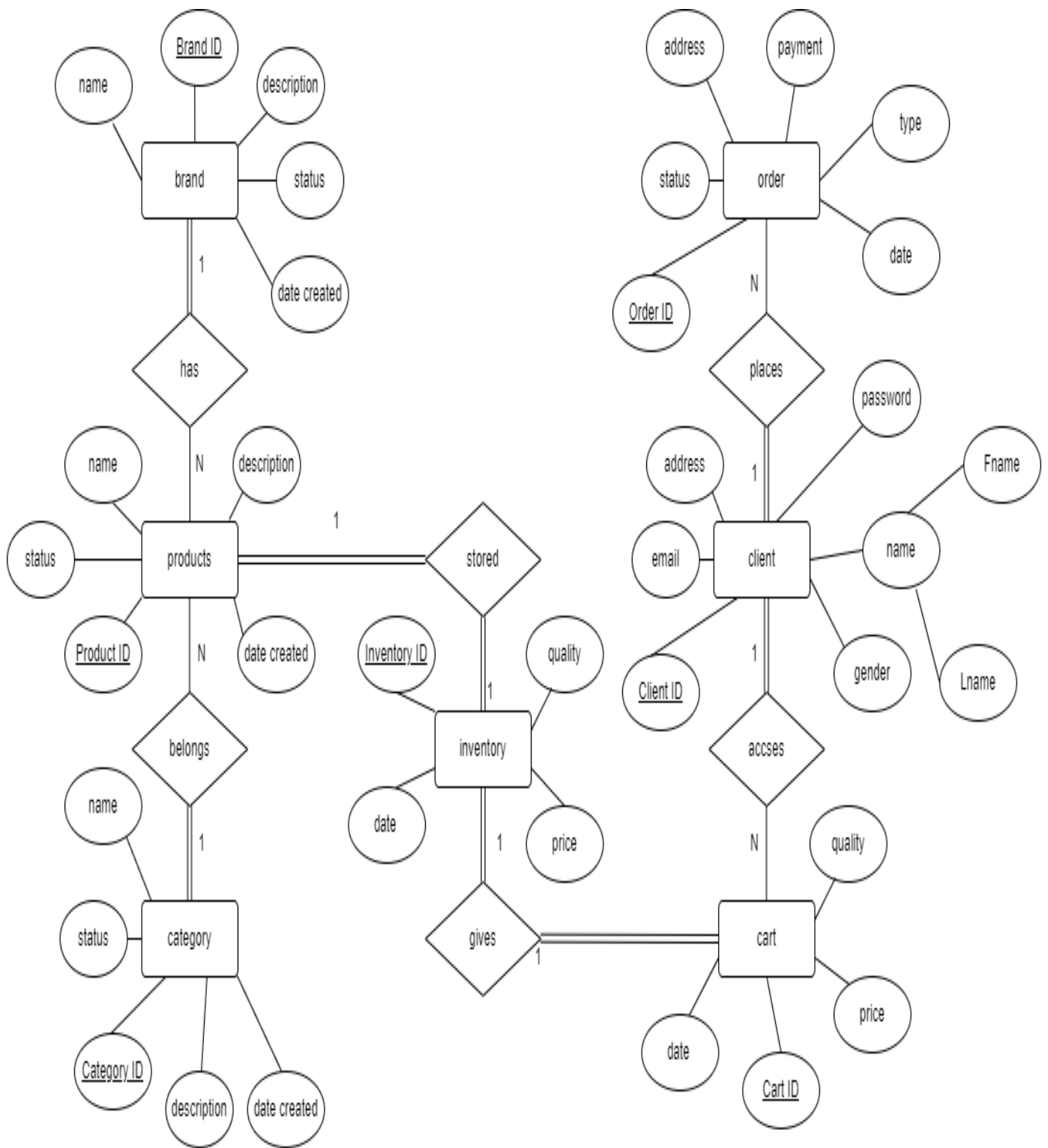
An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three-schema approach to software engineering. While useful for organizing data that can be re presented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional.



E-R Diagram for Mobile Store Management

RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.

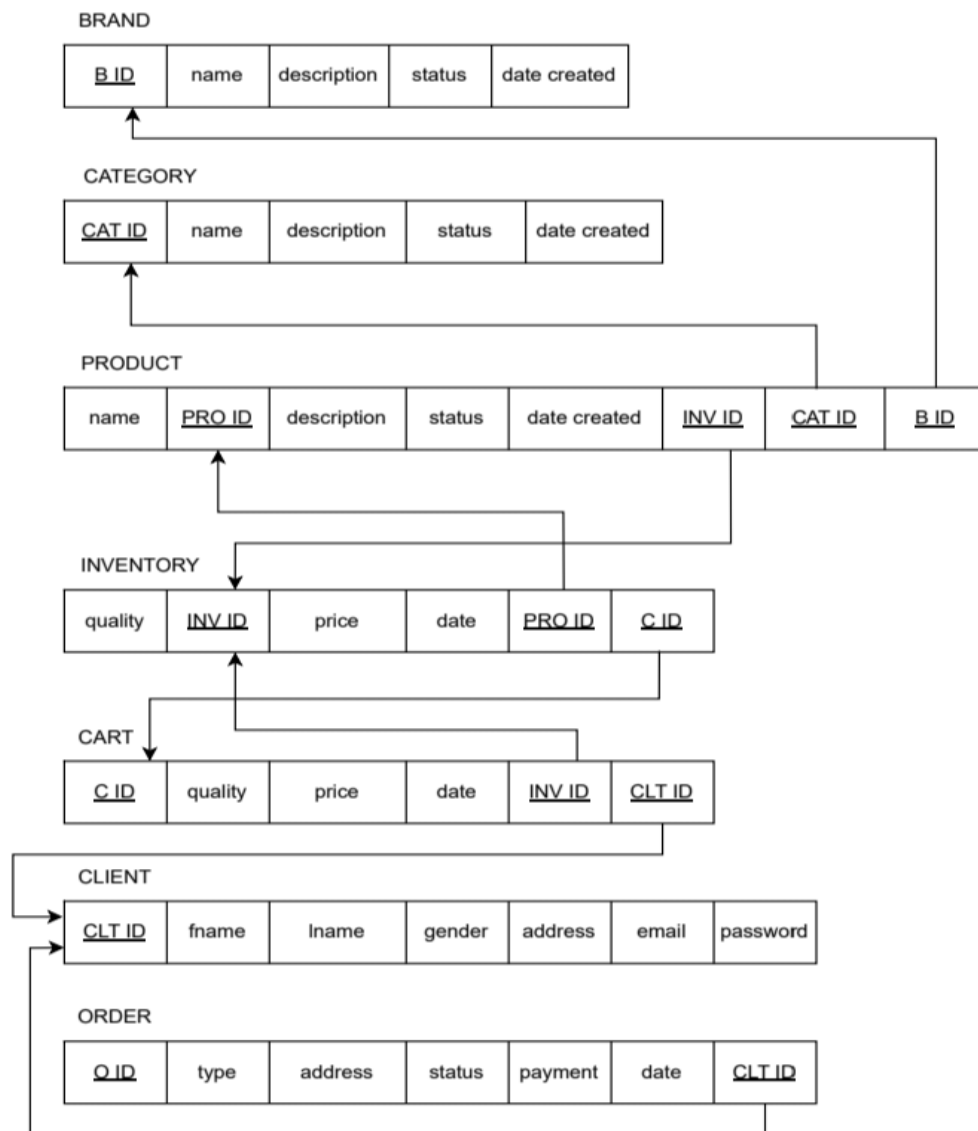


Fig. 3.4: Schema diagram

DESCRIPTION OF TABLES

Brand: It stores the brand details.

Brand_id : Unique brand id done by auto increment.

Brand_name : Name of the brand.

Description : It has the brand description.

Status : It has the status of the brand.

Date_created : The date of the product created.

Product_id : Unique product id done by auto increment.

Product_name : Name of the product.

Specifications : It has the specifications of the product.

User : It stores the user details.

User_id : Unique user id done by auto increment.

Firstname : It stores the users first name.

lastname : It stores the users last name.

Password : password associated with the user to login into the system.

Last_login : details of the date and time of the last login.

Date_added : date of the details added.

Date_updated : date of the details updated.

Cart : it stores the details of the cart.

Cart_ID : unique card ID done by auto increment.

Inventory_id : it has the details of the inventory.

Price : it has the details of the product price.

Quality : it has the details of the quality.

Order_list : it has the list of orders added to the cart.

Date_created : it has the details of the product created.

Date_updated : it has the details of the inventory updated.

Payment : it stores the details of the payment and order.

Delivery_address : it has the details of the delivery address the product.

Payment_method : it has types of payment is done.

order_type : it describes the order type.

Amount : it describes the price of the order.

Status : it describes status of the product or order.

Chapter 4

TESTING

SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

MODULE TESTING AND INTEGRATION

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 recommends testing the smaller building blocks of the program. It is largely 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 simple mentoring of parallelism to the testing process by giving the opportunity to test multiple modules simultaneously.

The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

LIMITATIONS

- ☐ Does not able to handle multiple users simultaneously.
- ☐ User's session timing is not recorded.
- ☐ No integration of other database systems.
- ☐ Only restricted to limited number of operations to be performed.

Chapter 5

IMPLEMENTATION

HOME

```
<style>
.carousel-item>img{
    object-fit:fill !important;
}
#carouselExampleControls .carousel-inner{
    height:280px !important;
}
</style>
<?php
$brands = isset($_GET['b']) ? json_decode(urldecode($_GET['b'])) : array();
?>
<section class="py-0">
    <div class="container-fluid">
        <div class="row">
            <div class="col-lg-2 px-1 border-right text-sm position-sticky ">
                <h4><b>Brands</b></h4>
                <ul class="list-group">
                    <a href="" class="list-group-item list-group-item-action">
                        <div class="icheck-primary d-inline">
                            <input type="checkbox" id="brandAll" >
                            <label for="brandAll">
                                All
                            </label>
                        </div>
                    </a>
                <?php
                $qry = $conn->query("SELECT * FROM brands where status =1 order by name asc");
                while($row=$qry->fetch_assoc()):
                ?>
                <li class="list-group-item list-group-item-action">
                    <div class="icheck-primary d-inline">
                        <input type="checkbox" id="brand-item-<?php echo $row['id'] ?>" <?php echo
in_array($row['id'],$brands) ? "checked" : "" ?> class="brand-item" value="<?php echo $row['id'] ?>">
                        <label for="brand-item-<?php echo $row['id'] ?>">
                            <?php echo $row['name'] ?>
                        </label>
                    </div>
                </li>
                <?php endwhile; ?>
```

```

</ul>
</div>
<div class="col-lg-10 py-2">
  <div class="row">
    <div class="col-md-12">
      <div id="carouselExampleControls" class="carousel slide bg-dark" data-ride="carousel">
        <div class="carousel-inner">
          <?php
            $upload_path      =      "uploads/banner";
            if(is_dir(base_app.$upload_path)):
              $file= scandir(base_app.$upload_path);
              $_i = 0;
              foreach($file      as      $img):
                if(in_array($img,array(' ','..')))
                  continue;
                $_i++;

              ?>
              <div class="carousel-item h-100 <?php echo $_i == 1 ? "active" : " ?>">
                ">
              </div>
            <?php endforeach; ?>
            <?php endif; ?>
          </div>
          <button class="carousel-control-prev" type="button" data-target="#carouselExampleControls"
data-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Previous</span>
          </button>
          <button class="carousel-control-next" type="button" data-target="#carouselExampleControls"
data-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Next</span>
          </button>
        </div>
      </div>
    </div>
  </div>
  <div class="container px-4 px-lg-5 mt-5">
    <div class="row gx-4 gx-lg-4 row-cols-md-3 row-cols-xl-4 ">
      <?php
        $where      =      "";
        if(count($brands)>0)
          $where = " and p.brand_id in (".implode(",",$brands).") ";

        $products = $conn->query("SELECT p.*,b.name as bname FROM products p inner join brands b on
p.brand_id = b.id where p.status = 1 {$where} order by rand() ");
        while($row = $products->fetch_assoc()):
          $upload_path = base_app.'/uploads/product_'.$row['id'];
          $img = "";

```

Dept. of CSE, KSSEM | 2023-24
14

```
if(checked == total)
    location.href="/?";
else
    location.href="/?b="+encodeURIComponent(_b);
}
function check_filter(){
    var checked = $('.brand-item:checked').length
    var total = $('.brand-item').length
    if(checked == total){
        $('#brandAll').attr('checked',true)
    }else{
        $('#brandAll').attr('checked',false)
    }
    if('<?php echo isset($_GET['b']) ?>' == "")
        $('#brandAll,.brand-item').attr('checked',true)
}
$(function(){
    check_filter()
    $('#brandAll').change(function(){
        if($(this).is(':checked') == true){
            $('.brand-item').attr('checked',true)
        }else{
            $('.brand-item').attr('checked',false)
        }
        _filter()
    })
    $('.brand-item').change(function(){
        _filter()
    })
})
</script>
```

INDEX

```
<?php require_once('config.php'); ?>
<!DOCTYPE html>
<html lang="en">
<?php require_once('inc/header.php') ?>
<?php if($_settings->chk_flashdata('success')): ?>
<script>
    $(function(){
        alert_toast("<?php echo $_settings->flashdata('success') ?>", 'success')
    })
</script>
<?php endif;?>
<body>
<?php require_once('inc/topBarNav.php') ?>
<?php $page = isset($_GET['p']) ? $_GET['p'] : 'home'; ?>
<?php
    if(!file_exists($page.".php") && !is_dir($page)){
        include '404.html';
    }else{
        if(is_dir($page))
            include $page.'/index.php';
        else
            include $page.'.php';
    }
?>
<?php require_once('inc/footer.php') ?>
<div class="modal fade" id="confirm_modal" role='dialog'>
    <div class="modal-dialog modal-md modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Confirmation</h5>
            </div>
            <div class="modal-body">
                <div id="delete_content"></div>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-primary" id='confirm' onclick="">Continue</button>
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
            </div>
        </div>
    </div>
</div>
<div class="modal fade" id="uni_modal" role='dialog'>
    <div class="modal-dialog rounded-0 modal-md modal-dialog-centered" role="document">
        <div class="modal-content rounded-0">
            <div class="modal-header">
```

```

    <h5 class="modal-title"></h5>
</div>
<div class="modal-body">
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-primary" id='submit' onclick="$('#uni_modal
form').submit()">Save</button>
    <button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>
</div>
</div>
</div>
</div>
<div class="modal fade" id="uni_modal_right" role='dialog'>
    <div class="modal-dialog rounded-0 modal-full-height modal-md" role="document">
        <div class="modal-content rounded-0">
            <div class="modal-header">
                <h5 class="modal-title"></h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span class="fa fa-arrow-right"></span>
                </button>
            </div>
            <div class="modal-body">
</div>
</div>
</div>
</div>
<div class="modal fade" id="viewer_modal" role='dialog'>
    <div class="modal-dialog modal-md" role="document">
        <div class="modal-content">
            <button type="button" class="btn-close" data-dismiss="modal"><span class="fa fa-
times"></span></button>
            <img src="" alt="">
        </div>
</div>
</div>
</div>
</body>
</html>

```


LOGIN

```
<style>
  #uni_modal .modal-content>.modal-footer,#uni_modal .modal-content>.modal-header{
    display:none;
  }
</style>
<div class="container-fluid">
  <div class="row">
    <h3 class="float-right">
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
    </h3>
    <div class="col-lg-12">
      <h3 class="text-center">Login</h3>
      <hr>
      <form action="" id="login-form">
        <div class="form-group">
          <label for="" class="control-label">Email</label>
          <input type="email" class="form-control form" name="email" required>
        </div>
        <div class="form-group">
          <label for="" class="control-label">Password</label>
          <input type="password" class="form-control form" name="password" required>
        </div>
        <div class="form-group d-flex justify-content-between">
          <a href="javascript:void()" id="create_account">Create Account</a>
          <button class="btn btn-primary btn-flat">Login</button>
        </div>
      </form>
    </div>
  </div>
</div>
<script>
  $(function(){
    $('#create_account').click(function(){
      uni_modal("", "registration.php", "mid-large")
    })
    $('#login-form').submit(function(e){
      e.preventDefault();
      start_loader()
      if($('.err-msg').length > 0)
        $('.err-msg').remove();
      $.ajax({
        url:_base_url_+"classes/Login.php?f=login_user",
        method:"POST",
        data:$(this).serialize(),
```

```
dataType:"json",
error:err=>{
  console.log(err)
  alert_toast("an error occured",'error')
  end_loader()
},
success:function(resp){
  if(typeof resp == 'object' && resp.status == 'success'){
    alert_toast("Login Successfully",'success')
    setTimeout(function(){
      location.reload()
    },2000)
  }else if(resp.status == 'incorrect'){
    var _err_el = $('<div>')
    _err_el.addClass("alert alert-danger err-msg").text("Incorrect Credentials.")
    $('#login-form').prepend(_err_el)
    end_loader()

  }else{
    console.log(resp)
    alert_toast("an error occured",'error')
    end_loader()
  }
}
})
})
})
</script>
```

PRODUCTS

```
<?php
$title = "";
$sub_title = "";
if(isset($_GET['c']) && isset($_GET['s'])){
    $cat_qry = $conn->query("SELECT * FROM categories where md5(id) = '{$_GET['c']}'");
    if($cat_qry->num_rows > 0){
        $result = $cat_qry->fetch_assoc();
        $title = $result['category'];
        $cat_description = $result['description'];
    }
    $sub_cat_qry = $conn->query("SELECT * FROM sub_categories where md5(id) = '{$_GET['s']}'");
    if($sub_cat_qry->num_rows > 0){
        $result = $sub_cat_qry->fetch_assoc();
        $sub_title = $result['sub_category'];
        $sub_cat_description = $result['description'];
    }
}
elseif(isset($_GET['c'])){
    $cat_qry = $conn->query("SELECT * FROM categories where md5(id) = '{$_GET['c']}'");
    if($cat_qry->num_rows > 0){
        $result = $cat_qry->fetch_assoc();
        $title = $result['category'];
        $cat_description = $result['description'];
    }
}
elseif(isset($_GET['s'])){
    $sub_cat_qry = $conn->query("SELECT * FROM sub_categories where md5(id) = '{$_GET['s']}'");
    if($sub_cat_qry->num_rows > 0){
        $result = $sub_cat_qry->fetch_assoc();
        $sub_title = $result['sub_category'];
        $sub_cat_description = $result['description'];
    }
}
?>
<!-- Header-->
<header class="bg-dark py-5" id="main-header">
    <div class="container px-4 px-lg-5 my-5">
        <div class="text-center text-white">
            <h1 class="display-4 fw-bolder"><?php echo $title ?></h1>
            <p class="lead fw-normal text-white-50 mb-0"><?php echo $sub_title ?></p>
        </div>
    </div>
</header>
<!-- Section-->
<section class="py-5">
<div class="container-fluid row">
<?php if(isset($_GET['c'])): ?><div
class="col-md-3 border-right mbpb-
3">
```

```

<h3><b>Sub Categories</b></h3>
<div class="list-group">
    <a href="/?p=products&c=<?php echo $_GET['c'] ?>" class="list-group-item <?php echo
!isset($_GET['s']) ? "active" : "" ?>">All</a>
    <?php
    $sub_cat = $conn->query("SELECT * FROM sub_categories where md5(parent_id) =
'{$_GET['c']}' ");
    while($row = $sub_cat->fetch_assoc()):
    ?>
        <a href="/?p=products&c=<?php echo $_GET['c'] ?>&s=<?php echo md5($row['id']) ?>"
class="list-group-item <?php echo isset($_GET['s']) && $_GET['s'] == md5($row['id']) ? "active" : ""
?>"><?php echo $row['sub_category'] ?></a>
        <?php endwhile; ?>
    </div>
    <hr>
</div>
<?php endif; ?>
<div class="<?php echo isset($_GET['c'])? 'col-md-9': 'col-lg-10 offset-md-1' ?>">
    <div class="container-fluid p-0">
        <?php if(isset($_GET['search'])): ?>
            <h4 class="text-center py-5"><b>Search Results for '<?php echo $_GET['search'] ?>'</b></h4>
            <?php endif; ?>
        <div class="row gx-2 gx-lg-2 row-cols-1 row-cols-md-3 row-cols-xl-4">

            <?php
            $whereData = "";
            if(isset($_GET['search']))
                $whereData = " and (p.name LIKE '%{$_GET['search']}'% or b.name LIKE
'%{$_GET['search']}'% or p.specs LIKE '%{$_GET['search']}'% )";
            elseif(isset($_GET['c']) && isset($_GET['s']))
                $whereData = " and (md5(category_id) = '{$_GET['c']}' and md5(sub_category_id) =
'{$_GET['s']}' )";
            elseif(isset($_GET['c']) && !isset($_GET['s']))
                $whereData = " and md5(category_id) = '{$_GET['c']}' ";
            elseif(isset($_GET['s']) && !isset($_GET['c']))
                $whereData = " and md5(sub_category_id) = '{$_GET['s']}' ";
            $products = $conn->query("SELECT p.*,b.name as bname FROM products p inner join brandsb
on p.brand_id = b.id where p.status = 1 {$whereData} order by rand() ");
            while($row = $products->fetch_assoc()):

```

Chapter 6

SNAPSHOTS

This chapter consists of working screenshots of the project.

LOGIN PAGE

This is the login page for existing users and is the first page shown to any customer.

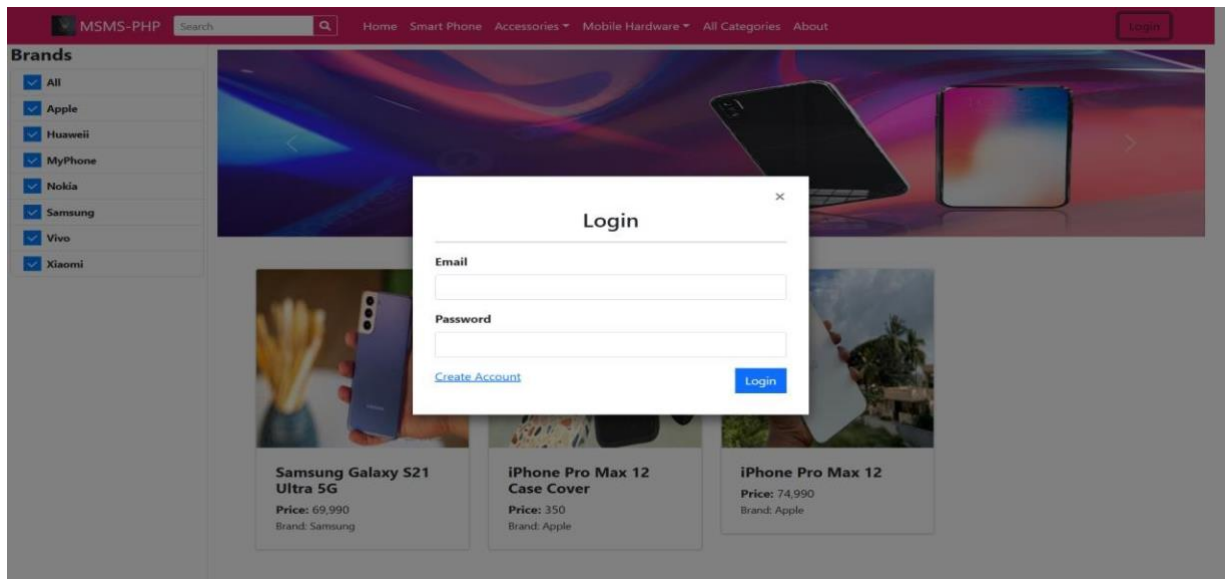


Fig 6.1: Login page

REGISTRATION

This page helps the user to create a new account or register.

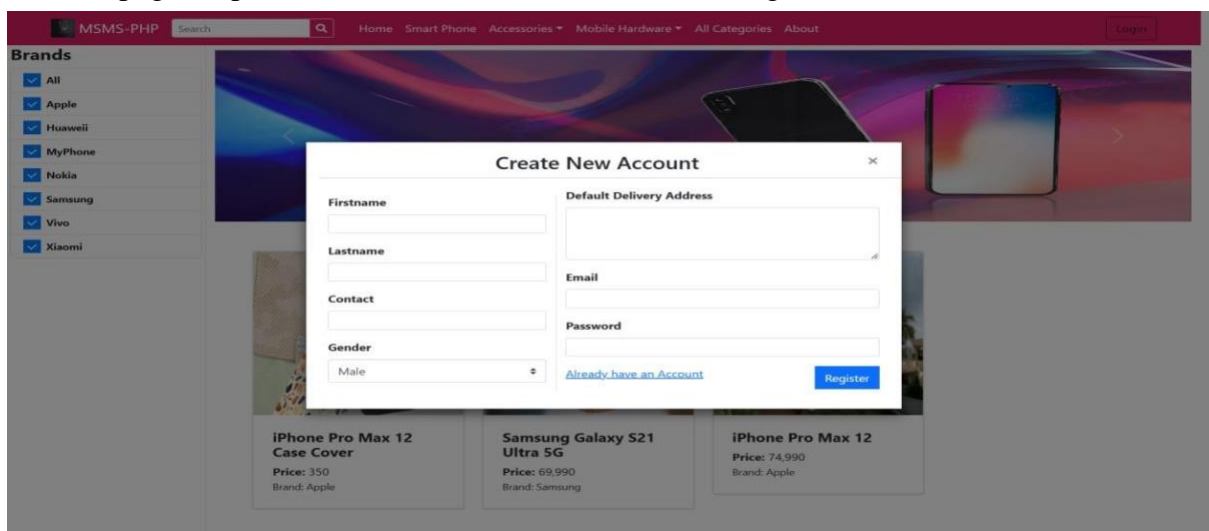


Fig 6.2: Registration Page

HOME PAGE

First home page shown to customers after login.

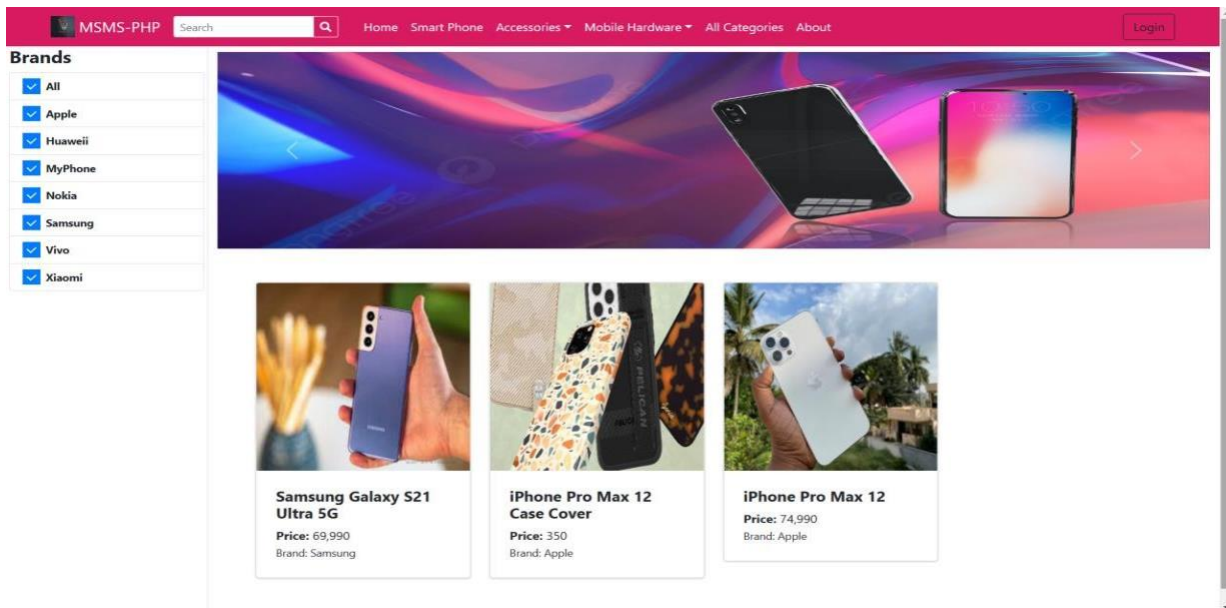


Fig 6.3: Home page

SMART PHONE

List of Smart Phones

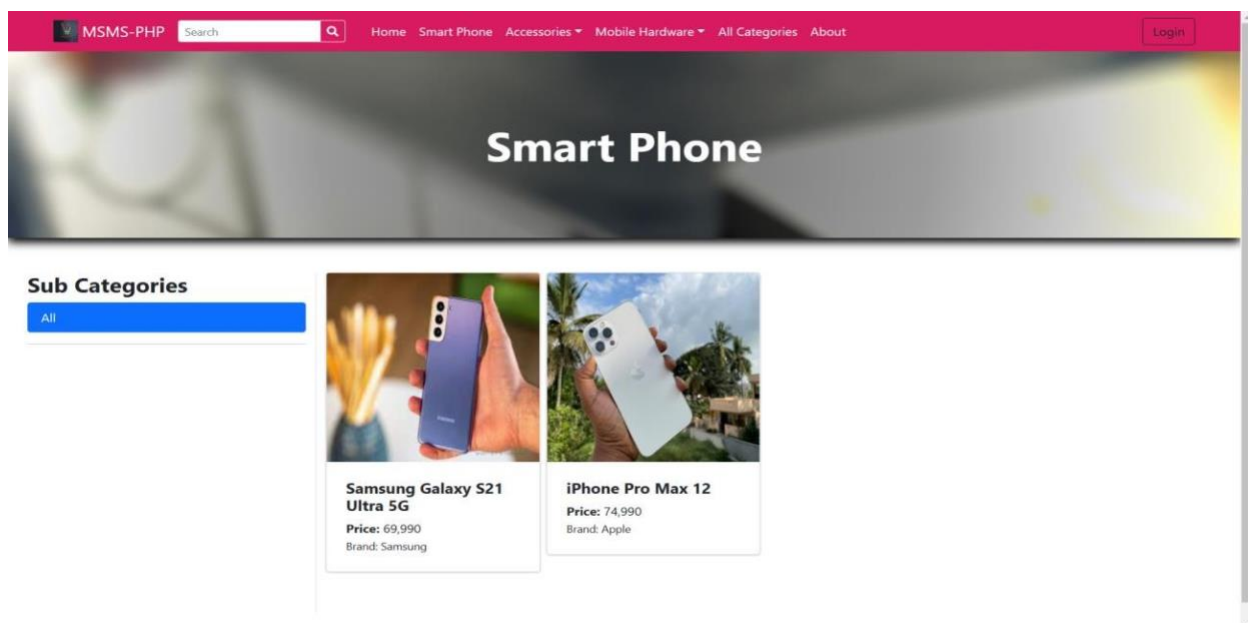


Fig 6.4: Smart Phones

ACCESSORIES

List Of Accessories

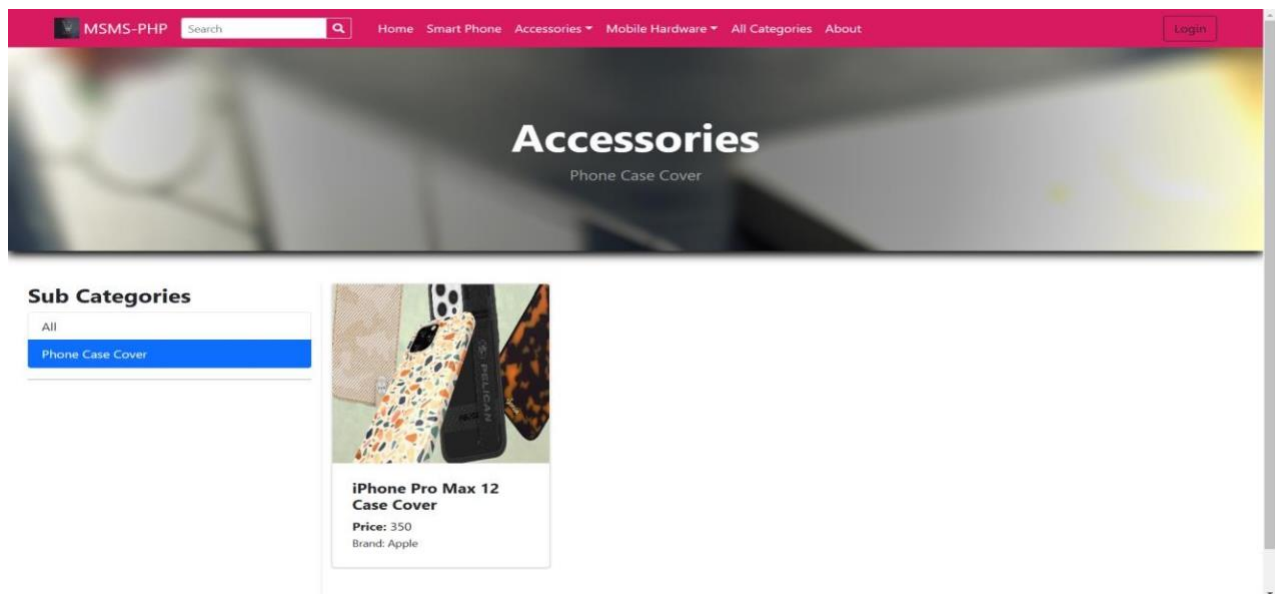


Fig 6.4: Accessories

ALL CATEGORIES

This page is helps to display the all the categories.

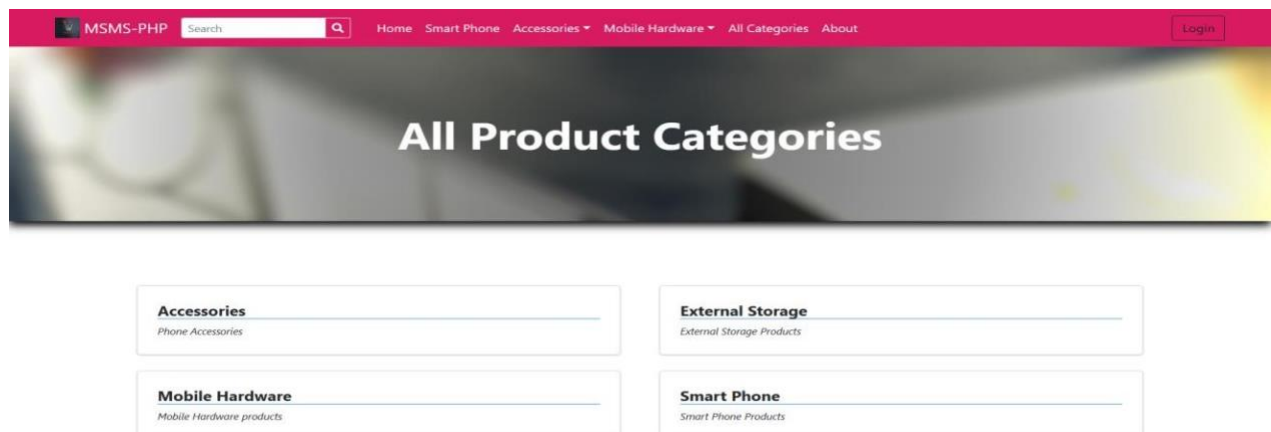


Fig 6.5: Accesories

This allows users to see all mobiles and mobiles accessories

CART

This is used to place a product to the cart before the order.

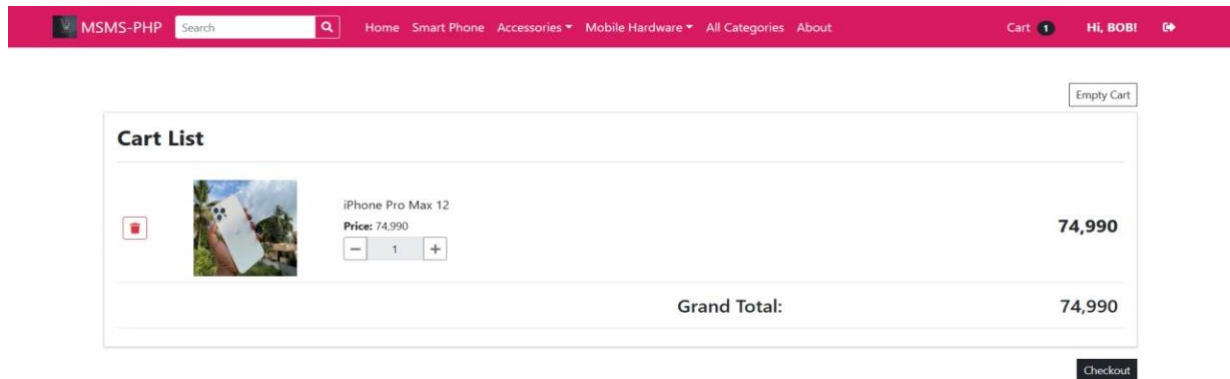


Fig 6.7: Cart

PAYMENTS

This page is use to make the payment of the product.

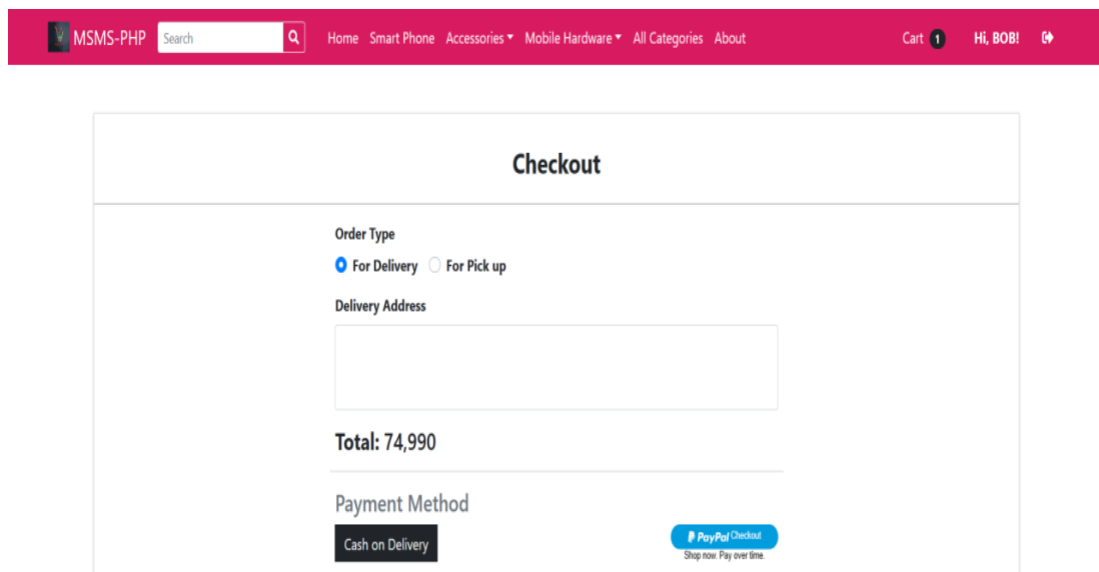


Fig 6.8: Payments

ABOUT US

This page allows you to see about store details.

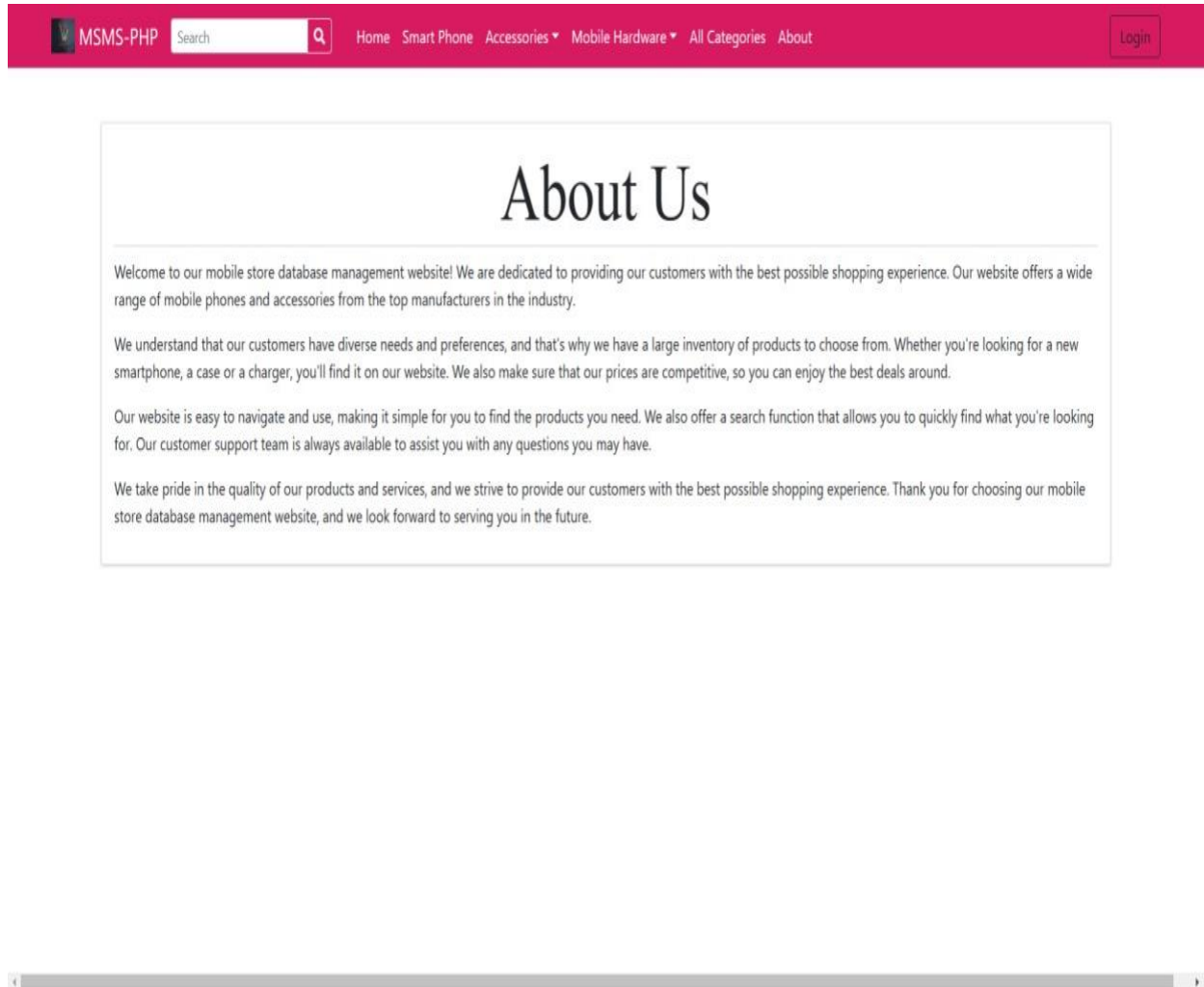


Fig 6.6: About us

RESULT

The resulting system is able to:

- ☐ Authenticate user credentials during login.
- ☐ Salted encryption for security of user passwords.
- ☐ Register new users and enabling them to purchase.
- ☐ Allow users to view their cart.
- ☐ Allow user to see the various products available in the catalog.

Chapter 7

CONCLUSION

A mobile store management system is a software application that allows businesses to manage and track inventory, sales, and customer data on mobile devices. The system can also include features such as inventory forecasting, customer relationship management, and financial reporting. Implementing a mobile store management system can improve the efficiency and productivity of a business by streamlining operations and providing real-time data and insights. Overall, a mobile store management system can be a valuable tool for any business enterprises looking to improve their mobile operations.

Chapter 8

FUTURE ENHANCEMENTS

Future upgrades to this project will implement:

- There are several potential areas for enhancement in a mobile store management system project, including:
- Integration with other systems: Integrating the mobile store management system with other systems such as inventory management, accounting, and customer relationship management can provide a more holistic view of the business and streamline processes.
- Real-time data and analytics: Incorporating real-time data and analytics capabilities can help store managers make more informed decisions and improve efficiency.
- Mobile payments: Adding mobile payment capabilities can make it more convenient for customers to make purchases and can help increase sales.
- Personalization: Incorporating personalization features such as customized product recommendations and targeted marketing can help improve customer engagement and increase sales.
- Inventory management: Enhancing the inventory management capabilities of the system can help store managers keep track of stock levels, reorder products, and avoid stockouts.
- Cloud-based: Migrating the system to cloud-based architecture can increase scalability, accessibility, and security while reducing costs.

REFERENCES

- [1] Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.
- [2] Monson-Haefel, R. (2007). J2EE Web services. Boston, Mass: Addison-Wesley.
SilberschatzA., KorthH.F.,&SudarshanS.(2011).
- [3] Database systems concepts.Estados Unidos:McGraw-Hill Companies, Inc.
- [4] Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill Education.
- [5] David. (2011).JavaScript: The Definitive Guide Sixth edition.
- [6] <https://www.w3schools.com>
- [7] <https://www.canvasjs.com>
- [8] <https://getbootstrap.com/>

