

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**"JNANA SANGAMA" BELGAUM-590018**



2017-2018

A  
Mini Project Report  
On

**Inventory Management System**

Submitted in partial fulfillment of the requirements in 5<sup>th</sup> Semester of  
BACHELOR OF ENGINEERING  
IN

**COMPUTER SCIENCE AND ENGINEERING**

By

**Pramod P  
[1JB15CS116]**

UNDER THE GUIDANCE OF

**Bindiya M K  
Associate Professor  
Department of CSE  
SJBIT, Bengaluru**

**Umashankar M L  
Assistant Professor  
Department of CSE  
SJBIT, Bengaluru**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**S. J. B INSTITUTE OF TECHNOLOGY  
B G S HEALTH AND EDUCATION CITY  
Kengeri, Bengaluru-560060.**

|| Jai Sri Gurudev ||  
Sri Adichunchanagiri Shikshana Trust ®

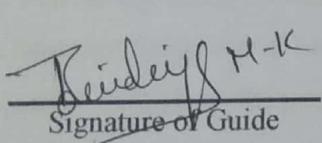
**S. J. B INSTITUTE OF TECHNOLOGY**  
BGS Health & Education City, Kengeri, Bengaluru-560060.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

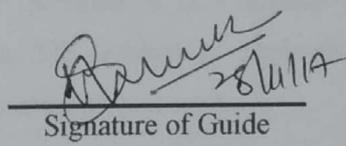


**CERTIFICATE**

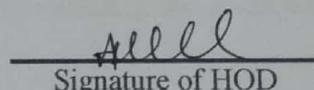
Certified that the mini project work entitled "**Inventory Management System**" carried out by **Pramod P [1JB15CS116]** is a bonafide student of S J B Institute of Technology in partial fulfillment for the award of "**BACHELOR OF ENGINEERING**" in **COMPUTER SCIENCE AND ENGINEERING** as prescribed by **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM** during the academic year **2017-2018**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work.

  
\_\_\_\_\_  
Signature of Guide

**Bindiya M K**  
Associate Professor  
Dept. of CSE, SJBIT

  
\_\_\_\_\_  
Signature of Guide

**Umashankar M L**  
Assistant Professor  
Dept. of CSE, SJBIT

  
\_\_\_\_\_  
Signature of HOD

**Dr. Krishna A N**  
Professor and Head  
Dept. of CSE, SJBIT

**EXTERNAL VIVA**

Name of the Examiners

Signature with Date

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_



## ACKNOWLEDGEMENT

I would like to express our profound gratefulness to His Divine Soul **Jagadguru Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Maha Swamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji** for providing us an opportunity to complete our academics in this esteemed Institution.

I would like to express my profound thanks to **Reverend Sri Sri Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this project in this admired institution.

I express my gratitude to **Dr. Puttaraju, Principal**, SJB Institute of Technology, for providing me excellent facilities and academic ambience, which have helped me in the satisfactory completion of the mini project work.

I extend my sincere thanks to **Dr. Krishna A N**, Professor and Head of the Department of Computer Science and Engineering, SJB Institute of Technology, for providing me invaluable support throughout the period of my mini project work.

I express my deepest gratitude and sincere thanks to mini project guides **Bindiya M K**, Associate Professor, and **Umashankar M L**, Assistant Professor, Department of Computer Science and Engineering, SJB Institute of Technology, Bengaluru, for their valuable guidance and support.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, the teaching & non-teaching staff of the department, the library staff and all my friends, who have directly or indirectly supported me during the period of my mini project work.

Regards,  
**Pramod P (1JB15CS116)**

---

## **Abstract**

Inventory management is the constant process of moving parts and products into and out of a company's location. Companies manage their inventory on a daily basis as they place new orders for products and maintain those orders. It is important that the company owners gain a firm grasp of everything involved in their inventory and maintain their stock of the products in their management process. That way, they can figure out creative ways to solve their inventory management challenges by finding the right solutions.

Without a doubt, inventory management is an important part of doing business. Companies remain stocked with all of the essential goods they need to sell to customers and complete daily operation by the means.

---

## Table of Contents

|  |          |
|--|----------|
| Acknowledgement.....                                     | I        |
| Abstract.....  | II       |
| Table of Contents.....                                   | III      |
| List of Figures.....                                     | V        |
| List of Tables .....                                     | VI       |
| Chapter 1 .....  | 1        |
| <b>Introduction.....</b>                                 | <b>1</b> |
| 1.1 Inventory Management System.....                     | 1        |
| Chapter 2 .....  | 2        |
| <b>Literature Survey.....</b>                            | <b>2</b> |
| 2.1 Traditional File Systems.....                        | 2        |
| 2.2 Pros and Cons of the Traditional Approach.....       | 2        |
| 2.3 Downfall of Traditional Management System.....       | 5        |
| 2.4 Introduction to the Database Management System ..... | 5        |
| 2.5 Indicative areas for the use of a DBMS.....          | 6        |
| 2.6 Advantages of a DBMS.....                            | 6        |
| 2.7 Components of a DBMS .....                           | 7        |
| Chapter 3 .....  | 8        |
| <b>System Requirements .....</b>                         | <b>8</b> |
| 3.1 Hardware Requirements .....                          | 8        |
| 3.2 Software Requirements .....                          | 8        |
| Chapter 4.....   | 9        |
| <b>System Design.....</b>                                | <b>9</b> |
| 4.1 Schema Diagram .....                                 | 9        |
| 4.2 ER Diagram .....                                     | 10       |

---

|  |           |
|--|-----------|
| Chapter 5 .....                          | 11        |
| <b>Implementation .....</b>              | <b>11</b> |
| 5.1 HTML5.....                           | 11        |
| 5.2 JavaScript .....                     | 12        |
| 5.3 SQL (Structured Query Language)..... | 13        |
| 5.4 Bootstrap.....                       | 13        |
| 5.5 Code Snippets .....                  | 15        |
| Chapter 6 .....                          | 33        |
| <b>Snapshots .....</b>                   | <b>33</b> |
| 6.1 The Index Page .....                 | 33        |
| 6.2 The Products Page.....               | 33        |
| 6.3 The Categories Page.....             | 34        |
| 6.4 The Brands Page .....                | 34        |
| 6.5 The Orders Page.....                 | 35        |
| 6.5.1 The AddOrders Page .....           | 35        |
| 6.5.2 The ManageOrders Page.....         | 36        |
| 6.6 The Report Page.....                 | 36        |
| 6.7 Tables .....                         | 37        |
| Chapter 7 .....                          | 40        |
| <b>Conclusion .....</b>                  | <b>40</b> |
| <b>References .....</b>                  | <b>41</b> |

## List of Figures

| <b>Fig No.</b> | <b>Description</b>                                | <b>Page No.</b> |
|----------------|---|-----------------|
| Fig 2.1        | <b>Components of a DBMS</b>                       | 7               |
| Fig 4.1        | <b>Schema Diagram</b>                             | 9               |
| Fig 4.2        | <b>ER Diagram for Inventory Management System</b> | 10              |
| Fig 6.1        | <b>The Index Page</b>                             | 33              |
| Fig 6.2        | <b>The Products Page</b>                          | 33              |
| Fig 6.3        | <b>The Categories Page</b>                        | 34              |
| Fig 6.4        | <b>The BrandsPage</b>                             | 34              |
| Fig 6.5        | <b>The Orders Page</b>                            | 35              |
| Fig 6.5.1      | <b>The AddOrders Page</b>                         | 35              |
| Fig 6.5.2      | <b>The ManageOrders Page</b>                      | 36              |
| Fig 6.6        | <b>The Report Page</b>                            | 36              |

---

## List of Tables

| Table No. | Description            | Page No. |
|-----------|------------------------|----------|
| Table 6-1 | Brands Table .....     | 37       |
| Table 6-2 | Categories Table ..... | 37       |
| Table 6-5 | Product Table.....     | 39       |
| Table 6-3 | Orders Table .....     | 38       |
| Table 6-4 | Order_item Table ..... | 40       |
| Table 6-6 | Users Table .....      | 40       |

---

## Chapter 1

# Introduction

## 1.1 Inventory management system

Inventory management refers to the process of ordering, storing and using a company's inventory: raw materials, components and finished products.

A company's inventory is one of the most valuable assets. In retail , manufacturing, food service and other inventory-intensive sectors, a company's inputs and finished products are the core of its business, and a shortage of inventory when and where it's needed can be extremely detrimental. At the same time, inventory can be thought of as a liability\*if not is an accounting sense). A large inventory carries the risk of spoilage, theft, damage, or shifts in demand. Inventory must be insured, and if it is not sold in time it may have to be disposed of at clearance prices – or simply destroyed.

For these reasons, inventory management is important for business of any size. Knowing when to restock certain items, what amounts to purchase or produce, what price to sell and when to sell – can easily become complex decisions. Small business will often keep track of it manually where as the large corporations use software applications. Appropriate inventory management strategies vary depending upon the industry.

Good stock management will help you maintain the right level of stock to meet customer demands. When managing the stock, considering quantities, sales and storage is essential

Hence, inventory management is an important part of doing business. It is the means by which companies remain stocked with all of the essential goods they need to sell to customers and complete daily operations

“One of the great responsibilities that I have is to manage my assets wisely, so that they create a value” –Alice Walton.

---

## Chapter 2

# Literature Survey

### 2.1 Traditional File Systems

In the early days of computing, data management and storage was a very new concept for organizations. The traditional approach to data handling offered a lot of the convenience of the manual approach to business processes (e.g. handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

Separate information systems for each business function also led to conflicts of interest within the company. Departments felt a great deal of ownership for the data that they collected, processed, and managed which caused many issues among company-wide collaboration and data sharing. This separation of data also led to unnecessary redundancy and a high rate of unreliable and inconsistent data.

### 2.2 Pros and Cons of the Traditional Approach

#### Pros

- Simple
    - Matched existing business processes and functions
-

- Companies were not as interested in funding complicated information systems
- Initially low-cost
  - Early computing was not viewed as beneficial for large funding
  - Systems were designed to be cheap in order to save on cost

**Cons**

- Separated ownership
  - Business functions had a high sense of data ownership
  - Departments unwilling to share data for fear of minimizing their superiority
- Unmanaged redundancy
  - Multiple instances of the same data appeared throughout various files, systems, and databases
  - Information updated in one place was not replicated to the other locations
  - Disk space was very expensive, and redundancy had a big impact on storage
- Data inconsistency
  - Redundant data stored in various locations was usually never stored the same way
  - Formatting was not centrally managed
- Lack of data sharing
  - Same data stored in multiple locations
  - Caused unnecessary doubling of efforts for processing and managing data
- High costs in the long run

- Hiring data processors for each department was very expensive, and each position was typically working on the same thing just for a different area
- Doubling of work as well as excessive maintenance costs

### **2.3 Downfall of Traditional Management System**

Conceived in a relatively centralized era when software was deployed in static environments, legacy database architectures fail to support an increasingly mobile world where applications are accessed anytime, anywhere. Today software users want consistent improvements in usability and expect SaaS vendors to deliver new features and functionalities needed to achieve their business objectives.

However, legacy database technologies fall short in serving the needs of today's distributed and cloud environments for the following reasons:

- Inadequate failover capabilities
- Latency issues
- Insufficient provisions during peak demands
- Lack of high availability at all times
- Increasing operational costs
- Inability to meet the demands of global markets

For all of these reasons, traditional databases are unable to deliver results in a rapidly growing environment where the workload is geographically distributed across heterogeneous datacentres. Upgrading to a more distributed data model is costly and complicated and your DBAs can't just sit back and give up on this situation. Hence, due to these various reasons, the downfall of the traditional system was inevitable.

### **2.4 Introduction to the Database Management System**

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of

database. A datum is a unit of data. Meaningful data combines to form information. Hence, information is interpreted data – data provided with semantics. MS ACCESS is one of the most common examples of database management software.

Database systems are meant to handle large collection of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

### **2.5 Indicative areas for the use of a DBMS**

- Airlines: reservations, schedules etc.
- Telecom: calls made, customer details, network usage etc.
- Universities: registration, results, grades etc.
- Sales: products, purchases, customers etc.
- Banking: all transactions etc.

### **2.6 Advantages of a DBMS**

A Database Management System has many advantages over the traditional file system used in the earlier days, such as:

- Data independence: Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
- Efficient data access: DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
- Data integrity and security: If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.

- Data administration: When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

## 2.7 Components of a DBMS

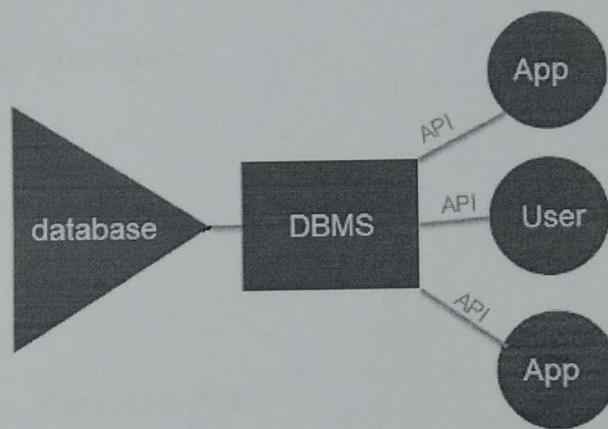


Fig 2.1 Components of a DBMS

- **Users:** Users may be of any kind, such as database administrators, system developers or database users.
- **Database application:** Database application may be Departmental, Personal, Organizational and /or Internal.
- **DBMS:** Software that allows users to create and manipulate database access.
- **Database:** Collection of logical data as a single unit.

---

## Chapter 3

# System Requirements

### 3.1 Hardware Requirements

- Processor: Pentium IV or above
- RAM – 2 GB or more
- Hard Disk – 2GB or more

### 3.2 Software Requirements

Technologies used:

- Front End: HTML, CSS, PHP
- Connection/Controller: PHP
- Back-End/Database: MySQL

Software:

- Text Editor: Brackets
- Server: Apache (on XAMPP 7)
- Operating System: Windows 10
- Database Support: MySQL 5.7
- Back-End: PHP 7.1.9

## Chapter 4

# System Design

### 4.1 Schema Diagram

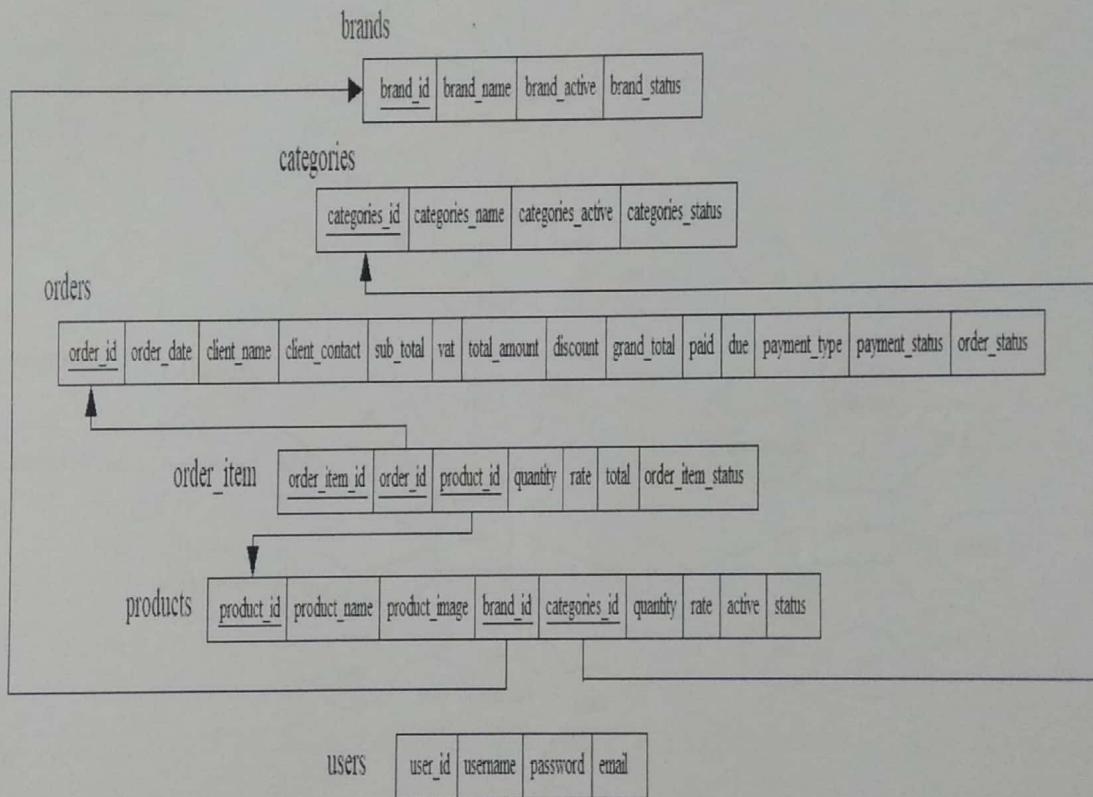


Fig 4.1 Schema Diagram

# System Design

## 4.1 Schema Diagram

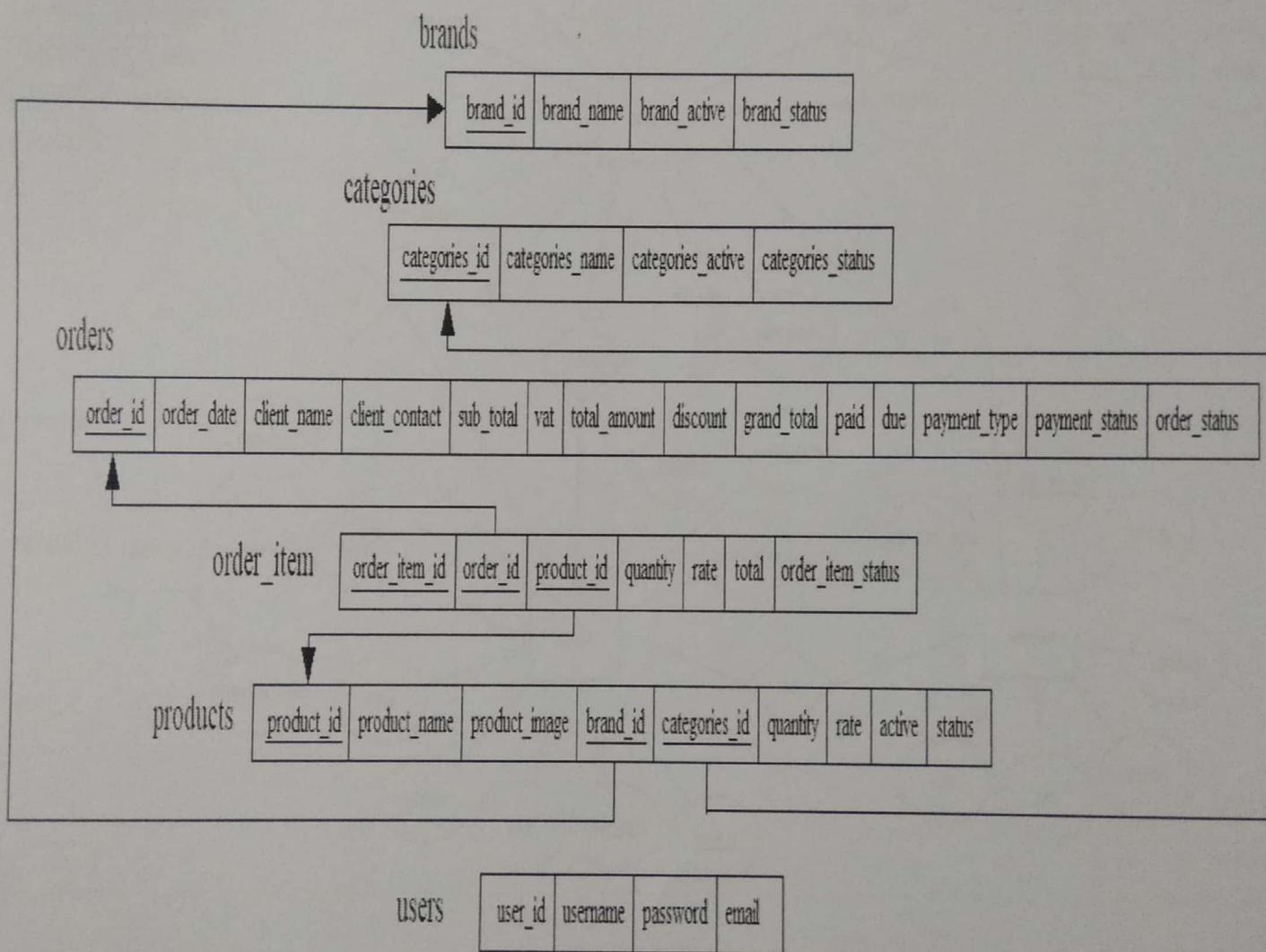


Fig 4.1 Schema Diagram

## 4.2 ER Diagram

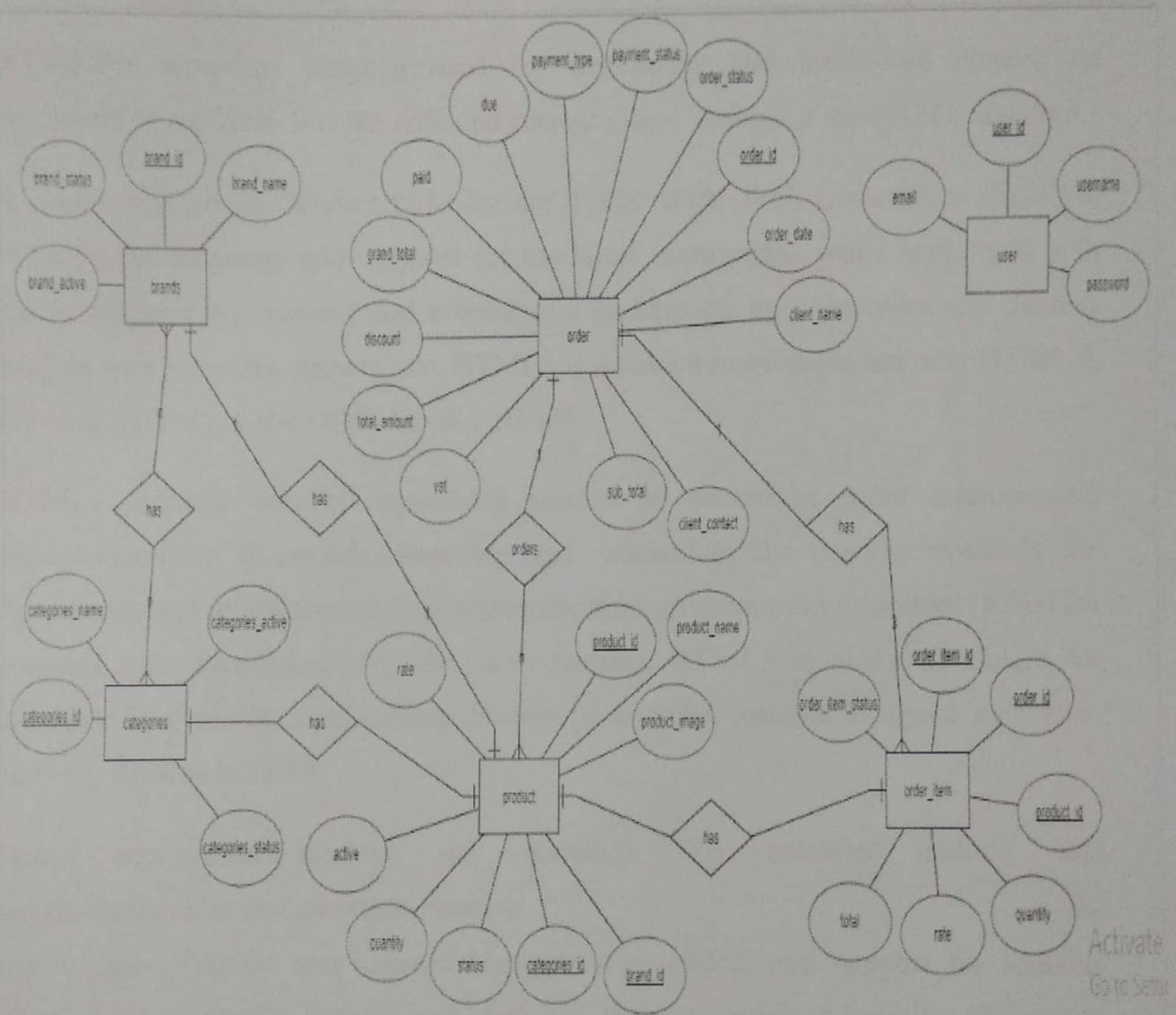


Fig 4.2 ER Diagram for Art Gallery

---

## Chapter 5

# Implementation

### 5.1 HTML5

**HTML5** is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard.

It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

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. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new `<video>`, `<audio>` and `<canvas>` elements were added, and support for scalable vector graphics (SVG) content and Math ML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as `<main>`, `<section>`, `<article>`, `<header>`, `<footer>`, `<aside>`, `<nav>` and `<figure>`, are added. New attributes are introduced, some elements and attributes have been removed, and others such as `<a>`, `<cite>` and `<menu>` have been changed, redefined or standardized.

The APIs and Document Object Model (DOM) are now fundamental parts of the HTML5 specification and HTML5 also better defines the processing for any invalid documents.

---

## 5.2 JavaScript

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

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

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

### 5.3 SQL (Structured Query Language)

**SQL (Structured Query Language)** is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like ISAM or VSAM, SQL offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify *how* to reach a record, e.g. with or without an index.

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. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks".<sup>[10]</sup> Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

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.

#### **5.4 Bootstrap**

Bootstrap is a free and open-source front-end web framework for designing websites and web application. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as other JavaScript extension. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap, originally names Twitter Blueprint, was developed by Mark Otto, and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden.

Bootstrap provides a set of stylesheets that provide basic style definition for all key HTML components. It also comes with several JavaScript components in the form of jQuery pulgins. They provide additional user interface elements such as dialog box, tooltips and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields. In version 1.3, the following JavaScript plugins are supported: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Typehead.

## 5.5 Code Snippets

### 5.5.1 Config – To establish connection

This is a code snippet to show how PHP is used to connect to the local MySQL database using the localhost server.

```
<?php  
  
$localhost = "127.0.0.1";  
  
$username = "root";  
  
$password = "";  
  
$dbname = "yo";  
  
// db connection  
  
$connect = new mysqli($localhost, $username, $password, $dbname);  
  
// check connection  
  
if($connect->connect_error) {  
  
    die("Connection Failed : " . $connect->connect_error);  
  
} else {  
  
    // echo "Successfully connected";  
  
}  
  
?>
```

### 5.5.2 The Login page

#### Login Page snippet (validation)

```
<?php
```

```
require_once 'php_action/db_connect.php';
session_start();

if(isset($_SESSION['userId'])) {

    header('location: /project1/product.php');

}

$errors = array();

if($_POST) {

    $username = $_POST['username'];

    $password = $_POST['password'];

    if(empty($username) || empty($password)) {

        if($username == "") {

            $errors[] = "Username is required";

        }

        if($password == "") {

            $errors[] = "Password is required";

        }

    } else {

        $sql = "SELECT * FROM users WHERE username = '$username'";

        $result = $connect->query($sql);

        if($result->num_rows == 1) {

            $password = md5($password);

            // exists

        }

    }

}

Scanned by CamScanner
```

```
$mainSql = "SELECT * FROM users WHERE username =
    '$username' AND password = '$password'";

$mainResult = $connect->query($mainSql);

if($mainResult->num_rows == 1) {

    $value = $mainResult->fetch_assoc();

    $user_id = $value['user_id'];

    $_SESSION['userId'] = $user_id;

    header('location:/project1/product.php');

} else{

    $errors[] = "Incorrect username/password
combination";

} // /else

} else {

    $errors[] = "Username does not exists";

} // /else

} // /else not empty username // password

} // /if $_POST

?>
```

### **5.5.3 The Product Page**

**Card snippet in PHP for the table in product page**

```
<?php require_once 'php_action/db_connect.php' ?>
```

```
<?php require_once 'includes/header.php'; ?>

<div class="row">
    <div class="col-md-12">
        <ol class="breadcrumb">
            <li class="active">Product</li>
        </ol>
        <div class="panel panel-default">
            <div class="panel-heading">
                <div class="page-heading"><i class="glyphicon glyphicon-edit"></i> Manage Product</div>
            </div> <!-- /panel-heading -->
            <div class="panel-body">
                <div class="remove-messages"></div>
                <div class="div-action pull pull-right" style="padding-bottom:20px;">
                    <button class="btn btn-default button1" data-toggle="modal" id="addProductModalBtn" data-target="#addProductModal"> <i class="glyphicon glyphicon-plus-sign"></i> Add Product </button>
                </div> <!-- /div-action -->
            </div>
        </div>
        <table class="table" id="manageProductTable">
            <thead>
                <tr>
```

```
<th
style="width:10%;">Photo</th>

<th>Product Name</th>
<th>Rate</th>

<th>Quantity</th>
<th>Brand</th>
<th>Category</th>
<th>Status</th>

<th
style="width:15%;">Options</th>
</tr>

</thead>
</table>

</div>           </div>
</div>
</div>
```

**Bootstrap code snippet for Header in each page**

```
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav navbar-right">

    <li id="navBrand"><a href="brand.php"><i class="glyphicon glyphicon-btc"></i> Brand</a></li>

    <li id="navCategories"><a href="categories.php"> <i class="glyphicon glyphicon-th-list"></i> Category</a></li>

    <li id="navProduct"><a href="product.php"> <i class="glyphicon glyphicon-ruble"></i> Product </a></li>

<li class="dropdown" id="navOrder">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"> <i class="glyphicon glyphicon-shopping-cart"></i> Orders <span class="caret"></span></a>
    <ul class="dropdown-menu">
        <li id="topNavAddOrder"><a href="orders.php?o=add"> <i class="glyphicon glyphicon-plus"></i> Add Orders</a></li>
        <li id="topNavManageOrder"><a href="orders.php?o=manord"> <i class="glyphicon glyphicon-edit"></i> Manage Orders</a></li>
    </ul>
</li>
```

```
<li id="navReport"><a href="report.php"> <i class="glyphicon glyphicon-check"></i> Report </a></li>

<li class="dropdown" id="navSetting">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"> <i class="glyphicon glyphicon-user"></i> <span class="caret"></span></a>
    <ul class="dropdown-menu">
        <li id="topNavSetting"><a href="setting.php"> <i class="glyphicon glyphicon-wrench"></i> Setting</a></li>
        <li id="topNavLogout"><a href="logout.php"> <i class="glyphicon glyphicon-log-out"></i> Logout</a></li>
    </ul>
</li>

</ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
```

#### 5.5.4 The Brands page

##### SQL snippet in PHP to fetch details of brands

```
<?php
require_once 'core.php';

$sql = "SELECT brand_id, brand_name, brand_active, brand_status FROM brands
WHERE brand_status = 1";
```

```
$result = $connect->query($sql);

$output = array('data' => array());

if($result->num_rows > 0) {

    // $row = $result->fetch_array();

    $activeBrands = "";

    while($row = $result->fetch_array()) {

        $brandId = $row[0];

        // active

        if($row[2] == 1) {

            // activate member

            $activeBrands = "<label class='label label-success'>Available</label>";

        } else {

            // deactivate member

            $activeBrands = "<label class='label label-danger'>Not Available</label>";

        }

    }

    $button = '<!-- Single button -->

<div class="btn-group">

<button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

    Action <span class="caret"></span>

</button>

<ul class="dropdown-menu">


```

```
<li><a type="button" data-toggle="modal" data-target="#editBrandModel"
onclick="editBrands('.$brandId.')"> <i class="glyphicon glyphicon-edit"></i>
Edit</a></li>

<li><a type="button" data-toggle="modal" data-
target="#removeMemberModal"
onclick="removeBrands('.$brandId.')"> <i class="glyphicon
glyphicon-trash"></i> Remove</a></li>

</ul>

</div>';

$output['data'][] = array(
    $row[1],
    $activeBrands,
    $button
);

} // /while

} // if num_rows

$connet->close();

echo json_encode($output);
```

### 5.5.5 The Categories Page

```
<div class="modal fade" id="addCategoriesModal" tabindex="-1" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
```

```
<form class="form-horizontal" id="submitCategoriesForm"
action="php_action/createCategories.php" method="POST">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="fa fa-plus"></i> Add Categories</h4>

</div>

<div class="modal-body">

<div id="add-categories-messages"></div>

<div class="form-group">

<label for="categoriesName" class="col-sm-4 control-
label">Categories Name: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-7">

<input type="text" class="form-control"
id="categoriesName" placeholder="Categories
Name" name="categoriesName"
autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="categoriesStatus" class="col-sm-4 control-label">Status:</label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-7">

<select class="form-control" id="categoriesStatus"
name="categoriesStatus">

<option value="">~~SELECT~~</option>

<option value="1">Available</option>

<option value="2">Not Available</option>


```

```
</select>

</div>
</div> <!-- /form-group-->

</div> <!-- /modal-body -->

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal"><i
    class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="submit" class="btn btn-primary" id="createCategoriesBtn"
    data-loading-text="Loading..." autocomplete="off"><i
    class="glyphicon glyphicon-ok-sign"></i> Save Changes</button>

</div> <!-- /modal-footer -->

</form> <!-- /.form -->

</div> <!-- /modal-content -->

</div> <!-- /modal-dailog -->

</div>
```

### 5.5.6 The Orders Page

```
<?php

require_once 'php_action/db_connect.php';

require_once 'includes/header.php';

if($_GET['o'] == 'add') {

    // add order

    echo "<div class='div-request div-hide'>add</div>";

} else if($_GET['o'] == 'manord') {
```

```
echo "<div class='div-request div-hide'>manord</div>";  
} else if($_GET['o'] == 'editOrd') {  
  
    echo "<div class='div-request div-hide'>editOrd</div>";  
  
} // /else manage order  
  
?>  
  
<ol class="breadcrumb">  
  
    <li>Order</li>  
  
    <li class="active">  
  
        <?php if($_GET['o'] == 'add') { ?>  
  
            Add Order  
  
        <?php } else if($_GET['o'] == 'manord') { ?>  
  
            Manage Order  
  
        <?php } // /else manage order ?>  
  
    </li>  
  
</ol>  
  
<h4>  
  
    <i class='glyphicon glyphicon-circle-arrow-right'></i>  
  
    <?php if($_GET['o'] == 'add') {  
  
        echo "Add Order";  
  
    } else if($_GET['o'] == 'manord') {  
  
        echo "Manage Order";  
  
    } else if($_GET['o'] == 'editOrd') {  
  
        echo "Edit Order";  
  
    }>
```

```
    }  
?  
</h4>  
  
<div class="panel panel-default">  
  <div class="panel-heading">  
    <?php if($_GET['o'] == 'add') { ?>  
      <i class="glyphicon glyphicon-plus-sign"></i> Add Order  
    <?php } else if($_GET['o'] == 'manord') { ?>  
      <i class="glyphicon glyphicon-edit"></i> Manage Order  
    <?php } else if($_GET['o'] == 'editOrd') { ?>  
      <i class="glyphicon glyphicon-edit"></i> Edit Order  
    <?php } ?>
```

**PHP snippet for calling Stored Procedure for fetching orders**

```
<?php  
  
require_once 'core.php';  
  
$sql = "SELECT order_id, order_date, client_name, client_contact, payment_status  
       FROM orders WHERE order_status = 1";  
  
$result = $connect->query($sql);  
  
  
$output = array('data' => array());
```

```
if($result->num_rows > 0) {  
  
    $paymentStatus = "";  
    $x = 1;  
  
    while($row = $result->fetch_array()) {  
        $orderId = $row[0];  
  
        $countOrderItemSql = "call cto($orderId)";  
        $itemCountResult = $connect->query($countOrderItemSql);  
        $itemCountRow = $itemCountResult->fetch_row();  
  
        // active  
        if($row[4] == 1) {  
            $paymentStatus = "<label class='label label-success'>Full  
                            Payment</label>";  
        } else if($row[4] == 2) {  
            $paymentStatus = "<label class='label label-info'>Advance  
                            Payment</label>";  
        } else {  
            $paymentStatus = "<label class='label label-warning'>No  
                            Payment</label>";  
        } // /else  
  
        $button = '<!-- Single button -->  
        <div class="btn-group">  
        ...  
    }  
}
```

```

<button type="button" class="btn btn-default dropdown-toggle" data-
    toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Action <span class="caret"></span>
</button>

<ul class="dropdown-menu">
    <li><a href="orders.php?o=editOrd&i=$orderId." id="editOrderModalBtn" onclick="editOrder($orderId.)"> <i class="glyphicon glyphicon-edit"></i> Edit</a></li>

    <li><a type="button" data-toggle="modal" id="paymentOrderModalBtn" data-target="#paymentOrderModal" onclick="paymentOrder($orderId.)"> <i class="glyphicon glyphicon-save"></i> Payment</a></li>

    <li><a type="button" onclick="printOrder($orderId.)" > <i class="glyphicon glyphicon-print"></i> Print </a></li>

    <li><a type="button" data-toggle="modal" data-target="#removeOrderModal" id="removeOrderModalBtn" onclick="removeOrder($orderId.)"> <i class="glyphicon glyphicon-trash"></i> Remove</a></li>
</ul>
</div>';

```

```

$output['data'][] = array(
    // image
    $x,
    // order date
    $row[1],
    // client name
    $row[2],

```

```
// client contact  
$row[3],  
$itemCountRow,  
$paymentStatus,  
// button  
$button  
);  
  
$x++;  
} // /while  
  
}// if num_rows  
  
$connect->close();  
  
echo json_encode($output);
```

### 5.5.7 The Report Page

```
<?php require_once 'includes/header.php'; ?>  
  
<div class="row">  
  <div class="col-md-12">  
    <div class="panel panel-default">  
      <div class="panel-heading">  
        <i class="glyphicon glyphicon-check"></i> Order  
Report  
      </div>
```

```
<!-- /panel-heading -->

<div class="panel-body">

    <form class="form-horizontal"
        action="php_action/getOrderReport.php"
        method="post" id="getOrderReportForm">

        <div class="form-group">

            <label for="startDate" class="col-sm-2 control-
                label">Start Date</label>

            <div class="col-sm-10">

                <input type="text" class="form-control"
                    id="startDate" name="startDate"
                    placeholder="Start Date" />

            </div>

        </div>

        <div class="form-group">

            <label for="endDate" class="col-sm-2 control-
                label">End Date</label>

            <div class="col-sm-10">

                <input type="text" class="form-control"
                    id="endDate" name="endDate"
                    placeholder="End Date" />

            </div>

        </div>

        <div class="form-group">

            <div class="col-sm-offset-2 col-sm-10">

                <button type="submit" class="btn btn-success"
                    id="generateReportBtn"> <i class="glyphicon
                    glyphicon-ok-sign"></i> Generate
                Report</button>

            </div>

        </div>

    </form>

</div>
```

```
</div>

</form>

</div>

<!-- /panel-body -->

</div>

</div>

<!-- /col-md-12 -->

</div>

<!-- /row -->

<script src="custom/js/report.js"></script>

<?php require_once 'includes/footer.php'; ?>
```

#### **5.5.8 Stored Procedure**

CREATE DEFINER = 'root'@'localhost' Type 'PROCEDURE'

DIRECTION = 'in' NAME = 'abc' TYPER = 'int'

BEGIN

SELECT count(\*) FROM order\_item WHERE order\_id = abc;

END

Security Type = DEFINER

## Chapter 6

# Snapshots

### 6.1 The Index Page

This is the initial page that greets a visitor. An admin of the company can enter their login details, which will take them to another page.

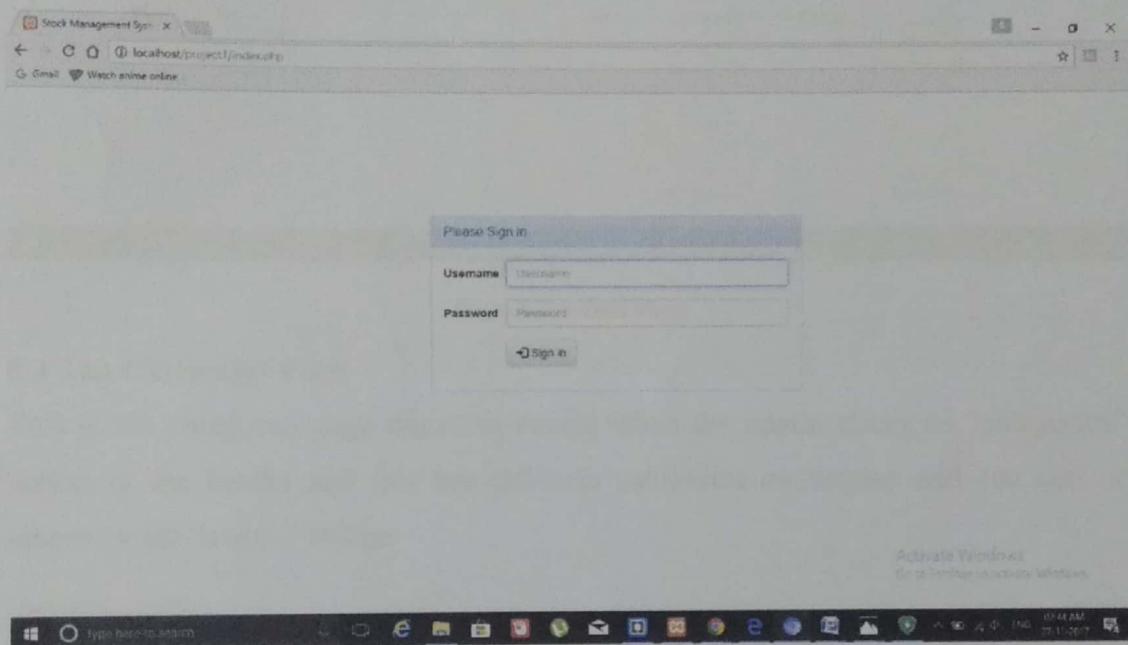


Fig 6.1 Login Page

### 6.2 The Product Page

The Home page is the initial page after a user logs in, and has different products on display. The admin can click the “Add Products” button to add more products, and edit or remove it by clicking on “action” button.

The screenshot shows a web-based inventory management system. The header includes links for Home, Category, Product, Orders, Report, and Log Out. The main content area is titled "Manage Product" and displays a table of products. The columns are: Photo, Product Name, Rate, Quantity, Brand, Category, Status, and Options. There are two entries: "half shirt" and "pant". Both items have a status of "Available" and an "Action" button. A search bar and a "Show 10 entries" dropdown are at the top of the table. The bottom of the page shows navigation links for Previous, Next, and a total of 2 entries.

| Photo | Product Name | Rate | Quantity | Brand  | Category    | Status    | Options |
|-------|--------------|------|----------|--------|-------------|-----------|---------|
|       | half shirt   | 9    | 3        | adidas | casual wear | Available | Action  |
|       | pant         | 6    | 3        | yoyoso | casual wear | Available | Action  |

Fig 6.2 Product Page

### 6.3 The Categories Page

This is the categories page which is visited when the admin clicks on “categories” button in the header and this has different categories on display and can edit or remove it via “action” button.

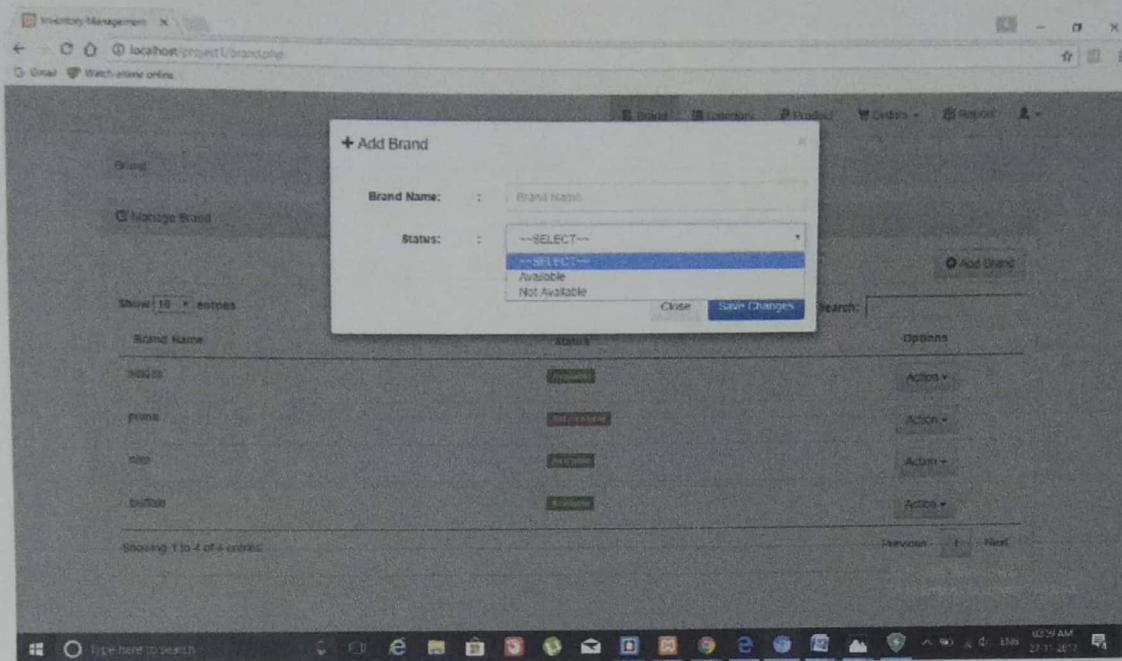
The screenshot shows the "Manage Categories" page. The header includes links for Home, Category, Product, Orders, Report, and Log Out. The main content area is titled "Manage Categories" and displays a table of categories. The columns are: Categories Name, Status, and Options. There are two entries: "casual wear" and "formal wear". The "casual wear" entry has a status of "Available" and an "Action" button. The "formal wear" entry has a status of "Not Available" and buttons for "Edit" and "Remove". A search bar and a "Show 10 entries" dropdown are at the top of the table. The bottom of the page shows navigation links for Previous, Next, and a total of 2 entries.

| Categories Name | Status        | Options     |
|-----------------|---------------|-------------|
| casual wear     | Available     | Action      |
| formal wear     | Not Available | Edit Remove |

Fig 6.3 Categories page

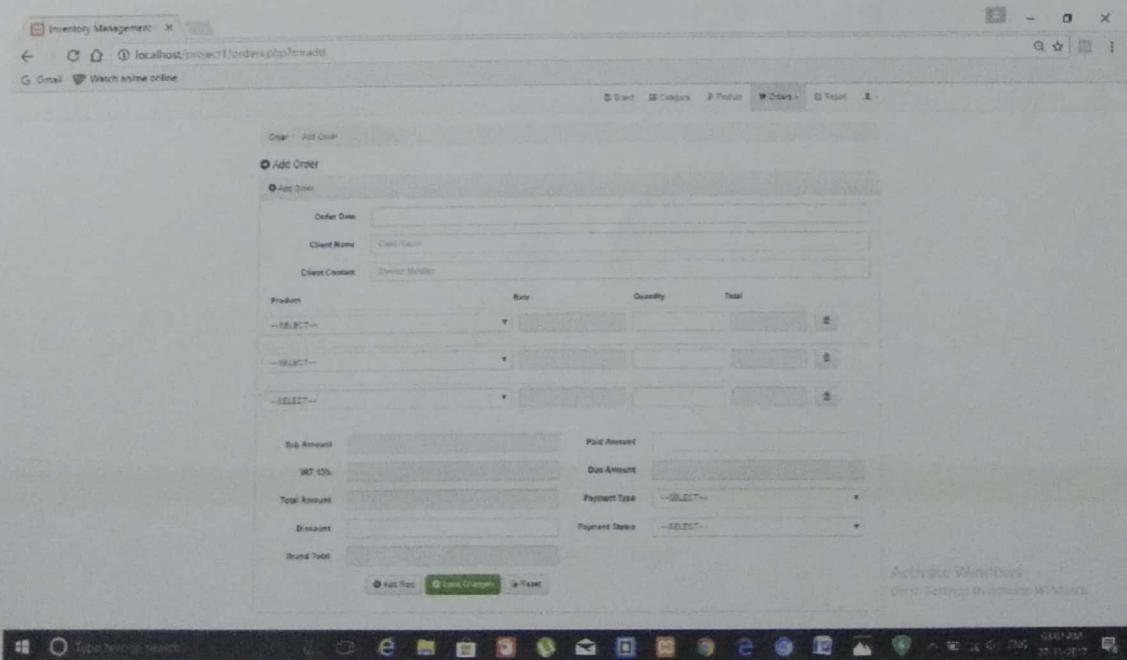
### 6.4 The Brands Page

The Brands page contains different brands on display. The admin can add more brands and remove or edit the existing brands by clicking on “Add Brand” and “action” button respectively.

**Fig 6.4 The BrandsPage**

## 6.5 The Orders Page

The admin adds the orders by entering the required details

**Fig 6.5.1 Add order Page**

The manage orders page shows all the orders of the user, along with the total quantity of the products bought, and the total amount of money spent on those products.

A screenshot of a web browser displaying the 'Manage Orders' page of an inventory management system. The URL in the address bar is `localhost/project1/orders.php?o=manord`. The page header includes links for Brand, Category, Product, Orders, Report, and User. Below the header, there are two tabs: 'Manage Order' (selected) and 'Manage Order'. A search bar and a dropdown for 'Show 10 entries' are present. The main content area displays a table with one row of data:

| # | Order Date | Client Name | Contact | Total Order Item | Payment Status | Option |
|---|------------|-------------|---------|------------------|----------------|--------|
| 1 | 2017-11-18 | pramod      | 795406  | 0                | Full Payment   | Action |

Below the table, it says 'Showing 1 to 1 of 1 entries'. At the bottom right, there are 'Previous' and 'Next' buttons. The status bar at the bottom indicates 'Activate Windows' and shows the date as 27-11-2017 and time as 03:03 AM.

Fig 6.5.2 Manage Orders Page

## 6.6 The Report Page

This page is to fill in a report which contains the start date and the end date of report

A screenshot of a web browser displaying the 'Report' page of the inventory management system. The URL in the address bar is `localhost/project1/report.php`. The page header includes links for Brand, Category, Product, Orders, Report, and User. Below the header, there is a section titled 'Order Report' with two input fields: 'Start Date' and 'End Date', both currently empty. A green button labeled 'Generate Report' is positioned below the date fields. The status bar at the bottom indicates 'Activate Windows' and shows the date as 27-11-2017 and time as 03:05 AM.

Fig 6.7 Report page

## 6.7 Tables

The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar lists databases: test, information\_schema, inventorymanagement, mysql, performance\_schema, phpmyadmin, and test. Under the 'Tables' section, 'brands' is selected. The main area displays the 'brands' table with the following data:

|                          | brand_id | brand_name | brand_active | brand_status |
|--------------------------|----------|------------|--------------|--------------|
| <input type="checkbox"/> | 15       | adidas     | 1            | 1            |
| <input type="checkbox"/> | 16       | puma       | 2            | 1            |
| <input type="checkbox"/> | 17       | nike       | 1            | 1            |
| <input type="checkbox"/> | 18       | lacoste    | 1            | 1            |

Below the table, there are buttons for 'Edit', 'Copy', 'Delete', 'Export', and 'Import'. The status bar at the bottom right shows '03:12 AM 27-1-2017'.

Table 6-1 Brands Table

The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar lists databases: test, information\_schema, inventorymanagement, mysql, performance\_schema, phpmyadmin, and test. Under the 'Tables' section, 'categories' is selected. The main area displays the 'categories' table with the following data:

|                          | categories_id | categories_name | categories_active | categories_status |
|--------------------------|---------------|-----------------|-------------------|-------------------|
| <input type="checkbox"/> | 10            | casual wear     | 1                 | 1                 |
| <input type="checkbox"/> | 11            | formal wear     | 2                 | 1                 |

Below the table, there are buttons for 'Edit', 'Copy', 'Delete', 'Export', and 'Import'. The status bar at the bottom right shows '03:13 AM 27-1-2017'.

Table 6-2 CategoriesTable

# Inventory Management System

snapshots

The screenshot shows the phpMyAdmin interface for the 'products' table. The table has 18 rows and 10 columns. The columns are: product\_id, product\_name, product\_image, brand\_id, categories\_id, taxedBy, rate, active, i, and status. The data includes various product names like 'abc', 'Vans', 'shirt', 'T-Shirts', 'Half Pant', 'T-Shirt', 'Half Pant', 'India shirt', 'pant', 'Half shirt', and 'jacket', each with a unique image URL and specific values for the other columns.

| product_id | product_name | product_image                                    | brand_id | categories_id | taxedBy | rate | active | i | status |
|------------|--------------|--|----------|---------------|---------|------|--------|---|--------|
| 11         | abc          | /assets/images/stock/15324821355a5f8e3e5ed26.png | 15       | 10            | 1       | 200  | 2      | 2 | 2      |
| 12         | Vans         | /assets/images/stock/15701522545e189e58704ef.png | 17       | 10            | 5       | 50   | 2      | 2 | 2      |
| 15         | shirt        | /assets/images/stock/136221285a1b08125ddc3.jpg   | 15       | 11            | 3       | 8    | 2      | 1 | 1      |
| 2          | T-Shirts     | /assets/images/stock/1034872825261575.jpg        | 2        | 2             | 6       | 1200 | 1      | 1 | 1      |
| 3          | Half Pant    | /assets/images/stock/132745705273249746.jpg      | 5        | 3             | 17      | 1200 | 1      | 1 | 1      |
| 4          | T-Shirt      | /assets/images/stock/11296168221cc9465.jpg       | 4        | 3             | 28      | 1600 | 1      | 1 | 1      |
| 7          | Half Pant    | /assets/images/stock/177625780341357467.jpg      | 11       | 2             | 28      | 1200 | 1      | 1 | 1      |
| 9          | India shirt  | /assets/images/stock/14058805795a62d1c5805.png   | 11       | 9             | 2       | 4000 | 1      | 1 | 1      |
| 14         | pant         | /assets/images/stock/1766034053a1aaad09fd.jpg    | 17       | 10            | 3       | 0    | 1      | 1 | 1      |
| 16         | Half shirt   | /assets/images/stock/12934675a1aaab2ca5818.jpg   | 15       | 10            | 3       | 0    | 1      | 1 | 1      |
| 17         | jacket       | /assets/images/stock/127117295a1b035dc09584.jpg  | 18       | 10            | 8       | 8    | 1      | 1 | 1      |

Table 6-3 Products Table

B

The screenshot shows the phpMyAdmin interface for the 'orders' table. The table has 6 rows and 14 columns. The columns are: order\_id, order\_date, client\_name, client\_contact, sub\_total, vat, total\_amount, discount, grand\_total, paid, due, payment\_type, payment\_status, and order\_status. The data includes orders from clients named 'pramod', 'pramod', 'bill gates', and 'golu' with various total amounts and payment details.

| order_id | order_date | client_name | client_contact | sub_total | vat    | total_amount | discount | grand_total | paid    | due     | payment_type | payment_status | order_status |
|----------|------------|-------------|----------------|-----------|--------|--------------|----------|-------------|---------|---------|--------------|----------------|--------------|
| 3        | 2017-11-10 | pramod      | 009816153      | 5200.00   | 877.04 | 5305.04      | 0        | 5305.04     | 5305.04 | 0.00    | 2            | 1              | 2            |
| 4        | 2017-11-10 | pramod      | 795496         | 5800.00   | 480.00 | 6280.00      | 0        | 6280.00     | 4065.00 | 4065.00 | 2            | 1              | 1            |
| 5        | 2017-11-04 | bill gates  | 9739426450     | 2490.00   | 313.04 | 2721.04      | 0        | 2721.04     | 2721.04 | 0.00    | 2            | 1              | 1            |
| 6        | 2017-11-02 | golu        | 6974908205     | 1200.00   | 157.17 | 1356.17      | 6.17     | 1356.00     | 1356.00 | 8.00    | 2            | 1              | 1            |

Table 6-4 Orders Table

## Inventory Management System

snapshots

The screenshot shows the phpMyAdmin interface for the 'Inventory' database. The left sidebar lists tables such as 'New', 'information\_schema', 'inventory', 'mysql', 'performance\_schema', 'preplayader', 'test', and 'Orders\_item'. The 'Orders\_item' table is selected, displaying its data in a grid format. The grid has columns: order\_item\_id, order\_id, product\_id, quantity, rate, total, and order\_item\_status. There are six rows of data. Below the grid, there are options for 'Edit', 'Copy', 'Delete', 'Check all', and 'With selected'. At the bottom, there are buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

Table 6-5 Orders\_item Table

The screenshot shows the phpMyAdmin interface for the 'Inventory' database. The left sidebar lists tables including 'New', 'information\_schema', 'inventory', 'mysql', 'performance\_schema', 'preplayader', 'test', and 'Users'. The 'Users' table is selected, displaying its data in a grid format. The grid has columns: user\_id, username, password, and email. There is one row of data. Below the grid, there are options for 'Edit', 'Copy', 'Delete', 'Check all', and 'With selected'. At the bottom, there are buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

Table 6-6 Users Table

---

## Chapter 7

### **Conclusion**

With the theoretical inclination of our syllabus, it becomes very essential to take the utmost advantage of any opportunity of gaining practical experience that comes along. The construction of this mini project “Inventory Management System” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts, thus making us more competent.

In the present system, we cannot sell products, which can be implemented in future versions of the website. This could be done with future iterations of the mini project.

## References

- The Bootstrap Website (for design) – <https://getbootstrap.com/>
- W3Schools (HTML, CSS and PHP reference) – <https://www.w3schools.com>
- PHP Official Documentation – <http://php.net/docs.php>
- The MySQL Documentation – <https://dev.mysql.com/doc>
- Stack Overflow – <https://stackoverflow.com>
- Stack Exchange – <https://stackexchange.com/>
- Wikipedia – <https://www.wikipedia.org>
- <https://www.youtube.com>