

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT

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

1.2 PROJECT OBJECTIVE

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

1.3 PROJECT OVERVIEW

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

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

1.4 PROJECT SCOPE

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

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

Shopping cart is one of the important facility provided in online shopping, this lets customer to browse different goods and services and once they select an item to purchase they can place the item in shopping cart, and continue browsing till the final selection. Customers can even remove the items from shopping cart that were selected earlier before they place the final order. It reminds us of shopping basket that we carry in departmental store.

1.5 INTRODUCTION TO DATABASE

Databases and database technology have had a major impact on the growing use of computers. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. Nowadays, this data is typically stored in mobile phones, which have their own simple database software. In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. A database can be of any size and complexity. For example, the list of names and addresses referred to earlier may consist of only a few hundred records, each with a simple structure. On the other hand, the computerized catalogue of a large library may contain half a million entries organized under different categories.

A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. Changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

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

1.4.1 ADVANTAGES OF DBMS:

Compared to the File Based Data Management System, Database Management System has many advantages.

1.Reducing Data Redundancy

The file-based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which leads to data redundancy. This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

2.Data Integrity

Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data that is visible to multiple users. So, it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

3.Data Security

Data Security is vital concept in a database. Only authorised users should be allowed to access the database and their identity should be authenticated using a username and password.

Unauthorised users

should not be allowed to access the database under any circumstances as it violates the integrity constraints.

4.Privacy

The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

5.Backup and Recovery

Database Management System automatically takes care of backup and recovery. The users don't need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

6.Data Consistency

Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

1.4.2 COMPONENTS OF DBMS

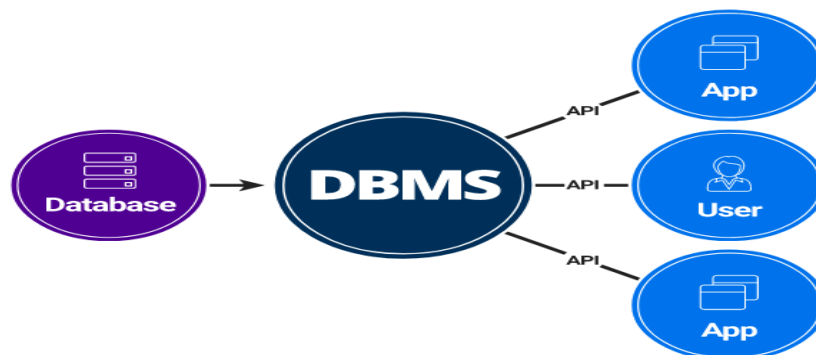


Fig 1.1 Components of DBMS

The figure 1.1 shows the components of DBMS which describe the different parts that work together for creating, managing the database that forms a complete system named DBMS(database management system).

- **Users:**

Users may be of any kind, such as data base 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.

1.4.3 INDICATIVE AREAS FOR THE USE OF DBMS

<u>SECTOR</u>	<u>USE OF DBMS</u>
BANKING	For customer information, account activities, payments, deposits, loans, etc.
AIRLINES	For reservations and schedule information.
UNIVERSITIES	For student information, course registrations, colleges, and grades.
TELECOMMUNICATION	It helps to keep call records, monthly bills, maintaining balances, etc.
FINANCE	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
SALES	Use for storing customer, product & sales information.

1.5 CONSTRAINTS

Mainly Constraints on the relational database are of 4 types:

- Domain constraints
- Key constraints
- Entity Integrity constraints
- Referential integrity constraints

DOMAIN CONSTRAINTS:

- Every domain must contain atomic values (smallest indivisible units) it means composite and multivalued attributes are not allowed.
- We perform datatype check here, which means when we assign a data type to a column we limit the values that it can contain.
- Eg. If we assign the datatype of attribute age as int, we can't give it values other than int datatype.

KEY CONSTRAINTS OR UNIQUENESS CONSTRAINTS:

- These are called uniqueness constraints since it ensures that every tuple in the relation should be unique.
- A relation can have multiple keys or candidate keys(minimal superkey), out of which we choose one of the keys as primary key, we don't have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key with less number of attributes.
- Null values are not allowed in the primary key, hence Not Null constraint is also a part of key constraint.

ENTITY INTEGRITY CONSTRAINTS:

Entity Integrity constraints says that no primary key can take NULL value, since using primary key we identify each tuple uniquely in a relation.

REFERENTIAL INTEGRITY CONSTRAINTS:

- The Referential integrity constraints is specified between two relations or tables and used to maintain the consistency among the tuples in two relations.
- This constraint is enforced through foreign key, when an attribute in the foreign key of relation R1 have the same domain(s) as the primary key of relation R2, then the foreign key of R1 is said to reference or refer to the primary key of relation R2.
- The values of the foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take NULL values, but can't be empty.

1.6 NORMALISATION

Normalizing tables accordingly based on the normal forms.

NORMAL FORMS:

Of a relation refers to the highest normal form condition that it meets and hence indicates the degree to which it has been normalized.

- **First NF:**

States that the domain of the attribute must only include atomic values and the value of any attribute in a tuple must be a single value.

- **Second NF:**

Is based on the concept of full functional dependency I.e. if removal of any attribute A from X in FD $X \rightarrow Y$ the dependency does not hold anymore.

- **Third NF:**

A relation schema R is in 3NF if, whenever a non-trivial functional dependency $X \rightarrow A$ holds in R either: X is a super key of R or A is prime attribute of R.

- **Boyce-Codd NF:**

A relation schema R is in BCNF if whenever a non-trivial functional dependency $X \rightarrow A$ holds in R, then X is a super key of R.

- **Fourth NF:** A relation schema R is in 4NF w.r.t a set of dependencies F if, for every non-trivial

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

1. Operating System: Windows XP/VISTA/7/8/10
2. XAMPP Server 3.2.2 with the following services
 - Apache 2.4
 - MYSQL 5.5
 - PHP 7.1.23
3. HTML Text Editor: Visual Studio code/Sublime Text
4. Browser: Chrome/firefox/Internet Explorer
5. Full Administrator access

2.2 HARDWARE REQUIREMENTS

6. RAM:512 MB and above
7. HDD:500 MB and above
8. PROCESSOR: Intel® Core™ i5-8250U CPU @ 1.6GHz 1.80GHz
9. SYSTEM TYPE: 64-bit operating system, x64-based processor

CHAPTER 3

SYSTEM DESIGN

3.1 FUNCTIONAL REQUIREMENTS

Various functional modules can be implemented by the system. it provides a requirement overview of the system.

- **Maintenance:-**

Is the system that maintains the detail of the products and their hierarchy attributes (size, weight, cost etc) The two main components of master maintenance are as follows.

- **Product :-**

It's include the information of the products, item no, size, categories etc. it's is also the admin of vendor or seller where they can put the information of the products in websites.

- **Price :-**

Only for the price of the products and applicable discount of the products.

- **Transaction:-**

Transaction is a payment method in which the transfer of money of buying products. This process is secure and password protected. Three steps involved in the online transaction are Registration, Placing an order, and, Payment.

- **Reporting:-**

After ordering for the product, the system will sent one copy of the bill to the customer's Email-address and another one for the system data base.

- **Stock Report:-**

Produce the quantity of the products available and status of the products.

- **Order Report:-**

List of the products that can be buy by the customer.

- **Delivery Report:-**

List of the products that can be delivered to the customer. Customer wants to buy the product then he/she must be registered, unregistered user can't go to the shopping cart.

- **Login:-**

Customer logs in to the system by entering valid user id and password for the shopping.

- **Changes to Cart:-**

Changes to cart means the customer after login or registration can make order or cancel order of the product from the shopping cart.

- **Payment:-**

In this system we are dealing the mode of payment by Cash. We will extend this to credit card, debitcard etc. in the future.

- **Logout:-**

After ordering or surfing for the product customer has to logout.

3.2 E R DIAGRAM

An **entity relationship diagram (ERD)** shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

A relationship is how the data is shared between entities. There are three types of relationships between entities that we observe in below diagram:

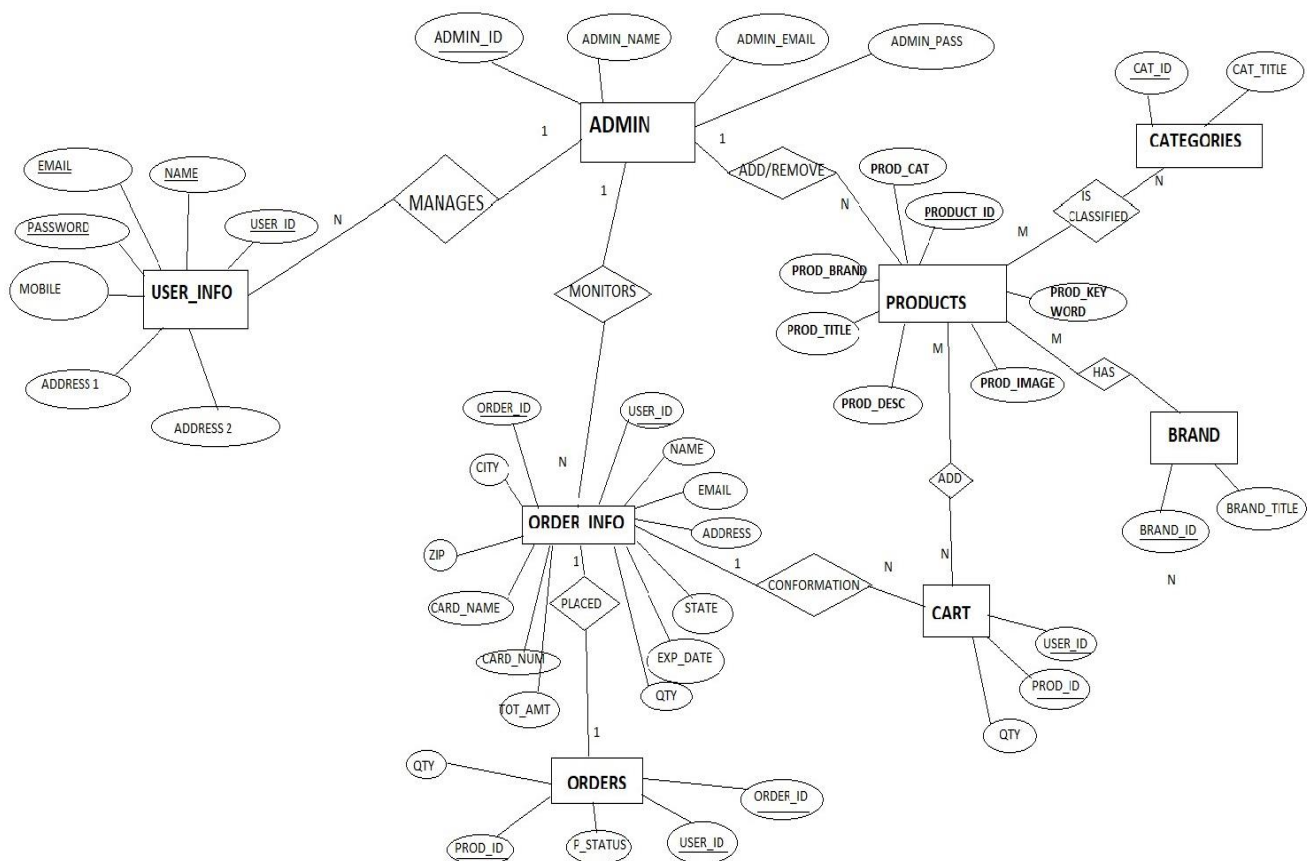


FIG 3.1 E-R Diagram of E-Commerce database

- Fig 3.1 represents the E-R diagram of ecommerce which contains entities and relationship between attributes.

3.3 SCHEMA DIAGRAM

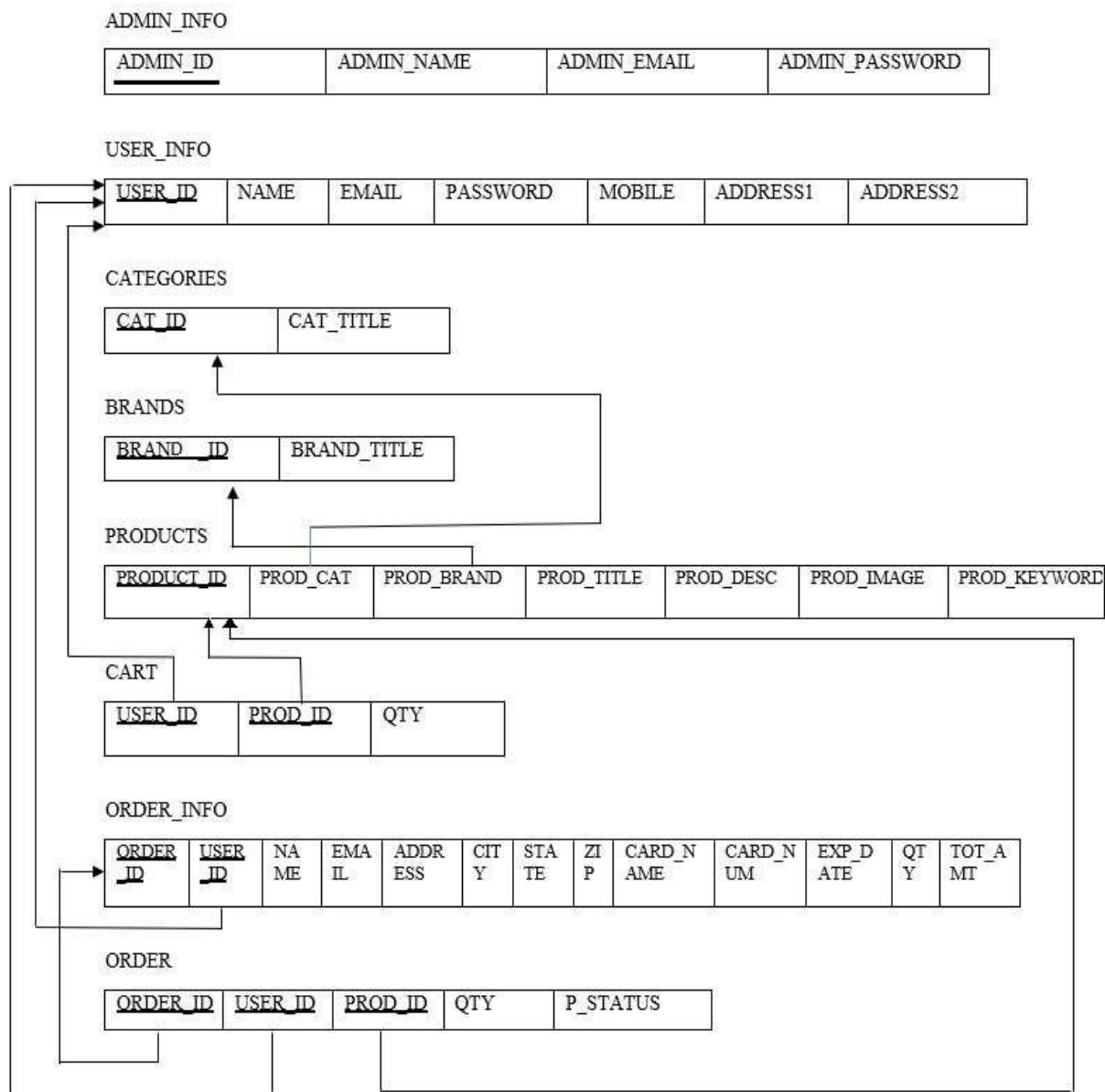


FIG 3.2 SCHEMA DIAGRAM of E-Commerce database

- The figure 3.2 describes the Relational Schema diagram of this project with all entities and attributes used.
- The underlined attributes specifies that they are primary key, and the arrow marks connects the foreign key with its primary key.

CHAPTER 4

IMPLEMENTATION

4.1 TECHNOLOGIES USED IN BUILDING THE PROJECT

4.1.1 HTML

HTML stands for Hypertext Markup Language, it is the standard markup language for creating web pages and web applications. With Cascading Style Sheets(CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a Web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages, with HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create standard documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, return using angle brackets. Browsers do not display the HTML tags, but use them to interpret the content of the page.

4.1.2 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML. Separation of formatting and content also makes feasible to present the same markup page in different styles rendering methods such as on screen in print by voice.

CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. What background images or colors are used, layout designs, variations.

4.1.3 JAVA SCRIPT

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

4.1.4 PHP

PHP is a servlet-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor. PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

4.1.5 SQL

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 [SAM 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. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Since then, the standard has been revised to include a larger set of features. 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".⁹¹ 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.

4.1 TABLE CREATION

SQL comprises both data definition and data manipulation languages. Using the data definition properties of SQL, one can design and modify database schema.

CREATE: Creates new databases, tables and views from RDBMS.

Syntax: CREATE TABLE table_name(attributes list with their datatypes);

CREATE TABLE COMMANDS:

```
CREATE TABLE ADMIN_INFO(  
  
ADMIN_ID NUMBER(5) PRIMARY  
KEY, ADMIN_NAME VARCHAR(30),  
ADMIN_EMAIL VARCHAR(30)  
ADMIN_PASSWORD VARCHAR(15)  
  
);
```

```
CREATE TABLE USER_INFO(  
USER_ID NUMBER(5) PRIMARY  
KEY, NAME VARCHAR(30),  
  
MOBILE NUMBER(10),  
EMAIL VARCHAR(20),  
PASSWORD  
VARCHAR(15),
```

```
ADDRESS1
VARCHAR(30),
ADDRESS2
VARCHAR(30)

);
```

```
CREATE TABLE CATAGORIES(
CAT_ID NUMBER(4) PRIMARY
KEY,CAT_TITLE VARCHAR(20),

);
```

```
CREATE TABLE BRANDS(
BRAND_ID NUMBER(4) PRIMARY
KEY,BRAND_TITLE VARCHAR(20),

);
```

```
CREATE TABLE PRODUCT(
PRODUCT_ID NUMBER(4) PRIMARY KEY,
PRODUCT_CAT NUMBER(4) REFERENCES CAT_ID(CATEGORIES) ON DELETE
CASCADE, PRODUCT_BRAND  NUMBER(4)  REFERENCES
BRAND_ID(CATEGORIES) ON DELETE CASCADE,
PROD_TITLE VARCHAR(30),
PROD_DESC VARCHAR(30),
PROD_IMAGE VARCHAR(30),
PROD_KEYWORD
VARCHAR(30)

);
```

```
CREATE TABLE CART(
USER_ID NUMBER(5) REFERENCES USER_ID(USER_INFO) ON DELETE
CASCADE,PROD_ID NUMBER(5) REFERENCES PROD_ID(PRODUCT) ON
DELETE CASCADE, QTY NUM(2)
```


);

```
CREATE TABLE ORDER_INFO(  
ORDER_ID NUMBER(5)  
PRIMARYKEY,  
USER_ID NUMBER(5) REFERENCES USER_ID(USER_INFO) ON DELETE  
CASCADE,NAME VARCHAR(30),  
  
EMAIL VARCHAR(20),  
ADDRESS  
VARCHAR(30),CITY  
VARCHAR(15), STATE  
VARCHAR(15), ZIP  
NUMBER(6),  
CARD_NAME VARCHAR(15),  
CARDNUMBER NUMBER(16),  
  
EXPDATE VARCHAR(6),  
PROD_COUNT NUMBER(4),  
TOT_AMT NUMBER(6), CVV  
NUMBER(3),  
  
);
```

```
CREATE TABLE ORDERS(  
ORDER_ID NUMBER(5) REFERENCES ORDER_ID(ORDER_INFO) ON DELETE  
CASCADE,USER_ID NUMBER(5) REFERENCES USER_ID(USER_INFO) ON  
DELETE CASCADE, PROD_ID NUMBER(5) REFERENCES PROD_ID(PRODUCT)  
ON DELETE CASCADE,  
  
QTY NUMBER(2),  
P_STATUS  
VARCHAR(8)  
  
);
```

4.2 CODE SNIPPETS

4.2.1 CONFIG – TO ESTABLISH CONNECTION

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$db = "onlineshop";
// Create connection
$con = mysqli_connect($servername, $username, $password,$db);
// Check connection
if (!$con) {
    die("Connection failed: " . mysqli_connect_error());
}
```

4.2.2 ADMIN ADD PRODUCT

```
<?php
session_start();
include("../db.php");

if(isset($_POST['btn_save']))
{
    $product_name=$_POST['product_name'];
    $details=$_POST['details'];
    $price=$_POST['price'];
    $c_price=$_POST['c_price'];
    $product_type=$_POST['product_type'];
    $brand=$_POST['brand'];
    $tags=$_POST['tags'];

    //picture coding
    $picture_name=$_FILES['picture']['name'];
    $picture_type=$_FILES['picture']['type'];
    $picture_tmp_name=$_FILES['picture']['tmp_name'];
    $picture_size=$_FILES['picture']['size'];

    if($picture_type=="image/jpeg" || $picture_type=="image/jpg" || $picture_type=="image/png" ||
    $picture_type=="image/gif")
    {

        if($picture_size<=50000000)
```

```
$pic_name=time()." ".$picture_name;
move_uploaded_file($picture_tmp_name,"../product_images/".$pic_name);

mysqli_query($con,"insert into products (product_cat, product_brand,product_title,product_price,
product_desc, product_image,product_keywords) values
('$product_type','$brand','$product_name','$price','$details','$pic_name','$tags')") or die ("query
incorrect");

header("location: sumit_form.php?success=1");
}

mysqli_close($con);
}
include "sidenav.php";
include "topheader.php";
?>
```

4.2.3 ADMIN EDIT USER

```
<?php
session_start();
include("../db.php");
$user_id=$_REQUEST['user_id'];

$result=mysqli_query($con,"select user_id,first_name,last_name, email, password from user_info
where user_id='$user_id'")or die ("query 1 incorrect..... ");

list($user_id,$first_name,$last_name,$email,$user_password)=mysqli_fetch_array($result);

if(isset($_POST['btn_save']))
{

$first_name=$_POST['first_name'];
$last_name=$_POST['last_name'];
$email=$_POST['email'];
$user_password=$_POST['password'];

mysqli_query($con,"update user_info set first_name='$first_name', last_name='$last_name',
email='$email', password='$user_password' where user_id='$user_id'")or die("Query 2 is inncorrect
");

header("location: manageuser.php");
mysqli_close($con);
}
```

```
include "sidenav.php";
include "topheader.php";
?>
```

4.2.4 USER LOGIN

```
<?php
include "db.php";

session_start();

if(isset($_POST["email"]) && isset($_POST["password"])){
    $email = mysqli_real_escape_string($con,$_POST["email"]);
    $password = $_POST["password"];
    $sql = "SELECT * FROM user_info WHERE email = '$email' AND password = '$password'";
    $run_query = mysqli_query($con,$sql);
    $count = mysqli_num_rows($run_query);
    $row = mysqli_fetch_array($run_query);
    $_SESSION["uid"] = $row["user_id"];
    $_SESSION["name"] = $row["first_name"];
    $ip_add = getenv("REMOTE_ADDR");

    if($count == 1){

        if (isset($_COOKIE["product_list"])) {
            $p_list = stripslashes($_COOKIE["product_list"]);

            $product_list = json_decode($p_list,true);
            for ($i=0; $i < count($product_list); $i++) {

                $verify_cart = "SELECT id FROM cart WHERE user_id =
$_SESSION[uid] AND p_id = ".$product_list[$i];
                $result = mysqli_query($con,$verify_cart);
                if(mysqli_num_rows($result) < 1){

                    $update_cart = "UPDATE cart SET user_id =
'$_SESSION[uid]' WHERE ip_add = '$ip_add' AND user_id = -1";
                    mysqli_query($con,$update_cart);
                }else{

                    $delete_existing_product = "DELETE FROM cart
WHERE user_id = -1 AND ip_add = '$ip_add' AND p_id = ".$product_list[$i];
                    mysqli_query($con,$delete_existing_product);
                }
            }
        }
    }
}
```

```
    }

    setcookie("product_list","",strtotime("-1 day"),"/");

    echo "cart_login";
    exit();

}
echo "login_success";
$BackToMyPage = $_SERVER['HTTP_REFERER'];
if(!isset($BackToMyPage)) {
    header('Location: '.$BackToMyPage);
    echo "<script type='text/javascript'>

        </script>";
} else {
    header('Location: index.php'); // default page
}

exit;

} else {
    $email = mysqli_real_escape_string($con,$_POST["email"]);
    $password = md5($_POST["password"]);
    $sql = "SELECT * FROM admin_info WHERE admin_email = '$email' AND
admin_password = '$password'";
    $run_query = mysqli_query($con,$sql);
    $count = mysqli_num_rows($run_query);

    if($count == 1){
        $row = mysqli_fetch_array($run_query);
        $_SESSION["uid"] = $row["admin_id"];
        $_SESSION["name"] = $row["admin_name"];
        $ip_add = getenv("REMOTE_ADDR");

        echo "login_success";

        echo "<script> location.href='admin/addproduct.php'; </script>";
        exit;

    } else {
        echo "<span style='color:red;'>Please register before login..!</span>";
        exit();
    }
}
```

```
    }  
  }  
?>
```

4.2.5 USER REGISTER

```
if (isset($_POST['reg_user'])) {  
    // receive all input values from the form  
    $username = mysqli_real_escape_string($db, $_POST['username']);  
    $email = mysqli_real_escape_string($db, $_POST['email']);  
    $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);  
    $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);  
  
    // form validation: ensure that the form is correctly filled ...  
    // by adding (array_push()) corresponding error unto $errors array  
    if (empty($username)) {  
        array_push($errors, "Username is required");  
    }  
    if (empty($email)) {  
        array_push($errors, "Email is required");  
    }  
    if (empty($password_1)) {  
        array_push($errors, "Password is required");  
    }  
    if ($password_1 != $password_2) {  
        array_push($errors, "The two passwords do not match");  
    }  
}
```

4.2.6 USER LOGOUT

```
<?php  
  
session_start();  
  
unset($_SESSION["uid"]);  
  
unset($_SESSION["name"]);  
  
$BackToMyPage = $_SERVER['HTTP_REFERER'];  
if(isset($BackToMyPage)) {  
    header('Location: '.$BackToMyPage);  
} else {  
    header('Location: index.php'); // default page  
}  
?>
```

4.2.7 CATEGORIES

```

<?php
session_start();
$ip_add = getenv("REMOTE_ADDR");
include "db.php";
if (isset($_POST["category"])) {
    $category_query = "SELECT * FROM categories";

    $run_query = mysqli_query($con, $category_query) or die(mysqli_error($con));
    echo "

                                <div class='aside'>
                                <h3 class='aside-title'>Categories</h3>
                                <div class='btn-group-vertical'>

    ";
    if (mysqli_num_rows($run_query) > 0) {
        $i = 1;

        while ($row = mysqli_fetch_array($run_query)) {

            $cid = $row["cat_id"];
            $cat_name = $row["cat_title"];
            $sql = "SELECT COUNT(*) AS count_items FROM products WHERE
product_cat=$i";

            $query = mysqli_query($con, $sql);
            $row = mysqli_fetch_array($query);
            $count = $row["count_items"];
            $cho "

                                <div type='button' class='btn navbar-btn category' cid='$cid'>

    <a href='#'>
    <span ></span>
    $cat_name
    <small class='qty'>($count)</small>
    </a>
    </div>“;
                                echo "</div>";
        }
    }
}

```

4.2.8 PRODUCTS

```

<?php

```

```
include 'db.php';
$product_id = $_GET['p'];

$sql = " SELECT * FROM products ";
$sql = " SELECT * FROM products WHERE product_id = $product_id";
if (!$con) {
die("Connection failed: " . mysqli_connect_error());
}
$result = mysqli_query($con, $sql);
if (mysqli_num_rows($result) > 0) {
while ($row = mysqli_fetch_assoc($result)) {
echo '
<div class="product-preview">

</div>

?>
```

4.2.9 ADD TO CART

```
if (isset($_POST["addToCart"])) {

    $p_id = $_POST["proId"];

    if (isset($_SESSION["uid"])) {

        $user_id = $_SESSION["uid"];

        $sql = "SELECT * FROM cart WHERE p_id = '$p_id' AND user_id = '$user_id'";
        $run_query = mysqli_query($con, $sql);
        $count = mysqli_num_rows($run_query);
        if ($count > 0) {
            echo "
                <div class='alert alert-warning'>
                    <a href='#' class='close' data-dismiss='alert' aria-
label='close'>&times;</a>
                    <b>Product is already added into the cart Continue

        $sql = "SELECT id FROM cart WHERE ip_add = '$ip_add' AND
p_id = '$p_id' AND user_id = -1";
        $query =
        mysqli_query($con,
        $sql); if
        (mysqli_num_rows($que
```



```
        ry) > 0) {
            echo "
                                <div class='alert alert-warning'>
                                <a href='#' class='close' data-dismiss='alert'
aria-label='close'>&times;</a>
                                <b>Product is already added into the cart
Continue Shopping..!</b>
                                </div>";
            exit();
        }

        $sql = "INSERT INTO `cart`
                (`p_id`, `ip_add`, `user_id`, `qty`)
                VALUES ('$p_id','$ip_add','-1','1')";
        if (mysqli_query($con, $sql)) {
            echo "
                                <div class='alert alert-success'>
                                <a href='#' class='close' data-dismiss='alert' aria-
label='close'>&times;</a>
                                <b>Your product is Added Successfully..!</b>
                                </div>
                                ";
            exit();
        }
    }
}
```

4.3.0 CHECKOUT

```
<?php
session
_start()
include
"db.ph
p";
if (isset($_SESSION["uid"])) {

    $f_name = $_POST["firstname"];
    $email = $_POST['email'];
    $address = $_POST['address'];
    $city = $_POST['city'];
    $state = $_POST['state'];
    $zip= $_POST['zip'];
    $cardname= $_POST['cardname'];
    $cardnumber= $_POST['cardNumber'];
    $expdate= $_POST['expdate'];
    $cvv= $_POST['cvv'];
    $user_id=$_SESSION["uid"];
    $cardnumberstr=(string)$cardnumber;
    $total_count=$_POST['total_count'];
    $prod_total = $_POST['total_price'];

    $sql0="SELECT order_id from `orders_info`;
    $runquery=mysqli_query($con,$sql0);
    if
        (mysqli_num_rows($runq
        uery) == 0) {echo(
        mysqli_error($con));
        $order_id=1;
    }else if (mysqli_num_rows($runquery) > 0) {
        $sql2="SELECT MAX(order_id) AS max_val from `orders_info`;
        $runquery1=mysqli_query($con,$sql2);
        $row = mysqli_fetch_array($runquery1);
        $order_id= $row["max_val"];
        $order_id=$or
        der_id+1;
        echo(
        mysqli_error($
        con));
    }

    $sql = "INSERT INTO `orders_info`
    (`order_id`,`user_id`,`f_name`,
    `email`,`address`,
    `city`,`state`,`zip`,
    `cardname`,`cardnumber`,`expdate`,`prod_count`,`total_amt`,`cvv`)
    VALUES ($order_id, '$user_id','$f_name','$email',
    '$address', '$city', '$state',
```

```
'$zip','$cardname','$cardnumberstr','$expdate','$total_count','$prod_to
tal','$scvv');";
```

```
if(mysqli_query($con,$sql)){
    $i=1;
    $prod_id_=0;
    $prod_price_=0;
    $prod_qty_=
    0;
    while($i<=$t
otal_count){
        $str=(string)$i;
        $prod_id_+$str = $_POST['prod_id_'].$i;
        $prod_id=$prod_id_+$str;
        $prod_price_+$str = $_POST['prod_price_'].$i;
        $prod_price=$prod_price_+$str;
        $prod_qty_+$str = $_POST['prod_qty_'].$i;
        $prod_qty=$prod_qty_+$str;
        $sub_total=(int)$prod_price*(int)$prod_qty;
        $sql1="INSERT INTO `order_products`
(`order_pro_id`,`order_id`,`product_id`,`qty`,`amt`)
VALUES (NULL,
'$order_id','$prod_id','$prod_qty','$sub_total')";
        if(mysqli_query($con,$sql1)){
            $del_sql="DELETE from cart where
            user_id=$user_id";
            if(mysqli_query($con,$del_sql)){
                echo"<script>window.location.href='store.php'</script>";
            }else{
                echo(mysqli_error($con));
            }
        }else{
            echo(mysqli_error($con));
        }
        $i++;
    }
}else{
    echo(mysqli_error($con));
}
}
?>
```

4.3.1 TRIGGERS

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in database. The trigger is most used for maintaining the integrity of the information on the database. For example, when a new record is added to the patient table, if he needs room, room table is automatically updated. Triggers can also be used to log historical data, for example keeping track of patient and their medical history.

```
{CREATE | RECREATE | CREATE OR ALTER} TRIGGER name FOR {table name | view
name}
{ACTIVE/INACTIVE}
{BEFORE|AFTER}
{INSERT [OR UPDATE] [OR DELETE] | UPDATE [OR INSERT] [OR
DELETE] | DELETE[OR|UPDATE][OR INSERT] }
[POSITION]ASBEGIN END
CREATE TRIGGER USER_TRIGBEFORE INSERT
ON orders FOR EACH ROW
INSERT INTO TRIGGER_TIME VALUES (NOW ())
```

```
CREATE TRIGGER `after_user_info_insert` AFTER INSERT ON `user_info` FOR
EACH ROW BEGIN
INSERT INTO user_info_backup
VALUES(new.user_id,new.first_name,new.last_name,new.email,new.password,new.
mobile,new.address1,new.address2);
END
```

CHAPTER 5

TESTING AND RESULTS

5.1 TESTING

Testing performs a very crucial role in quality assurance and ensuring the reliability of the software. During testing, the program to be tested is executed as a set of test cases and the output is performed as expected. The service of testing for errors in a program depends critically on the test cases.

5.1.1 TESTING PROCESS

Basic levels of testing include

- Unit testing.
- Integration testing.
- System testing.

5.1.2 UNIT TESTING

Unit testing focuses verification on the smallest unit of the software design, the software components or module. The unit testing was implemented in the project to test the following:

- Each module was tested to ensure that the information flows properly into and out of the program unit under test.
- Each unit was tested to ensure its proper functioning. Different modules are tested independently to ensure that the desired output is obtained by each module.

5.1.3 INTEGRATION TESTING

Integration testing is the systematic technique for constructing the program structures. While at the same time, tests are conducted to uncover errors associated with the interfacing. The objective is, to use the unit tested components and build a program structure that has been designed. During the testing phase, a bottom-up approach is implemented. The bottom-up approach of integration testing is carried out in the following way:

- The components at the bottom level which have no subordinates are tested.
- After combining, different components are linked together to the main menu to ensure

5.1.4 SYSTEM TESTING

Here we test the system according to the requirements that are specified in the requirements analysis. In this level, all the modules are integrated into a system and then they are tested. Thus, testing starts from the main menu and ends with end results.

System testing involves testing the software with following:

- Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called the End to End testing scenario.
- Verify thorough testing of every input in the application to check for desired outputs. Testing of the user's experience with the application.

WEBSITE SNAPSHOTS

5.1 HOME PAGE

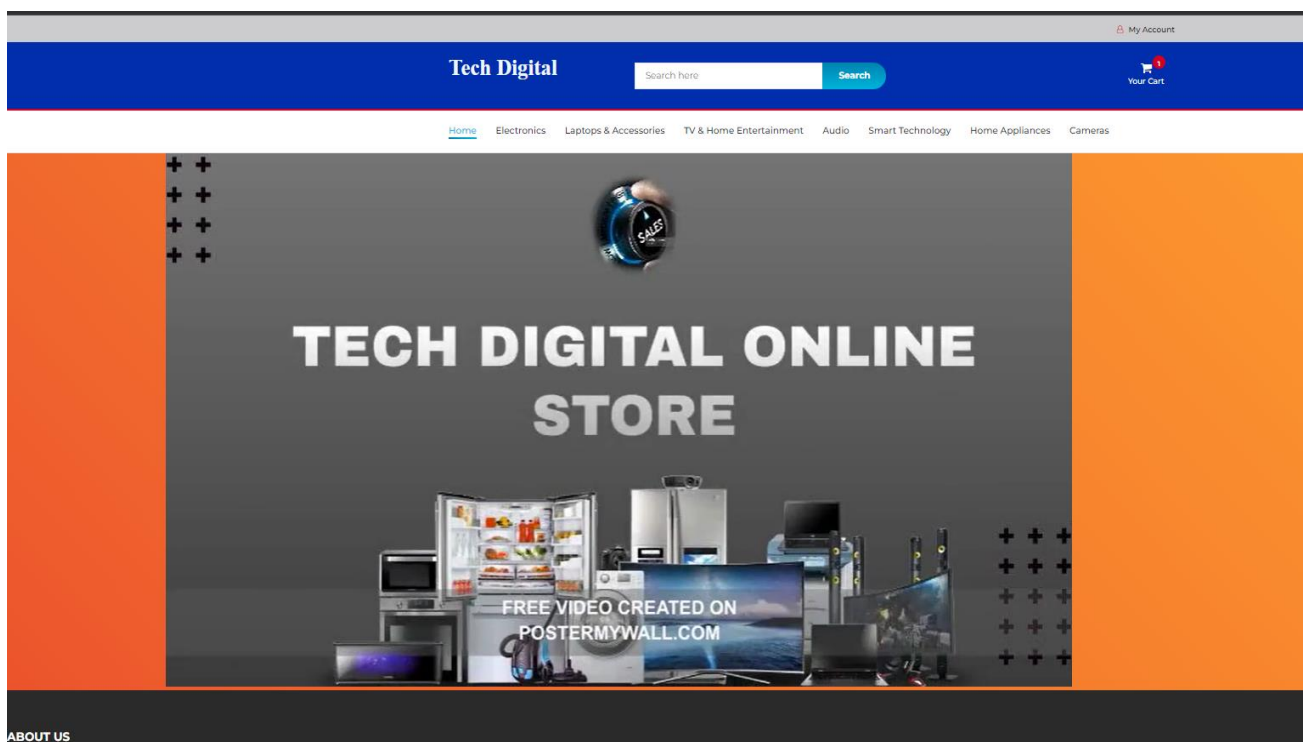


Fig 5.1 Home and Login Page

- The above figure 5.1 shows the screenshot of the Homepage of our Project.
- From this window, the user can navigate to the various other windows (like Electronics, Laptops, audio etc) using the navigation bar.

5.2 LOGIN PAGE

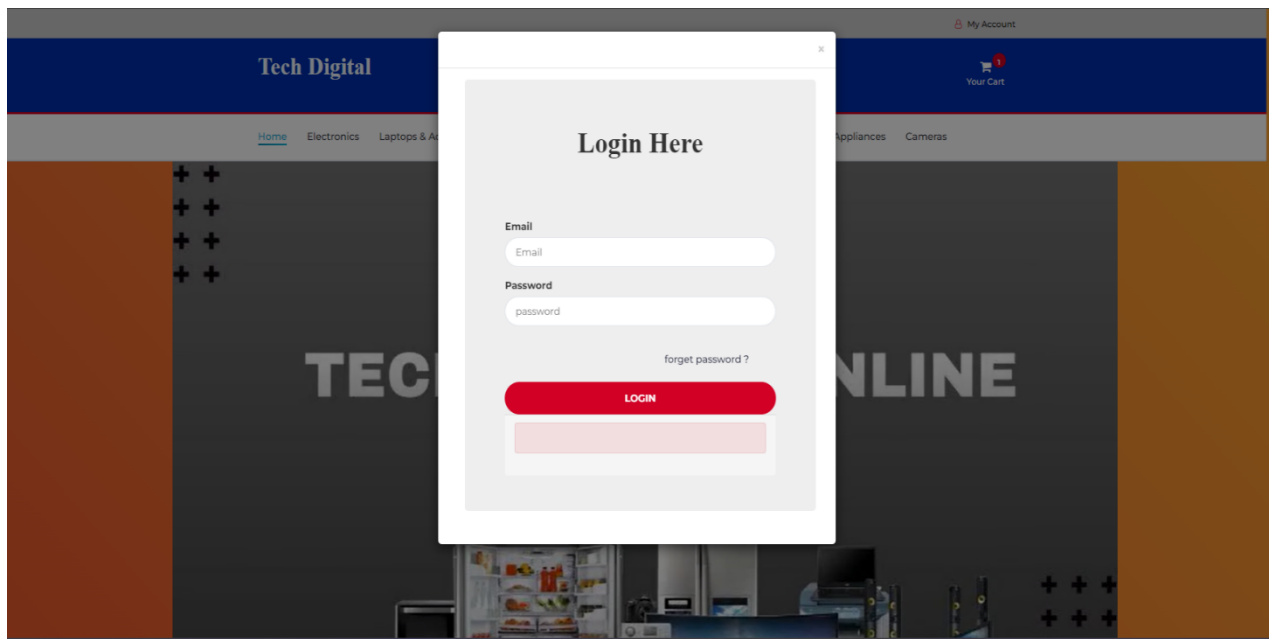


Fig 5.2 Login Page

- The figure 5.2 shows the login page of our project.
- This login page allows a user to enter their username and password to access a particular service or website.

5.3 SIGNUP PAGE

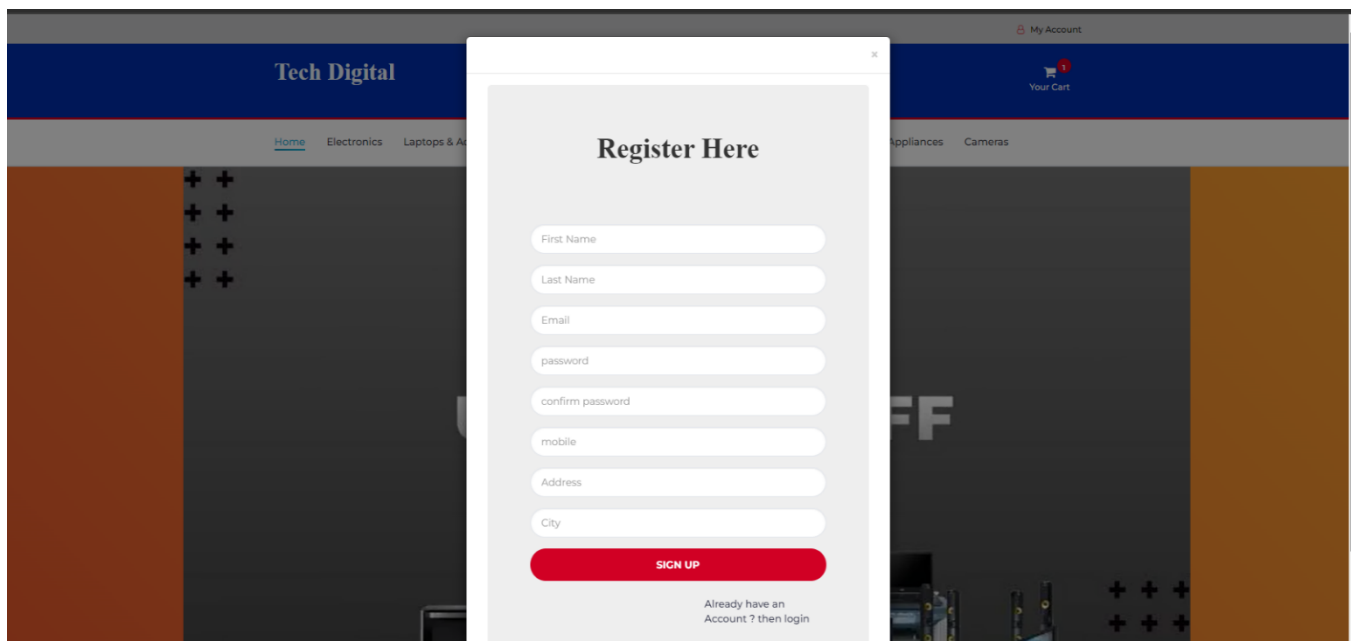


Fig 5.3 Signup Page

The figure 5.3 shows the signup page of the project.

- This signup page allows users to create a new account on a website or application.
- It typically includes a form where the user can enter their personal information, such as their name, email address, and password.
- Once the user submits the form, their account will be created, and they will be able to log in to the website or application with the credentials they provided.
- The purpose of a signup page is to allow users to create an account so they can access the features and content of the website or application.

5.4 USER CATEGORY PAGE

- The figure 5.4 shows the categories of products of this project.
- Here in this window, the user is able to view different categories of the electronic products such as tv, laptops, cameras etc.

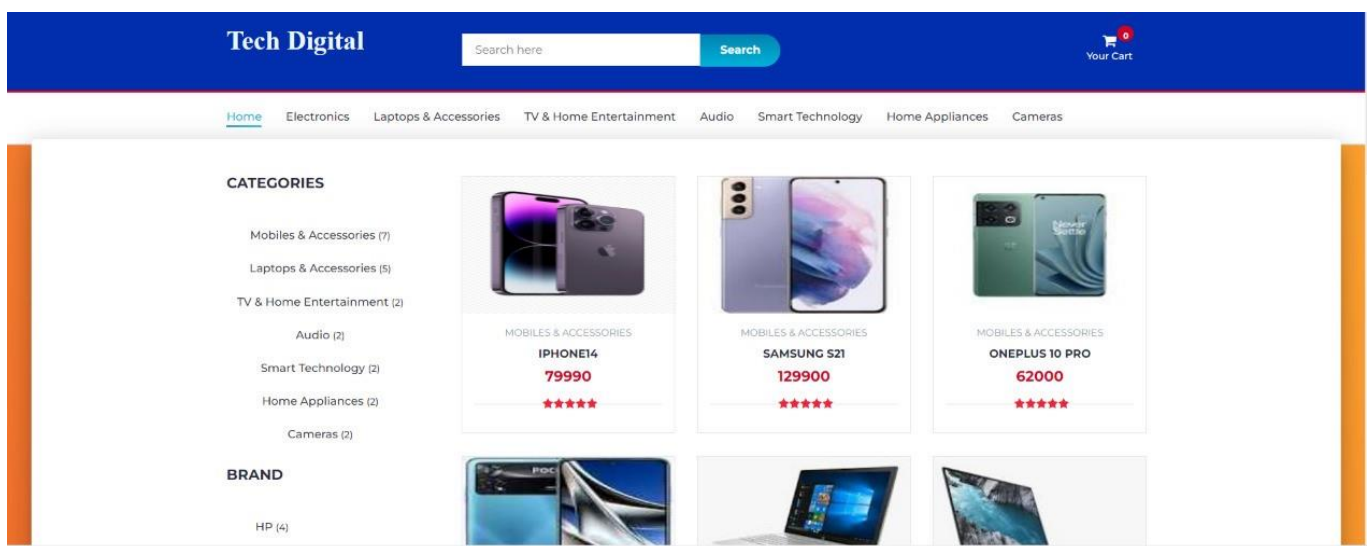


Fig 5.4 USER CATEGORY PAGE

5.5 USER CART PAGE

- The user is able to add the product to the cart which he/her wishes to buy and can press the ready to checkout button.

Tech Digital Search here Search Your Cart

Home Electronics Laptops & Accessories TV & Home Entertainment Audio Smart Technology Home Appliances Cameras

Product	Price	Quantity	Subtotal
<p>Samsung s21</p> <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,</p>	129900	1	129900

Total : \$ 129900 Ready to Checkout

ABOUT US

ONLINE SHOPPING (ELECTRONICS)

Fig 5.5 USER CART PAGE

- The above fig 5.5 show the book now page this project.

5.6 USER PAYMENT INTERFACE PAGE

Tech Digital Search here Search Your Cart

Home Electronics Laptops & Accessories TV & Home Entertainment Audio Smart Technology Home Appliances Cameras

Billing Address

Full Name

Email

Address

City

State

Zip

☐ Shipping address same as billing

Payment

Accepted Cards

Name on Card

Card Number

Exp Date

CVV

Cart

no	product title	qty	amount
1	Samsung s21	1	129900

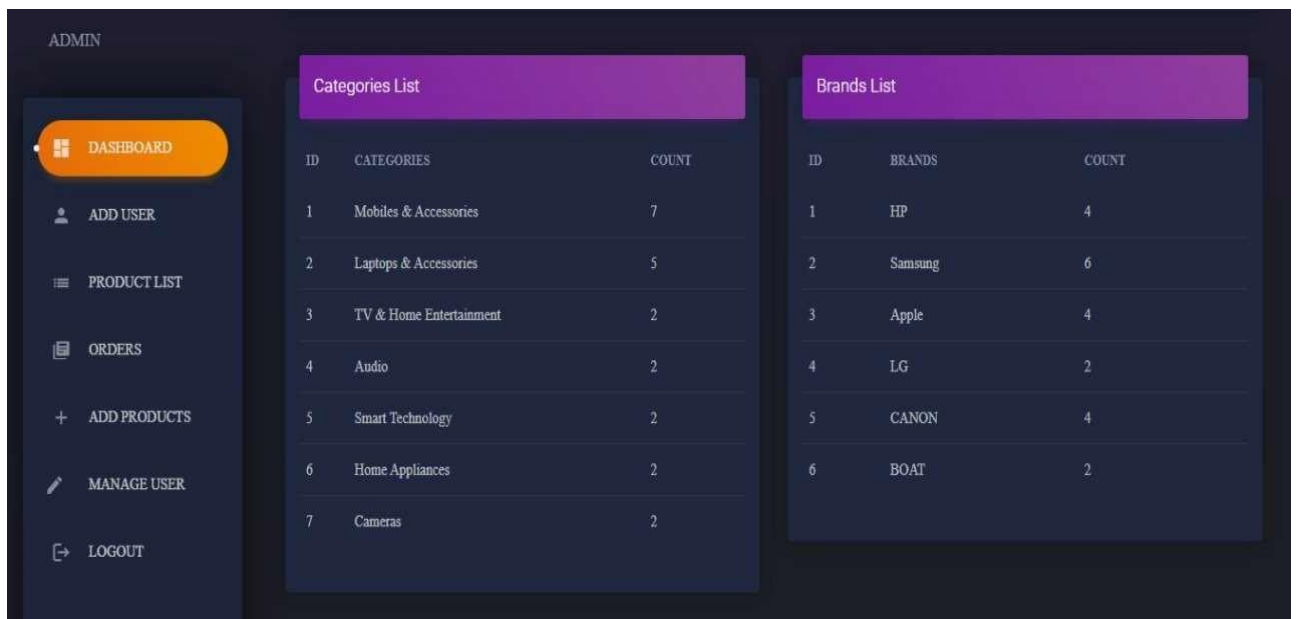
total \$129900

Continue to checkout

Fig 5.6 USER PAYMENT INTERFACE PAGE

- The above fig 5.6 show the user payment interface page this project.
- User is able to fill the billing address such as city, state, pincode and make the payment transactions by entering the cardnumber and cvv.

5.7 ADMIN DASHBOARD PAGE



The Admin Dashboard Page features a sidebar with navigation options: DASHBOARD (selected), ADD USER, PRODUCT LIST, ORDERS, ADD PRODUCTS, MANAGE USER, and LOGOUT. The main content area displays two tables: 'Categories List' and 'Brands List'.

ID	CATEGORIES	COUNT
1	Mobiles & Accessories	7
2	Laptops & Accessories	5
3	TV & Home Entertainment	2
4	Audio	2
5	Smart Technology	2
6	Home Appliances	2
7	Cameras	2

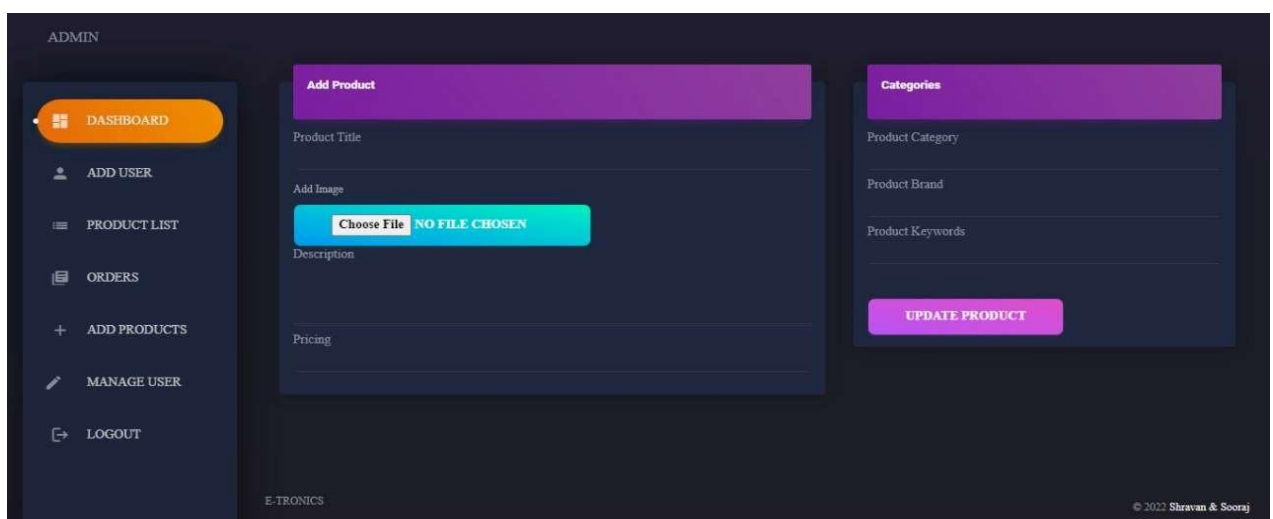
ID	BRANDS	COUNT
1	HP	4
2	Samsung	6
3	Apple	4
4	LG	2
5	CANON	4
6	BOAT	2

Fig 5.7 Admin Dashboard Page

- The above fig 5.7 shows the Admin Dashboard Page of this project.
- An admin dashboard page for a E-COMMERCE system would typically allow the administrator to manage various aspects of the electronic items.

5.8 ADMIN ADD PRODUCTS

- This page would typically allow an administrator to add new products to the sys.



The Admin Add Product Form includes a sidebar with navigation options: DASHBOARD (selected), ADD USER, PRODUCT LIST, ORDERS, ADD PRODUCTS, MANAGE USER, and LOGOUT. The main content area displays a form for adding a new product.

Add Product

Product Title

Add Image

NO FILE CHOSEN

Description

Pricing

Categories

Product Category

Product Brand

Product Keywords

E-TRONICS © 2022 Shravan & Sooraj

Fig 5.8 ADMIN ADD PRODUCT

- The figure 5.8 shows the admin add products.

5.9 ADMIN LOGIN PAGE

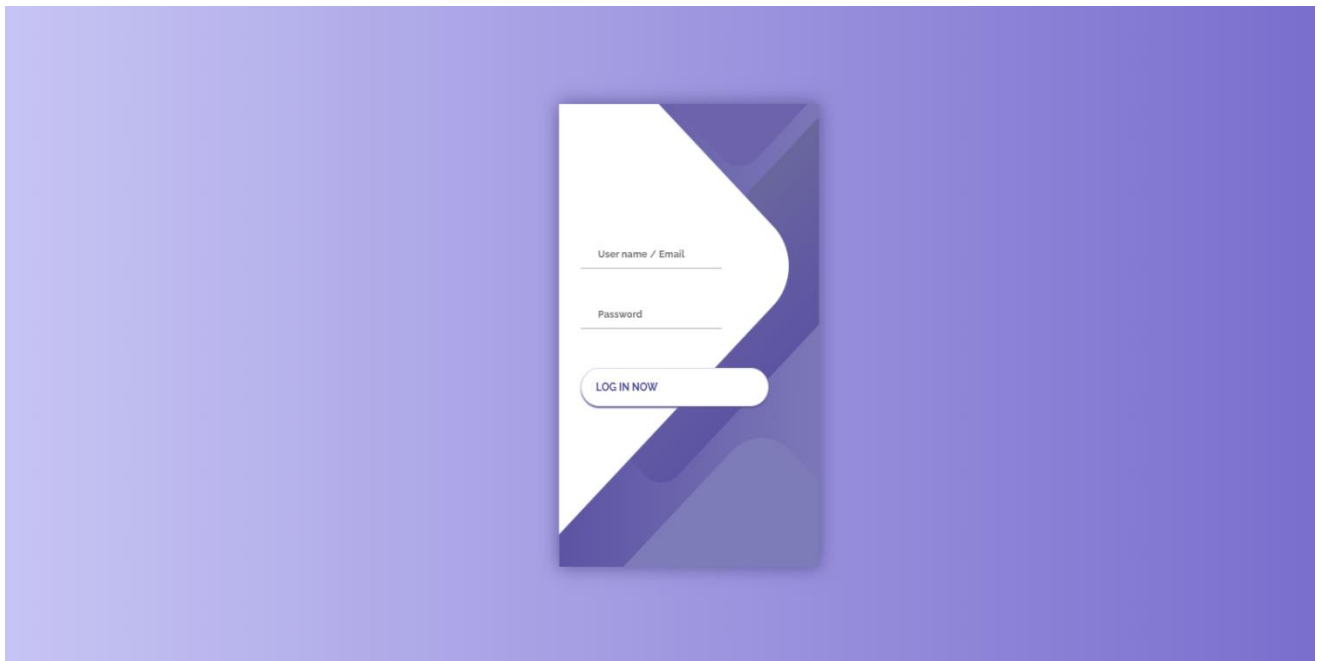


Fig 5.9 Admin login page

- The above figure 5.9 describes the Booking page of the admin page in this project.
- This page allows an administrator with the appropriate permissions to add, update and delete products and can view various users.

SNAPSHOTS OF DATABASE:

5.10 ADMIN TABLE

Server: 127.0.0.1 » Database: onlineshop » Table: admin_info

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

`SELECT * FROM `admin_info``

Number of rows: 25 Filter rows: Search this table Sort by key: None

	admin_id	admin_name	admin_email	admin_password
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	monik	monik@gmail.com	123456
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	pavansg	pavan12gowdru@gmail.com	123456

Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations: Print Copy to clipboard Export Display chart Create view

Fig 5.10 Admin Details

- The figure 5.10 shows the admin details, where we can see the usernames and passwords stored in the database for the admin login.

5.11 BRANDS TABLE

Server: 127.0.0.1 » Database: onlineshop » Table: brands

Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

`SELECT * FROM `brands``

Number of rows: 25 Filter rows: Search this table Sort by key: None

	brand_id	brand_title
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	HP
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Samsung
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Apple
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	LG
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	CANON
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	BOAT

Number of rows: 25 Filter rows: Search this table Sort by key: None

Fig 5.11 Brands

- The figure 5.11 shows various brands in the electronic items.
- These users have access, they can view various images of the product and buy the gadgets which he/her wish.

5.12 CART TABLE

Server: 127.0.0.1 » Database: onlineshop » Table: cart

Showing rows 0 - 18 (19 total, Query took 0.0003 seconds.)

`SELECT * FROM `cart``

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	p_id	ip_add	user_id	qty
<input type="checkbox"/> Edit Copy Delete	6	26	::1	4	1
<input type="checkbox"/> Edit Copy Delete	9	10	::1	7	1
<input type="checkbox"/> Edit Copy Delete	10	11	::1	7	1
<input type="checkbox"/> Edit Copy Delete	11	45	::1	7	1
<input type="checkbox"/> Edit Copy Delete	44	5	::1	3	0
<input type="checkbox"/> Edit Copy Delete	46	2	::1	3	0
<input type="checkbox"/> Edit Copy Delete	48	72	::1	3	0
<input type="checkbox"/> Edit Copy Delete	49	60	::1	8	1
<input type="checkbox"/> Edit Copy Delete	50	61	::1	8	1

Fig 5.12 Cart

- The figure 5.12 shows the products in the cart.
- These are the products which are stored in the database along with the quantity.

5.13 CATEGORIES TABLE

- The fig 5.13 contains various categories.
- This categories include tv, mobile, laptop, audio, camera and various home appliances.

Server: 127.0.0.1 » Database: onlineshop » Table: categories

Showing rows 0 - 6 (7 total, Query took 0.0003 seconds.)

`SELECT * FROM `categories``

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	cat_id	cat_title
<input type="checkbox"/> Edit Copy Delete	1	Mobiles & Accessories
<input type="checkbox"/> Edit Copy Delete	2	Laptops & Accessories
<input type="checkbox"/> Edit Copy Delete	3	TV & Home Entertainment
<input type="checkbox"/> Edit Copy Delete	4	Audio
<input type="checkbox"/> Edit Copy Delete	5	Smart Technology
<input type="checkbox"/> Edit Copy Delete	6	Home Appliances
<input type="checkbox"/> Edit Copy Delete	7	Cameras

Fig 5.13 Categories

5.14 ORDERS TABLE

Server: 127.0.0.1 » Database: onlineshop » Table: orders

Showing rows 0 - 0 (1 total, Query took 0.0002 seconds)

SELECT * FROM `orders`

Number of rows: 25 Filter rows: Search this table

order_id	user_id	product_id	qty	trx_id	p_status
2	14	2	1	07M47684BS5725041	Completed

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view

Fig 5.14 Orders

- The figure 5.14 shows the orders of the product.
- It contains the status of the ordered product along with the quantity.

5.15 ORDERS_INFO TABLE

Server: 127.0.0.1 » Database: onlineshop » Table: orders_info

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds)

SELECT * FROM `orders_info`

Number of rows: 25 Filter rows: Search this table Sort by key: None

order_id	user_id	f_name	email	address	city	state	zip	cardname	cardnumber	expdate	prod_count	total_amt	cvv
2	26	ABC	abc@gmail.com	ABC122	Narasimharajapura	Karnataka	577134	abc	5465 4552 5585	12/22	2	30000	232
3	26	abc d	abc@gmail.com	goa	goa	Karnataka	577240	abc	852045236985	12/25	1	5000	210
4	27	shravan	shravanhn@gmail.com	nrpura.chickmaglur	nrpura	Karnataka	577134	shravanhn	4521 0125 8569 1114	12/23	2	35000	111
5	29	ABC	asd@gmail.com	ABC122	Narasimharajapura	Karnataka	577134	abc	852142501142	12/25	1	5000	123
6	30	pavan	pavan12goudru@gmail.com	ARASINAGATTA LINGADAHALLI POST CHANNAGIRI TALUK, D.	CHANNAGIRI	Karnataka	577221	ppp	1232321456543	12/24	1	25000	1233

Fig 5.15 Orders info

- The figure 5.15 shows the information of the ordered products.
- It contains various fields of the address billing and payment transactions.

5.16 ORDER PRODUCTS TABLE

Server: 127.0.0.1 > Database: onlineshop > Table: order_products

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

`SELECT * FROM `order_products``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	order_pro_id	order_id	product_id	qty	amt
<input type="checkbox"/> Edit Copy Delete	96	5	1	1	5000
<input type="checkbox"/> Edit Copy Delete	97	6	2	1	25000

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Fig 5.16 Order products

- The figure 5.16 shows the information of the order products.
- It contains fields such as amount of the product and the quantity bought.

5.17 PRODUCTS TABLE

Server: 127.0.0.1 > Database: onlineshop > Table: products

Showing rows 0 - 21 (22 total, Query took 0.0003 seconds.)

`SELECT * FROM `products``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	product_id	product_cat	product_brand	product_title	product_price	product_desc	product_image	product_keywords
<input type="checkbox"/> Edit Copy Delete	1	1	2	Samsung S21	50000	Samsung S21	s21.jpg	samsung mobile electronics
<input type="checkbox"/> Edit Copy Delete	2	1	3	iPhone 5s	25000	iphone 5s	http___pluspng.com_img-png_iphone-hd-png-iphone-ap...	mobile iphone apple
<input type="checkbox"/> Edit Copy Delete	3	1	3	iPad air 2	30000	ipad apple brand	da4371ffa192a115f922b1c0d808193.png	apple ipad tablet
<input type="checkbox"/> Edit Copy Delete	4	1	3	iPhone 6s	32000	Apple iPhone	http___pluspng.com_img-png_iphone-6s-png-iphone-6s...	iphone apple mobile
<input type="checkbox"/> Edit Copy Delete	8	1	4	Sony	40000	Sony Mobile	530201353846AM_635_sony_xperia_z.png	sony mobile
<input type="checkbox"/> Edit Copy Delete	33	6	2	Refrigerator	35000	Refrigerator	CT_WM_BTS-BTC-AppliancesHome_20150723.jpg	refrigerator samsung
<input type="checkbox"/> Edit Copy Delete	36	6	5	Iron	15000	gj	iron.JPG	iron
<input type="checkbox"/> Edit Copy Delete	46	1	2	Samsung Galaxy Note 3	10000		samsung_galaxy_note3_note3neo.JPG	samsung galaxy note 3 neo
<input type="checkbox"/> Edit Copy Delete	71	1	2	Samsung galaxy s7	5000	Samsung galaxy s7	product07.png	samsung mobile electronics

Fig 5.17 products

- The figure 5.18 shows the information of the products.
- Such as product title, description and rating of the products.

5.18 USER_INFO TABLE

The screenshot shows the phpMyAdmin interface for the 'onlineshop' database. The 'user_info' table is selected, and its structure is displayed. The table has 9 columns: user_id, first_name, last_name, email, password, mobile, address1, address2, and address3. The data is shown in a table format with 7 rows. The interface includes a sidebar with a tree view of the database structure, a top navigation bar with tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. The main area shows the table's structure and data, with a status bar at the bottom indicating 'Showing rows 0 - 6 (7 total, Query took 0.0003 seconds)'.

	user_id	first_name	last_name	email	password	mobile	address1	address2	address3
<input type="checkbox"/>	26	abc	d	abc@gmail.com	123456789	1234567891	goa	goa	
<input type="checkbox"/>	27	shravan	hn	shavanhn@gmail.com	123456789	9632604496	Nrpora, chickmaglur	nrpora	
<input type="checkbox"/>	28	aaa	bbb	aa@gmail.com	123456789	0963260449	Narasimharajapura	India	
<input type="checkbox"/>	29	asdf	dgthg	asdf@gmail.com	123456789	9632604496	, karnataka	karnataka 5	
<input type="checkbox"/>	30	PAVAN	aasas	p00@gmail.com	Pavan@123	7411771796	ARASINAGATTA LINGADAHALLI POST CHANNAGIRI TALUK DAVANGERE D		
<input type="checkbox"/>	31	PAVANSG	sg	p@gmail.com	Vikibidre@21	7411771765	ARASINAGATTA LINGADAHALLI POST CHANNAGIRI TALUK DAVANGERE D		
<input type="checkbox"/>	32	sppp	raju	monik@gmail.com	123456	9525567551	CHANNAGIRI	India	

Fig 5.18 user info

- The figure 5.18 shows the user information.
- It contains fields of various users who have signed up the website along with their basic details.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

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

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

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

There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like providing moderator more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of history of purchases of each customer and provide suggestions based on their history. These features could have been implemented unless the time did not limit us.

The project made here is just to ensure that this product could be valid in today's challenging world. Here all the facilities are made and tested. Currently the system works for a limited number of administrators to work. In near future it will be extended for many types of insurance policies so that efficiency can be improved.