

# **PROJECT REPORT ON E-COMMERCE STORE**

Submitted in partial fulfilment  
of the requirement for B.tech 6<sup>th</sup> Semester



**SUBMITTED TO: -**

**DR. CHANDER DIWAKAR**

ASSISTANT PROFESSOR

CSE DEPARTMENT

**SUBMITTED BY: -**

**SUMAN**

ROLL NO.: - 251902174

CSE-B (VI SEMESTER)

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY,  
KURUKSHETRA

(2019-2023)

# CONTENTS

Chapters	<u>Particulars</u>	
	Declaration_____	3
	Abbreviations_____	4
1.	Introduction_____	5-6
	1.1 OBJECTIVE	
	1.2 Goals	
	1.3 SCOPE	
2.	Layout & Architecture Of E-Commerce Store_____	7-13
	2.1 Description	
	2.2 Structure of the Lifestyle	
	2.3 Data Flow Diagram	
3.	Software & Hardware Requirements & Analysis_____	14-24
	3.1 Analysis	
	3.2 Software Requirements	
	3.3 Hardware Requirements	
	3.4 Language and Technologies Used	
4.	Elements Of E-Commerce Store_____	25-36
	4.1 File Description	
	4.2 Output	
5.	Conclusion_____	36-37
6.	Reference_____	38
7.	Code Of Project_____	39-58

## **DECLARATION**

I hereby declare that the work which is being presented in the My Project Report on “Web Development”, is an authentic record of my own work carried under the supervision of “Dr. Chander Diwakar”. During the semester I worked at Web Development and completed my project entitled “Ecommerce-store”. All the information furnished in this project report is based on my own intensive work and is genuine.

**Suman**

**251902174**

## **Abbreviations**

CSS	Cascading Style Sheets
DBMS	Database Management System.
DDL	Data Definition Language
DML	Data Manipulation Language
GUI	Graphical User Interface
HTML	HyperText Markup Language
PHP	PHP is a general-purpose scripting language geared towards web development.
SQL	Structured Query Language, used to create the database.
SQL Server	SQL server is a database management system in which database is created and manages the database like “update”, “Delete” and insert a new record in the database.

# **1. INTRODUCTION**

## **Project Title: E-Commerce Store**

### **1.1 OBJECTIVE:**

1. An E-commerce portal which will allow formal and informal merchants in developing countries to advertise and sell their goods on the internet.
2. This would permit rural communities to make their wares available to the rest of the world via the world wide web.
3. The objective of this project is to create an e-commerce web portal with a content management system which would allow product information to be updated securely using a mobile device.
4. The web portal will have an online interface in the form of an e-commerce website that will allow users to buy goods from the merchants.

### **1.2 Goals:**

The goal of this website is to develop an e-commerce store where product like cameras, watches and clothing can be bought from the comfort of home through the Internet. However, for implementation purposes, this paper will deal with an online shopping.

An E-store is a virtual store on the Internet where customers can browse the catalogue and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. Usually, the customer will be asked to fill or select a

billing address, a shipping address, a shipping option. The main goal of this website is to develop an easy to use web-based interface where Customer can search for products, view a complete description of the product and order the product.

### **1.3 SCOPE:**

Scope of this project is very broad in terms of other manually functioning:

- a) It can be used by user to get information about the latest brands.
- b) It can be used for small businesses to sell their products.
- c) It is based on web application we can easily access it from anywhere.

## **2. Layout & Architecture of E-Commerce Store**

### **2.1 Description**

“**LIFESTYLE STORE**” is a shopping website which is to be created in the given format. The following points describe completely the functionality of the website from the user’s point of view. The website will contain the following pages:

1. Home Page
2. Sign up Page
3. Login Page
4. About Us Page
5. Products Page
6. Cart Page
7. Order Page

#### **2.1.1 Home page:**

The Home page or index page is usually the first page that a user sees on any website.

The page needs to be divided into three parts:

**Navbar (fixed navbar at the top):** It is a collapsing navbar. It contains the following options:

- a) Name of store
- b) Products
- c) 3.About us
- d) Login
- e) Signup

If user has logged in, then the nav-bar will be different:

- a) Name of store
- b) Products
- c) About us
- d) Cart
- e) Orders

f) Logout

### **Body:**

- a) The body part consists of two parts. The upper part has a background image and a semi-transparent box inside which we have a button (Shop now).
- b) When you click the button it redirects the visitor to the login page.

### **Footer:**

The footer contains text: Copyright © Lifestyle Store. All Rights Reserved | Contact Us: +91 90000 00000

### **2.1.2 SIGNUP PAGE:**

- a) On the signup page, new users can register using the ‘Sign Up’ form and create accounts, while existing users can log in using the ‘Login’ page.
- b) The signup form has four fields i.e. ‘name’, ‘email’, ‘password’, ‘phone number’. What you cannot see is the backend or the functionality of the form, which comprises of form validation and form injection.

### **FORM VALIDATION**

In order for the data (entered by any user) to be valid input, we have to force the data to obey certain rules. We have listed the rules for this signup form as follows: 1. Each user on the website will be uniquely identified by his email id, so we must ensure that no two users have the same email ids. 2. We must ensure that the user enters a valid e-mail address and not just any string. (e.g. user@gmail.com) 3. We must ensure that a valid 10-digit mobile number is entered by the user. 4. We must ensure that the password contains a minimum of 6 characters. Also, the password stored in the DB should be encrypted. For this, we need to use an encryption technique that you have learned during the program. 5. We must ensure that the user submits the form only after she fills in all the fields.

If a user enters valid data, she is logged in and is taken to the Home page. Using Form Validation, we check whether the user has entered the correct values. If the



user enters an email which is invalid, she is shown a message that the email entered is invalid. Similarly, if the user enters an email address that has already been registered, a message is shown telling her that this email address is already registered. If the user enters a contact number which is not a valid 10-digit mobile number, a message is shown that the contact number is invalid. Also, if the password is less than 6 characters, it should display the corresponding error message. The user may enter any value in the remaining fields.

## **FORM INJECTION**

When you allow a user to enter an input on your website, there is always the risk that a hacker may try to compromise your server or website. This can be done by entering a PHP code instead of just plain text in the input field of the form. Form injection is a basic security feature that makes it very difficult for hackers to do this. Using form injection, we identify whether a user is trying to pass a PHP code in the input field of a form. This is basic security against potential hackers.

How to show the message:

We are going to use the Javascript alert statement for showing user messages like Email Already Exists and redirect to pages using **location.href**.

Alert: This is used to display a message on your screen and it accepts a parameter which is your message. You can use it in your .php file using echo statement like:

```
Echo("<script>alert('your message here')</script>");
```

Location.href: This can be used to redirect to any page.

You can use it like:

```
echo ("script>location.href='signin.php'</script>");
```

signup.php is the page it is redirecting to.

### **2.1.3 LOGIN PAGE:**

- a) In the login form, there are two fields, one is the email address and the other is a password. When the user enters her credentials, the website checks the database for the email and password. If both are found corresponding to the same user on the database, the user is logged in.

- b) Again, we use form validation and form injection just as we did it in the signup form. If a user enters an email id that is not registered in the database, she is not allowed to login and an error message is displayed. If the email id is registered and the user enters a wrong password, an error message is shown indicating that the password entered is incorrect. If she enters the correct email address and password, she is logged in and is redirected to the Home page
- c) An additional safety feature being implemented in these forms is that the passwords entered by the users are encrypted. Therefore, even if the database is compromised by a hacker, they cannot determine the password. For this, we need to use an encryption technique that you have learned during the program. Once the user's input is validated, the data entered by her is accepted and stored in a database (MySQLi is used for this purpose). The user is redirected to the home page, where she can create the plan.

#### **2.1.4 ABOUT US PAGE:**

The navbar and the footer remain the same. In the body section, it contains two div elements one of which contains an image and other contains the content.

#### **2.1.5 PRODUCTS PAGE:**

- a) The navbar and the footer remain the same. In the body section, it contains different products with captions and prices. Every product must contain an add to cart button.
- b) The products are saved in database and are fetched from the database using PHP script.
- c) It contains script and code with which we can view the details of a single product.
- d) The product can easily be added to cart.

### **2.1.6 CART PAGE:**

- a) The navbar and the footer remain the same.
- b) This page records the details of products which are added to cart by the user.
- c) It will display the records in form of a table and it contains the grand total below the table.

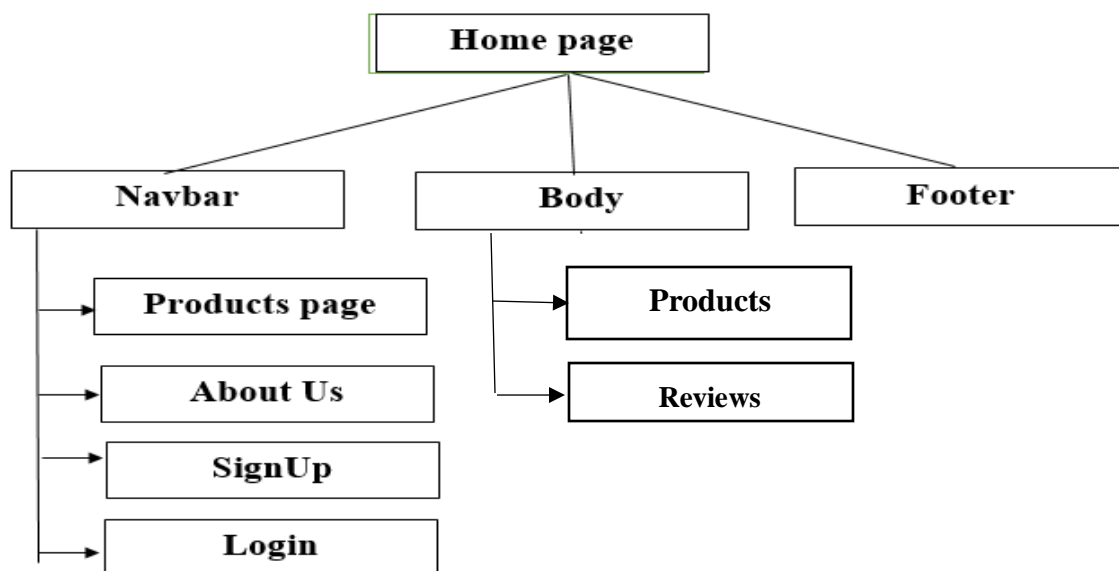
It also contains two buttons:

- One for confirming the order of product.
- Second for removing the product from the cart.

### **2.1.7 ORDERS PAGE:**

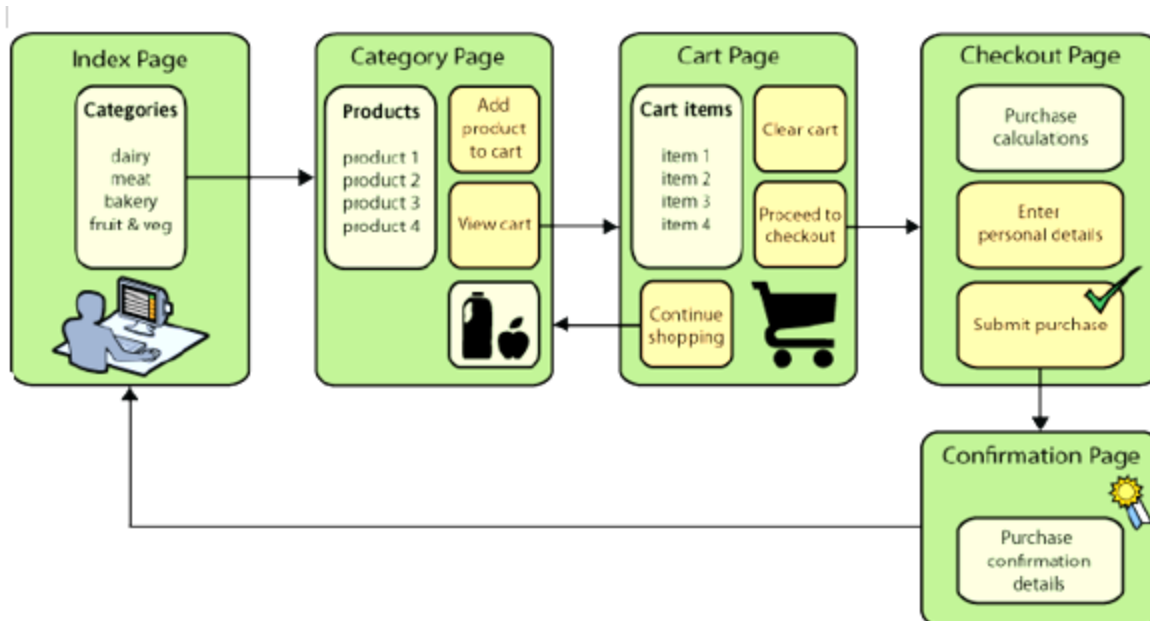
- a) The navbar and the footer remain the same.
- b) This page records the details of products which are ordered by the user.
- c) It will display the records in form of a table and it contains the grand total below the table.
- d) It contains a button for cancelling an order.

## **2.2 Structure of the Lifestyle Store is shown below:**

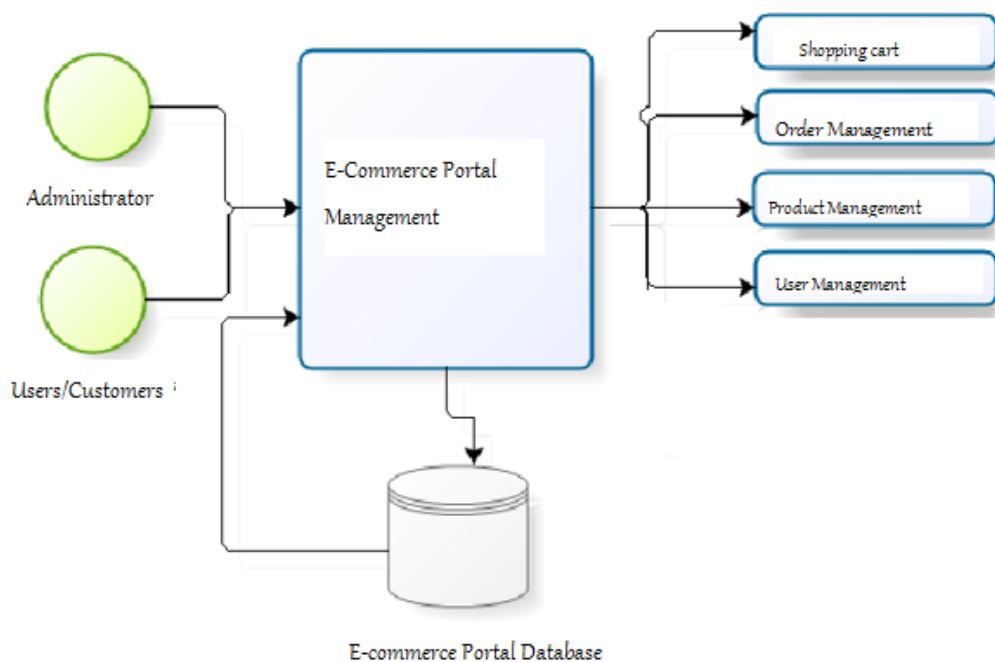


## 2.3 Data Flow Diagram

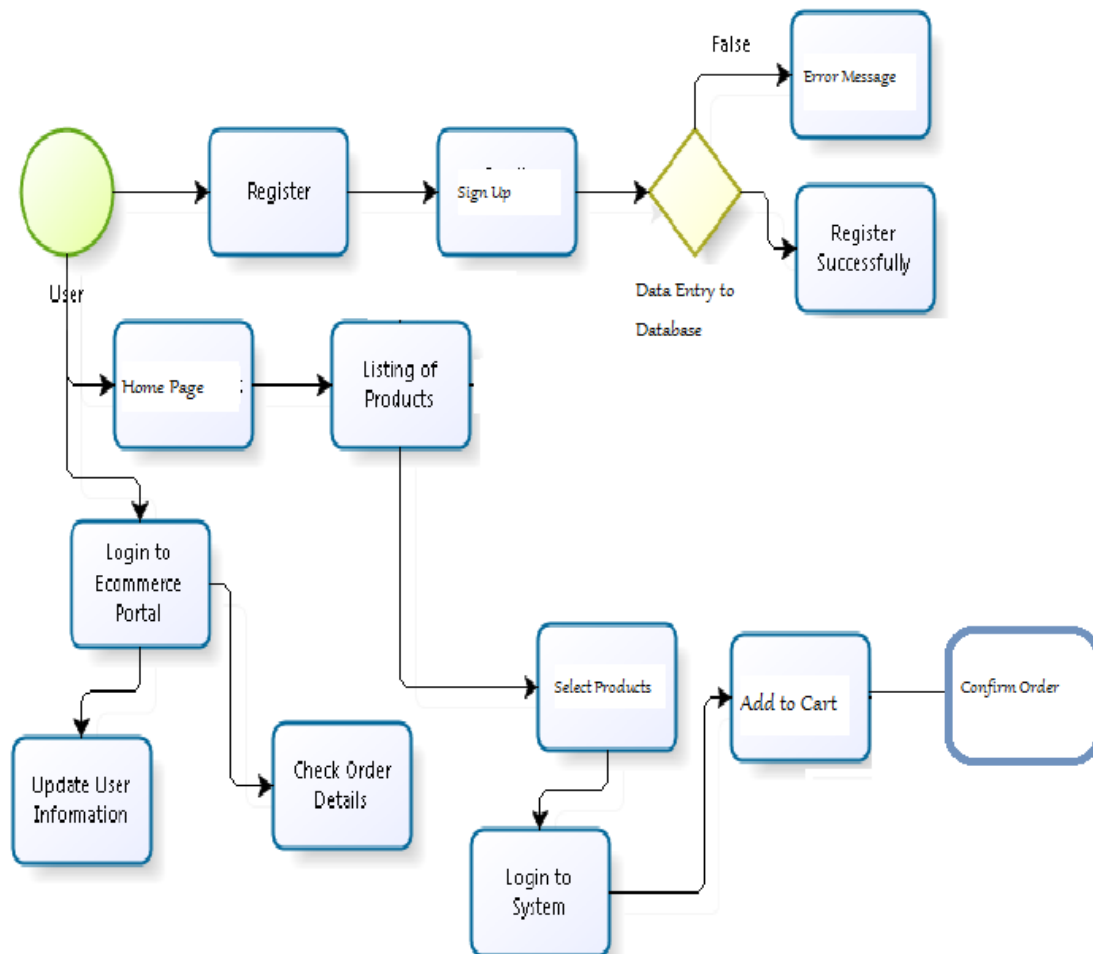
### E-commerce high level process workflow:



### 2.3.1 Data Flow Diagram Level 0:



### 2.3.2 Data Flow Diagram Level 1:



## **3. Software & Hardware Requirements & Analysis**

### **3.1 Analysis**

The purpose of this document is to build an online website is to shop online which also stores your login info and other credentials. This project is a prototype for the shopping website and it is restricted within the college premises.

#### **OPERATING ENVIRONMENT**

Web-based applications run on a **web application server** and access data on an enterprise information system, such as a DB2® database server. The application server code, which provides the run time environment, supports the appropriate interface for interacting with the web server. So, the operating environment includes:

- a) Distributed database
- b) Client/server system
- c) Operating system: Windows.
- d) Database: mysql database
- e) Platform: Java/PHP

#### **3.1.1 Technical Details**

- a) Front end is designed using HTML, CSS and Bootstrap.
- b) Backend is based on PHP + MySQLi based RDB (Relational Data Base) model.
- c) The SQL queries are run using the CI SQL library functions.
- d) Backend online host includes a centralized database resident on the server, the script which is built in PHP used to SQL query the database on user's request for transaction of data.
- e) The forms are made using the HTML, Bootstrap for designing and Php, SQLi` for back-end.

### **3.2 Software Requirements**

#### **A. User Interfaces:**

Front-end software: VsCode

Back-end software: SQL+, PHP

### **3.3 Hardware Requirements:**

#### **Hardware Interfaces:**

Windows.

A browser which supports HTML & Javascript.

#### **For VsCode:**

##### **Hardware**

Visual Studio Code is a small download (< 200 MB) and has a disk footprint of < 500 MB. VS Code is lightweight and should easily run on today's hardware.

We recommend:

1.6 GHz or faster processor

1 GB of RAM

##### **Platforms**

VS Code has been tested on the following platforms:

OS X El Capitan (10.11+)

Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10, 11 (32-bit and 64-bit)

Linux (Debian): Ubuntu Desktop 16.04, Debian 9

Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 8, Fedora 24

##### **System Requirements for XAMPP**

Operating System: Windows 7/8/8.1/10

Memory (RAM): 2 GB of RAM required.

Hard Disk Space: 250 MB of free space required for full installation.

Processor: Intel Pentium 4 Dual Core GHz or higher.

### **3.4 Language and Technologies Used**

- a. HTML
- b. CSS
- c. Bootstrap.
- d. Scripting Technologies like PHP.
- e. Database using MySQLi

### **3.4.1 Introduction to HTML**

HTML Hypertext Markup Language (HTML) 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 webserver 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 structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

HTML elements are delineated by tags, written using angle brackets. Tags such as `<img />` and `<input />` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

The following is an example of the classic Hello world program, a common test employed for comparing programming languages, scripting languages and markup languages. This example is made using 9 lines of code:

#### **GENERAL SYNTAX OF HTML:**

```
<!DOCTYPE html>
<html>
<head>
<title>This is a title</title>
</head>
<body>
<p>Hello world!</p>
</body>
</html>
```

### **3.4.2 Introduction to CSS**



CSS Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

### **TYPES OF CSS:**

- **Inline CSS:** In this CSS is applied in between the tags. E.g.:

*<tag style = " styling " > Hello World </tag>*

- **Internal CSS:** In this the css code is defined inside the style tag in the head section of the HTML page. General Syntax:

```
<html>
<head>
<style> <! -- CSS STYLING -- > </style>
</head>
</html>
```

- **External CSS:** In this the CSS code is written on another page and is linked to the HTML page. It is advantageous to use this type of styling as we can use the same file to style various HTML pages. External CSS uses the extension .css and is applied using the following syntax:

```
<html>
<head>
<link relation= " stylesheet " type= "css" href= "URL to the page">
</head>
</html>
```

All the CSS style types are important but can be used in different situations.

- **Inline CSS** is used when only small changes are to be done to the HTML tag and the changes are to be reflected only to that specific tag
- **Internal CSS** is used when the individual HTML pages have to be designed differently. This also slows the page load system if the internal styling is long.
- **External CSS** files are maintained to design multiple pages and use common styles over various pages. It is useful as it helps in managing the resources in an easy manner.

Both HTML and CSS are used to create a UI but CSS behaves like a makeup on the face of an actress which makes her look even more beautiful than she is in reality.

### **3.4.3 Introduction to BOOTSTRAP**

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

Bootstrap was created at Twitter in mid-2010 by [@mdo](#) and [@fat](#). Prior to being an open-sourced framework, Bootstrap was known as *Twitter Blueprint*. A few months into development, Twitter held its [first Hack Week](#) and the project exploded as developers of all skill levels jumped in without any external guidance. It served as the style guide for internal tools development at the company for over a year before its public release, and continues to do so today.

Originally [released](#) on Friday, August 19, 2011, we've since had over [twenty releases](#), including two major rewrites with v2 and v3. With Bootstrap 2, we added responsive functionality to the entire framework as an optional stylesheet. Building on that with Bootstrap 3, we rewrote the library once more to make it responsive by default with a mobile first approach.

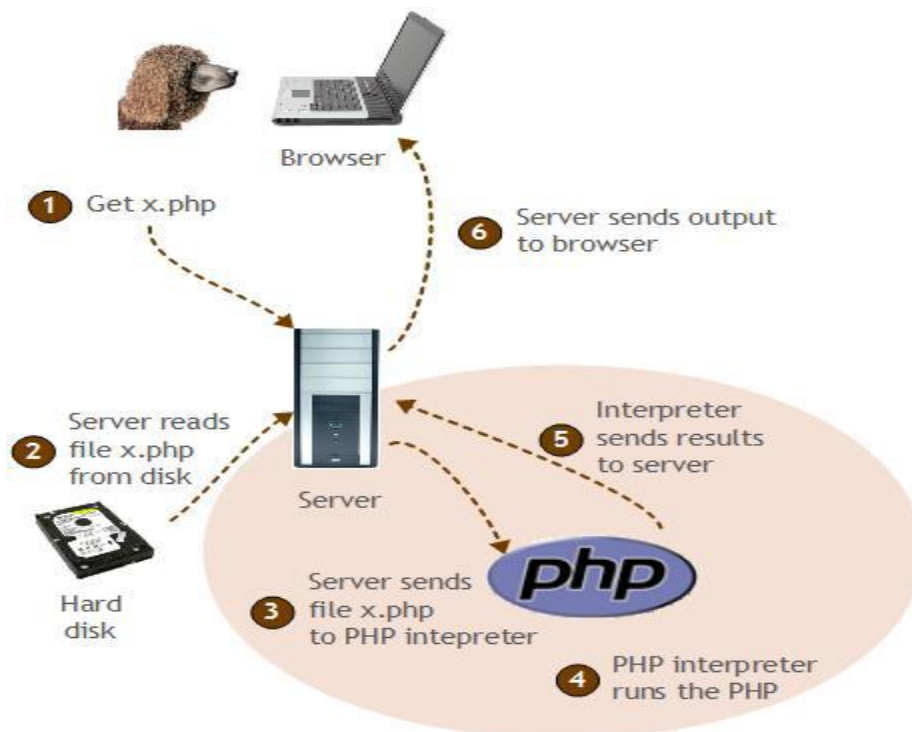
With Bootstrap 4, we once again rewrote the project to account for two key architectural changes: a migration to Sass and the move to CSS's flexbox. Our intention is to help in a small way to move the web development community forward by pushing for newer CSS properties, fewer dependencies, and new technologies across more modern browsers.

## INSTALLING AND LINKING BOOTSTRAP:

- Install bootstrap from <https://getbootstrap.com/>
- Copy the bootstrap.min.css file to your CSS folder and link it to the HTML page in the similar manner to how any other CSS file is linked.
- Link the bootstrap.min.js file which is present in the JS folder of the bootstrap. It can be linked using script tag. E.g.:  
`<script src="URL to bootstrap.min.js"></script>`
- Now use bootstrap classes to reduce the work of designing which was earlier done through CSS.

### 3.4.4 PHP

PHP is a server-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 Pre-processor.



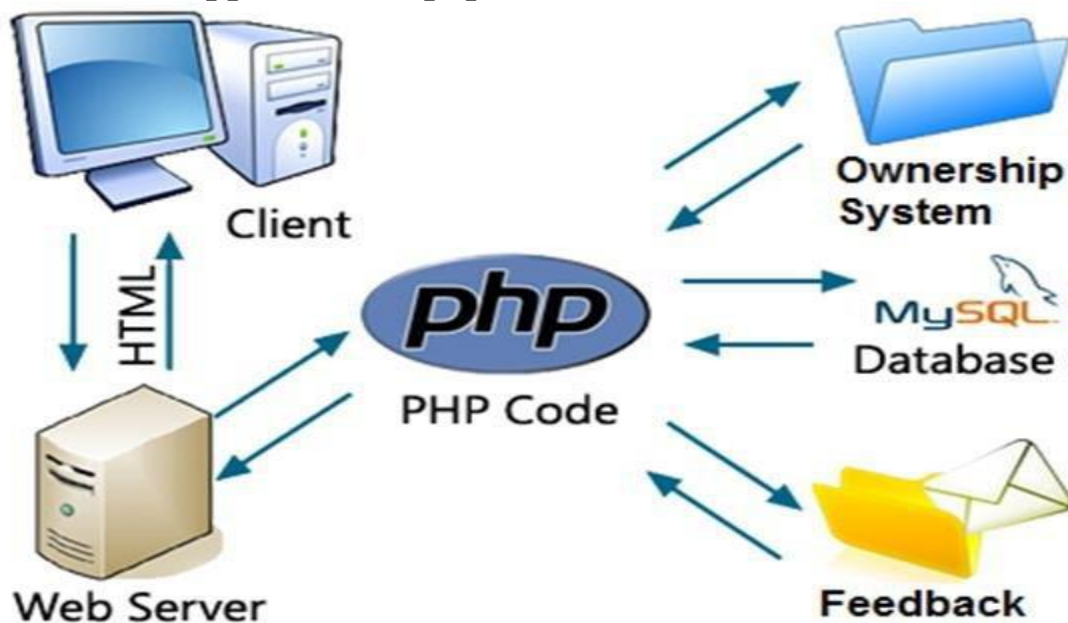
**Figure: PHP outputs HTML**

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 stand-alone graphical applications.

#### **INSTALLING PHP:**

- i. Step 1:** Download the files. Download the latest PHP 5 ZIP package from [www.php.net/downloads.php](http://www.php.net/downloads.php).
- ii. Step 2:** Extract the files.
- iii. Step 3:** Configure php.ini.
- iv. Step 4:** Add C: php to the path environment variable.
- v. Step 5:** Configure PHP as an Apache module.
- vi. Step 6:** Test a PHP file.
- vii. Step 7:** Or we can install Wamp which have inbuilt php, MySQLi, apache server

**We have used xampp to run the php files.**



**Figure: PHP**

### **3.4.5 Database**

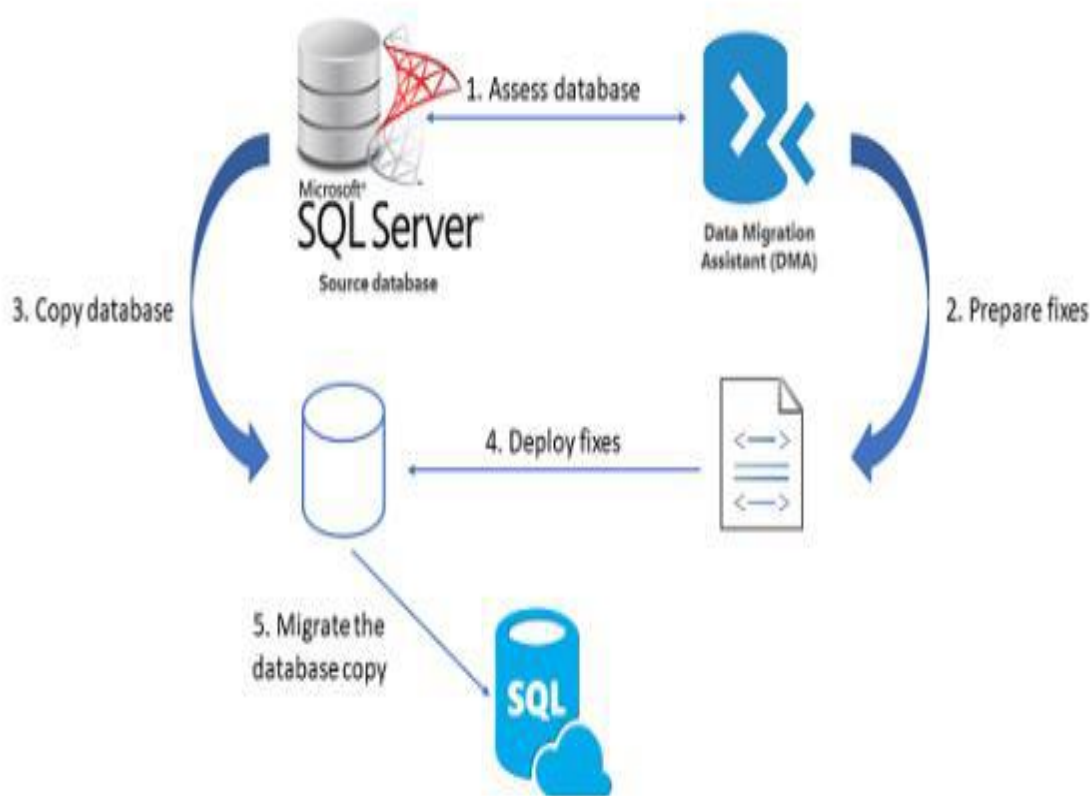
A database is an organized collection of data. It is the collection of schemas, tables, queries, reports, views, and other objects. The data are typically organized to model

aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database management system (DBMS) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyse data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MemSQL and IBM DB2.

### 3.4.6 SQL

SQL 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.



**Figure: SQL database Connection**

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." 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.

## **4.Elements Of E-Commerce Store**

### **4.1 File description**

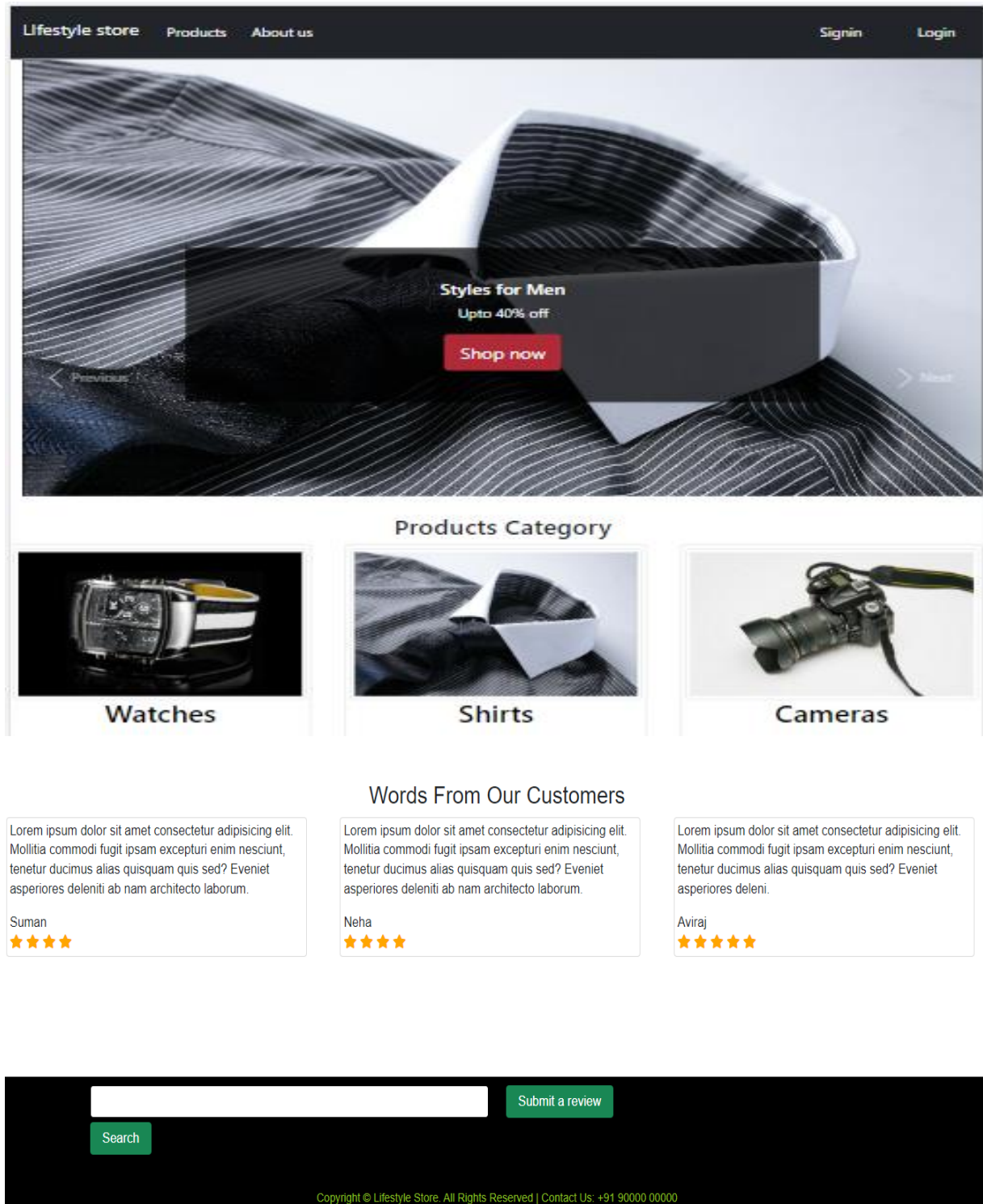
The site is created using HTML, CSS, JS and PHP.

All of the files are saved with .php extension.

- 1) Header.php file contains the code for Navbar. The navbar contains the links of products page, about us page, login page and sign up page.
- 2) Footer.php file contains the code for footer.
- 3) The index.php file contains the code for Home page.
- 4) Products.php contains the code for products page.
- 5) Aboutus.php contains the code for About us page.
- 6) login.php contains the code for login page.
- 7) signin.php contains the code for signin page.
- 8) Design.css contains the css for all pages.
- 9) Inc: It contains four files:
  - a. Auth\_check.php: It contains the code for checking the session of a user.
  - b. Session.php: It contains the code for starting the session.
  - c. Errormessage.php: It conatins error message if the signin went wrong.
  - d. Successmessage.php: It contains success message after successful signin.
- 10) Db: It contains three file:
  - a. Conn.php: It is used to establish connection with the database.
  - b. Crud.php: It has code of all SQL queries used to made Various actions.
  - c. User.php: It has code for user authentication.

## 4.2 Output:

# Home Page





A dark, atmospheric street scene at night. The image is dominated by deep blues and blacks, with some light reflecting off wet surfaces and building facades. In the center, there is a semi-transparent white rectangular box containing three lines of placeholder text in a serif font. To the left, a street sign with the word 'ON' and a right-pointing arrow is visible. Below it, another sign reads '123 MAGICAL STREET' and 'WINTERBERRY BAKERY'. On the right, a building with a sign that says 'HOL' is partially visible. The overall mood is mysterious and urban.

# Signin Page

Lifestyle storeProductsAbout us

SigninLogin

Signin

Enter details to create account

Name

Name

Email address

Email

Contact

Contact

Password

Password

Submit

Copyright © Lifestyle Store. All Rights Reserved | Contact Us: +91 90000 00000

# Login Page

Lifestyle store Products About us

Signin Login

Login

Login to make a purchase

Email address

Email

Password

Password

Login

Don't have an account? Register

Copyright © Lifestyle Store. All Rights Reserved | Contact Us: +91 90000 00000

# Products Page

Lifestyle store Products About us

Signin Login

## Welcome to our Lifestyle Store!

We have the best cameras, watches and shirts for you. No need to hunt around, we have all in one place.



Item:Colors-Women-Embellished-Saree

Price:1200

Details:Beautiful pattern and variety of colors

[View Details](#)



Item:Jewelry-Color-diamond-Love-font

Price:2000

Details:Beautiful pattern and variety of colors

[View Details](#)



Item:Jewelry-full metal earring

Price:2000

Details:Beautiful pattern and variety of colors

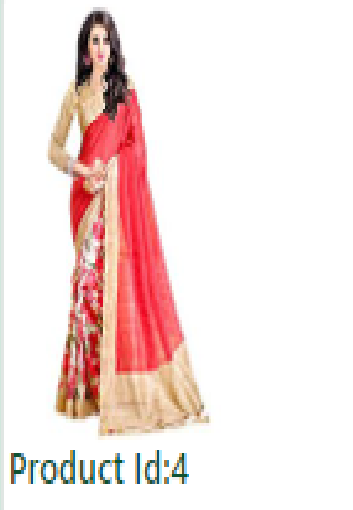
[View Details](#)

# Add to Cart

[Lifestyle store](#) [Products](#) [About us](#) [Cart](#) [Orders](#) [Logout](#) [Hello neha@gmail.com!](#)

X

ITEM ADDED



Product Id:4

Item:Colors-Women-Embellished-Saree

Price:1200




Details:Beautiful pattern and variety of colors

Quantity :1


Back

Copyright © Lifestyle Store. All Rights Reserved | Contact Us: +91 90000 00000

# View Cart Page

Lifestyle store				Products	About us	Cart	Orders	Logout	Hello neha@gmail.com!
#	Product Name	Quantity	Price	Image	Actions				
28	Steel-Bracelet 2	1	399		<div>Confirm order</div> <div>Delete</div>				
11	fancy-look-attractive-saree-2-original	1	1200		<div>Confirm order</div> <div>Delete</div>				
4	Colors-Women-Embellished-Saree	1	1200		<div>Confirm order</div> <div>Delete</div>				
			Grand Total	2799					
Copyright © Lifestyle Store. All Rights Reserved   Contact Us: +91 90000 00000									

# View Order Page

Lifestyle store				Products	About us	Cart	Orders	Logout	Hello neha@gmail.com!
#	Product Name	Quantity	Price	Image	Actions				
28	Steel-Bracelet 2	1	399		<a href="#">Cancel Order</a>				
Grand Total			399						
Copyright © Lifestyle Store. All Rights Reserved   Contact Us: +91 90000 00000									

# Review Page

Lifestyle store Products About us Cart Orders Logout Hello neha@gmail.com!

## WE APPRECIATE YOUR REVIEW!

Your review will help us to improve our web hosting quality products, and customer services.

### Your personal info

Name

Email

Rate our overall services.

Write your feedback.

Submit



# Search Results Page

Lifestyle store   Products   About us   Signin   Login

Here are some matching results



Item:Jewelry-Color-diamond-Love-font

Price:2000

Details:Beautiful pattern and variety of colors

[View Details](#)



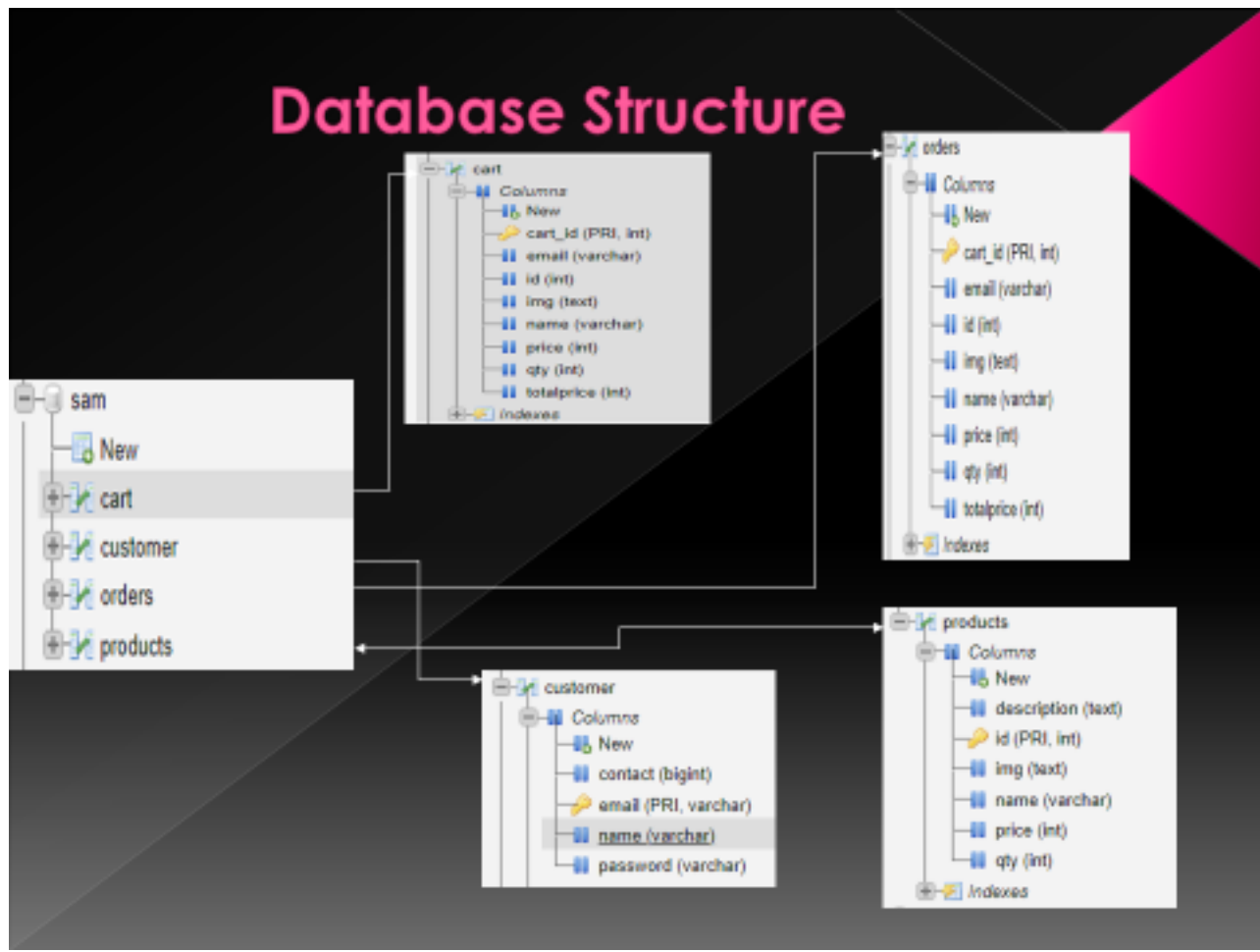
Item:Diamond-Setting-Watch-3

Price:700

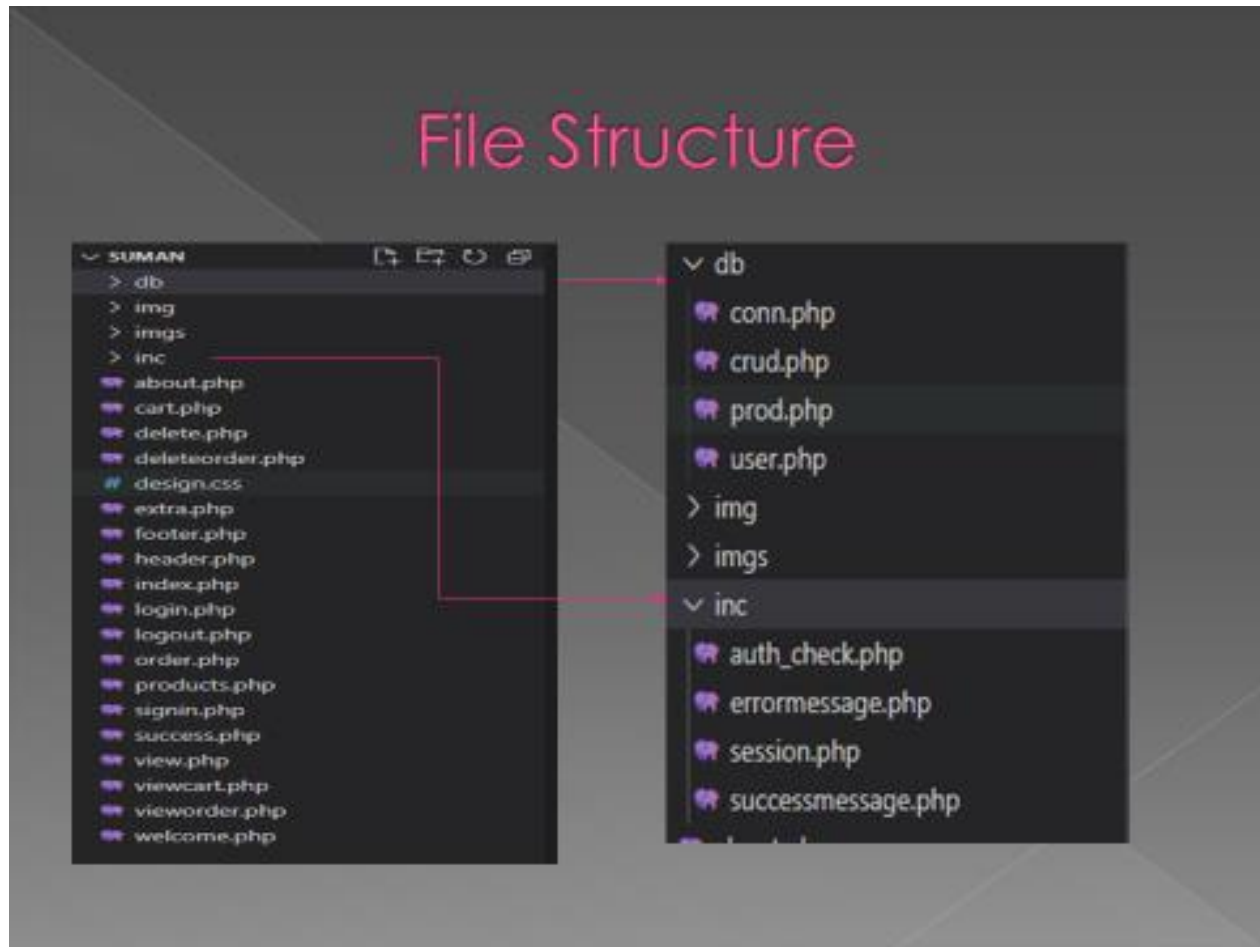
Details:2016-New-Electronic-Style-Women-Dress-  
New-Fashion-Watches-Imitation-Diamond-Setting-  
Watch-Casual-Wrist-Watch-3

[View Details](#)

# Database



# File Structure



## **5. Conclusion**

- a) The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur's but also from the customer's point of view.
- b) For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible.
- c) As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. "Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site. Hence, we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.
- d) In this project, the user is provided with an ecommerce web site that can be used to buy items online.
- e) To implement this as a web application we used PHP as the Technology. PHP has several advantages such as enhanced performance, scalability, built-in security and simplicity. A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a number of features that are designed to make the customer more comfortable.
- f) This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The building of the project has given me a precise knowledge about how PHP is used to develop a website, how it connects to the database to access the data and how the data and web pages are modified to provide the user with a shopping cart application.

### **Outcomes:**

- 1. Able to use the HTML programming language.
- 2. Resolves written HTML codes.
  - a. Runs the page designed using HTML codes
  - b. Be able to use CSS for styling the web pages

3. Able to modify Site Designs to look how we Want Them to Look.
4. Able to use Simple and impressive design techniques, from basics till advanced to focus on goal oriented and user centric designs.
5. Able to plan for website & actually build excellent web sites
6. Able to create web elements like buttons, banners & Bars and of course complete UI designs.
7. Publishes the site designed
8. Be able to make changes on the Site.
9. Setting up page layout, color schemes, contract, typography in the designs
10. Able to create Forms and validations for our website.

## **6. REFERENCE**

- 1) <http://www.w3schools.com/>
- 2) <https://www.javatutpoint.com/>
- 3) <https://www.getbootstrap.com/>
- 4) <https://www.stackoverflow.com/>
- 5) Wikipedia

## 7.CODE OF PROJECT

### Conn.php

```
<?php
    // Development Connection
    $host = '127.0.0.1:3308';
    $db = 'sam';
    $user = 'root';
    $pass = '';
    $charset = 'utf8mb4';
    $dsn = "mysql:host=$host;dbname=$db;charset=$charset";
    try{
        $pdo = new PDO($dsn, $user, $pass);
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    } catch(PDOException $e) {
        throw new PDOException($e->getMessage());
    }
    require_once 'crud.php';
    require_once 'user.php';
    $crud = new crud($pdo);
    $user = new user($pdo);
    ?>
```

### Crud.php

```
<?php
class crud{
    // private database object\
    private $db;

    //constructor to initialize private variable to the database connection
    function __construct($conn){
        $this->db = $conn;
    }

    public function insertcustomer($name, $email,$contact,$password){
        try {
            // define sql statement to be executed
            $sql = "INSERT INTO customer (name,email,contact,password) VALUES
(:Name,:Email,:Contact,:Password)";
            //prepare the sql statement for execution
            $stmt = $this->db->prepare($sql);
            // bind all placeholders to the actual values
            $stmt->bindParam(':Name',$name);
            $stmt->bindParam(':Email',$email);
            $stmt->bindParam(':Contact',$contact);
```

```

        $stmt->bindParam(':Password',$password);
        // execute statement
        $stmt->execute();
        return true;
    }
    catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

public function insertproduct($id,$name, $price, $description, $img,$email){
    try{
        $qty=1;
        $sql= "INSERT INTO cart (email,name, price , img, qty, totalprice, id) VALUES (:email,:name,
:price, :img, :qty, :price, :id)";
        $insert_stmt= $this->db->prepare($sql);
        $insert_stmt->bindParam(":email", $email);
        $insert_stmt->bindParam(":name", $name);
        $insert_stmt->bindParam(":price", $price);
        $insert_stmt->bindParam(":img", $img);
        $insert_stmt->bindParam(":qty", $qty);
        $insert_stmt->bindParam(":totalprice", $totalprice);
        $insert_stmt->bindParam(":id", $id);
        $insert_stmt->execute();
        return true;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

public function getAllproducts(){
    try{
        $sql = "SELECT id,name, price, description,img FROM `products` ";
        $result = $this->db->query($sql);
        return $result;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

public function getAllcart($email){
    try{
        $sql = "select * from cart where email= :email";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':email', $email);

```



```

        $stmt->execute();
        $result = $stmt->fetchall(PDO::FETCH_ASSOC);
        return $result;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

    public function getAllorder($email){
    try{
        $sql = "select * from orders where email= :email";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':email', $email);
        $stmt->execute();
        $result = $stmt->fetchall(PDO::FETCH_ASSOC);
        return $result;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

public function getProductDetails($id){
    try{
        $sql = "select * from products where id = :id";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':id', $id);
        $stmt->execute();
        $result = $stmt->fetch();
        return $result;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

    public function getSum($email){
    try{
        $sql = "select SUM(price) AS sum from cart where email = :email";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':email', $email);
        $stmt->execute();
        $result = $stmt->fetch();
        $sum= $result['sum'];
        return $sum;
    }catch (PDOException $e) {
        echo $e->getMessage();
    }
}

```

```

        return false;
    }
}

public function getSum1($email){
    try{
        $sql = "select SUM(price) AS sum from orders where email = :email";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':email', $email);
        $stmt->execute();
        $result = $stmt->fetch();
        $sum= $result['sum'];
        return $sum;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

public function deleteproduct($id){
    try{
        $sql = "delete from cart where id = :id";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':id', $id);
        $stmt->execute();
        return true;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

public function deleteorder($id){
    try{
        $sql = "delete from orders where id = :id";
        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':id', $id);
        $stmt->execute();
        return true;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}

public function orderproduct($id){
    try{
        $sql = "insert into orders select * from cart where id = :id";

```

```

        $stmt = $this->db->prepare($sql);
        $stmt->bindParam(':id', $id);
        $stmt->execute();
        return true;
    }catch (PDOException $e) {
        echo $e->getMessage();
        return false;
    }
}
}
?>

```

## User.php

```

<?php
class user{
    // private database object\
    private $db;

    //constructor to initialize private variable to the database connection
    function __construct($conn){
        $this->db = $conn;
    }

    public function getUser($email,$password){
        try{
            $sql = "select email,password  from customer where email = :Email AND password = :Password";

            $stmt = $this->db->prepare($sql);
            $stmt->bindParam(':Email', $email);
            $stmt->bindParam(':Password', $password);
            $stmt->execute();
            $result = $stmt->fetch();
            return $result;
        }catch (PDOException $e) {
            echo $e->getMessage();
            return false;
        }
    }
}
?>

```

## Auth\_check.php:

```

<?php
if(!isset($_SESSION['Email'])){
    header("Location: login.php");
}

```

```
?>
```

## Errormessage.php

```
<div class="alert alert-danger" role="alert">
    Either the Email_Id or contact number is already registered. Please try with a new one
</div>
```

## Session.php

```
<?php
    session_start();
?>
```

## Successmessage.php

```
<div class="alert alert-success" role="alert" text-align="center">
    You are registered successfully!!
    Your details are as shown.
</div>
```

## About.php

```
<?php
    include 'header.php';
?>

<div id="background_image">
    <div class="container">
        <div id="banner_content">

            <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aliquid, ea consequatur. Cum
            incidunt excepturi, debitis tenetur doloribus dolorum vero aperiam, beatae consectetur deleniti provident
            unde laudantium maxime saepe culpa ratione.

            </p>
        </div>
    </div>
</div>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia commodi fugit ipsam excepturi
    enim nesciunt, tenetur ducimus alias quisquam quis sed? Eveniet asperiores deleniti ab nam architecto
    laborum quis reprehenderit quo, praesentium dicta quas.

    Lorem ipsum dolor sit amet consectetur adipisicing elit. Mollitia commodi fugit ipsam excepturi enim
    nesciunt, tenetur ducimus alias quisquam quis sed? Eveniet asperiores deleniti ab nam architecto laborum
    quis reprehenderit quo, praesentium dicta quas.
<p>

    <?php
    include 'footer.php';
?>
```

## Header.php

```
<?php include 'inc/session.php';?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOmLASjC"
        crossorigin="anonymous">
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
        type="text/javascript"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-MrcW6ZMFY1zcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
        crossorigin="anonymous">
    </script>

    <link rel="stylesheet" href="design.css" type="text/css">
</head>

<body>
    <div class="navbar navbar-fixed-top navbar-dark bg-dark">
        <div class="container-fluid">
            <ul class="nav">
                <li class="nav-item">
                    <a href="index.php" class="navbar-brand">Lifestyle store</a></li>
                <li class="nav-item"> <a class="btn btn-dark" href="products.php">Products </a></li>
                <li class="nav-item"> <a class="btn btn-dark" href="about.php">About us </a></li>
            </ul>

            <ul class="nav justify-content-end">

                <?php
if (isset($_SESSION['Email'])) {
    ?>

    <li class="nav-link active"><a class="btn btn-dark" href="viewcart.php"> Cart </a></li>
    <li><br></li>
    <li class="nav-link active"><a class="btn btn-dark" href="vieworder.php"> Orders </a></li>
    <li><br></li>
    <li class="nav-link active" style="float: right"> <a href="logout.php"><span class="btn
btn-dark">
        Logout</a></li>
            </ul>
        </div>
    </div>
</body>
</html>
```

```

        <span class="navbar-text">Hello
            <?php echo $_SESSION['Email'] ?>!
        </span>
        <?php
    } else { ?> <li class="nav-link active"> <a class="btn btn-dark" href="signin.php">Signin</a></li>
        <li class="nav-link active"> <a class="btn btn-dark" href="login.php">Login </a></li>
        <?php } ?>
    </ul>

</div>
</div>

```

## Footer.php

```

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DktIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-
JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1" crossorigin="anonymous"></script>
<footer >
<div class="container">
    <p> <center>Copyright © Lifestyle Store. All Rights Reserved | Contact Us: +91 90000 00000
        </center></p>
</div>
</footer>
</body>

```

## Index.php

```

<?php require_once ( "db/conn.php" ); ?>

<?php
    include 'header.php';
    ?>

    <div class="container-fluid">
<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
    <div class="carousel-inner">
        <div class="carousel-item active">
            
            <div class="carousel-caption d-none d-md-block">
                <div id="banner_content">
                    <h5>WE sell lifestyle</h5>
                    <p>Buy Now! Huge Selection</p>
                    <a class="btn btn-danger btn-lg active" href="products.php">Shop now</a>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
    <div class="carousel-item">
        
        <div class="carousel-caption d-none d-md-block">
            <div id="banner_content">
                <h5> Styles for Men</h5>
                <p>Upto 40% off</p>
                <a class="btn btn-danger btn-lg active" href="products.php">Shop now</a>
            </div>
        </div>
    </div>
    <div class="carousel-item">
        
        <div class="carousel-caption d-none d-md-block">
            <div id="banner_content">
                <h5>Shop by Category</h5>
                <p>Explore a vast selection of products</p>
                <a class="btn btn-danger btn-lg active" href="products.php">Shop now</a>
            </div>
        </div>
    </div>
</div>
<div>
    <a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="sr-only">Previous</span>
    </a>
    <a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="sr-only">Next</span>
    </a>
</div>

<br>
<div>
    <h3 style="text-align: center;">Products Category</h3>
</div>
<div class="row">
    <div class="items">
        <a href="products.php">
            
            <div class="caption">
                <h2>Watches</h2>
                <p>Original watches from the best brands</p>
            </div>
        </a>
    </div>
</div>

```

```

        </div>
    </a>
</div>
<div class="items">
    <a href="products.php">
        
        <div class="caption">
            <h2>Shirts</h2>
            <p>Our exquisite collection of shirts</p>
        </div>
    </a>
</div>
<div class="items">
    <a href="products.php">
        
        <div class="caption">
            <h2>Cameras</h2>
            <p>Choose among the best available in the world</p>
        </div>
    </a>
</div>
</div>
<div class="row">
    <div class="items">
        <a href="products.php">
            
            <div class="caption">
                <h2>Footwear</h2>
                <p>Stylish footwear from the best brands</p>
            </div>
        </a>
    </div>

    <div class="items">
        <a href="products.php">
            
            <div class="caption">
                <h2>Jeans</h2>
                <p>Our exquisite collection of shirts</p>
            </div>
        </a>
    </div>

```



```

        </a>
    </div>

    <div class="items">
        <a href="products.php">
            
            <div class="caption">
                <h2>Jewelry</h2>
                <p>Wide range of best quality bags</p>
            </div>
        </a>
    </div>
</div>
</div>
<?php
    include 'footer.php';
?>
</html>

```

## Products.php

```

<?php require_once 'header.php';
require_once 'db/conn.php';
//get all products
$result= $crud->getAllproducts();
?>
<div class="container">
    <div class="jumbotron">
        <h1 class="display-4"> <b>Welcome to our Lifestyle Store!</b></h1>
        <p class="my-4">We have the best cameras, watches and shirts for you. No need to hunt
            around, we have all in one place.</p>
    </div>
</div>

    <div class="row">

<?php
    while ($r= $result->fetch(PDO::FETCH_ASSOC)){?>
        <div class="items">
            
            <div class="caption">
                <h5>Item:<?php echo $r['name'] ?></h5>
                <p>Price:<?php echo $r['price'] ?></p>
                <p>Details:<?php echo $r['description'] ?></p>
            </div>
        </div>
    </div>
}

```

```

                <a class="btn btn-primary btn-block" href="view.php?id=<?php echo $r['id'] ?>">
View Details </a>
            </div>
        </div>

        <?php
        }
        ?>

    </div>
    <?php
include 'footer.php';
?>
</html>

```

### Delete.php

```

<?php
require_once 'header.php';
    require_once 'inc/auth_check.php';
    require_once 'db/conn.php';
    if(!isset($_GET['id'])){
        include 'inc/errormessage.php';
        header("Location: viewcart.php");
    }else{
        // Get ID values
        $id = $_GET['id'];
        //Call Delete function
        $result = $crud->deleteproduct($id);
        //Redirect to list
        if($result)
        {
            header("Location: viewcart.php");
        }
        else{
            include 'inc/errormessage.php';
        }
    }
?>

```

### Deleteorders.php

```

<?php
require_once 'header.php';
    require_once 'inc/auth_check.php';
    require_once 'db/conn.php';
    if(!isset($_GET['id'])){
        include 'inc/errormessage.php';
        header("Location: vieworder.php");
    }

```

```

    }else{
        // Get ID values
        $id = $_GET['id'];
        //Call Delete function
        $result = $crud->deleteorder($id);
        //Redirect to list
        if($result)
        {
            header("Location: vieworder.php");
        }
        else{
            include 'inc/errormessage.php';
        }
    }
}
?>

```

## Login.php

```

<?php
require_once 'header.php';
require_once 'db/conn.php';
require_once 'db/user.php';
//If data was submitted via a form POST request, then...
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $email = $_POST['Email'];
    $password = $_POST['Password'];
    $result = $user->getUser($email,$password);
    if(!$result){
        echo '<div class="alert alert-danger">Username or Password is incorrect! Please try again.
</div>';
    }else{
        $_SESSION['Email'] = $email;
        header("Location: index.php");
    }
}

?>
<div class="col-md-6">
    <div class="panel panel-primary">
        <div class="panel heading">
            <h4>Login</h4>
        </div>
        <div class="panel-body">
            <p class="text-warning">Login to make a purchase</p>
        </div>
        <form action="<?php echo htmlentities($_SERVER['PHP_SELF']); ?>" method="post">
            <div class="form-group">

```

```

        <label for="exampleInputEmail1">Email address</label>
        <input type="text" class="form-control" name="Email" placeholder="Email"
id="Email"
        value="<?php if($_SERVER['REQUEST_METHOD'] == 'post') echo
$_POST['email']; ?>">
    </div>
    <div class="form-group">
        <label for="exampleInputPassword1">Password</label>
        <input type="password" class="form-control" name="Password"
placeholder="Password"
        id="Password">
    </div>
    <input type="submit" value="Login" class="btn btn-primary btn-block">
</form>
<br>
<div class="panel-footer">Don't have an account?<a class="link"
href="signin.php">Register </a>
    </div>
</div>
</div>

<?php
include 'footer.php';
?>
</html>

```

## Logout.php

```

<?php
include_once 'inc/session.php';
?>
<?php
session_destroy();
header('location: index.php');
?>

```

## Order.php

```

<?php
require_once 'header.php';
    require_once 'inc/auth_check.php';
    require_once 'db/conn.php';
    if(!isset($_GET['id'])){
        include 'inc/errormessage.php';
        header("Location: viewcart.php");
    }else{
        // Get ID values
        $id = $_GET['id'];
    }

```

```

        //Call Delete function
        $result = $crud->orderproduct($id);
        //Redirect to list
        if($result)
        {
            header("Location: vieworder.php");
        }
        else{
            header("Location: viewcart.php");
        }
    }
}
?>

```

## Signin.php

```

<?php
    require_once 'header.php';
    require_once 'db/conn.php'

    ?>
    <div class="col-md-6">
        <div class="panel panel-primary">
            <div class="panel heading">
                <h4>Signin</h4>
            </div>
            <div class="panel-body">
                <p class="text-warning">Enter details to create account</p>
            </div>

            <form action="success.php" method="post" enctype="multipart/form-data">
                <div class="form-group">
                    <label for="exampleInputname">Name</label>
                    <input type="text" class="form-control" id="Name" name="Name" required
placeholder="Name">
                </div>
                <div class="form-group">
                    <label for="exampleInputEmail1">Email address</label>
                    <input type="email" class="form-control" id="Email" name="Email" required
placeholder="Email">
                </div>
                <div class="form-group">
                    <label for="exampleInputcontact">Contact</label>
                    <input type="text" class="form-control" id="Contact" name="Contact" required
type="text"
                    maxlength="10" placeholder="Contact">
                </div>
            </form>
        </div>
    </div>

```

```

        <div class="form-group">
            <label for="exampleInputPassword1">Password</label>
            <input type="password" class="form-control" id="Password" name="Password" required
minlength="6"
                placeholder="Password">
        </div>

        <button name="submit" class="btn btn-primary">Submit </button>

    </form>

</div>

</div>
<?php
    include 'footer.php';
    ?>
</html>

```

## Success.php

```

<?php
require_once 'header.php';
require_once 'db/conn.php';
?>
<?php
if(isset($_POST['submit'])){
    //extract values from the $_POST array
    $name = $_POST['Name'];
    $email = $_POST['Email'];
    $contact = $_POST['Contact'];
    $password = $_POST['Password'];
    //Call function to insert and track if success or not
    $isSuccess = $crud->insertcustomer($name, $email, $contact,$password);
    if($isSuccess){
        //    SendEmail::SendMail($email, 'Welcome to Lifestyle store', 'You have successfully registerted');
        require_once 'inc/successmessage.php';
    }
    else{
        require_once 'inc/errormessage.php';
    }
}

?>

<div class="card" style="width: 18rem;">
    <div class="card-body">

```

```

        <h5 class="card-title">
            Name: <?php echo $_POST['Name'] ; ?>
        </h5>
        <h6 class="card-title">
            <?php echo $_POST['Email']; ?>
        </h6>
        <p class="card-text">
            Contact Number: <?php echo $_POST['Contact']; ?>
        </p>

        <p class="card-text">
            Password: <?php echo $_POST['Password']; ?>
        </p>

    </div>
</div>
<?php
include 'footer.php';
?>
</html>

```

## View.php

```

<?php
    include 'header.php';
    // require_once 'inc/auth_check.php';
    require_once 'db/conn.php';

    if(!isset($_GET['id'])){
        include 'inc/errormessage.php';
    } else{
        $id = $_GET['id'];
        $result = $crud->getProductDetails($id);
        $id=$result['id'];
        $name=$result['name'];
        $price=$result['price'];
        $description=$result['description'];
        $img=$result['img'];
        $qty=$result['qty']; } ?>
<?php
echo '
<div class="row">
    <div class="items">
        <div> 
        <h3>Product Id: '.$id.'</h3>
    </div>

```

```

</div>
<div class="column">
<h3>Item: '.$name.'</h3>
<p>Price: '.$price.'</p>
<p>Details: '.$description.'</p>
    <p>Quantity available: '.$qty.'</p>
    <!-- <a class="btn btn-primary btn-block" href="cart.php"> Add to cart </a> -->
    <form method="post" action="cart.php" enctype="multipart/form-data">
    <div class="form-group">
    <input type="hidden" id="id" name="id" value="'.$id.'" >
    <input type="hidden" id="name" name="name" value="'.$name.'" >
    <input type="hidden" id="price" name="price" value="'.$price.'" >
    <input type="hidden" id="description" name="description" value="'.$description.'" >
    <input type="hidden" id="img" name="img" value="imgs/'.$img.'" >
    </div>

    <button name="submit" class="btn btn-primary">Add to cart </button>
    </form>
</div>
</div>'
?>
<?php
    include_once 'footer.php';
    ?>
</html>

```

## Viewcart.php

```

<?php
    require_once 'header.php';
    require_once 'db/conn.php';
    require_once 'inc/auth_check.php';
    $email=$_SESSION['Email'];
    $result = $crud->getAllcart($email);
    $sum= $crud->getSum($email);

    ?>
    <table class="table">
        <tr>
            <th>#</th>
            <th>Product Name</th>
            <th>Quantity</th>
            <th>Price</th>
            <th>Image</th>
            <th>Actions</th>

```





```

$email=$_SESSION['Email'];
$result = $crud->getAllorder($email);
$sum= $crud->getSum1($email);

?>
<table class="table">
    <tr>
        <th>#</th>
        <th>Product Name</th>
        <th>Quantity</th>
        <th>Price</th>
        <th>Image</th>
        <th>Actions</th>
    </tr>
    <?php
    if(count($result)){
        foreach($result as $row){
            echo '
            <tr>
            <td> '.$row['id'].' </td>
            <td> '.$row['name'].' </td>
            <td> '.$row['qty'].' </td>
            <td> '.$row['price'].'</td>
            <td>  </td>
            <td>
            <a href="deleteorder.php?id='.$row['id'].'" class="btn btn-danger">Cancel Order</a>
            </td>
            </tr>

            <tr>
            ';
        }}
    ?>

    <td> </td>
    <td> </td>
    <td>Grand Total</td>
    <td> <?php echo $sum ?></td>

    </tr>
</table>
<?php require_once 'footer.php'; ?>

```