Online Second Hand Book Buying & Selling Portal Software Project

Project report submitted
in partial fulfilment of the requirement for the award of the degree of

Bachelor of Technology in
Computer Science and Engineering

By

VENKATESH E	**(18UECS0922)**
SANTHA KUMAR D	**(18UECS0771)**
GURRAM RAJASHEKAAR	**(18UECS0301)**
RETURI SHYAM BHANODAY	**(18UECS0703)**

Under the guidance of
Mr. C.M.Chidambaranathan, B.E, M.Tech, (Ph.D)
ASST. PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING

VEL TECH RANGARAJAN Dr.SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
CHENNAI 600 062, TAMILNADU, INDIA
May 2021

CERTIFICATE

It is certified that the work contained in the project report titled “Online Secondhand Book Buying & Selling Portal Software Project” by “VENKATESH E(18UECS0922), SANTHA KUMAR D(18UECS0771),GURRAM RAJASHEKAAR(18UECS0301),RETURI SHYAM BHANODAY (18UECS0703)”has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Mr. C.M.Chidambaranathan
Assistant Profesor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science and Technology
May, 2021

Signature of Head of the Department
Dr. V. Srinivasa Rao Professor & Head
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science and Technology
May, 2021

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(VENKATESH E)

Date: / /

(SANTHA KUMAR D)

Date: / /

(GURRA

M RAJASHEKAAR)

Date: / /

(RETURI SHYAM

BHANODAY)

Date: / /

APPROVAL SHEET

This project report entitled Online Second Hand Book Buying & Selling Portal Software Project by VENKATESH E (18UECS0922), SANTHA KUMAR D(18UECS0771),GURRAM RAJASHEKAAR(18UECS0301),RETURI SHYAM BHANODAY (18UECS0703) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Mr. C.M.Chidambaranathan
B.E,M.Tech,(Ph.D)

Date: / /

Place: Chennai

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO).DSc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice-Chancellor Prof. V. S. S. Kumar, Ph.D.,** for providing us with an environment to complete our project successfully.

We are obligated to our beloved **Registrar Prof. Dr. E. KANNAN,** for providing immense support in all our endeavours.

We are thankful to our esteemed **Director Academics Dr. ANNE KOTESWARA RAO, Ph.D., for** providing a wonderful academic environment to complete our project successfully.

We record indebtedness to our **Dean & Head, Department of Computer Science and Engineering Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout this project.

We also take this opportunity to express a deep sense of gratitude to Our Internal Supervisor **Ms. Minu Inba Shanthini Watson Benjamin, M.E.,** for her cordial support, valuable information and guidance, he helped us in completing this project through various stages.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

VENKATESH E	(18UECS0922)
SANTHA KUMAR D	(18UECS0771)
GURRAM RAJASHEKAAR	(18UECS0818)
RETURI SHYAM BHANODAY	(18UECS0703)

ABSTRACT

Students have many course-based literature textbooks that they have stopped reading and these books are laying on their bookshelves unused. These books can be sold and the proceeds can be used to buy another book the student currently needs. This website will not only serve students but can also serve the entire population or anybody who wants to buy second-hand books or wants to empty their bookshelves and make financial gain from it.

These books can just be simply be uploaded to the website and another student or person can have access to buy the book by just visiting the website. Students also have lesser incomes compared to the rest of the working population. Hence any reasonable resource for getting income is always welcomed.

All these will be achieved by creating a user-friendly complete online shopping system for buying and selling second-hand books. Customers can access the bookstore website through the World Wide Web. Customers will be able to search the database to find the books they want, check the availability, and place the order to buy the book using their credit cards.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

The “Online Second Hand Book Buying & Selling Portal Software Project” helps the student to make use of their old books and materials which in return they gain knowledge of new books and materials in this online platform. This portal helps the student from various place to connect through our portal to post their used books to share among the peers and could connect to sell their materials also can buy materials from other peers. Our projects allow users and admin the same privileges except for the security purpose other than that all are given access to control to sell and buy because we have made it into a user-friendly platform.

1.2 Aim of the project

The aim of the project – “Online Second Hand Book Buying & Selling Portal Software Project” is to create a user-friendly platform to sell their used books and to buy the books in return. To gain knowledge during pandemic times as stayed from their home itself. To achieve many peers to the platform so that they may have many contents of knowledge sharing.

1.3 Project Domain

“Online Second Hand Book Buying & Selling Portal Software Project” is the web service in which its a buying and selling portal. The web service has been hosted in our service which we have been using “Amazon AWS service” with minimum configuration console maintained by our team members. Here we use 10

Frontend languages as :

- HTML
- CSS
- JAVASCRIPT
- PHP

Backend languages as :

- MYSQL
- PHP

1.4 Problem Statement

The present system includes a physical showcase where people search for their required book and purchase them afterwards. This procedure necessitates every feature like book, buyer, seller and price to be actually present at the sale. We aim to develop a website as a front end of a bookstore portal to eradicate this physical presence at sale. As the web based business get to be distinctly widespread, the Imperial Book Store is considering about extending its support of the services through the Internet. This project is to develop a Web application (called Imperial Web Book Store or IWBS) to assist the Imperial book shop with operating its business through the internet. The IWBS system supports online book selling and shopping. It additionally permits clients to offer their utilized books to the book shop by means of the Internet

CHAPTER 2
REQUIREMENT SPECIFICATION

Requirements analysis in systems engineering and software engineering encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users.

Requirements analysis is critical to the success of a development project. Requirements must be documented, actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Requirements can be architectural, structural, behavioural, functional, and non-functional.

The development of the project needs some requirement to make the project perform better and achieves the goal of the project. In developing Online Second Hand Book Buying & Selling Portal Software Project, the capabilities of computer and hardware play a big impact on project quality. The project maker should determine the minimum requirements of hardware and also software to be used to develop a good and attractive project.

There are two phases of requirement analysis as given below

1) Primary Research: Identifying the user requirements by conducting a survey based on a questionnaire.
2) Secondary Research: Comparing the identified requirements with already existing software having similar functionalities.

CHAPTER 3
PROJECT DESCRIPTION

3.1 Proposed System

We are using the automated website to overcome the disadvantages which are faced by the students and the teachers due to the manual paperwork. The proposed system is fully automated and requires only one person to maintain the functionalities. “Online Second Hand Book Buying & Selling Portal Software Project” is to take online test efficiently and no time wasting for checking the paper. It effectively estimates the candidate completely via an automated system which besides preserving time, offers a swifter outcome. Therefore, all these activities are stored in the “Online Second Hand Book Buying & Selling Portal Software Project” database. These data are stored automatically and we can easily retrieve the data.

There are two users on this website:

1. Admin
2. Buyer

3.1.1 Admin:

Admin will log in with a unique id and password. They add all the information regarding the quiz being held at the school/college. They can also modify the contents of the website. The admin provides a unique id and password to all the students.

3.1.2 Buyer:

Using the id and the password provided, users can view all the details of the books on the website. They can buy or sell books in the selling portal available on the main home page of our website and can also choose the categories according to their convenience. The database connectivity is provided to the website which helps to store the data securely. Therefore, nowadays almost all the schools and colleges are using this automated system. It makes the work easier and saves time and effort.

3.2 Modules included:

1. Admin Signup and Login.
2. Student Signup and Login.
3. Add details and Display results.
4. Register and Check results.
5. Logout.

3.3 Advantages:

1. Paperwork reduction.
2. Human effort can be reduced drastically.
3. Operations that are done manually can be done within a matter of seconds.
4. Data will be stored securely.
5. Data can be retrieved easily and quickly.

3.4 Feasibility study:

A feasibility study is an evaluation and analysis of a project or system that somebody has proposed. It is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable.

Feasibility means the degree or state of being easily, conveniently, or reasonably done. If something is feasible, it means that we can do it, make it, or achieve it. A feasibility analysis evaluates the project's potential for success. It aims to objectively and rationally uncover the strengths and weaknesses of an existing business in the natural environment.

The goal is to determine whether the project should go ahead, be redesigned, or else abandoned altogether. The five frames of analysis are: The frame of definition, the frame of contextual risks, the frame of potentiality, the parametric frame, the frame of dominant and contingency strategies.

Three key considerations involved in the feasibility analysis are:

1. Technical Feasibility
2. Economical Feasibility
3. Scheduling Feasibility

3.4.1 Technical Feasibility:

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

3.4.2 Economic Feasibility:

This assessment typically involves a cost benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3.4.3 Scheduling Feasibility:

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

3.5 Website Specifications:

There are two types of specifications:

1. Hardware Specifications.
2. Software Specifications.

3.5.1 Hardware Specifications:

3.5.1.1 Processor: Intel dual-core or above

3.5.1.2 Processor Speed: 1.0GHZ or above

3.5.1.3 RAM: 1GB RAM or above

3.5.1.4 Hard Disk: 20 GB hard disk or above

3.5.2 Software Specifications:

3.5.2.1 Languages: PHP, CSS, HTML

3.5.2.2 Operating System: Windows, Linux etc.

3.5.2.3 Server: XAMPP

3.5.2.4 Browser: Chrome, Firewall etc.

CHAPTER 4

MODULE DESCRIPTION

4.1 General Architecture:

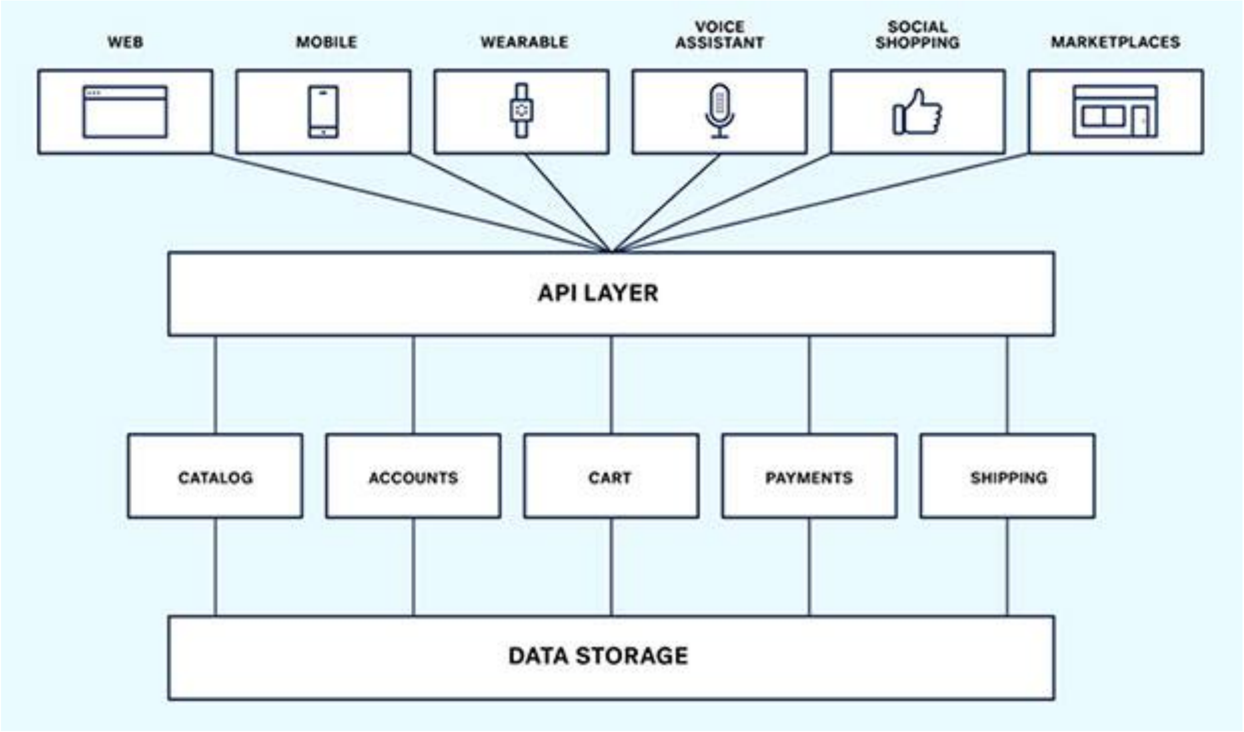


Fig 4.1 Architecture of Module Description

In the project implementation, we create two types of accounts one is for the user and the other for the admin. The administrator has the full rights to access the accounts and update the data He provides the information about the event taking place For a user he/she needs to sign up and then use the login page to choose events, register and checks results.

4.2 Module Description:

1. **Index:**
   This is the main and Home page of the Website.
2. **Registration:**
   This module allows user registration.
3. **Categories:**
   This module lists the categories of books available.
4. **Sell books:**
   This module allows users to add book details to sell their books.
5. **View Cart:**
   This module lists all the books added by the users for buying.
6. **Admin:**
   This module is used by the user to view queries posted by users.
7. **Contact Us:**
   This module is used by the users to send queries to Admin.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation:

I. Database overview:

The online bookstore database sit on the Oracle is made of 4 tables.

- **Fbooks:**

Define the inventory of books. It has the fields of: ISBN(primary key), Title, Author, ImageType, Price, Publisher and NumberInstock.

- **Fcutomers:**

It saves all the information of the customers. It has the fields of: CusomterID(primary key), User, Passwd, Fname, Lname, Address, City, State, ZipCode, Phone, Fax and Email.

- **Forders:**

It has the order information. The fields are: OrderID (primary key), CustomerID, TotalPrice, OrderDate, CardName, CreditCard, CreditType and ExpDate.

- **Forderdetail:**

It saves the information of every books on a multiple book order. The fields are: OrderID, ISBN, Quantity and ShipDate. Primary key is the combination of OrderID and ISBN.

II. Java class file overview:-

This project consists of a total of 13 java files.

- **BookDetails.java**

Define the object for an individual book. Every book has ISBN number, the Title and Price.

- **ShoppingCartItem.java**

Every shoppingCartItem holds an object, which is a book and the quantity of the book.

- **ShoppingCart.java**

ShoppingCart is implemented as a hashtable which can hold a book in ShoppingCartItem and number of different books in the hashtable. Add method is to add one ShoppingCartItem into the hashtable. Remove methold remove one from the hashtable.

- **BookpoolSearch.java**

This class worked as a bridge between the client's query and the online bookseller Bookpool.com. This can handle cilent's search by title and by author. After getting client input either by title or author, a query in the format of URL string is generated and sent to bookpool.com using java's standard URL class. Bookpool.com will respond to this query by returning a long string of html source code containing books of their database matching the query provided by the client.

- **FatbrainSearch.java**

Another bridge class works similar to the BookpoolSearch.java. It will connect to the online bookseller Fabrain.com. This can handle the user search query by title, author, subject and publisher.

- **BookSearch.java**

This class handle client's query to search the local Oracle database by Title, Author, ISBN and Publisher using Java JDBC. And using Java Servlet to generate html source code to display on client's screen. ShoppingCart was implemented by the Servlet Session Tracking technology. It takes the input from the bookstore main page, generates the searching query, searches the database and pastes the book found to user's screen. User can add any displayed books into the shopping cart by pressing the 'add to cart' icon.

- **ISBNSearch.java**

This class handles client's query to search by ISBN and display detailed book information for the user. This include the full title of the book, the image of the book and also the author, publisher and numbers in stock. The image is handle in a simple way. All the image files are saved in the server's /images directory.

- **ShowCard.java**

This class will display all the books user add to his shopping cart. Through the Servlet Session Tracking API, session was found by checking the session ID. Books information was extracted from the hashtable of the shoppingCart object. Html code was generated and send to user's screen by the Servlet doGet method.

- **FatbrainSearch.java**

Another bridge class works similar to the BookpoolSearch.java. It will connect to the online bookseller Fabrain.com. This can handle the user search query by title, author, subject and publisher.

- **BookSearch.java**

This class handle client's query to search the local Oracle database by Title, Author, ISBN and Publisher using Java JDBC. And using Java Servlet to generate html source code to display on client's screen. ShoppingCart was implemented by the Servlet Session Tracking technology. It takes the input from the bookstore main page, generates the searching query, searches the database and pastes the book found to user's screen. User can add any displayed books into the shopping cart by pressing the 'add to cart' icon.

- **ISBNSearch.java**

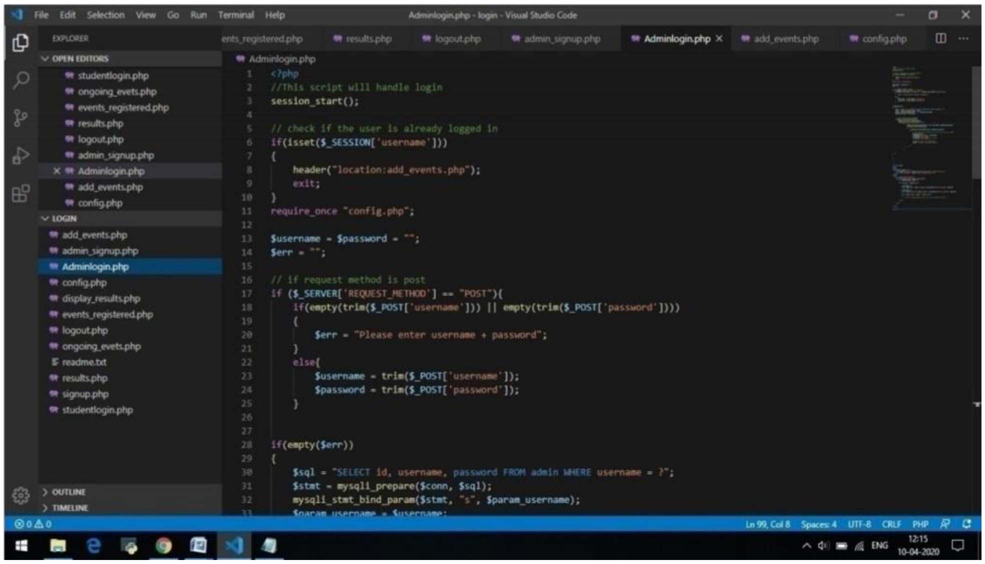
This class handles client's query to search by ISBN and display detailed book information for the user. This include the full title of the book, the image of the book and also the author, publisher and numbers in stock. The image is handle in a simple way. All the image files are saved in the server's /images directory.

- **ShowCard.java**

This class will display all the books user add to his shopping cart. Through the Servlet Session Tracking API, session was found by checking the session ID. Books information was extracted from the hashtable of the shoppingCart object. Html code was generated and send to user's screen by the Servlet doGet method.

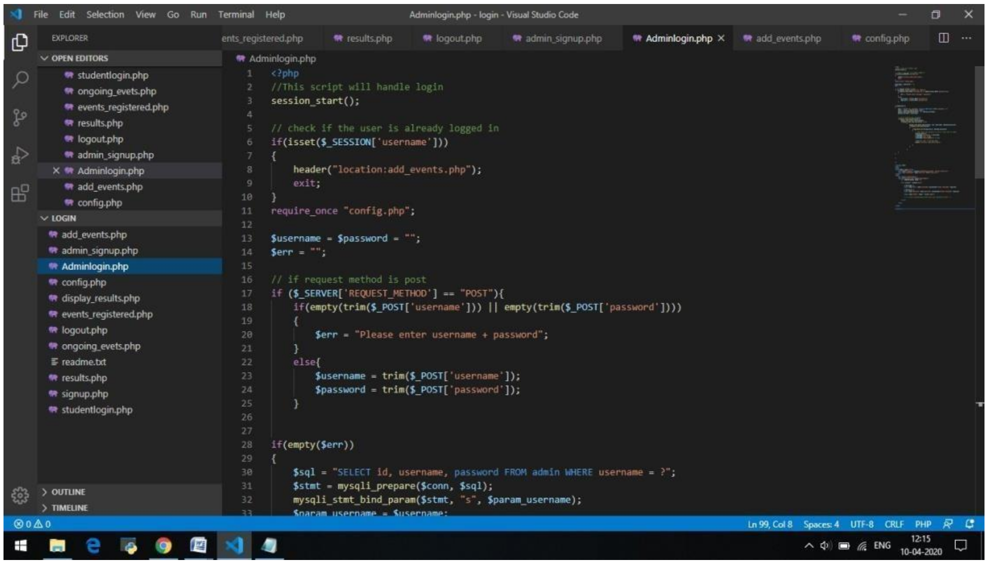
5.3 INPUT

5.3.1 PHP Code:



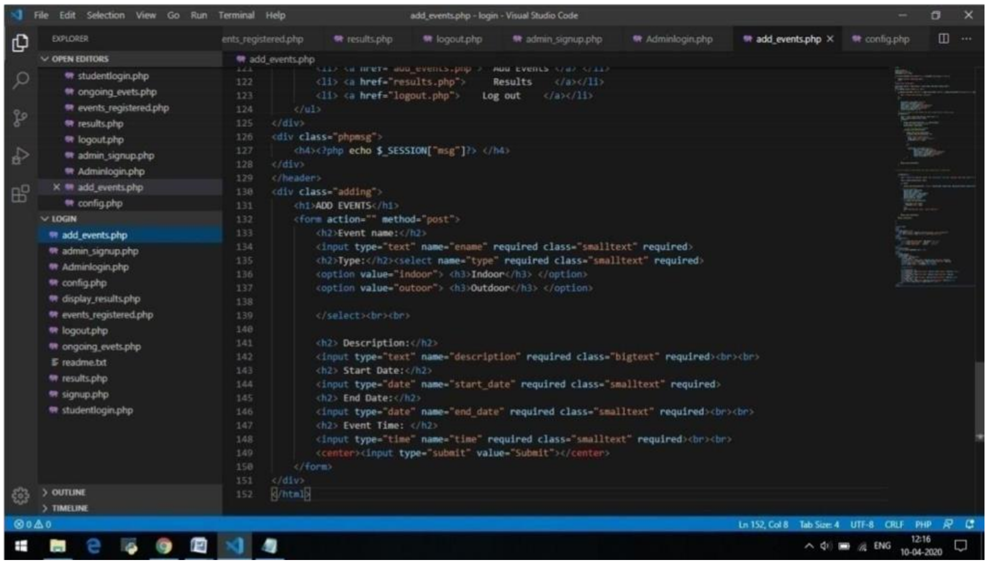
```php
<?php
//This script will handle login
session_start();

// check if the user is already logged in
if(isset($_SESSION['username']))
{
    header("location:add_events.php");
    exit;
}
require_once "config.php";

$username = $password = "";
$err = "";

// if request method is post
if ($_SERVER['REQUEST_METHOD'] == "POST"){
    if(empty(trim($_POST['username'])) || empty(trim($_POST['password'])))
    {
        $err = "Please enter username + password";
    }
    else{
        $username = trim($_POST['username']);
        $password = trim($_POST['password']);
    }


if(empty($err))
{
    $sql = "SELECT id, username, password FROM admin WHERE username = ?";
    $stmt = mysqli_prepare($conn, $sql);
    mysqli_stmt_bind_param($stmt, "s", $param_username);
    $param_username = $username;
```

Fig 5.3.1.1 Admin Signup



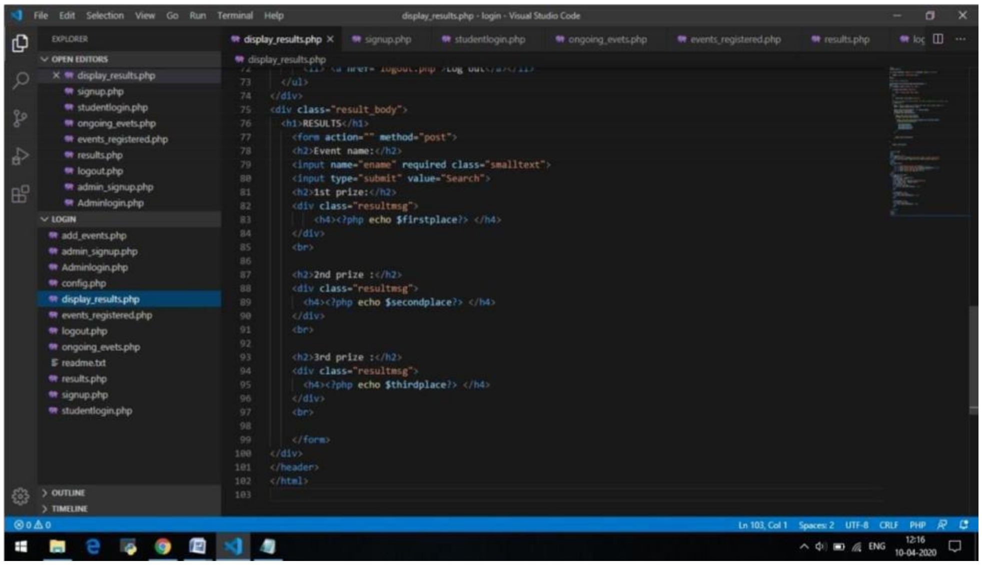
```php
<?php
//This script will handle login
session_start();

// check if the user is already logged in
if(isset($_SESSION['username']))
{
    header("location:add_events.php");
    exit;
}
require_once "config.php";

$username = $password = "";
$err = "";

// if request method is post
if ($_SERVER['REQUEST_METHOD'] == "POST"){
    if(empty(trim($_POST['username'])) || empty(trim($_POST['password'])))
    {
        $err = "Please enter username + password";
    }
    else{
        $username = trim($_POST['username']);
        $password = trim($_POST['password']);
    }


if(empty($err))
{
    $sql = "SELECT id, username, password FROM admin WHERE username = ?";
    $stmt = mysqli_prepare($conn, $sql);
    mysqli_stmt_bind_param($stmt, "s", $param_username);
    $param_username = $username;
```

Fig 5.3.1.2 Admin Login



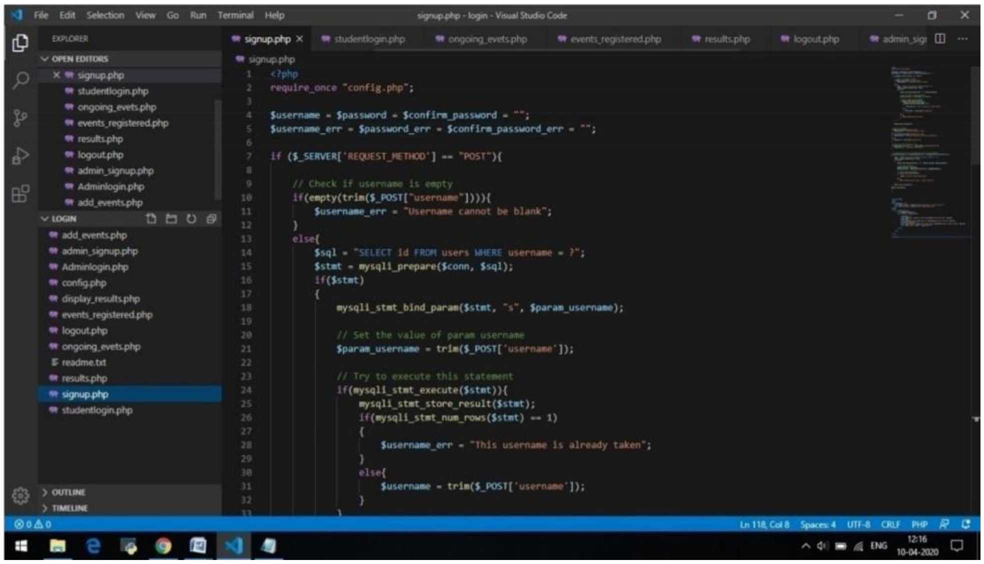
```
            <li> <a href="add_events.php"> Add Events </a> </li>
            <li> <a href="results.php">     Results    </a></li>
            <li> <a href="logout.php">    Log out    </a></li>
        </ul>
    </div>
    <div class="phpmsg">
        <h4><?php echo $_SESSION["msg"]?> </h4>
    </div>
    </header>
    <div class="adding">
        <h1>ADD EVENTS</h1>
        <form action="" method="post">
            <h2>Event name:</h2>
            <input type="text" name="ename" required class="smalltext" required>
            <h2>Type:</h2><select name="type" required class="smalltext" required>
            <option value="indoor"> <h3>Indoor</h3> </option>
            <option value="outoor"> <h3>Outdoor</h3> </option>

            </select><br><br>

            <h2> Description:</h2>
            <input type="text" name="description" required class="bigtext" required><br><br>
            <h2> Start Date:</h2>
            <input type="date" name="start_date" required class="smalltext" required>
            <h2> End Date:</h2>
            <input type="date" name="end_date" required class="smalltext" required><br><br>
            <h2> Event Time: </h2>
            <input type="time" name="time" required class="smalltext" required><br><br>
            <center><input type="submit" value="Submit"></center>
        </form>
    </div>
</html>
```

Fig 5.3.1.3 Registration Page



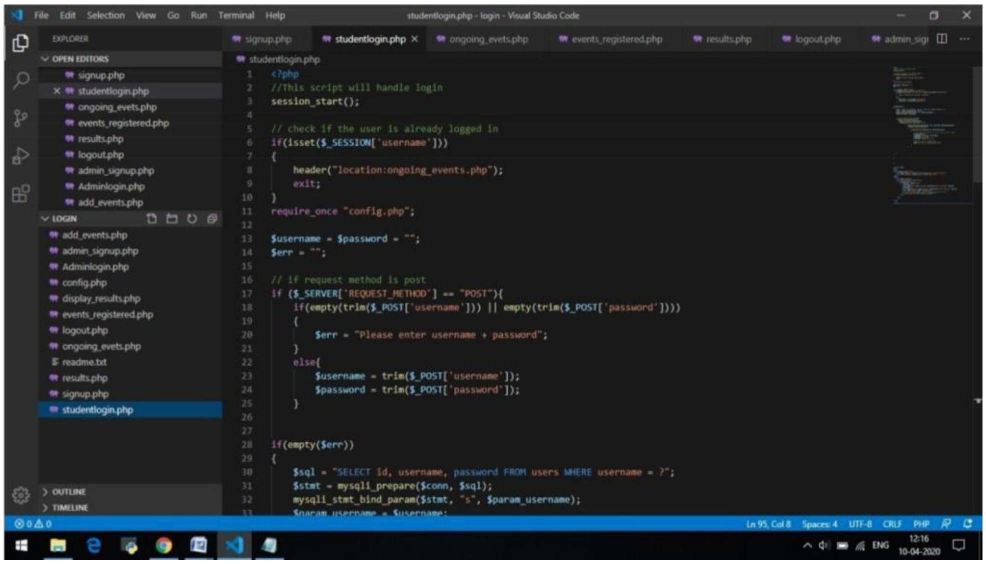
```
        <li> <a href="logout.php">Log out</a></li>
      </ul>
    </div>
    <div class="result_body">
      <h1>RESULTS</h1>
        <form action="" method="post">
        <h2>Event name:</h2>
        <input name="ename" required class="smalltext">
        <input type="submit" value="Search">
        <h2>1st prize:</h2>
        <div class="resultmsg">
            <h4><?php echo $firstplace?> </h4>
        </div>
        <br>

        <h2>2nd prize :</h2>
        <div class="resultmsg">
          <h4><?php echo $secondplace?> </h4>
        </div>
        <br>

        <h2>3rd prize :</h2>
        <div class="resultmsg">
          <h4><?php echo $thirdplace?> </h4>
        </div>
        <br>

        </form>
    </div>
    </header>
    </html>
```

Fig 5.3.1.4 Add Books



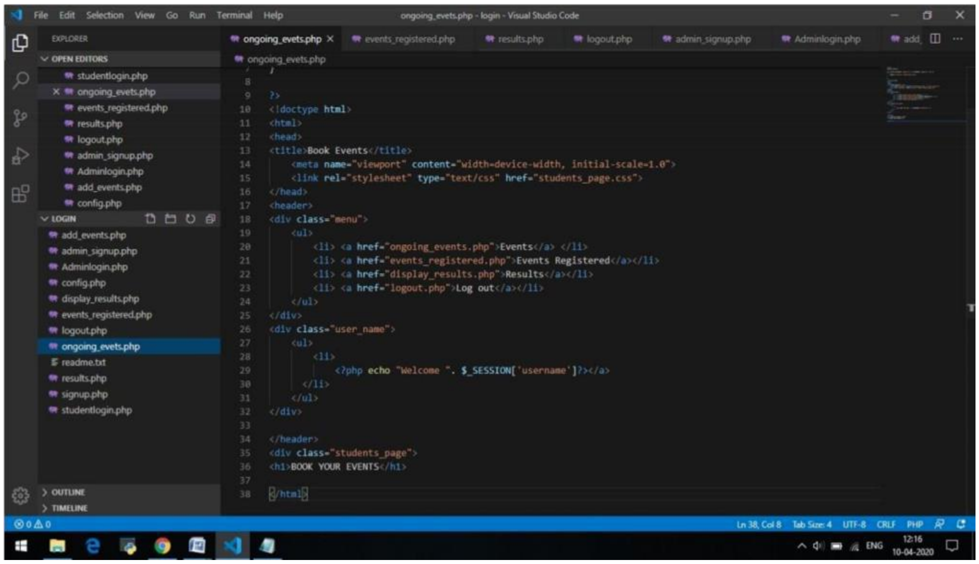
```php
<?php
require_once "config.php";

$username = $password = $confirm_password = "";
$username_err = $password_err = $confirm_password_err = "";

if ($_SERVER['REQUEST_METHOD'] == "POST"){

    // Check if username is empty
    if(empty(trim($_POST["username"]))){
        $username_err = "Username cannot be blank";
    }
    else{
        $sql = "SELECT id FROM users WHERE username = ?";
        $stmt = mysqli_prepare($conn, $sql);
        if($stmt)
        {
            mysqli_stmt_bind_param($stmt, "s", $param_username);

            // Set the value of param username
            $param_username = trim($_POST['username']);

            // Try to execute this statement
            if(mysqli_stmt_execute($stmt)){
                mysqli_stmt_store_result($stmt);
                if(mysqli_stmt_num_rows($stmt) == 1)
                {
                    $username_err = "This username is already taken";
                }
                else{
                    $username = trim($_POST['username']);
                }
            }
```

Fig 5.3.1.5 Buyer Signup



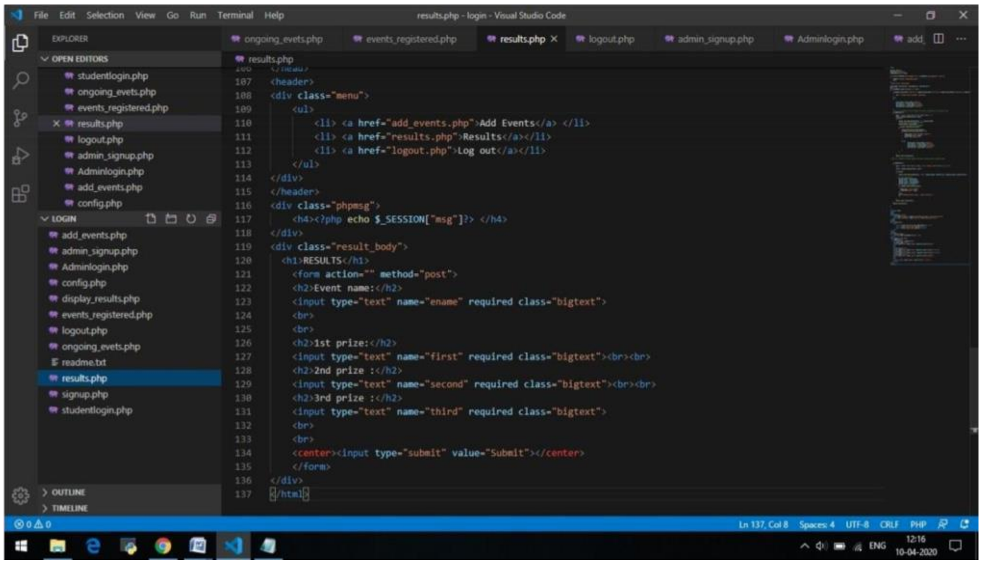
```php
<?php
//This script will handle login
session_start();

// check if the user is already logged in
if(isset($_SESSION['username']))
{
    header("location:ongoing_events.php");
    exit;
}
require_once "config.php";

$username = $password = "";
$err = "";

// if request method is post
if ($_SERVER['REQUEST_METHOD'] == "POST"){
    if(empty(trim($_POST['username'])) || empty(trim($_POST['password'])))
    {
        $err = "Please enter username + password";
    }
    else{
        $username = trim($_POST['username']);
        $password = trim($_POST['password']);
    }


if(empty($err))
{
    $sql = "SELECT id, username, password FROM users WHERE username = ?";
    $stmt = mysqli_prepare($conn, $sql);
    mysqli_stmt_bind_param($stmt, "s", $param_username);
    $param_username = $username;
```

Fig 5.3.1.6 Buyer Login



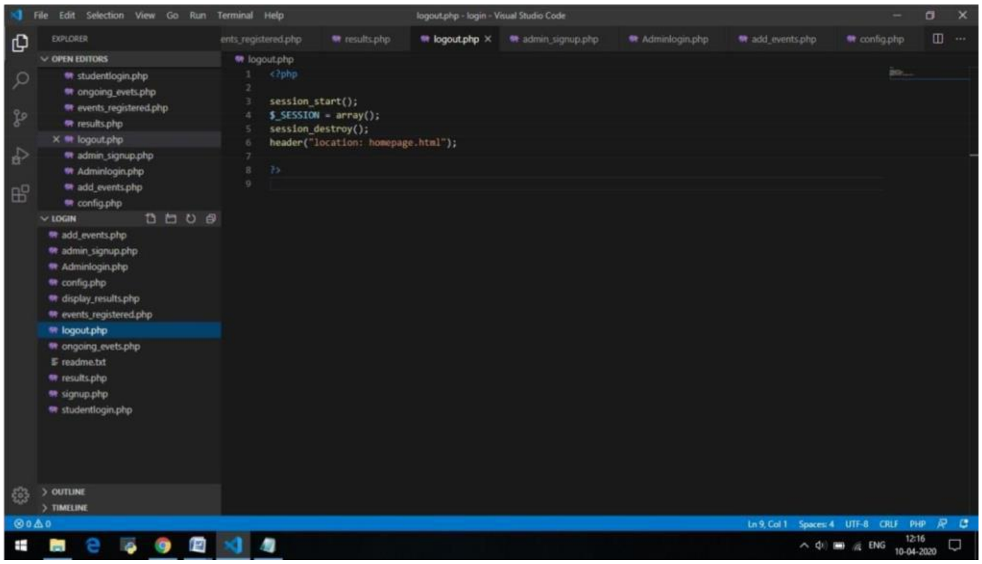
```
}

?>
<!doctype html>
<html>
<head>
<title>Book Events</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="students_page.css">
</head>
<header>
<div class="menu">
    <ul>
        <li> <a href="ongoing_events.php">Events</a> </li>
        <li> <a href="events_registered.php">Events Registered</a></li>
        <li> <a href="display_results.php">Results</a></li>
        <li> <a href="logout.php">Log out</a></li>
    </ul>
</div>
<div class="user_name">
    <ul>
        <li>
            <?php echo "Welcome ". $_SESSION['username']?></a>
          </li>
    </ul>
</div>

</header>
<div class="students_page">
<h1>BOOK YOUR EVENTS</h1>

</html>
```

Fig 5.3.1.7 View Cart



```
</head>
<header>
<div class="menu">
    <ul>
        <li> <a href="add_events.php">Add Events</a> </li>
        <li> <a href="results.php">Results</a></li>
        <li> <a href="logout.php">Log out</a></li>
    </ul>
</div>
</header>
<div class="phpmsg">
    <h4><?php echo $_SESSION["msg"]?> </h4>
</div>
<div class="result_body">
  <h1>RESULTS</h1>
    <form action="" method="post">
    <h2>Event name:</h2>
    <input type="text" name="ename" required class="bigtext">
    <br>
    <br>
    <h2>1st prize:</h2>
    <input type="text" name="first" required class="bigtext"><br><br>
    <h2>2nd prize :</h2>
    <input type="text" name="second" required class="bigtext"><br><br>
    <h2>3rd prize :</h2>
    <input type="text" name="third" required class="bigtext">
    <br>
    <br>
    <center><input type="submit" value="Submit"></center>
    </form>
</div>
</html>
```

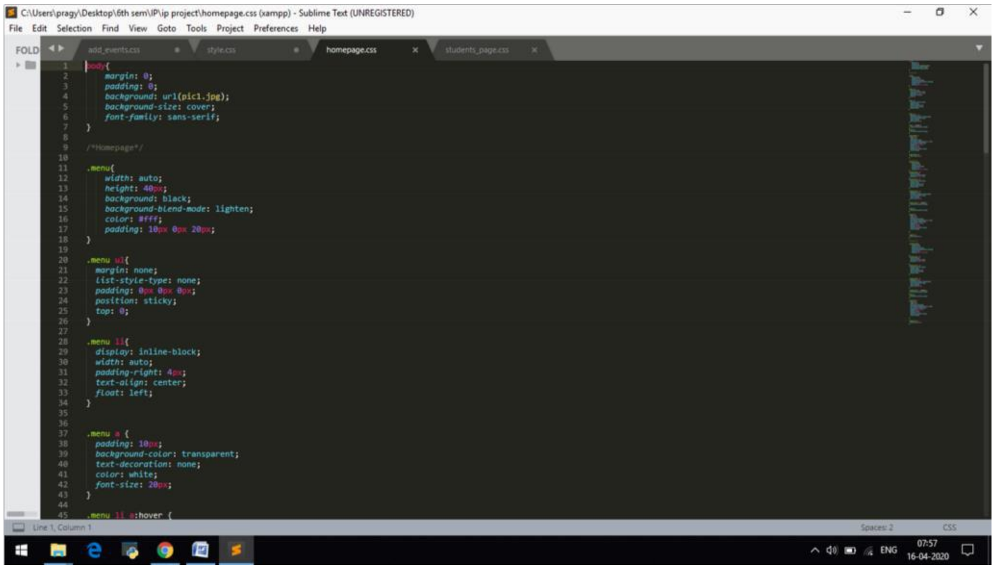
Fig 5.3.1.9 Results



```php
<?php

session_start();
$_SESSION = array();
session_destroy();
header("location: homepage.html");

?>

```

Fig 5.3.1.10 Logging out

5.3.2 CSS:

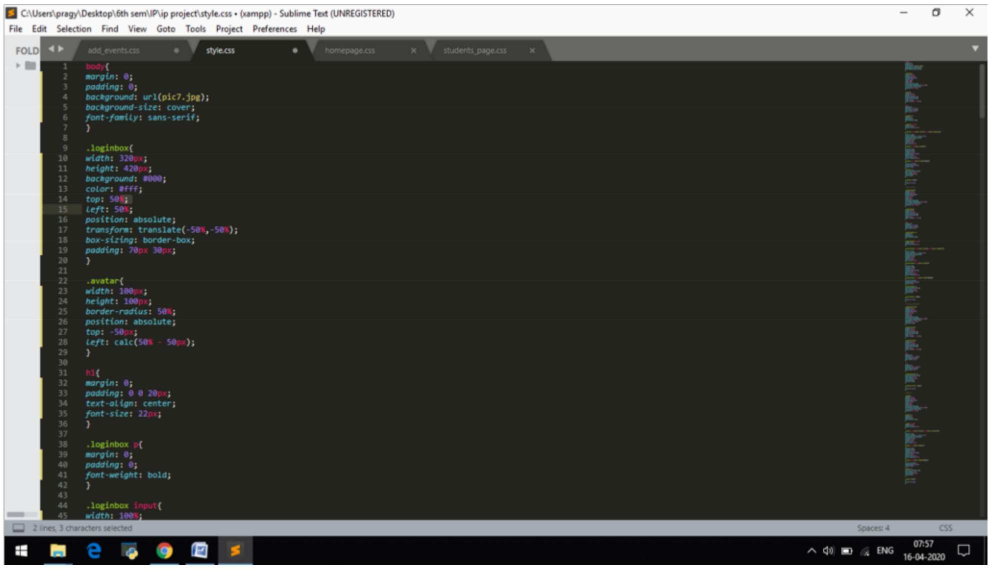


C:\Users\pragy\Desktop\6th sem\IP\ip project\homepage.css (xampp) - Sublime Text (UNREGISTERED)

```css
body{
    margin: 0;
    padding: 0;
    background: url(pic1.jpg);
    background-size: cover;
    font-family: sans-serif;
}

/*Homepage*/

.menu{
    width: auto;
    height: 40px;
    background: black;
    background-blend-mode: lighten;
    color: #fff;
    padding: 10px 0px 20px;
}

.menu ul{
  margin: none;
  list-style-type: none;
  padding: 0px 0px 0px;
  position: sticky;
  top: 0;
}

.menu li{
  display: inline-block;
  width: auto;
  padding-right: 4px;
  text-align: center;
  float: left;
}


.menu a {
  padding: 10px;
  background-color: transparent;
  text-decoration: none;
  color: white;
  font-size: 20px;
}

.menu li a:hover {
```

Line 1, Column 1

Fig 5.3.2.1 Homepage

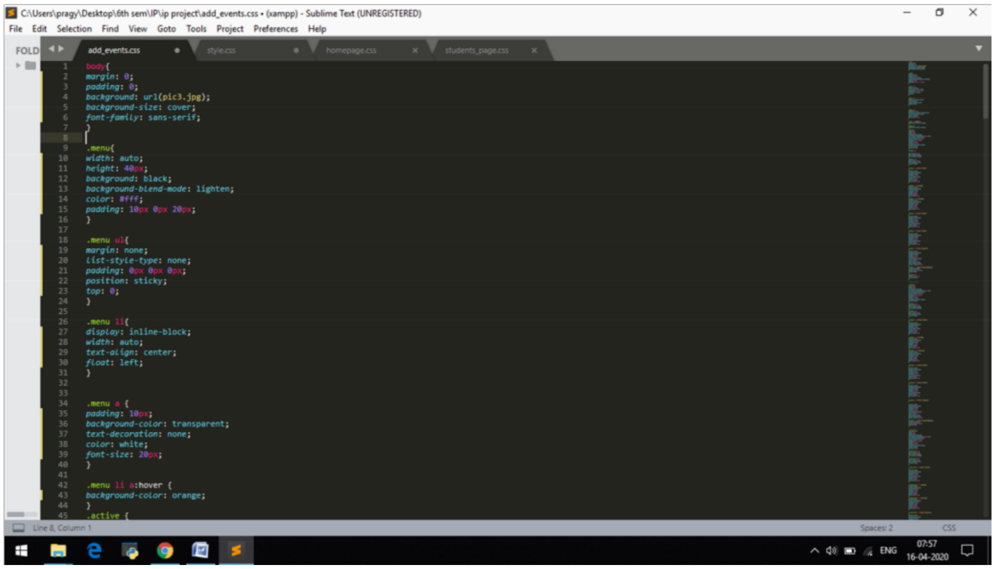


C:\Users\pragy\Desktop\6th sem\IP\ip project\style.css • (xampp) - Sublime Text (UNREGISTERED)

```css
body{
margin: 0;
padding: 0;
background: url(pic7.jpg);
background-size: cover;
font-family: sans-serif;
}

.loginbox{
width: 320px;
height: 420px;
background: #000;
color: #fff;
top: 50%;
left: 50%;
position: absolute;
transform: translate(-50%,-50%);
box-sizing: border-box;
padding: 70px 30px;
}

.avatar{
width: 100px;
height: 100px;
border-radius: 50%;
position: absolute;
top: -50px;
left: calc(50% - 50px);
}

h1{
margin: 0;
padding: 0 0 20px;
text-align: center;
font-size: 22px;
}

.loginbox p{
margin: 0;
padding: 0;
font-weight: bold;
}

.loginbox input{
width: 100%;
```

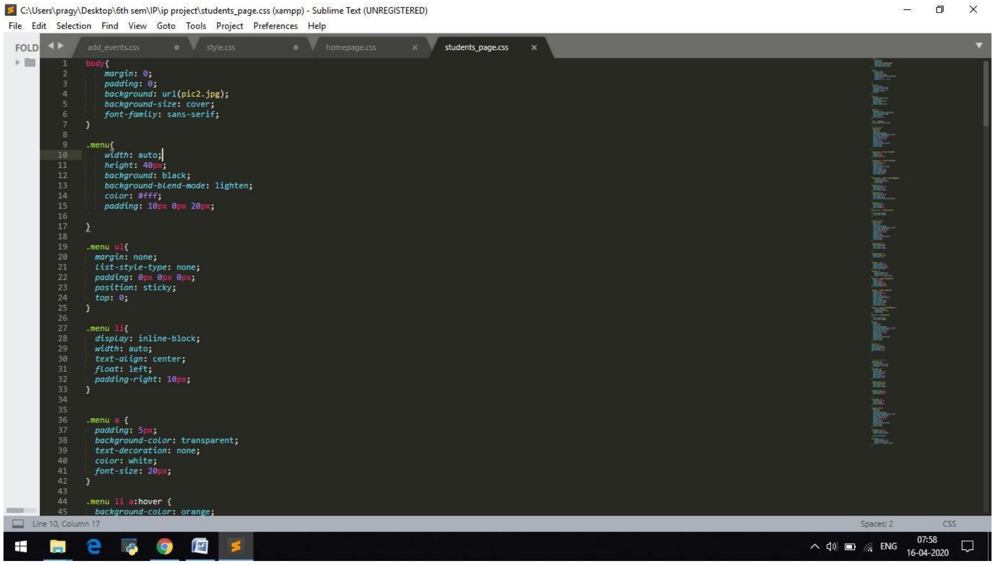
2 lines, 3 characters selected

Fig 5.3.2.2 Style


C:\Users\pragy\Desktop\6th sem\IP\ip project\add_events.css • (xampp) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLD
add_events.css
style.css
homepage.css
students_page.css
Line 8, Column 1
Spaces: 2
CSS
ENG
07:57
16-04-2020

```
body{
margin: 0;
padding: 0;
background: url(pic3.jpg);
background-size: cover;
font-family: sans-serif;
}

.menu{
width: auto;
height: 40px;
background: black;
background-blend-mode: lighten;
color: #fff;
padding: 10px 0px 20px;
}

.menu ul{
margin: none;
list-style-type: none;
padding: 0px 0px 0px;
position: sticky;
top: 0;
}

.menu li{
display: inline-block;
width: auto;
text-align: center;
float: left;
}


.menu a {
padding: 10px;
background-color: transparent;
text-decoration: none;
color: white;
font-size: 20px;
}

.menu li a:hover {
background-color: orange;
}
.active {
```

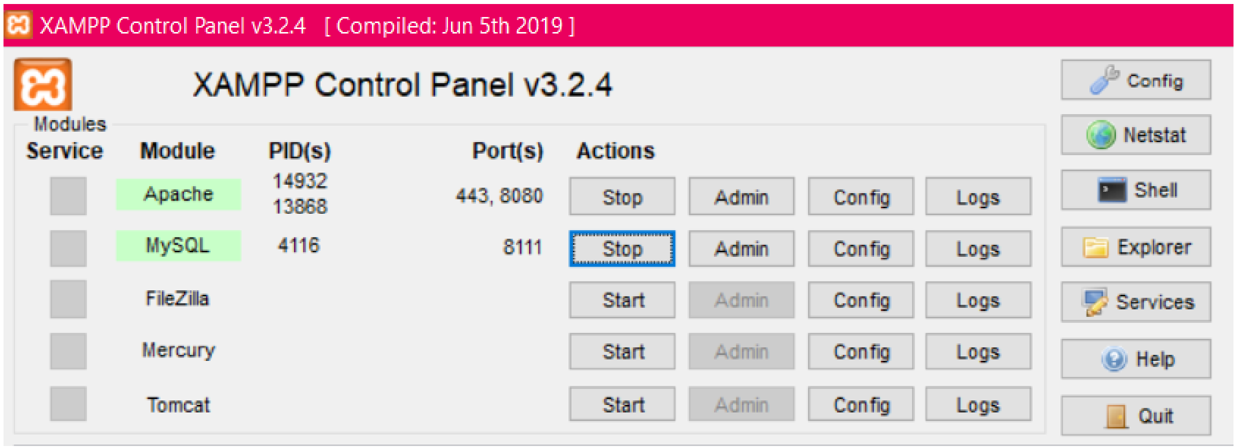
Fig 5.3.2.3 Add Book


C:\Users\pragy\Desktop\6th sem\IP\ip project\students_page.css (xampp) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLD
add_events.css
style.css
homepage.css
students_page.css
Line 10, Column 17
Spaces: 2
CSS
ENG
07:58
16-04-2020

```
body{
    margin: 0;
    padding: 0;
    background: url(pic2.jpg);
    background-size: cover;
    font-family: sans-serif;
}

.menu{
    width: auto;
    height: 40px;
    background: black;
    background-blend-mode: lighten;
    color: #fff;
    padding: 10px 0px 20px;

}

.menu ul{
  margin: none;
  list-style-type: none;
  padding: 0px 0px 0px;
  position: sticky;
  top: 0;
}

.menu li{
  display: inline-block;
  width: auto;
  text-align: center;
  float: left;
  padding-right: 10px;
}


.menu a {
  padding: 5px;
  background-color: transparent;
  text-decoration: none;
  color: white;
  font-size: 20px;
}

.menu li a:hover {
  background-color: orange;
```

Fig 5.3.2.4 Student Page

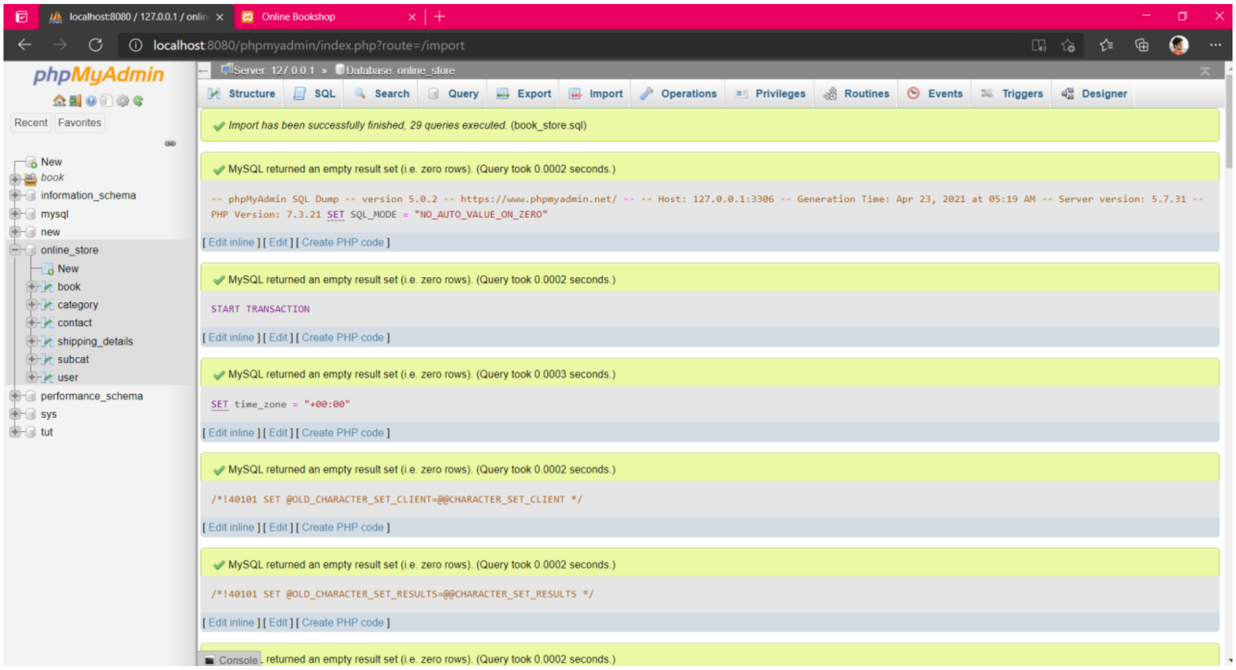
5.3.3 XAMPP:



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	14932 13868	443, 8080	Stop	Admin	Config	Logs
□	MySQL	4116	8111	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

Fig 5.3.3.1 Connection to start Apache and MySQL



✓ *Import has been successfully finished, 29 queries executed.* (book_store.sql)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
-- phpMyAdmin SQL Dump -- version 5.0.2 -- https://www.phpmyadmin.net/ -- -- Host: 127.0.0.1:3306 -- Generation Time: Apr 23, 2021 at 05:19 AM -- Server version: 5.7.31 --
PHP Version: 7.3.21 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
START TRANSACTION
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
SET time_zone = "+00:00"
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

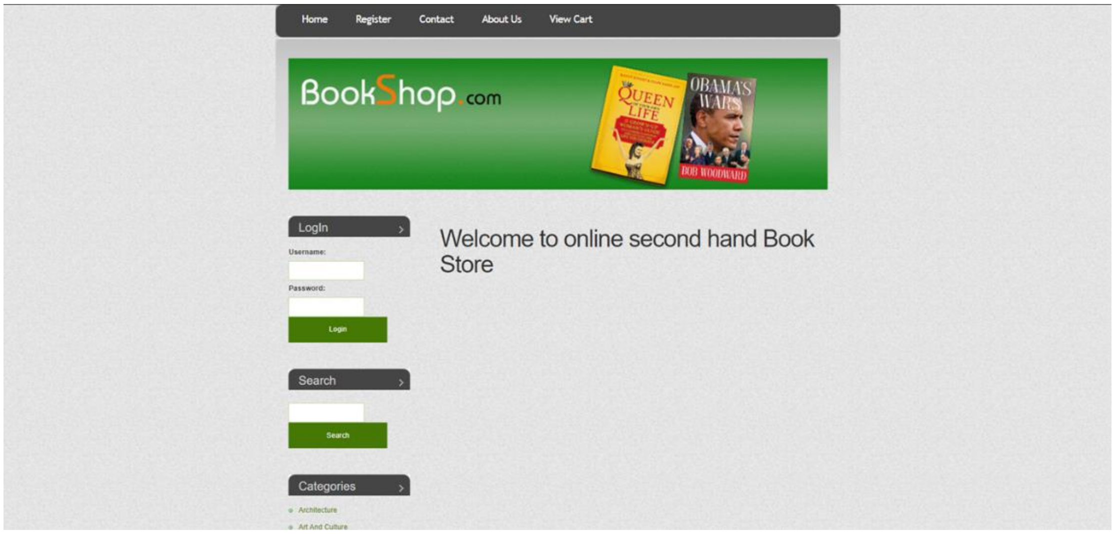
```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */
```

[Edit inline] [Edit] [Create PHP code]

returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

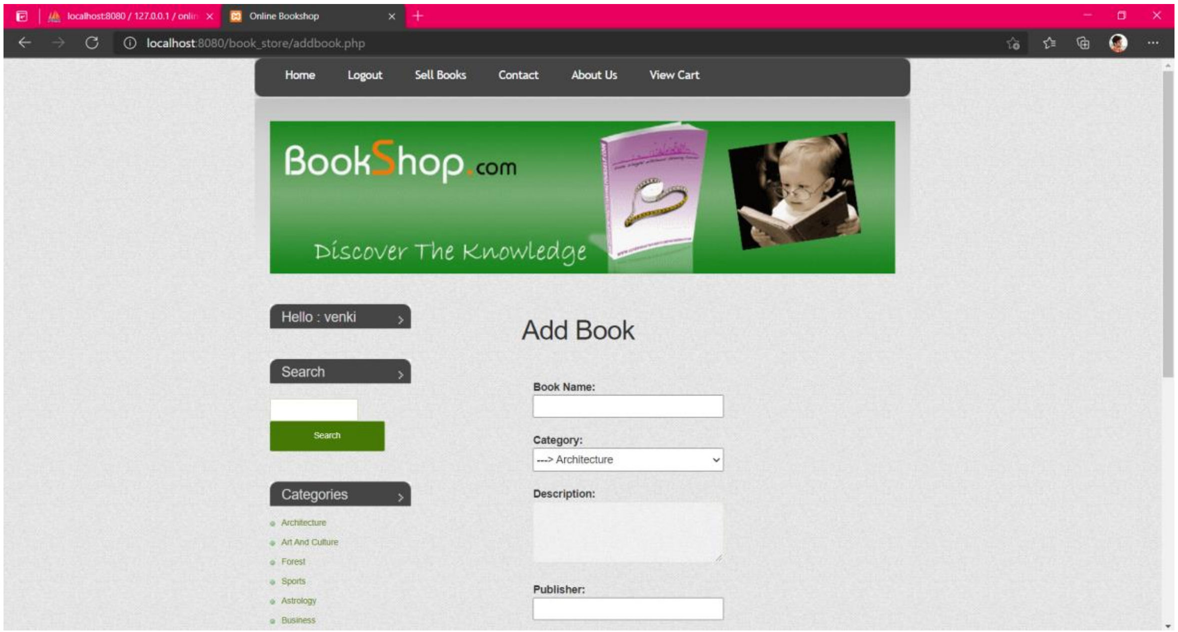
Fig 5.3.3.2 Database connectivity

5.4 OUTPUT:



Welcome to online second hand Book Store

Fig 5.4.1 Welcome Page



Add Book

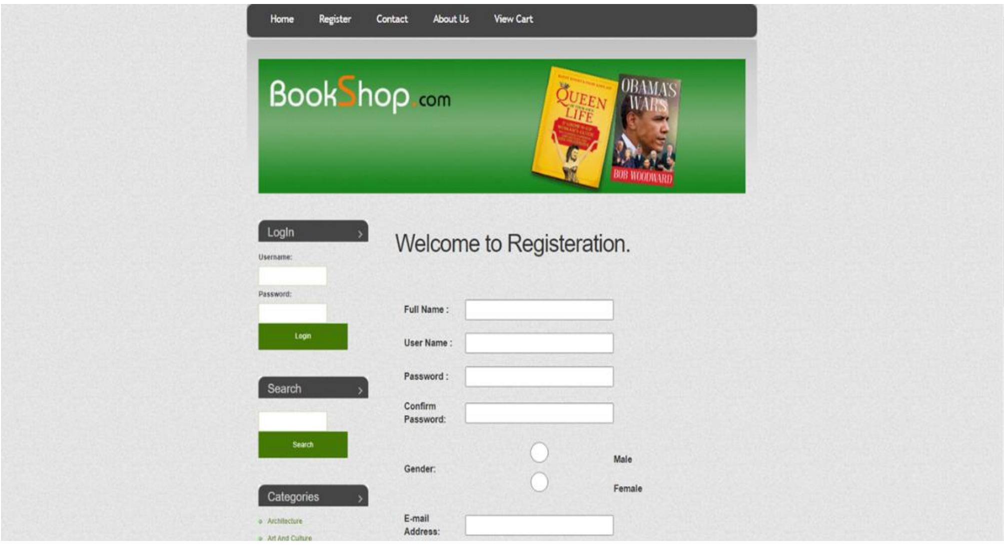
Book Name:

Category: ---> Architecture

Description:

Publisher:

Fig 5.4.2 Add Books



Welcome to Registeration.

Full Name :

User Name :

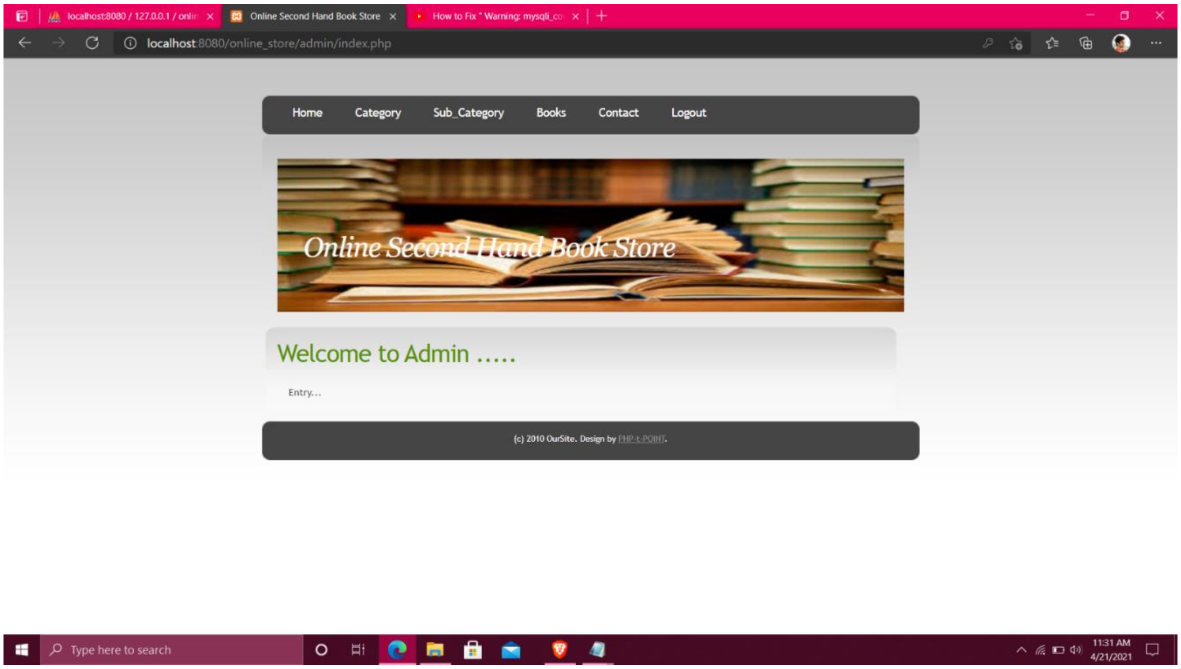
Password :

Confirm Password:

Gender: ○ Male ○ Female

E-mail Address:

Fig 5.4.3 Register

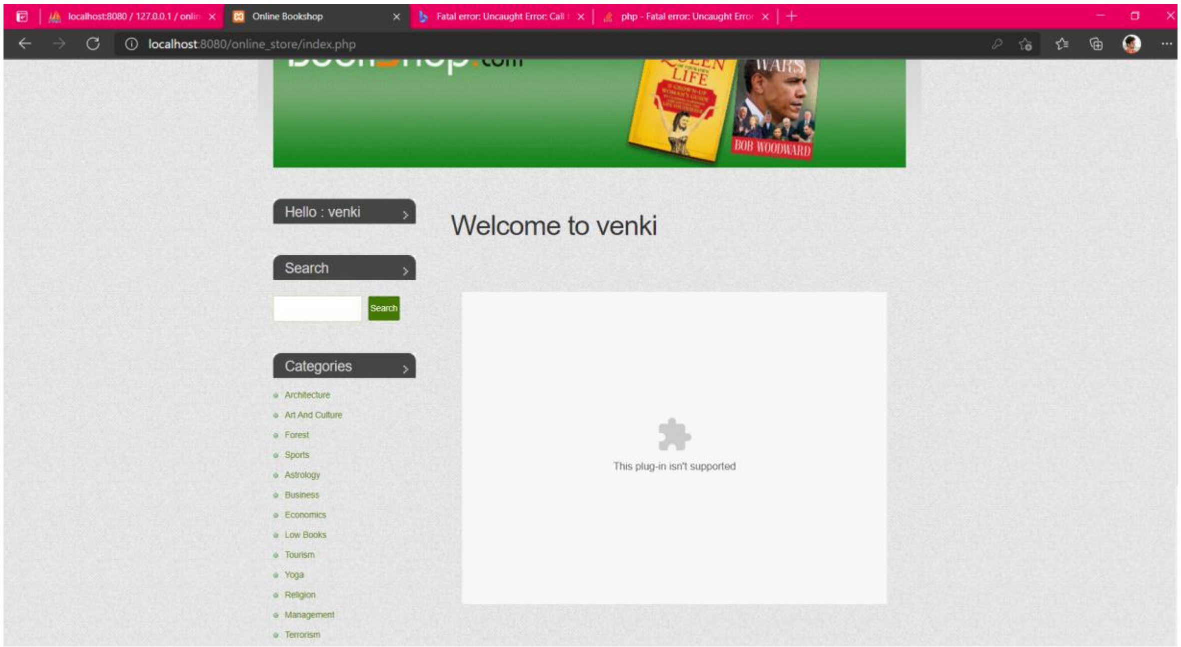

localhost:8080 / 127.0.0.1 / onlin
Online Second Hand Book Store
How to Fix " Warning: mysqli_co
localhost:8080/online_store/admin/index.php
Home
Category
Sub_Category
Books
Contact
Logout
Online Second Hand Book Store
Type here to search
11:31 AM
4/21/2021

Welcome to Admin .....

Entry...

(c) 2010 OurSite. Design by PHP-T-POINT.

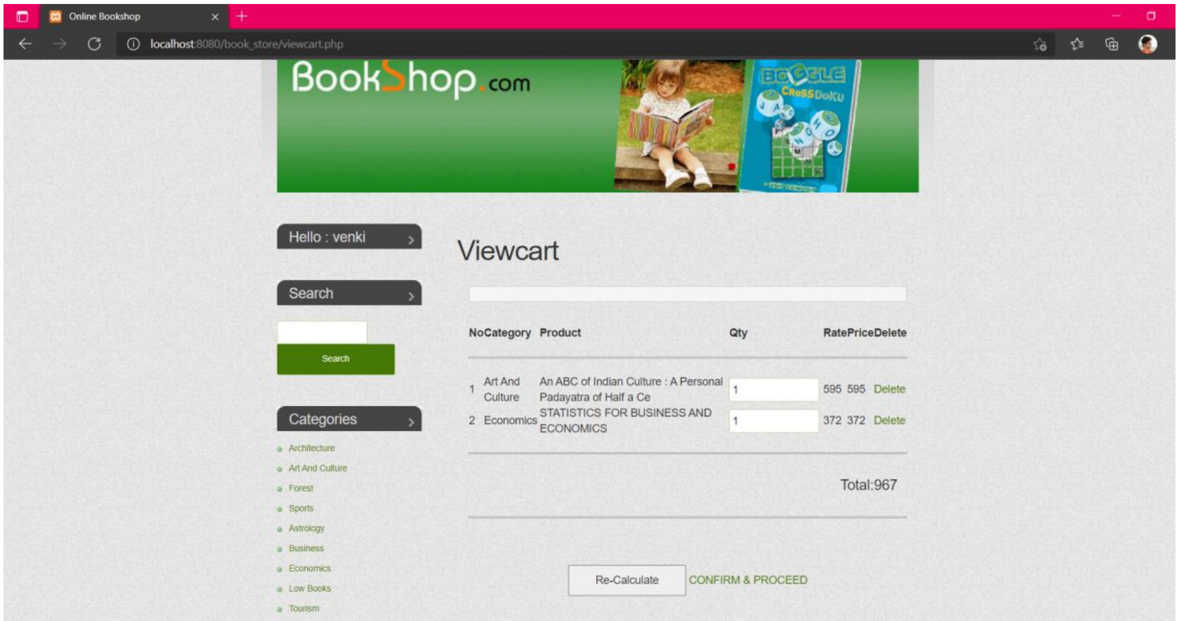
Fig 5.4.4 Admin Page



Welcome to venki

This plug-in isn't supported

Fig 5.4.5 User Page

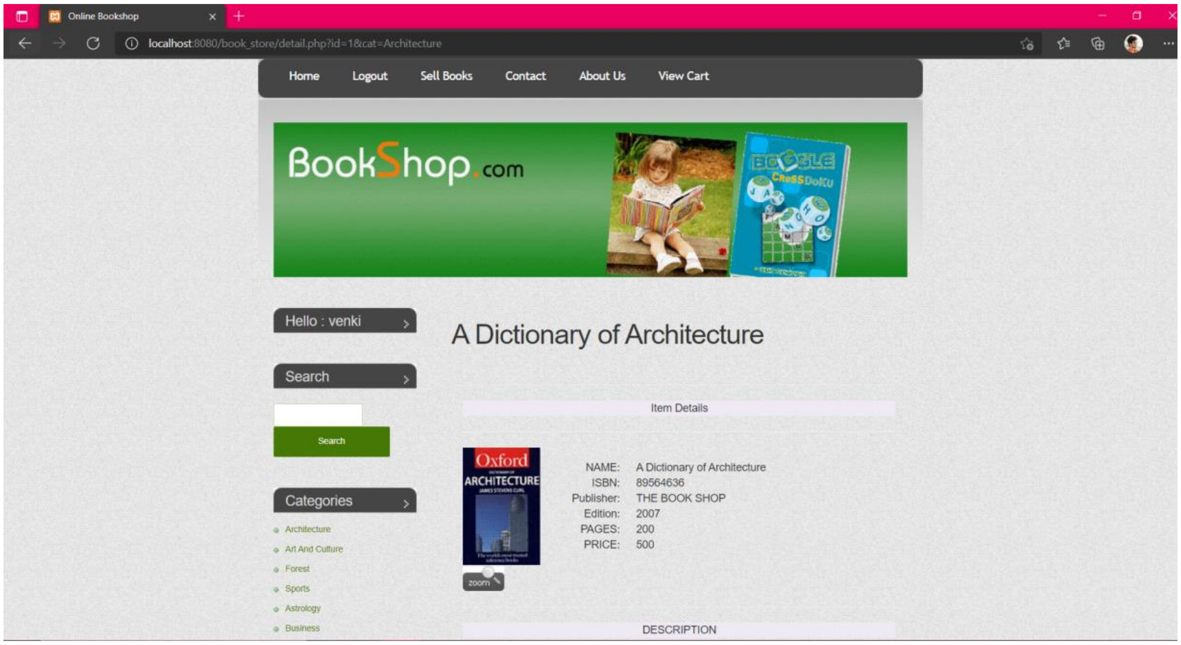


Viewcart

No	Category	Product	Qty	Rate	Price	Delete
1	Art And Culture	An ABC of Indian Culture : A Personal Padayatra of Half a Ce	1	595	595	Delete
2	Economics	STATISTICS FOR BUSINESS AND ECONOMICS	1	372	372	Delete

Total:967

Fig 5.4.6 Cart



A Dictionary of Architecture

Item Details

NAME: A Dictionary of Architecture
ISBN: 89564636
Publisher: THE BOOK SHOP
Edition: 2007
PAGES: 200
PRICE: 500

DESCRIPTION

Fig 5.4.7 Buy Books

CHAPTER 6

INSTALLATION INSTRUCTIONS

To create this website, one has to learn the basic HTML and PHP codes. Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS). CSS is used for describing the presentation of a document. PHP is a language to create dynamic interactive websites. We need to install Visual Studio Code and Xampp Control Panel in our system. VSC is a source-code editor for debugging and building different web applications. Xampp is a web server and it uses MySQL and Apache. We create database connectivity for the website where we store our data. This connectivity can be developed using the server.

We should open the XAMPP application and start the Apache and MySQL from there. After that, we need to open the XAMPP folder from the C directory. Then create a folder (eg-login) in the htdocs folder. The next step is to start the Visual Studio Code and open a new file (login) and we need to select a login and create different files with PHP format. In these files, we need to write our code for the websites we need to create. Firstly, we need to write a code for SQL connection so that there can be a database connection. After that, we have to create a database in the phpMyAdmin browser based on our codes to store the data.

When our codes are ready, we’ve to save them. After that we’ve to open the browser and type “localhost/login” and we will be able to check the result of our PHP files in the browser. Hence, we get our desired output.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion:

This Online Second Hand Book Buying & Selling Portal Software Project system provides a seller to sell second-hand books worldwide. It is automatically generated by the server. The administrator has the privilege to create, modify and delete the test papers and their particular questions. User can register, login and give the test with their specific id, and can see the results as well. We have worked and acquired more knowledge about the out project. The online system not only consumes less time but also helps to provide detailed information about its events. It can be used to check any information regarding the quiz event at any time. Manual paperwork is also reduced through this website. Our website is user-friendly and easy to use. With the use of HTML, PHP, CSS and XAMPP we've developed this website. It is an automatically working website hence it reduces human labour. Only one person can handle this website. It is developed in such a way that it can store a large number of data. Hence the data will be stored securely.

7.2 Future Enhancements:

- The main aim of our project is to create a good interaction between peer to peer.
- We are trying to do the project at the best level to satisfy all the end-users (i.e., sellers/buyers).
- In our future, we are decided to provide more security to our website which may not be hacked.
- It will be more empowering.
- Next, we are aiming to provide some online educational products to be sold in the portal.

REFERENCES

- IEEE Recommended Practice for Software Design Descriptions IEEE Std 1016- 1998
- IEEE Recommended Practice for Software Requirements Specifications - IEEE Std 830-1998
- IEEE Standard for Software Test Documentation IEEE Std 829-1998
- IEEE Guide for Software Quality Assurance Planning - IEEE Std 730.1-1995
- http://www.asp.net/learn/data-access/tutorial-16-vb.aspx • www.gliffy.com
- 7http://msdn.microsoft.com/ens/library/system.security.permissions.security permissionattribute(VS.71).aspx • http://www.locmetrics.com/
- http://nces.ed.gov/nceskids/createAGraph/
- http://en.wikipedia.org/wiki/Load_testing
- http://en.wikipedia.org/wiki/Unit_test
- http://www.viveo-oolobject.com/savoirfaire/ecmfinance/concepts/usine/3tiers.php