## Padmabhushan Vasantdada Patil Pratisthan’s College of Engineering



A MINI-PROJECT REPORT

ON

**“BookShare - A Website To Sell Your Books.”**

Submitted by

Karthik Pillai VU4F1718038

Mohit Saini VU4F1718045

Satyam Yadav VU4F1718040

**Department of Information Technology**

**Padmabhushan Vasantdada Patil Pratisthan’s College of Engineering**

Sion-Chunabhatti Eastern Express Highway

Mumbai 400022

2019-2020

**Padmabhushan Vasantdada Patil Pratisthan’s College of Engineering**

Sion-Chunabhatti Eastern Express Highway

Mumbai 400022

**Department of Information Technology**



CERTIFICATE

Certified that the mini-project work entitled **“BookShare - A Website To Sell Your Books.”** is a bonafide workcarried out by

Karthik PillaiVU4F1718038

` Satyam YadavVU4F1718040

Mohit SainiVU4F1718045

The report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed for the course.

……………………….. ........................... ....………………

**Prof. Medha Kulkarni Dr. Seema Ladhe Dr. Alam Shaikh**

Mini-Project Guide Head of Department Principal

TABLE OF CONTENTS

1. Abstract 1
2. Introduction 2
3. Existing System 3
4. Proposed System 3
5. Cascading StyleSheet 4
6. Responsive Web Design 9
7. Ajax
8. Php
9. Login Module 12
10. Register Module 14
11. Database Connectivity 15
12. Screenshot 17
13. Conclusion 20
14. References 21

**Abstract**

To enable Students to sell their used Books to other Students at a lesser price as compared to local shops. This will help the Students to sell their Books at a better price and to also buy Books at a lesser price.

This project is designed to overcome the above stated problem such that it will enable a platform for the students to Sell or Buy Books through the means of Internet which is growing rapidly and making it easy to reach mass audience.

Purchasing Books from local vendors may become expensive for some students. Second hand Books are nearly equivalent to the new ones but at a cheaper price. Thus it will help the students to buy books at a cheaper price.

This website helps students to communicate to their Seniors/Juniors to Buy/Sell books.

There is user authentication to maintain user’s individuality. This website includes Whatsapp API which can redirect the user to WhatsApp to communicate further.

**Introduction**

In the world of software development there are lots of improvement in the area of Architectural design and principles. The philosophies and implementation details are changing as the people guiding the development of the application.

In this fantastic and yet sometimes complex world of software development there are some tried and true architecture patterns and software development guidelines employed by most architects. Also your design must have an ability to turn towards innovation instead of lending itself to common practices. Web services are one such area where architects must lean on their creative side and hope that their solutions are still successful. In this report we will explain an exciting voyage down the road of Web services application. From requirements to use cases, to database design, to component frameworks, to user interfaces, we will cover each and every aspect of system design required to build an application with collaborative Web services.

The reason why we selected online Bookstore web service is everybody walking down the street has some idea about bookstores. The objective of this project is to develop an e- book store where books can be bought from the comfort of home through the Internet.

Online Book Store is a form of [electronic commerce](https://en.wikipedia.org/wiki/Electronic_commerce) (e-commerce) website which allows students to directly buy books from a seller over the [Internet](https://en.wikipedia.org/wiki/Internet) using a [web browser](https://en.wikipedia.org/wiki/Web_browser). Students can find a book of their interest simply by visiting the website and can directly communicate with the seller and bargain on the price of the book and can also choose a convenient place to exchange their products.

The traditional method of buying and selling a book through a local bookstore has many disadvantages such as:

* It involves a third party which can be easily avoided in the online system.
* The local book stores pays only about 40-50% of the original price of the book to the students.
* They take around 70-75% of the same book when they resell that book.
* Hence, the take around 30-40% in the transaction of one book.
* The student who wants to buy or sell the book has to go to the book store which can be done more conveniently through the online book store.
* They cannot bargain much on the price.

All these drawbacks or disadvantages of the traditional book selling method can be overcome by the online book store.

**Existing System**

Students who wants to sell their used books goes to a book store and sell their book. But these local book stores pays very less to the students as compared to the original price of the book. They only get around 40-50% of the original price.

When a student wants to buy a second hand book, the local book stores charges them around 70-75% of the original price of the book. In this way, the middle man, that is the local bookstores gets a lot of profit by paying less to the students who wants to sell their books and by charging around 30-40% more for what they have paid. This tradition system is more favourable of the book stores than the students.

**Proposed System**

With the help of these systems, the local book stores will be removed from the system and the students will be able to buy and sell books more conveniently and easily without any involvement of the third party.

Students who want to buy and sell their old and used books can come together at a single platform. They can provide the details of their books and can easily communicate with each other and bargain on the price at which they want to buy or sell the books. This system helps the students to get the most of their resources without paying any money to the third party sellers like the local book stores.

**Advantages:**

* Faster buying/selling procedure, as well as easy to find books.
* Low operational costs and better quality of services.
* No need of physical bookstore set-ups.
* Customers can easily select books from different book providers without moving around physically.
* Low transaction cost.
* No intermediary.
* Wide reach , more reach to customers, there is no theoretical geographic limitations.
* Round the clock availability , buying/selling 24/7.

**Disadvantages**:

* There is no guarantee of book quality.
* Mechanical failures can cause unpredictable effects on the total processes.
* There are many hackers who look for opportunities, and thus a web site, service, payment gateways, all are always prone to attack.
* Payment made has no guarantee.
* Notorious for scams.
* No standardisation.

**Cascading Style Sheets (CSS)**

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS is used for styling the HTML elements.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

**CSS Types**:-

There are the following three types of CSS:

1. Inline CSS.
2. Internal CSS.
3. External CSS.

**Inline** **Css**:

For Inline CSS every style content is in HTML elements. It is used for a limited section. Whenever our requirements are very small we can use inline CSS.

It will affect only single elements. In HTML we require that various HTML tag's views are different so then we use inline Cascading Style Sheets. There are disadvantages of inline Cascading Style Sheets. It must be specified on every HTML tag. There is very much time consumed by that and it is not the best practice for a good programmer and the code will be quite large and very complex.

Example:

<h1 style="color:blue;text-align:center;">This is a heading</h1>

<p style="color:red;">This is a paragraph.</p>

**Internal Css**:-

In internal CSS the style of CSS is specified in the <head> section. This is internal CSS, it affects all the elements in the body section. Internal CSS is used in the condition when we want a style to be used in the complete HTML body. For that we can use style in the head tag.

Example:

<head>

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

<style>

h1 {

color: orange;

}

</style>

</head>

**External Css:-**

In External CSS we create a .css file and use it in our HTML page as per our requirements. Generally external Cascading Style Sheets are used whenever we have many of HTML attributes and we can use them as required; there is no need to rewrite the CSS style again and again in a complete body of HTML that inherits the property of the CSS file. There are two ways to create a CSS file. The first is to write the CSS code in Notepad and save it as a css file, the second one is to directly add the stylesheet in our Solution Explorer and direct to use it on our HTML page.

HTML:

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

CSS: mystyle.css

body {

background-color: lightblue;

}

h1 {

color: navy;

margin-left: 20px;

}

**CSS**

**login.css**

body{

margin: 0;

padding: 0;

font-family: sans-serif;

background: url(assets/LoginRegisterBook.jpeg) no-repeat;

background-size: cover;

}

::-webkit-input-placeholder {

color: rgba(255,255,255,0.7);

}

h1{

font-size: 40px;

border-bottom: 6px solid #3cff00;

margin-bottom: 35px;

padding: 13px 0;

}

label{

font-weight: bold;

}

.login-box{

max-width: 280px;

margin: 110px auto;

color: white;

background: none;

border:2px solid #3cff00;

padding: 40px;

border-radius: 20px;

}

.textbox{

width: 100%;

overflow: hidden;

font-size: 20px;

padding: 8px 0;

margin: 8px 0;

border-bottom: 1px solid #3cff00;

}

.textbox input{

border: none;

outline: none;

background: none;

color: white;

font-size: 18px;

width: 65%;

float: left;

margin: 0 10px;

}

.btn{

width: 100%;

border: 2px solid #3cff00;

padding: 5px;

font-size: 18px;

/\*cursor: pointer;\*/

background: #3cff00;

border-radius: 30px;

color: black;

font-weight: bold;

}

#error{

background: #ff4046;

border:1px solid white;

width: 280px;

border-radius: 5px;

margin: 0 auto;

color: white;

font-weight: bold;

text-align: center;

}

#registerBtn{

margin-top: 15px;

width: 30%;

font-size: 12px;

color: black;

background: white;

margin: 15px auto;

}

#anchorSignup{

display: flex;

align-items: center;

justify-content: center;

}

**register.css**

body{

margin: 0;

padding: 0;

font-family: sans-serif;

background: url(assets/LoginRegisterBook.jpeg) no-repeat;

background-size: cover;

}

::-webkit-input-placeholder {

color: rgba(255,255,255,0.8 );

}

.reg-box{

max-width: 300px;

margin:75px auto;

color: white;

background: none;

border-radius: 20px;

border:2px solid #3cff00;

padding: 40px;

}

h1{

font-size: 40px;

border-bottom: 6px solid #3cff00;

margin-bottom: 35px;

padding: 13px 0;

}

.textbox{

width: 100%;

overflow: hidden;

font-size: 20px;

padding: 8px 0;

margin: 8px 0;

border-bottom: 1px solid #3cff00;

}

.textbox input{

border: none;

outline: none;

background: none;

color: white;

font-size: 18px;

width: 65%;

/\*float: left;\*/

margin: 0 10px;

}

.Rbtn{

font-size: 18px;

overflow: hidden;

background: none;

margin: 12px 0;

border-bottom: 1px solid #3cff00;

padding: 8px;

margin-bottom: 0;

}

.btn{

width: 100%;

border: 2px solid #3cff00;

padding: 5px;

font-size: 18px;

/\*cursor: pointer;\*/

background: #3cff00;

border-radius: 30px;

color: white;

font-weight: bold;

}

#error{

background: #ff4046;

border:1px solid white;

width: 280px;

border-radius: 5px;

margin: 0 auto;

color: white;

font-weight: bold;

text-align: center;

}

**index.css**

.btn-outline-success{

margin: 0px 10px;

color: white;

border: 1px solid white;

}

.secondnav{

margin-top: 56px;

}

.secondNavItems{

font-weight: bold;

margin-right: 40px;

}

.movingImg{

max-height: 250px;

}

.card{

margin: 10px;

}

#ourproducts{

text-align: center;

margin: 5px;

margin-top: 10px;

text-decoration: underline;

}

.navbar-text{

font-weight: bold;

margin-right: 15px;

}

.btn-danger{

background: #343a40;

}

#viewProducts{

font-weight: bold;

}

**RESPONSIVE WEB DESIGN (RWD)**

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones).

Responsive web design (RWD) is an approach in web development and design that is geared toward creating sites that provide an efficient and appealing visual experience with text that is easy to navigate without altering the resolution of a display screen.

This capability extends to any device or browser used to view a website, which means the website appearance and layout change according to the size of the display screen.

**RWD uses the following methods**:

* Fluid grids, where elements are resized through the use of relative units, like percentages, versus absolute units, like standard measurements and pixels
* Flexible images, which are also sized in relative units.
* Cascading Style Sheet (CSS) media queries, which allow a website to determine device type, screen size and browser capabilities, enabling the delivery of different style rules based on these characteristics.
* Server-side components with media queries, which enable faster loading websites – even with slower cellular data speeds.

**Code:**

.col-lg-4 .col-md-6 .col-sm-12{

position: relative;

width: 100%;

padding-right: 15px;

padding-left: 15px;

}

.col-lg-4 {

-ms-flex: 0 0 33.333333%;

flex: 0 0 33.333333%;

max-width: 33.333333%;

}

.col-md-6 {

-ms-flex: 0 0 50%;

flex: 0 0 50%;

max-width: 50%;

}

.col-sm-12 {

-ms-flex: 0 0 100%;

flex: 0 0 100%;

max-width: 100%; }

.row {

display: -ms-flexbox;

display: flex;

-ms-flex-wrap: wrap;

flex-wrap: wrap;

margin-right: -15px;

margin-left: -15px;

}

**PHP**

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP.

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

PHP is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP:

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

Five important characteristics make PHP's practical nature possible −

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

Code:

<?php

$user='root';

$pass='';

$db='bookshare';

$link=mysqli\_connect('localhost',$user,$pass,$db);

if(mysqli\_connect\_error()){

echo "There was an error connecting to DB!";

}else{

//connected to db successfully

if(isset($\_GET['branch'])){

// echo $\_GET['branch'];

$query="SELECT \* FROM books WHERE branch='".$\_GET['branch']."' ORDER BY bookid DESC";

}else{

$query="SELECT \* FROM books ORDER BY bookid DESC";

}

$result=mysqli\_query($link,$query);

while ($row=mysqli\_fetch\_array($result)) {

$bookid=$row['bookid'];

echo "

<div class=\"col-lg-3 col-md-4 col-sm-6\">

<div class=\"card\">

<img class=\"card-img-top\" src='images/".$row['image']."' alt=\"Card image cap\">

<div class=\"card-body\">

<h5 class=\"card-title\">".$row['title']."</h5>

<h6 class=\"card-title\">Rs.".$row['price']."</h6>

<p class=\"card-text\">".$row['description']."</p>

<a href=\"showitem.php?bookid=$bookid\" class=\"btn btn-primary btn-lg btn-block\">BUY</a>

</div>

</div>

</div>";

}

}

?>

**Login Module**

<?php

session\_start();

$user='root';

$pass='';

$db='bookshare';

$link=mysqli\_connect('localhost',$user,$pass,$db);

$error=null;

if(mysqli\_connect\_error()){

$error="There was an error connecting to DB!";

}else{

if(array\_key\_exists("email", $\_POST) AND array\_key\_exists("password", $\_POST)){

//page is submitted

if($\_POST["email"]=="" OR $\_POST["password"]==""){

$error="Please enter everything!";

}else{

//USER HAS ENTERED EVERYTHING

$query="SELECT \* FROM users WHERE email='".$\_POST['email']."'"." AND password='".$\_POST['password']."'";

$result=mysqli\_query($link,$query);//IF SUCH USER EXISTS HE WILL BE STORED IN $RESULT

if(mysqli\_num\_rows($result)>0){//USER EXISTS WITH GIVE EMAIL AND PASSWORD

//STORING SESSION VARIABLES AND REDIRECTING TO productpage.php

$row=mysqli\_fetch\_array($result);

$\_SESSION['name']=$row['name'];

$\_SESSION['id']=$row['id'];

$\_SESSION['email']=$row['email'];

$\_SESSION['gender']=$row['gender'];

header("Location: index.php");

}else{

$error="Please enter correct Email or Password!";//USER DOES NOT EXIST

}

}

}

}

?>

**Login Form:**

<form method="post"> <h1>Login</h1>

<div class="textbox"> <label>

Email<br>

<input type="Email" placeholder="abc@bookshare.com" name="email"> </label>

</div>

<div class="textbox"> <label>

Password<br>

<input type="password" placeholder="\*\*\*\*\*\*\*\*" name="password" value=""> </label>

</div>

<input class="btn" type="submit" name="" value="Sign-in"> </form>

**Register Module**

<?php

session\_start();

$user='root';

$pass='';

$db='bookshare';

$link=mysqli\_connect('localhost',$user,$pass,$db);

$error=null;

if(mysqli\_connect\_error()){

$error="There was an error connecting to DB!";

}else{

//SUCCESSFULLY CONNECTED TO DB

if(array\_key\_exists("name", $\_POST) AND array\_key\_exists("email", $\_POST) AND array\_key\_exists("password", $\_POST) AND array\_key\_exists("cpassword", $\_POST) AND array\_key\_exists("gender", $\_POST)){

//user has entered everything or blank page is submitted

if($\_POST['email']=="" OR $\_POST['password']=="" OR $\_POST['cpassword']=="" OR $\_POST['name']=="" OR $\_POST['gender']==""){

$error="Please enter all the fields!";

}else if($\_POST['password']!=$\_POST['cpassword']){//TO CHECK IF PASSWORD AND CONFIRM PASSWORD MATCH

$error="Your passwords should be same!";

}else{

//user has entered everything

$query="SELECT \* FROM users WHERE email='".mysqli\_real\_escape\_string($link,$\_POST['email'])."'";

$result=mysqli\_query($link,$query);//query to select users with the entered email

if(mysqli\_num\_rows($result)>0){

$error="Unfortunately Email is already taken!";//cant keep 2 or more same email ids

}else{

//EVERYTHING IS VALID, ADD THE USER INTO THE DATABASE

$query="INSERT INTO users (name,email,password,gender) VALUES ('".$\_POST['name']."','".$\_POST['email']."','".$\_POST['password']."','".$\_POST['gender']."')";//\*\*\*\*\*\*\*\*\*SQL PROTECTION REQUIRED\*\*\*\*\*\*\*

if(mysqli\_query($link,$query)){

// echo "<p>YOU HAVE BEEN SIGNED UP</p>";

header("Location: login.php");

}else{

$error="There was a problem signing you up!";

}

}

}

}

}?>

**Database Connectivity**

<?php

$user='root';

$pass='';

$db='bookshare';

$link=mysqli\_connect('localhost',$user,$pass,$db);

if(mysqli\_connect\_error()){

$error="There was an error connecting to DB!";

}else{

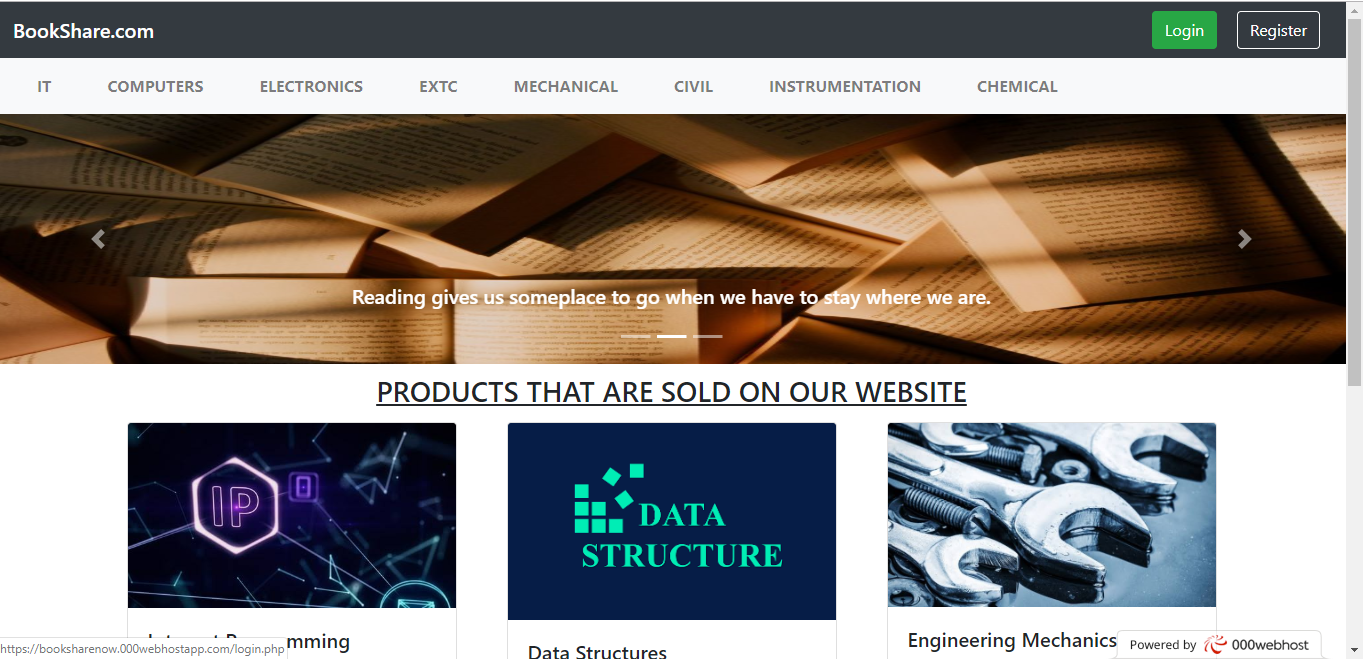
//PERFORM DATABASE OPERATIONS

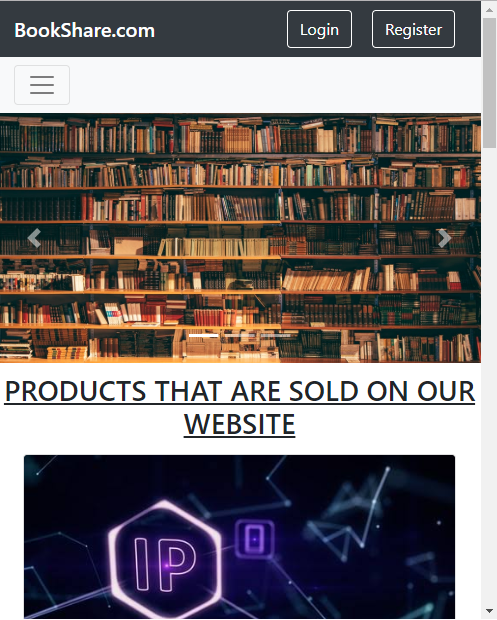
}

?>

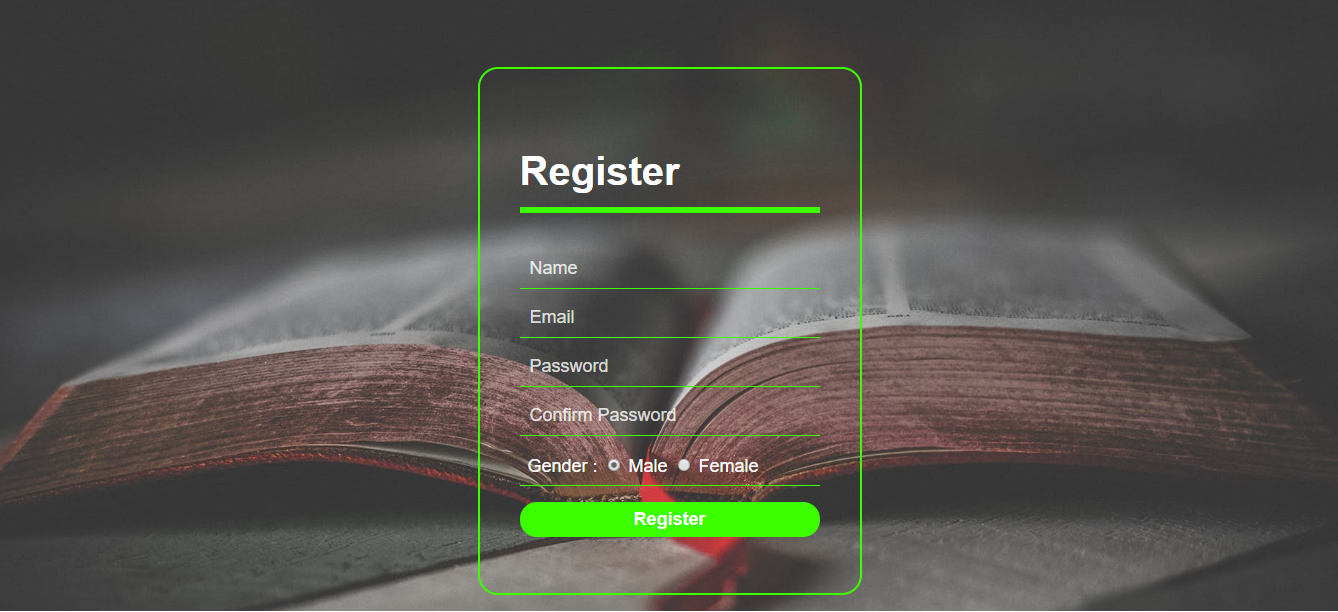
**SCREENSHOTS**

Landing Page - index.php

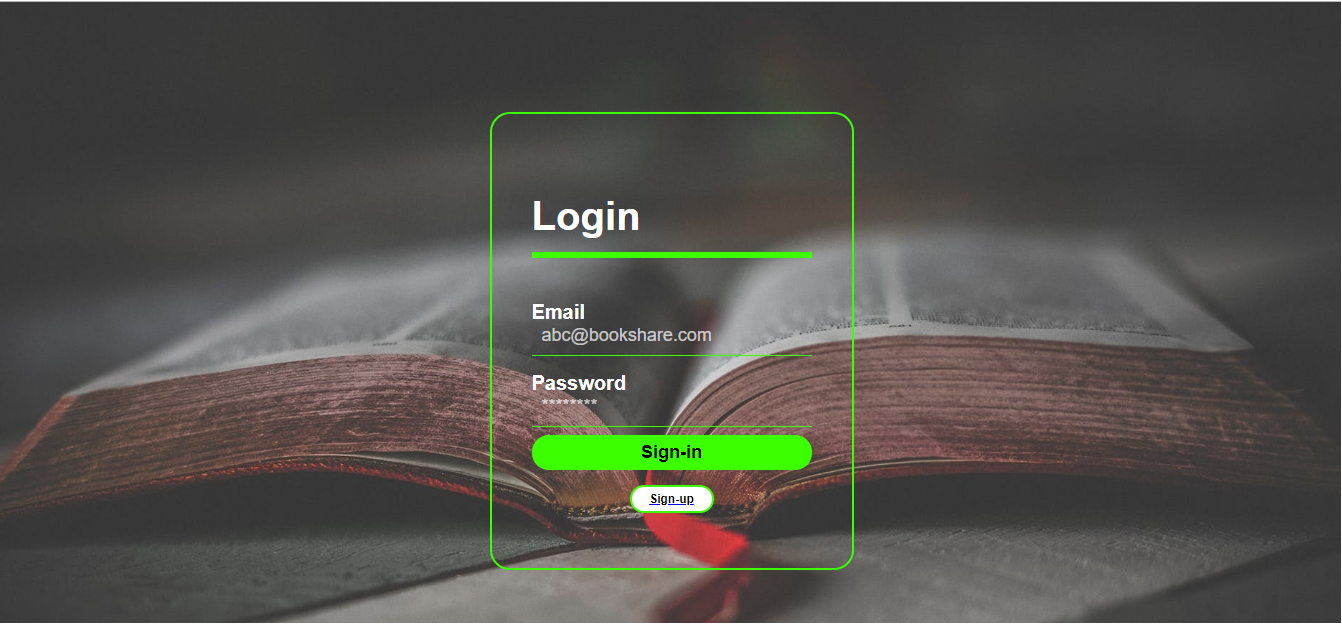




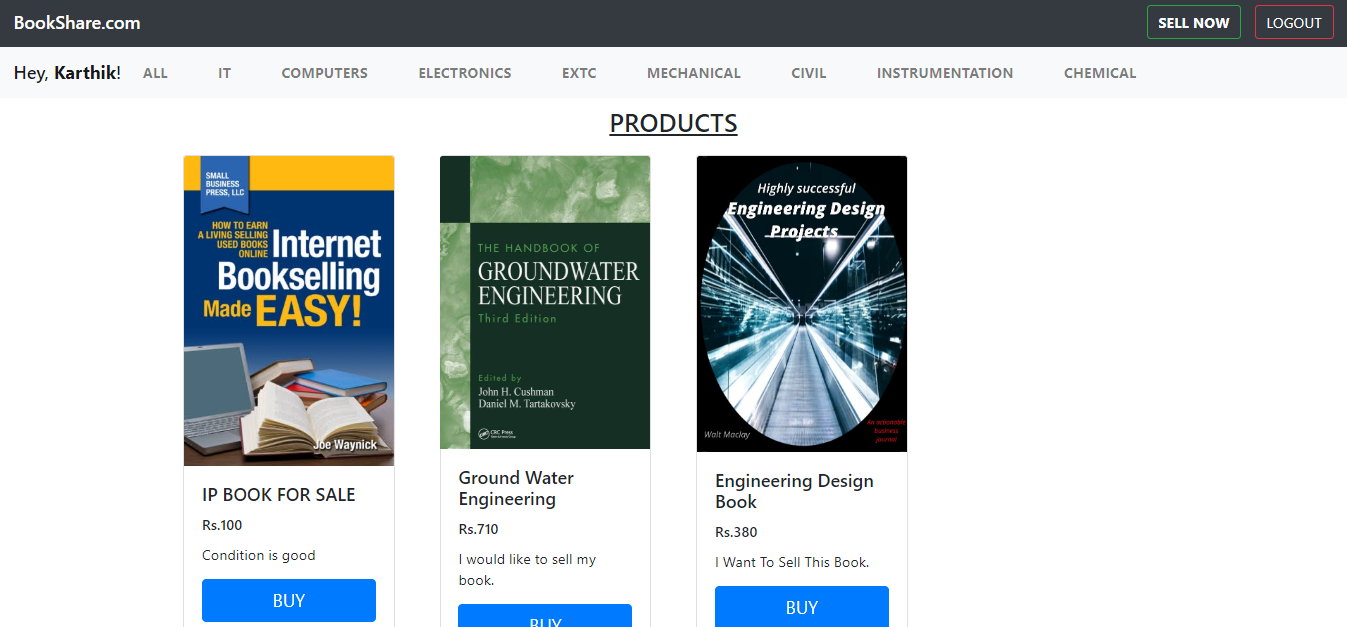
Registration Page - register.php



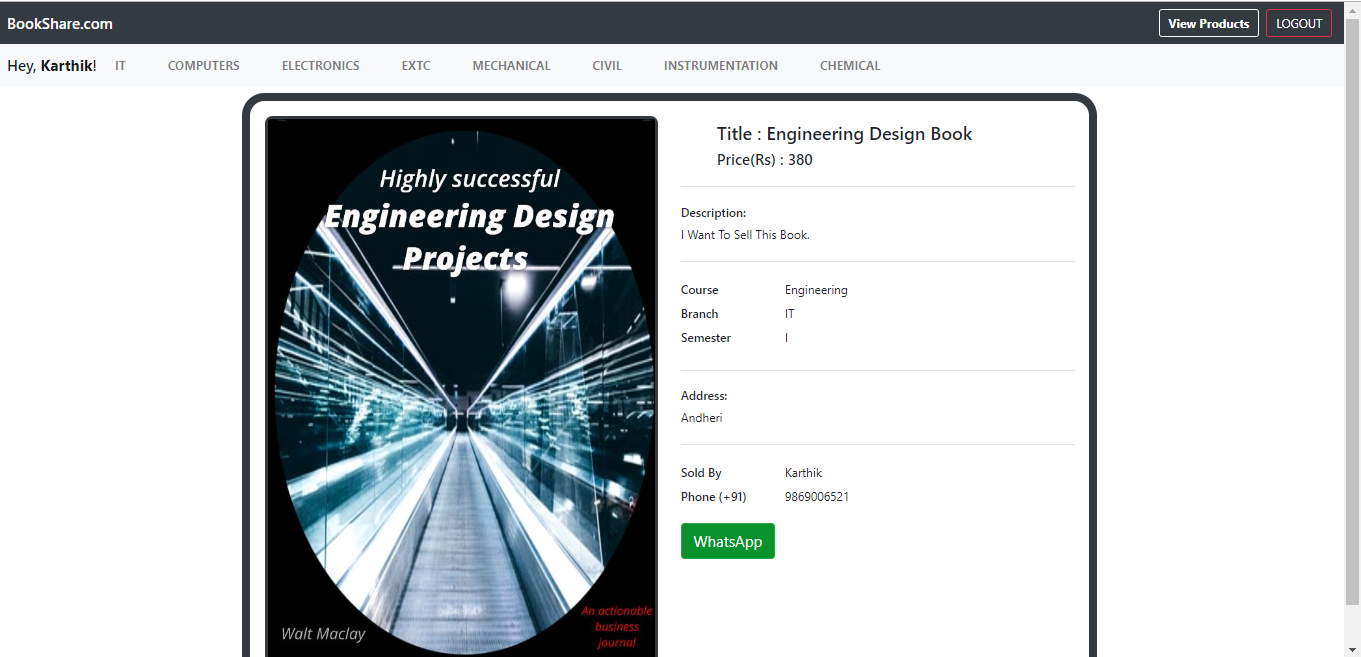
Login Page - login.php



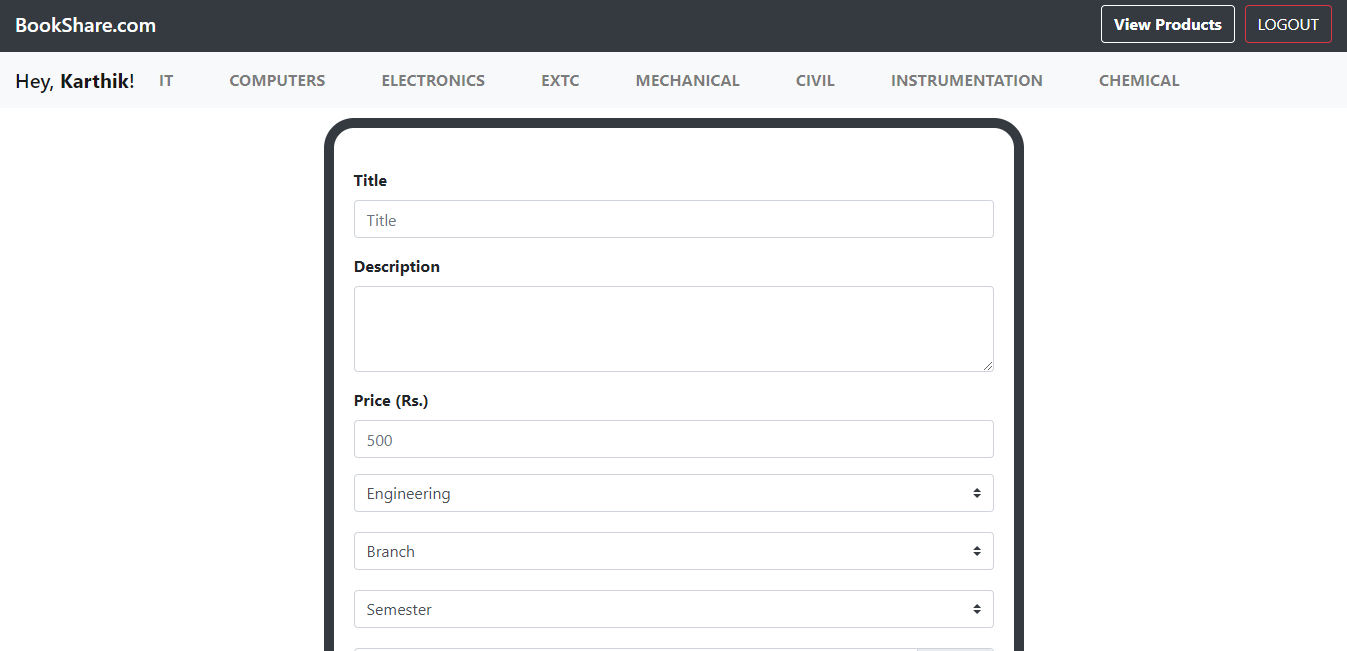
Product Page - productpage.php

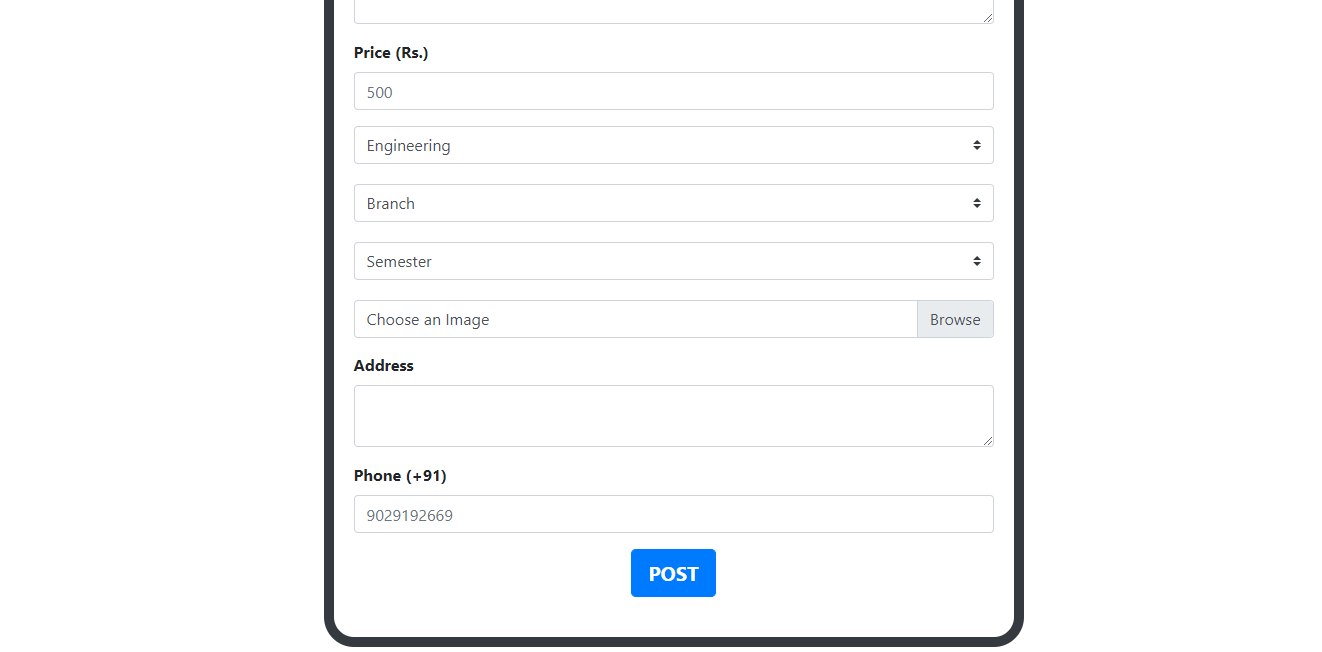


Individual Item Details - showitem.php



Selling Page - additempage.php





**Conclusion**

Online buying and selling is continuously progressing and is becoming more and more important to user as technology continues to advance and is something that should be taken advantage of and implemented.From the inception of the Internet and e-commerce, the possibilities have become endless for both buyers and sellers. However, just like anything else, e-commerce has its disadvantages including consumer uncertainties, but that can be resolved or avoided by good decision-making and business practices.

**References**

* [https://](https://www.sublimetext.com/)[www.google.com](http://www.google.com)
* [https://](https://www.sublimetext.com/)[www.w3school.com](http://www.w3school.com)
* [https://](https://www.sublimetext.com/)[www.apachefriends.org/download.html](http://www.apachefriends.org/download.html)
* [https://](https://www.sublimetext.com/)[www.getbootstrap.com](http://www.getbootstrap.com)
* [https://www.sublimetext.com](https://www.sublimetext.com/)
* [https://stackoverflow.com](https://stackoverflow.com/)