

Project Report



E-commerce Bookstore website

Abhayjeet Kumar



Table of Contents

S.No	Contents	Page No
1.	Introduction	3
2.	Purposes and Goals	4
3	Website Portals	5
4.	User Experience	8
5.	Technology Stack	9
6.	File Structure	11
7.	Database Schema	14
8.	Importance Routes	19
9.	Important Server Side Codes	20
10.	User Interfaces	22
11.	Authentication and Authorization	31
12.	Future Updates	35
13.	Conclusion	36
14.	Source Code and Contact Designer	37

Project Overview

Introduction

Abhay-eCommerce website, an online platform dedicated to connecting book enthusiasts and sellers in a seamless and secure environment. This project leverages Node.js, Pug, MySQL, and various other technologies to create a dynamic and user-friendly eCommerce experience for both buyers and sellers of books. Buyers can browse, purchase books, and track their orders, while sellers can list their books for sale, purchase books, and track their own orders as well.

Purpose and Goals

The primary purpose of this project is to provide an intuitive and feature-rich platform for book enthusiasts to discover, purchase, and explore a wide range of books. Additionally, it offers a convenient space for sellers to list their books for sale. The project's key goals include:


- **User-Friendly Interface:** To deliver an easy-to-navigate website that allows users to effortlessly browse and interact with books.
- **Authentication:** Implement a secure user authentication system, ensuring that only registered buyers and sellers can access certain features.
- **Buyer Features:** Enable buyers to explore the catalog of books, view book details, and make purchases.
- **Seller Features:** Provide sellers with a portal to add and manage books for sale, track orders, and interact with buyers.
- **Robust Database:** Develop a reliable database system (MySQL) to store book details, user information, orders, and transaction records.
- **Security:** Ensure data security and implement measures to protect user information, transactions, and system integrity.
- **Scalability:** Design the project with scalability in mind, allowing for future expansion and optimization.
- **Documentation:** Create comprehensive project documentation to aid in project maintenance, future development, and knowledge transfer.

Project Components

Buyer Portal

The Buyer Portal allows registered users to:

The screenshot shows the 'My Account' section of the Buyer Portal. On the left, there is a sidebar with a 'My Account' header and four buttons: 'My Details', 'My Location', 'My Orders', and 'My Accounts'. The main content area displays the user's profile information. At the top right of the main area is a green 'Profile' button. Below this is a large image of a cup of coffee with the text 'Daily Stories Thursday Hey' and a small profile picture. Below the image, the user's name 'Kamaljeet Roy' and date of birth 'Thu Jul 10 1997 00:00:00 GMT-0600 (Central Standard Time)' are displayed.

My Account	
My Details	 Name Kamaljeet Roy DOB Thu Jul 10 1997 00:00:00 GMT-0600 (Central Standard Time)
My Location	
My Orders	
My Accounts	

- Browse a vast collection of books organized by categories, authors, and publishers.
- View detailed information about each book, including price, name, category, author, publisher, and more.
- Add books to their shopping cart.
- Place orders and complete transactions securely.
- Manage their profile and order history.

The screenshot shows the 'My Orders' section of the Buyer Portal. On the left, there is a sidebar with a 'My Account' header and four buttons: 'My Details', 'My Location', 'My Orders', and 'My Accounts'. The main content area displays the user's order history. At the top right of the main area is a green 'Profile' button. Below this is a table titled 'My Order Description' with columns: 'Order Id', 'Order Date', 'Name', 'Price', 'Quantity', and 'Receipt'. The table contains one order with the following details: Order Id: 8781d746-a501-4f38-b3c8-80f2be0359aa, Order Date: Sun Aug 06 2023 08:28:12 GMT-0600 (Central Standard Time), Name: DBMS, Price: 565, Quantity: 65, and Receipt: view.

My Account													
My Details	My Order Description <table border="1"><thead><tr><th>Order Id</th><th>Order Date</th><th>Name</th><th>Price</th><th>Quantity</th><th>Receipt</th></tr></thead><tbody><tr><td>8781d746-a501-4f38-b3c8-80f2be0359aa</td><td>Sun Aug 06 2023 08:28:12 GMT-0600 (Central Standard Time)</td><td>DBMS</td><td>565</td><td>65</td><td>view</td></tr></tbody></table>	Order Id	Order Date	Name	Price	Quantity	Receipt	8781d746-a501-4f38-b3c8-80f2be0359aa	Sun Aug 06 2023 08:28:12 GMT-0600 (Central Standard Time)	DBMS	565	65	view
Order Id		Order Date	Name	Price	Quantity	Receipt							
8781d746-a501-4f38-b3c8-80f2be0359aa		Sun Aug 06 2023 08:28:12 GMT-0600 (Central Standard Time)	DBMS	565	65	view							
My Location													
My Orders													
My Accounts													

Seller Portal

The Seller Portal offers registered sellers the following functionalities:

The screenshot shows the 'Admin' section of the Seller Portal. On the left, there's a sidebar with 'My Account' and a list of links: 'My Details', 'My Location', 'My Orders', and 'My Accounts'. The main content area displays the user's profile for 'Abhayjeet Roy'. It includes a profile picture of a cup of coffee with the text 'Daily Stories Thursday Hey' and a small circular icon. Below the image, the profile details are listed: Name (Abhayjeet Roy), DOB (Thu Aug 21 2003 00:00:00 GMT-0600 (Central Standard Time)), and Location (null, null, null, null, null, null). At the top right of the main content area, there are buttons for 'Profile', 'Insert', and 'View items'. The browser's address bar shows the URL 'localhost:5001/profile/4a76b701-374e-437b-bcf9-5e8af31e5c45'.


- List books for sale, providing details such as price, name, category, author, publisher, etc.
- Manage their inventory, including adding, editing, and removing books.
- Track and process orders from buyers.
- Maintain a seller profile.

The screenshot shows the 'Add Item' form in the Seller Portal. The form is titled 'Item Name:' and has a text input field with the placeholder 'Enter Item Name'. Below this is a 'Category' dropdown menu with 'None' selected, followed by a text input field with the placeholder 'Enter Category'. The form continues with fields for 'Author:' (placeholder 'Enter Author'), 'Publisher:' (placeholder 'Enter Publisher'), 'Price:' (placeholder 'Enter Price'), and 'Quantity:' (placeholder 'Enter Quantity'). At the bottom, there is an 'Image:' section with a 'Choose File' button and the text 'No file chosen'. A large blue 'ADD' button is at the bottom center. At the top right of the form, there are buttons for 'Profile', 'Insert', and 'View items'. The browser's address bar shows the URL 'localhost:5001/profile/4a76b701-374e-437b-bcf9-5e8af31e5c45'.

Admin Portal

This Portal is for admin.

Here Admin manage Listed Books or Unlisted Books.

 **Abhay**

Search....


Abhay Home Categories▼ Products About

Admin Portal

Listed

Name	Seller	Price
c++	Abhayjeet Roy	954
Computer Networks	Abhayjeet Roy	345
Operating system	Abhayjeet Roy	563
Modern History	Abhayjeet Roy	654

When Seller add book to store, there is need of admin confirmation to add to sell. Admin decline selling of that book.

 **Abhay**

Search....

Abhay Home Categories▼ Products About

Admin Portal

Un - Listed

Name	Seller	Role	Price	Confirm
C#	Monu Kumar	Admin	564	<input type="button" value="Remove"/> <input type="button" value="Add"/>

 HelpDesk@gmail.com  123456789

User Experience

The project emphasizes a smooth and engaging user experience. Users can view book details without requiring login, fostering accessibility for potential buyers. However, to purchase books or add books for sale, users must log in, ensuring secure transactions and streamlined interactions.

Our eCommerce platform aims to provide a safe and enjoyable experience for all users while supporting the book industry by facilitating book discovery and trade.

Technology Stack

The Following Dependencies are used to build the project.

```
"dependencies": {  
  "bcrypt": "^5.1.0",  
  "body-parser": "^1.20.2",  
  "dotenv": "^16.3.1",  
  "express": "^4.18.2",  
  "express-async-handler": "^1.2.0",  
  "express-session": "^1.17.3",  
  "jsonwebtoken": "^9.0.1",  
  "multer": "^1.4.5-lts.1",  
  "mysql": "^2.18.1",  
  "nodemailer": "^6.9.5",  
  "otp-generator": "^4.0.1",  
  "pug": "^3.0.2",  
  "twilio": "^4.17.0",  
  "uuid": "^9.0.0",  
  "uuidv4": "^6.2.13"  
}
```

Server-Side Development:

- **Node.js:** A JavaScript runtime environment used for server-side development.
- **Express.js:** A web application framework for Node.js, used to build the backend of the eCommerce website.
- **MySQL:** A relational database management system (RDBMS) used for storing and managing data.
- **Express-Session:** Middleware for handling session management in Express.js.
- **Multer:** Middleware for handling file uploads in Express.js.
- **Dotenv:** A module for loading environment variables from a .env file.
- **Express-Async-Handler:** A utility for handling asynchronous errors in Express.js routes.
- **Bcrypt:** A library for hashing and salting user passwords securely.
- **JSON Web Tokens (JWT):** Used for user authentication and authorization.
- **OTP Generator:** A library for generating one-time passwords for authentication.
- **Nodemailer:** A module for sending email notifications and verification emails.
- **Twilio:** A cloud communications platform used for sending SMS notifications.
- **UUID and UUIDv4:** Libraries for generating universally unique identifiers for various purposes.
- **body-parser:**

Front-End Development:

- **Pug (formerly Jade):** A template engine for rendering dynamic HTML content.
- **CSS/Stylesheets:** Custom styles or CSS frameworks (e.g., Bootstrap) for styling the website.
- **JavaScript:** Client-side scripting for interactivity and dynamic behavior.

Database:

- **MySQL:** As the primary relational database for storing product information, user data, orders, and more.

Security:

- **BCrypt:** Used for securely hashing and salting user passwords to protect user data.
- **JSON Web Tokens (JWT):** Used for secure authentication and authorization.
- **Session Management:** Handled using Express-Session for user sessions.

File Uploads:

- **Multer:** Used to handle and manage file uploads, such as product images/user Profile Images..

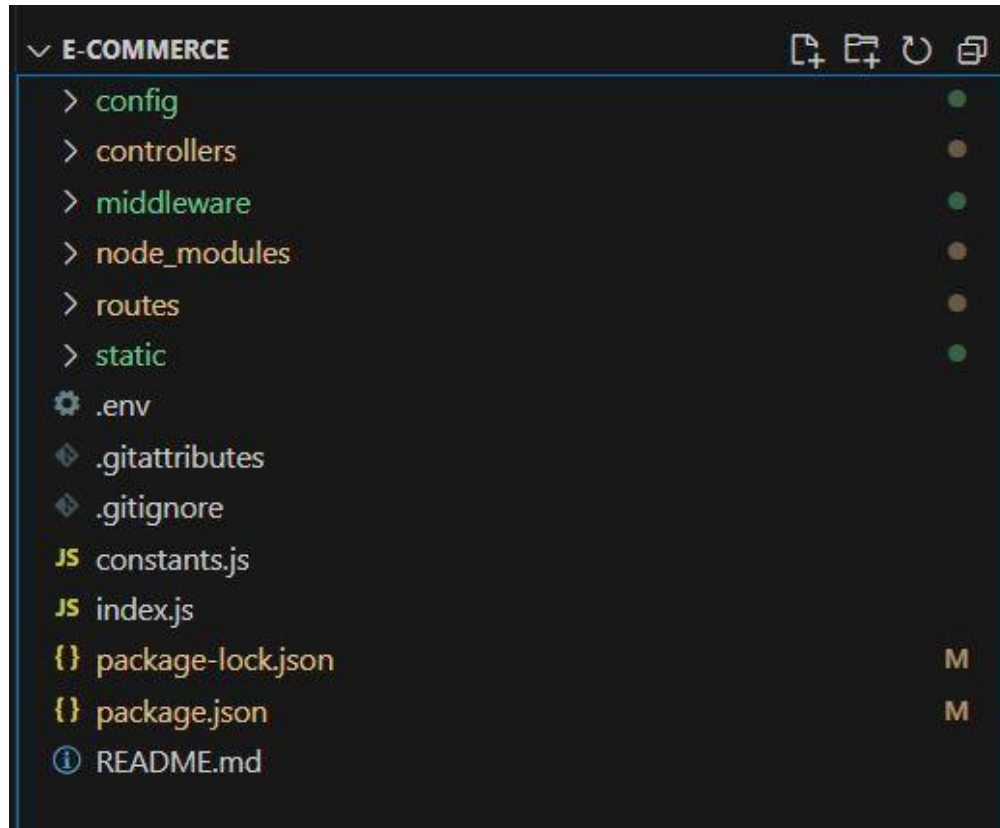
Email and SMS Integration:

- **Nodemailer:** Used for sending email notifications, such as order confirmations.
- **Twilio:** Used for sending SMS notifications, such as OTPs or order updates.

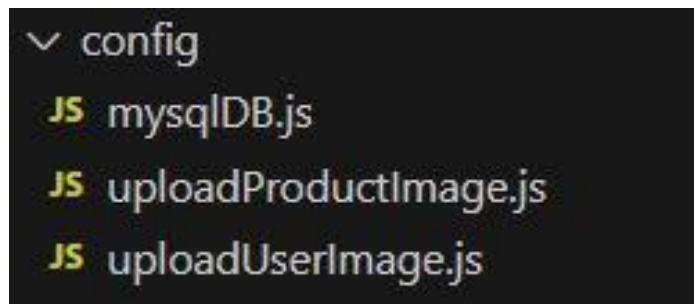
Project File Structure

Source-Code : <https://github.com/21-abhay/E-commerce>

E-COMMERCE File :



Config :



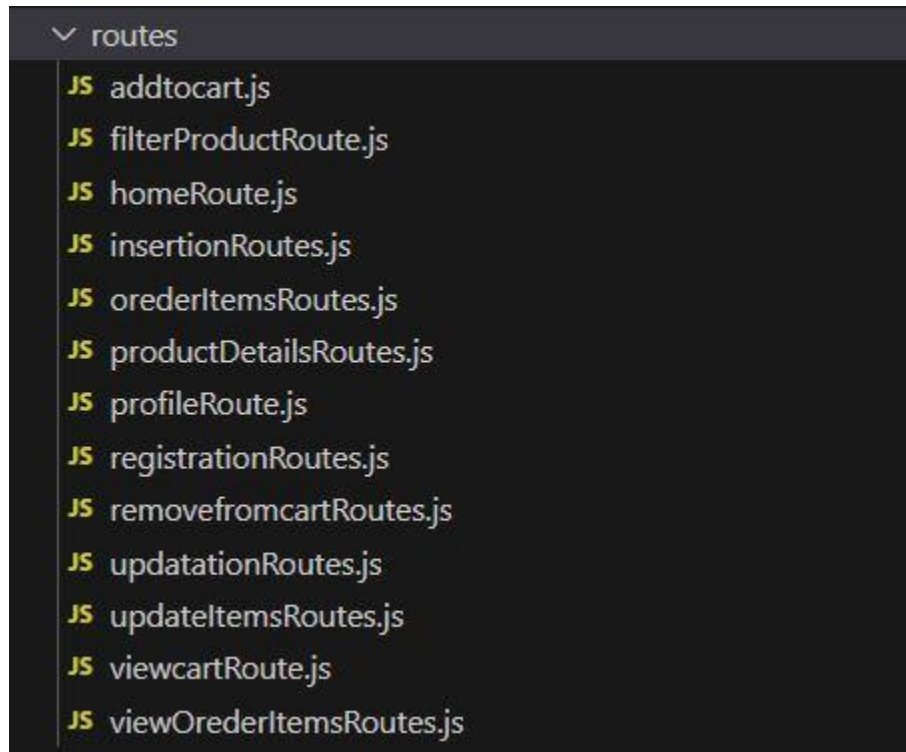
Controlles :

```
▼ controllers
  JS about.js
  JS addtocart.js
  JS filterProductController.js
  JS home.js
  JS insertController.js
  JS loginController.js
  JS logoutController.js
  JS orderController.js
  JS product.js
  JS productDetailsController.js
  JS profileControllers.js
  JS removefromcart.js
  JS signupController.js
  JS updateAccountController.js
  JS updateDetailsController.js
  JS updateLocationController.js
  JS viewcart.js
  JS viewOrderDetails.js
```

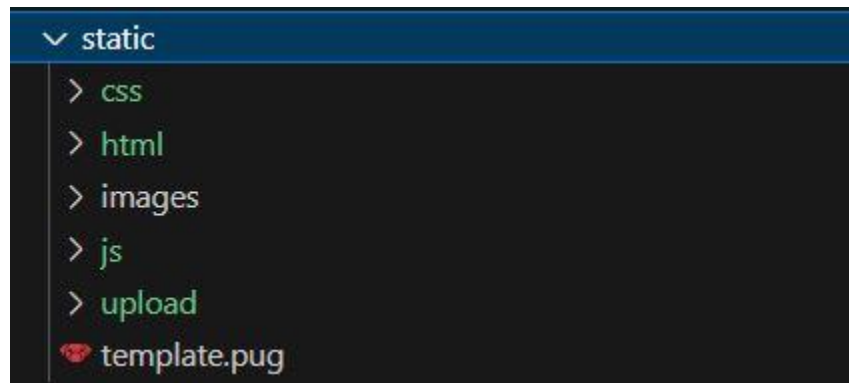
Middleware :

```
▼ middleware
  JS bcrypt.js
  JS otpGenerate.js
  JS sessionHandler.js
```

Routes :



Static :

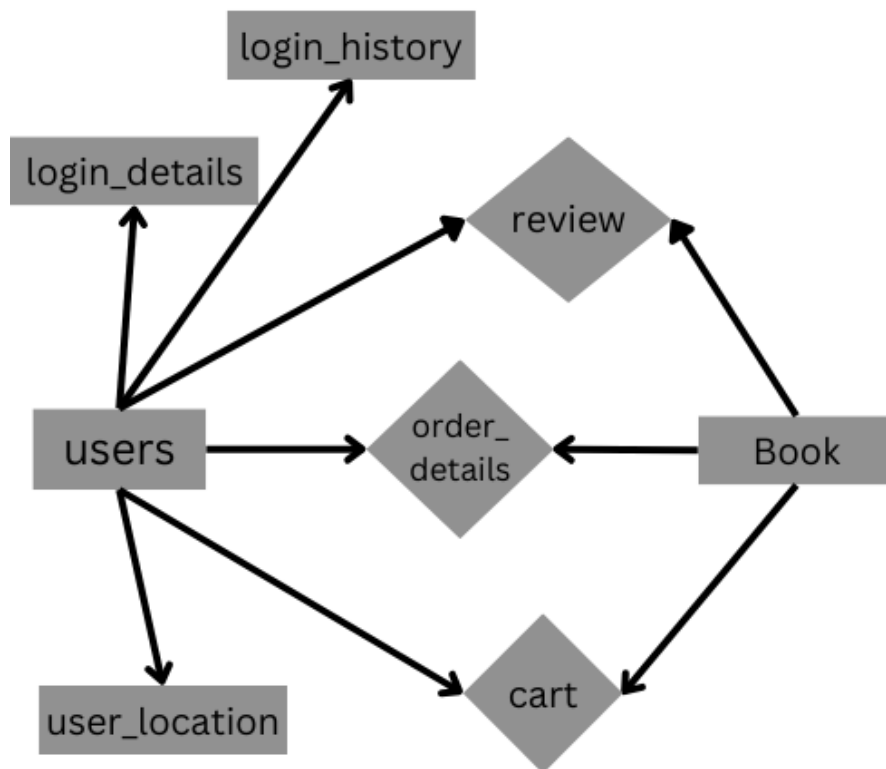


DataBase Schema

MySQL database is used to store user and book data along with login details

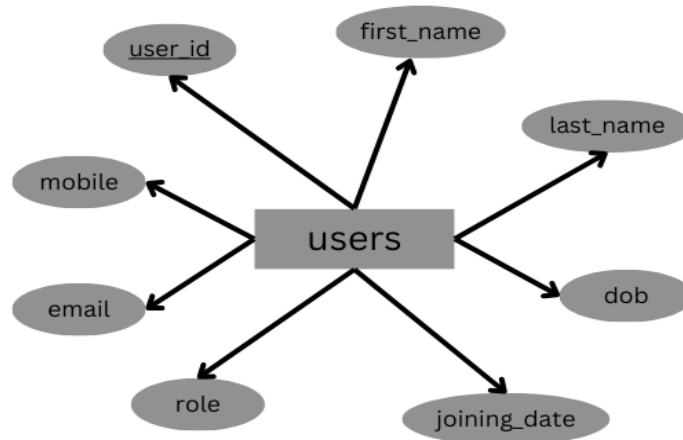
DataBase Tables

1. users
2. book
3. order_details
4. review
5. User_location
6. cart
7. login_details
8. login_history



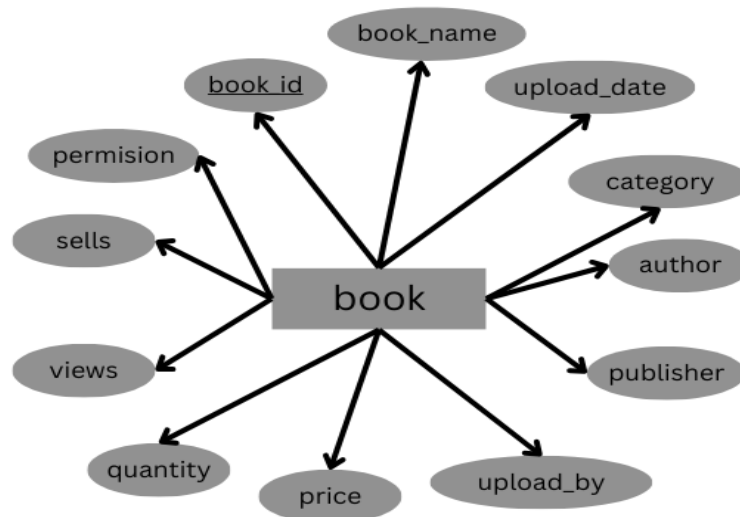
users :

'users' Table store info about user who login to buy or sell book by taking their following info.



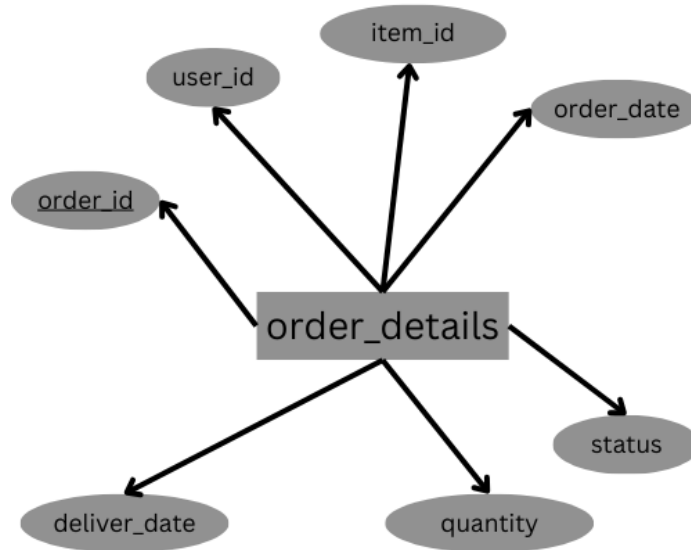
book :

'book' Table store info about books available to sell.it store the following data.



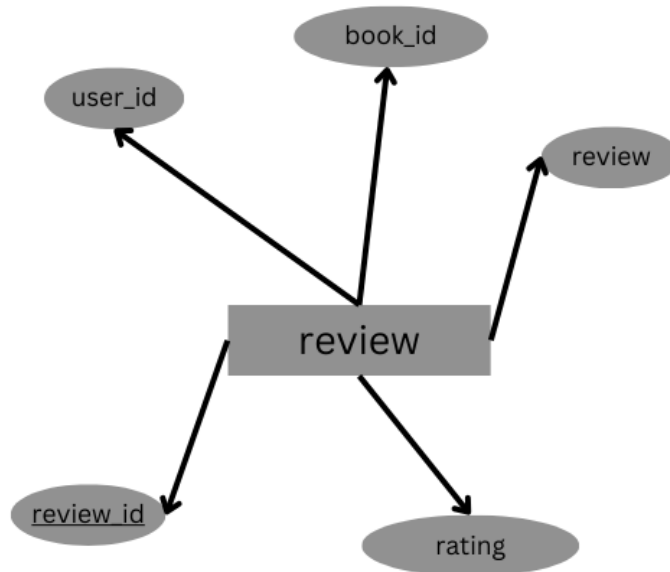
order_details

'order_details' Table store info about order book and who's order it.



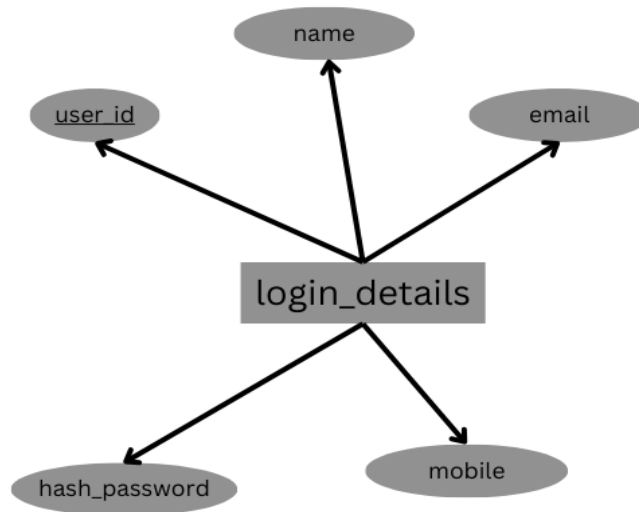
review :

'review' Table Stores reviews given by users on books.



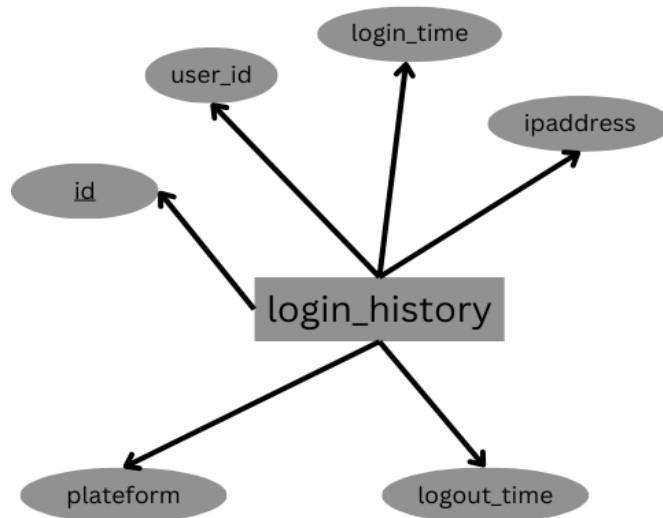
login_details :

'login_details' Table stores login info user by they can login to website.



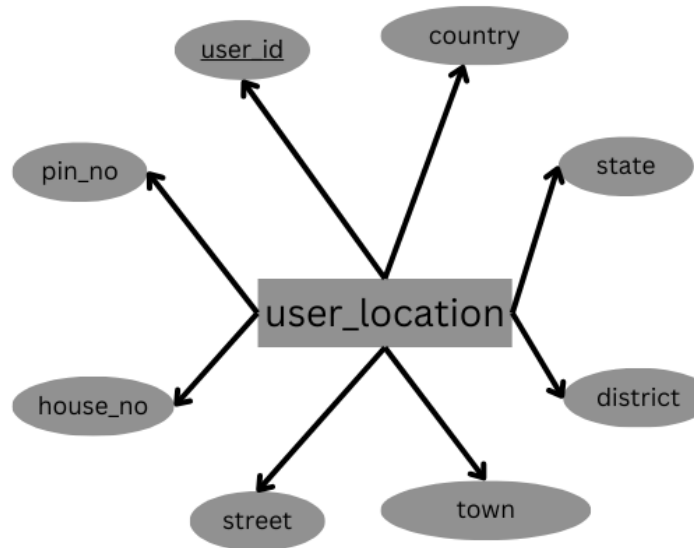
login_history :

'login_history' Table stores info about when does user login to account and from which device and browser.



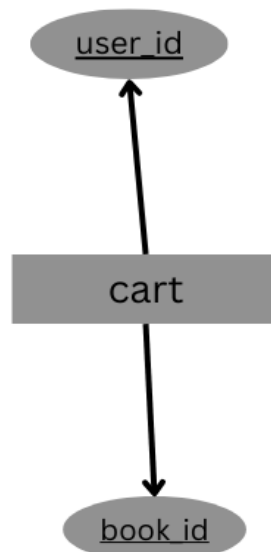
user_location :

'user_location' Table stores users location info.



Cart :

It store the book Id that user want to buy now or in near future.



Important Routes

The Followings are important route in this website :

1. **['/', '/home']**
2. **'/about'**
3. **'/product'**
4. **'/login'**
5. **'/singup'**
6. **'/product/product_Id'**
7. **'/viewcart/view'**
8. **'/order/product_Id'**
9. **'/order/cart/allbooks'**
10. **'/filter-product?category=History&price=all'**
11. **'/profile/user_Id'**
12. **'/view/order/order_Id'**
13. **'/update/update-details/user_id'**
14. **'/update/update-location/user_id'**
15. **'/update/update-account/user_id'**

Server Side Code.

- index.js

```
const express = require('express');
const dotenv = require('dotenv');
const app = express();
dotenv.config();

const errorHandler= require('./middleware/errorHandler');
app.use(errorHandler);
app.use(express.json());

app.use('/admin', require('./routes/adminRoutes'));
app.use('/', require('./routes/homeRoute'));
app.use('/', require('./routes/registrationRoutes'))
app.use('/profile', require('./routes/profileRoute'))
app.use('/product', require('./routes/productDetailsRoutes'))
app.use('/order', require('./routes/orederItemsRoutes'))
app.use('/view/order', require('./routes/viewOrederItemsRoutes'))
app.use('/insert', require('./routes/insertionRoutes'))
app.use('/update', require('./routes/updatationRoutes'))
app.use('/', require('./routes/filterProductRoute'))
app.use('/addtocart', require('./routes/addtocart'))
app.use('/viewcart', require('./routes/viewcartRoute'))
app.use('/removefromcart', require('./routes/removefromcartRoutes'))

const port = process.env.PORT;
app.listen(port, '0.0.0.0', () => {
  console.log(`Server is started at http://localhost:${port}`);
});
```

- **.env**

Storing Important Data in environment file

```
PORT = "5001"
# DataBase
database_host = "localhost"
database_user = "root"
database_port = 3306
database_password = ""
database_database = "store"
# Admin Credential
admin_email = "admin@abhay.com"
admin_password = "Admin-Of-Abhay"

access_token_secret = "this_is_my_secret_key_for_jwt_token"
session_secret = "thisismysecrtekeyfhrgrfgrferty84fwir767"
# session_time = 1000 * 60 * 60 * 24
```

- **mysqlDB.js**

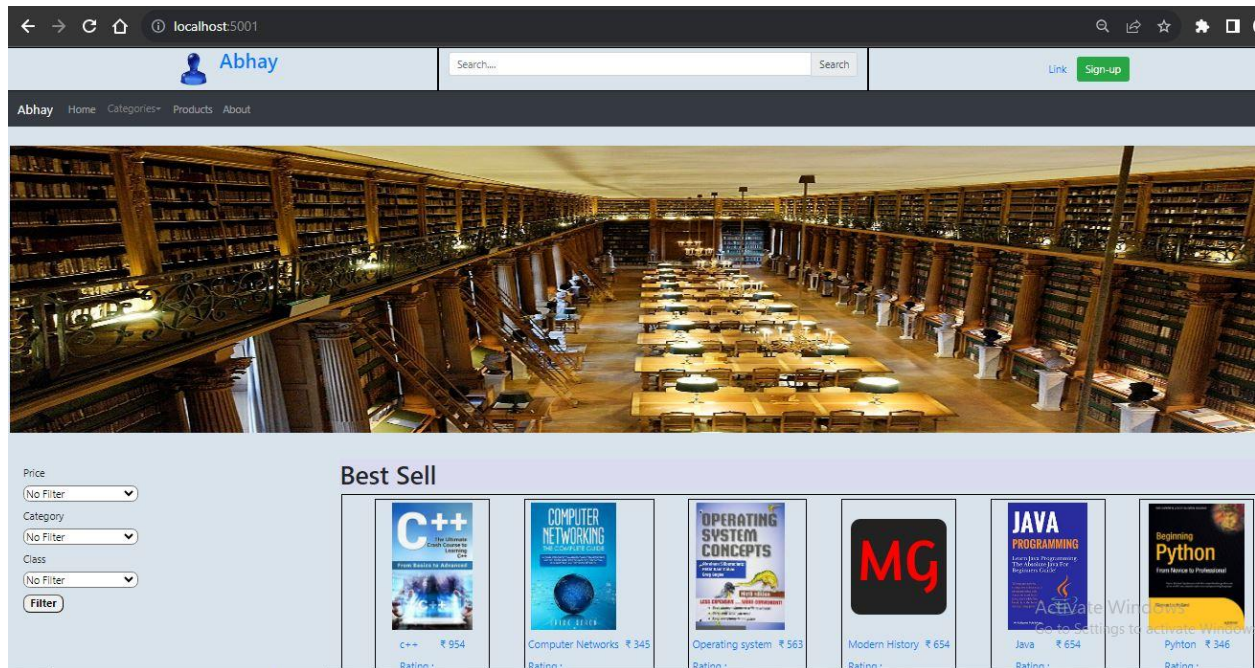
Connectin mySQL database with server.

```
const mysql = require('mysql');
const con = mysql.createConnection({
  host: process.env.database_host,
  user: process.env.database_user,
  port: process.env.database_port,
  password: process.env.database_password,
  database: process.env.database_database
});
con.connect((err) => {
  if (err) {
    return console.error('Error connecting to the database:');
  }
  else {
    console.log('Connected to the database!');
  }
});
module.exports = con;
```

User Interface

1. '/' or '/home'

User Authentication Not required to visit at home.



At 'home'

- you will get top 10 best sell book
- And there is filter component

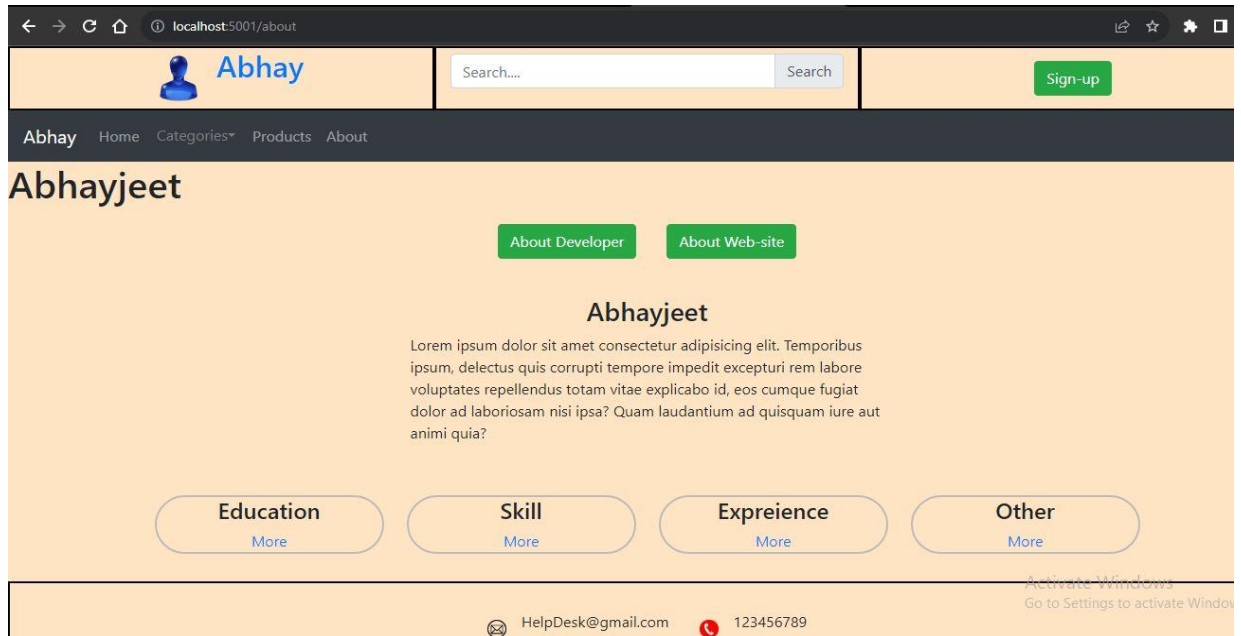
Filter Component



2. '/about'

Here, you will get about our website and owner.

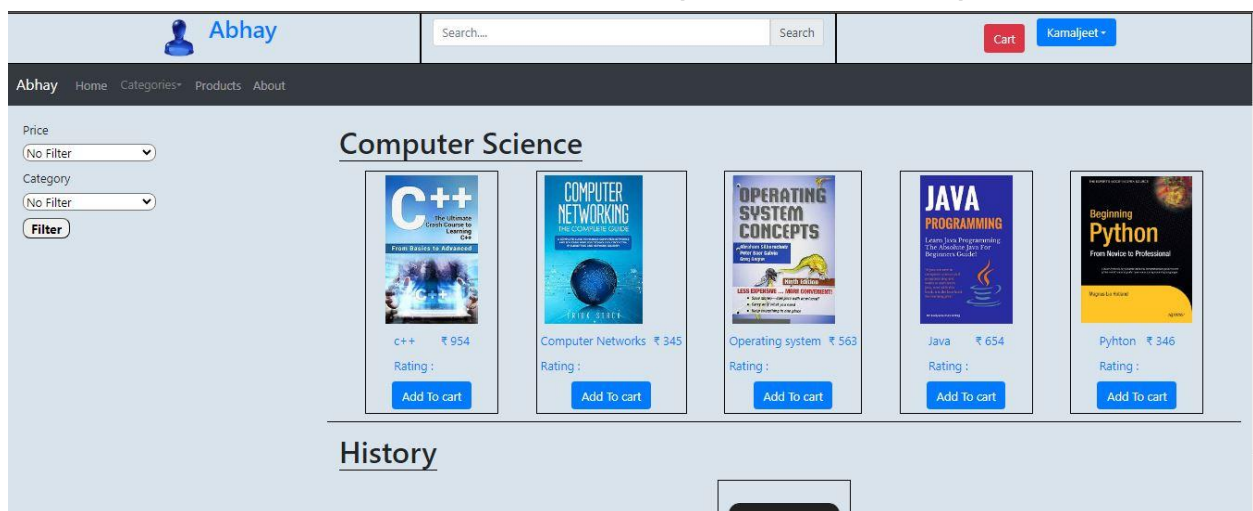
User Authentication Not required to visit at about.



3. '/product'

Here, you will get all books category wise

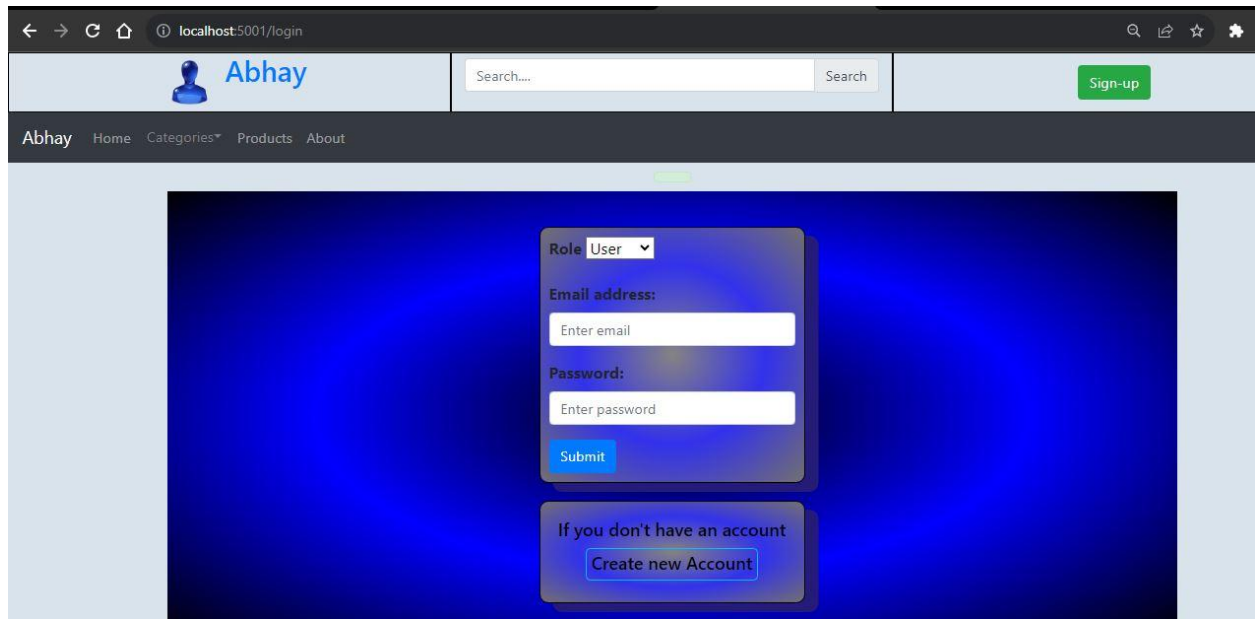
User Authentication Not required to visit at product.



4. '/login'

Here, only already registered user can login to view their accounts or to buy/sell books.

User Authentication Not required to visit at login.

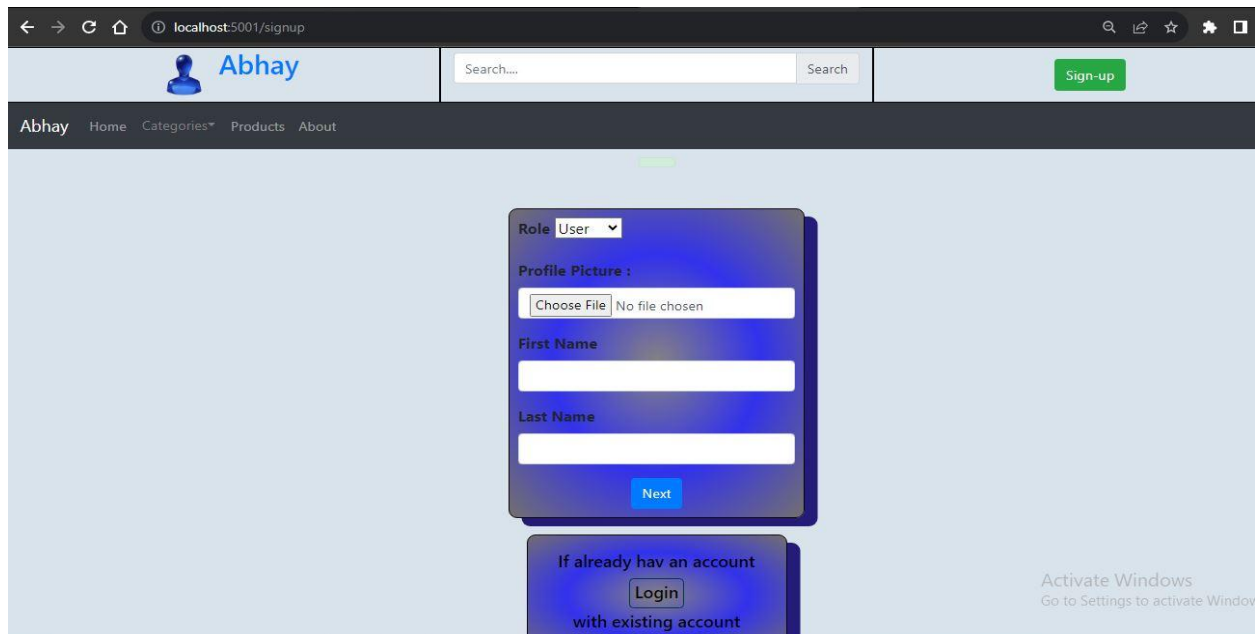


The screenshot shows a web browser at localhost:5001/login. The page has a header with the Abhay logo, a search bar, and a green 'Sign-up' button. Below the header is a navigation bar with links: Abhay, Home, Categories, Products, and About. The main content area features a login form with a 'Role' dropdown set to 'User'. The form includes fields for 'Email address' (placeholder: 'Enter email') and 'Password' (placeholder: 'Enter password'), followed by a blue 'Submit' button. Below the form is a link that says 'If you don't have an account' with a 'Create new Account' button.

5. '/signup'

Here, user can register themselves to buy/sell books.

User Authentication Not required to visit at signup.

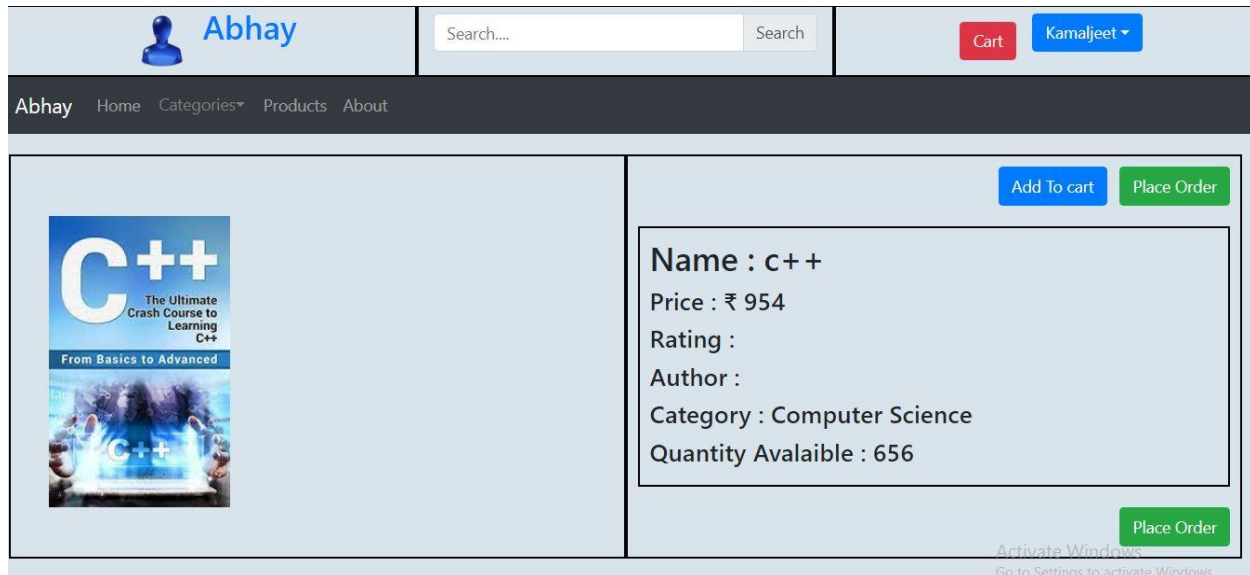


The screenshot shows a web browser at localhost:5001/signup. The page layout is identical to the login page, with the same header, navigation bar, and main content area. The main content area features a signup form with a 'Role' dropdown set to 'User'. The form includes a 'Profile Picture' section with a 'Choose File' button and 'No file chosen' text, followed by 'First Name' and 'Last Name' text input fields, and a blue 'Next' button. Below the form is a link that says 'If already hav an account' with a 'Login' button and the text 'with existing account'. In the bottom right corner, there is a watermark for 'Activate Windows'.

6. '/product/product_id'

Here you can view books details before buying the books.

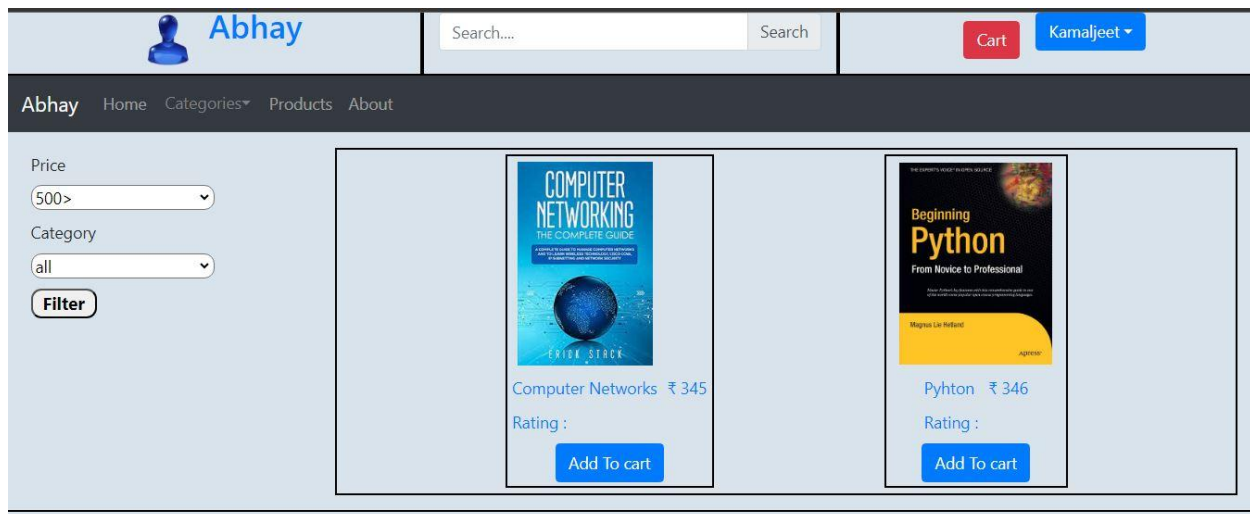
User Authentication Not required to view product details.



7. '/filter-product?category=History&price=all'

These routes used to filter out product

These routes also not required user authentication



8. '/order/product_Id'

Here, user can order book of respective book_Id
This route requires user Authentication.

localhost:5001/order/ccf86471-4674-4ed4-ae58-386fa939dde5

Name : Java

Author :

Price : ₹ 654

Rating :

Category : Computer Science

Quantity :

Buy

HelpDesk@gmail.com 123456789

9. '/order/cart/allbooks'

Here, user can confirm order of many books after adding books to cart.

This route requires user Authentication.

multiple

Name	price	qnt	Total price
Operating system	₹ 563	<input type="text" value="1"/>	₹ 563
Modern History	₹ 654	<input type="text" value="1"/>	₹ 654
Java	₹ 654	<input type="text" value="1"/>	₹ 654

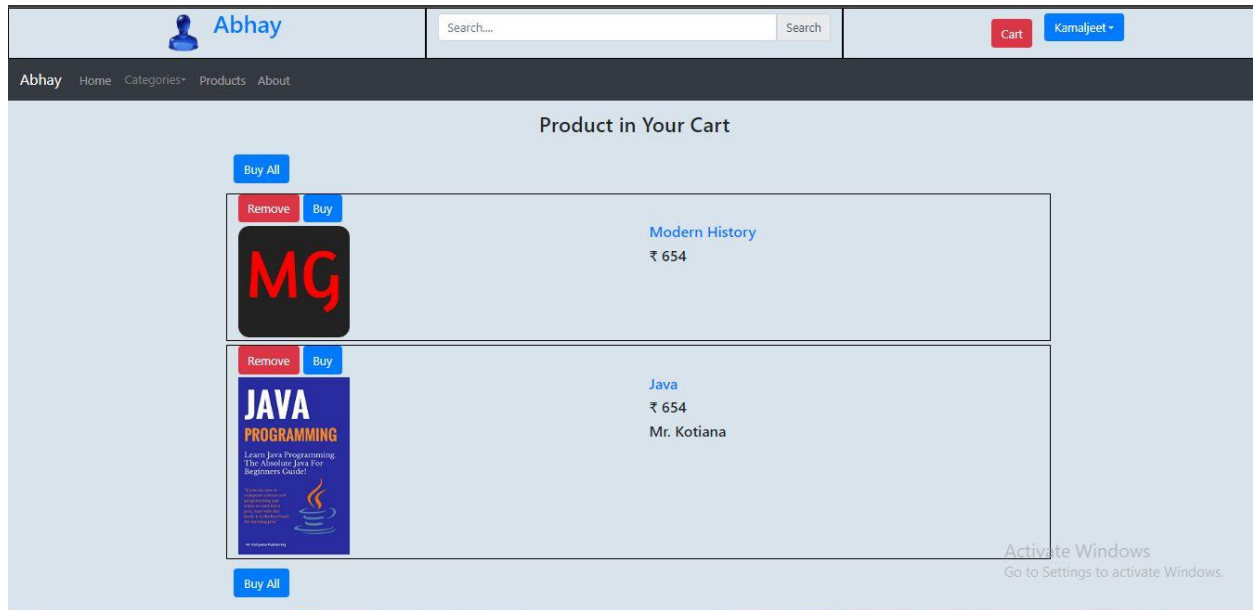
Total : ₹ 1871

Buy

10. '/viewcart/view'

Here, user can view books added in cart by themselves. And also they can remove the book from cart list.

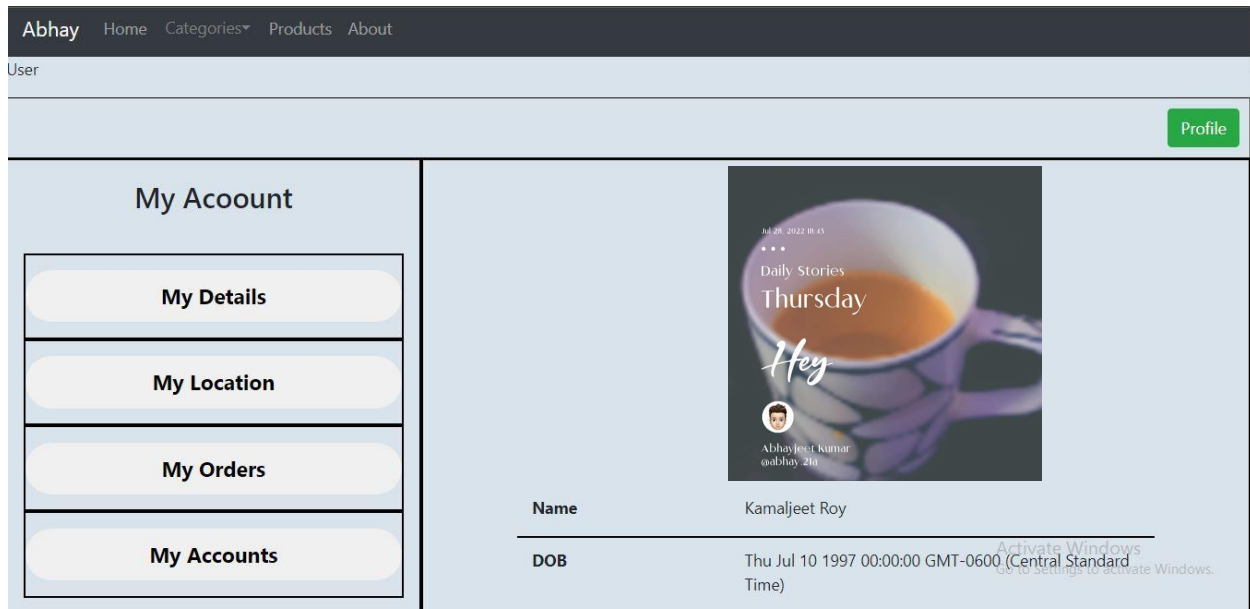
This routes also needs user authentication.



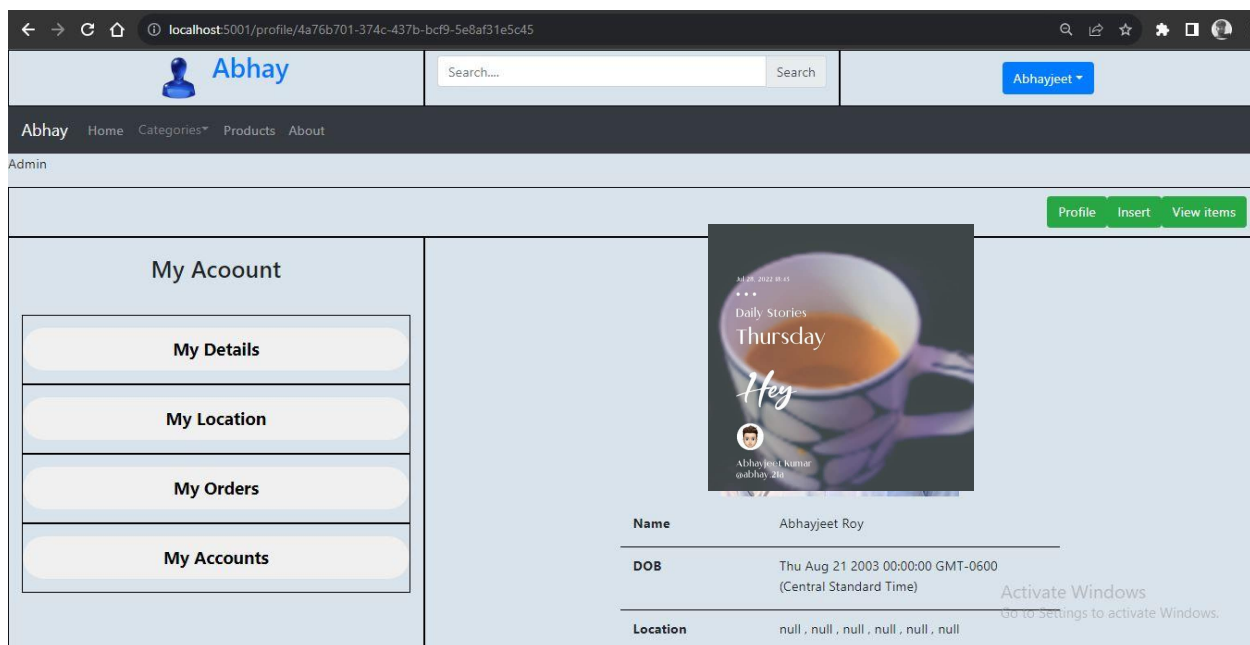
11. '/profile/user_id'

Here, user can view their profile details, order details
And Seller can also add books to sell and track his books details.
This routes also need user authentication.

User Portal : Buyer



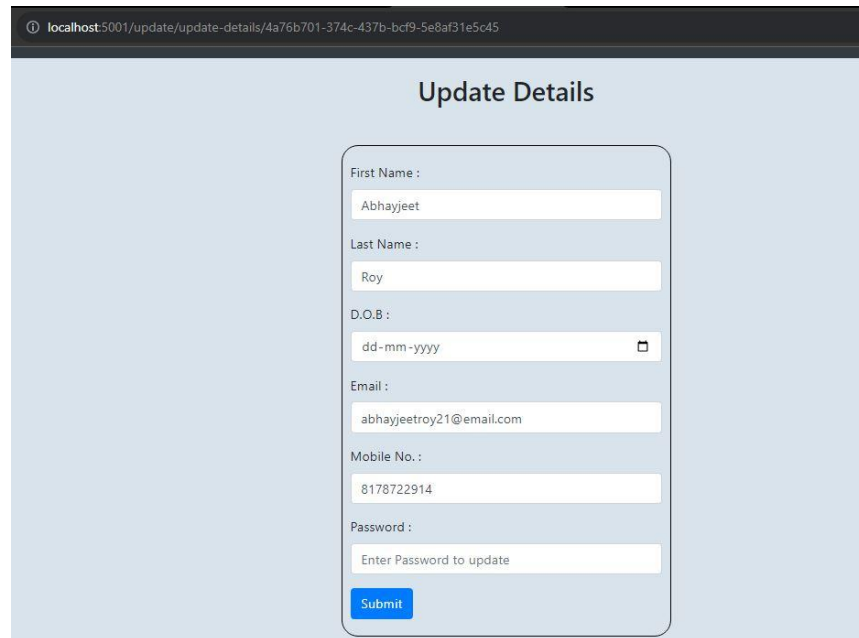
User Portal : Seller



12. '/update/update-details/user_id'

Here, user can update their profile details

This routes also needs user authentication

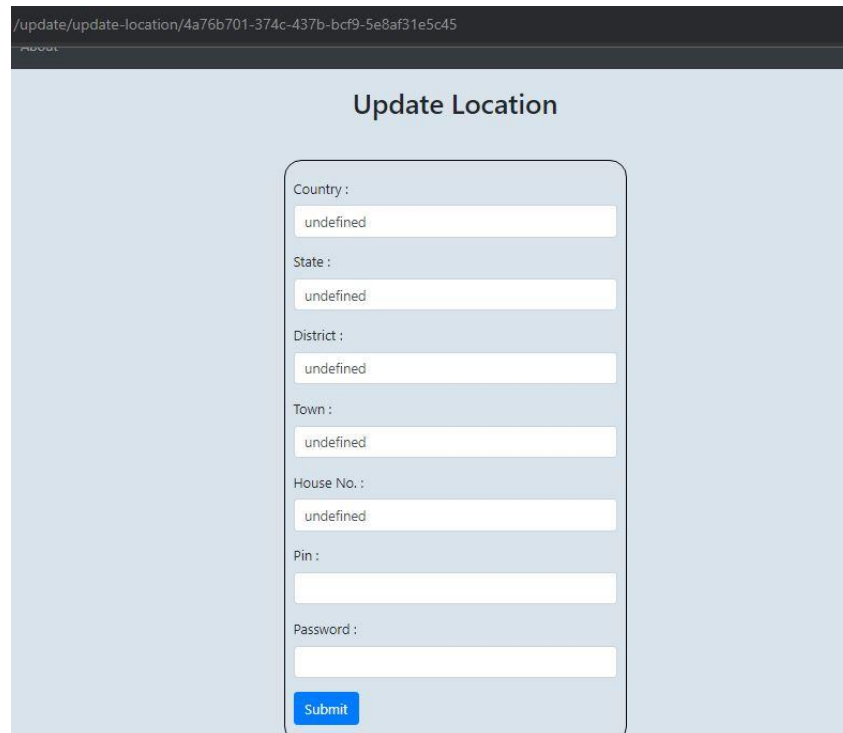


The screenshot shows a web browser window with the address bar displaying 'localhost:5001/update/update-details/4a76b701-374c-437b-bcf9-5e8af31e5c45'. The page title is 'Update Details'. The form contains the following fields: 'First Name' (filled with 'Abhayjeet'), 'Last Name' (filled with 'Roy'), 'D.O.B' (filled with 'dd-mm-yyyy' and a calendar icon), 'Email' (filled with 'abhayjeetroy21@email.com'), 'Mobile No.' (filled with '8178722914'), and 'Password' (filled with 'Enter Password to update'). A blue 'Submit' button is at the bottom.

13. '/update/update-location/user_id'

Here, user can update their location details

This routes also needs user authentication



The screenshot shows a web browser window with the address bar displaying '/update/update-location/4a76b701-374c-437b-bcf9-5e8af31e5c45'. The page title is 'Update Location'. The form contains the following fields: 'Country' (filled with 'undefined'), 'State' (filled with 'undefined'), 'District' (filled with 'undefined'), 'Town' (filled with 'undefined'), 'House No.' (filled with 'undefined'), 'Pin' (empty), and 'Password' (empty). A blue 'Submit' button is at the bottom.

14. '/update/update-account/user_id'

Here, user can update their login details

This routes also needs user authentication

update/update-account/4a76b701-374c-437b-bcf9-5e8af31e5c45

Update Account

Email :

Mobile :

Password :

Confirm Password :

Enter Password to confirm changes...

Authentication and Authorization

Authentication verifies the identity of users, while **authorization** controls their access to different parts of the site.

This documentation explains how these processes are implemented using **bcrypt.js** for secure password handling and **express-session** for managing user sessions.

Authentication Process :

The authentication process goes through three stage :

1. User registration
2. Password Hashing
3. Login

1.User Registration :

First User need to register himself by entering required personal data like (Name, DOB, mobile, email, location) and also set a difficult password.

As email and password are their login credentials.

After login, they can benefit some extra features :

The image displays two sequential steps of a user registration form. The first step (left) includes a 'Role' dropdown menu set to 'User', a 'Profile Picture' section with a 'Choose File' button and 'No file chosen' text, and input fields for 'First Name' and 'Last Name'. A 'Next' button is at the bottom. The second step (right) features input fields for 'Email' and 'Mobile No.', with 'Back' and 'Next' buttons at the bottom.

DOB

dd-mm-yyyy 

Back Next

Location

Enter Country

Enter State

Enter District

Enter Town-street

Enter House No.

Enter Pin No

Please Fill Absolute Address...

Back Next

Password

Confirm Password

Back

Submit

2.Password Hashing :

```
const bcrypt = require('bcrypt');

async function hashPassword(password) {
  try {
    const saltRounds = 10;
    const hashedPassword = await bcrypt.hash(password, saltRounds);
    return hashedPassword;
  } catch (error) {
    throw error;
  }
}
```



```

    }
  }
  async function verifyPassword(password, hash) {
    try {
      return await bcrypt.compare(password, hash);
    }
    catch (err) { throw err; }
  }
  module.exports = { hashPassword, verifyPassword };

```

This HashedPassword Saved in Database.

Bcrypt is provide much secure hashpassword. If Any one extract hashpassword,he can't access user data with that hashpassword. And Users only have real password.

3.Login

Now Only Registered users can login as they need email and password that they entered during registration.

In this stage,

The screenshot displays a web browser window at the URL `localhost:5001/login`. The page layout includes a top navigation bar with the 'Abhay' brand name, a search input field, and a green 'Sign-up' button. Below this is a secondary navigation bar with links for 'Abhay', 'Home', 'Categories', 'Products', and 'About'. The central focus is a login form with a 'Role' dropdown menu currently showing 'User'. The form contains two input fields: 'Email address' with the placeholder 'Enter email' and 'Password' with the placeholder 'Enter password'. A blue 'Submit' button is positioned below these fields. At the bottom of the form, there is a section titled 'If you don't have an account' with a button labeled 'Create new Account'.

- First user need give login email and password.
- Then at backend, system check the user exist with this email or not,
- If not exist, user redirect to login page with a message (“Please enter valid login credentials”)
- If user exist , then system verify the password enter by user is correct or not

```
async function verifyPassword(password, hash) {  
  try{  
    return await bcrypt.compare(password, hash);  
  }  
  catch(err) {throw err;}  
}  
module.exports={verifyPassword};
```

- If password is wrong, user will redirect to login page with a message (“Please enter valid login credentials”)
- If password is correct, then user get access to enjoy benefits of registered user with that role.

Authorization Process

Once users are logged in, Their user-id saved in session.
And They can enjoy different feature of website respective of their role.
Here are features of website respective of Role.

1. Buyers

- Can view books details
- Can add book to shopping cart.
- Can place order.
- Can manage their profile
- Can track order

2. Sellers

- Can access seller portal.
- Can add, edit and remove book from store.
- Can manage their seller profile.
- Can used as buyer portal.

Future Updates

1. Reviews and Ratings

Implementing a system for users to leave reviews and ratings for books and sellers, enhancing trust and decision-making.

2. Online Payment Integration

Integrating a payment gateway to facilitate online payments, such as upi, credit card and digital wallets.

3. Search and Recommendation System

Adding advanced search functionality and a recommendation system based on user preferences and browsing history.

Conclusion

1. Achievements

This e-commerce bookstore website successfully provides a platform for buying and selling books, with robust security measures and a user-friendly interface. It meets the initial project goals and requirements.

2. Lessons Learned

The project provided valuable insights into web development, security, and database design. It also highlighted the importance of user experience and scalability.

3. Project Significance

The e-commerce bookstore website contributes to the online book market, enabling book enthusiasts to access a wide range of books while providing a platform for sellers to reach a larger audience. The project has the potential to grow and adapt to changing market demands through future enhancements.

Source Code of this Project

GitHub : @21-abhay/E-commerce

Github - link : <https://github.com/21-abhay/E-commerce>

Contact Designer

Instagram : @21.abhay

GitHub : @21-abhay

Linkedin : @abhayjeet-kumar-186427263