

**Domain :** E-Commerce

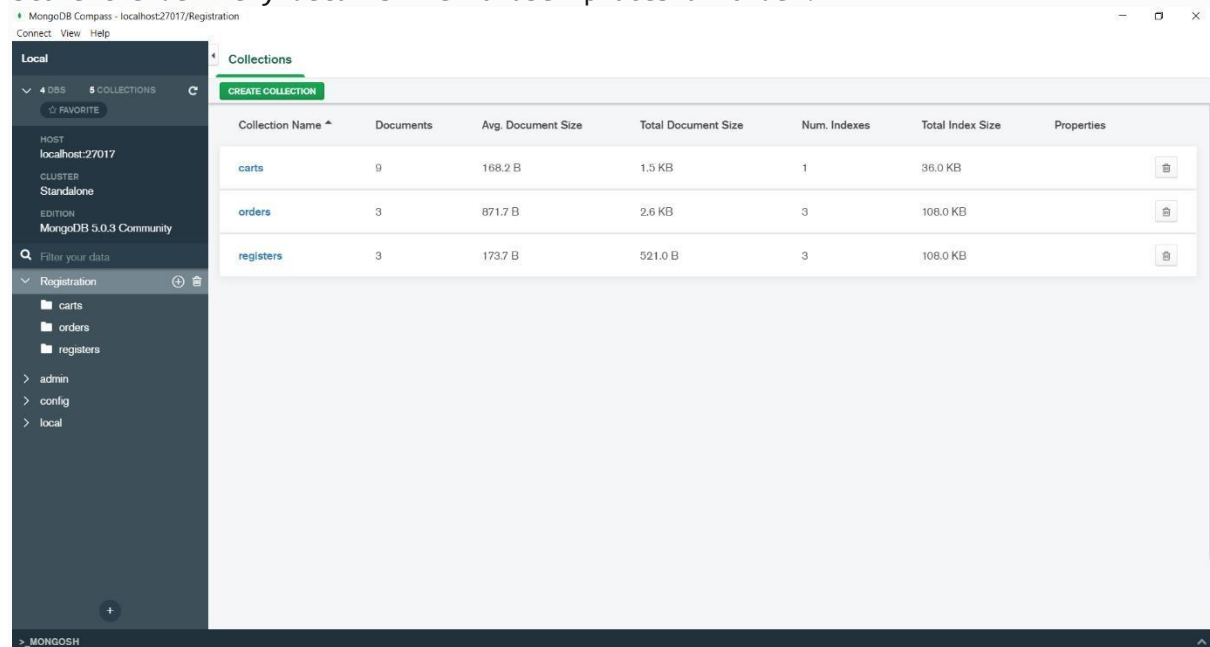
**Title Of Application :** Online Book Store

**Problem Statement :** The main objective of the online book store is to manage the details of all the books and customer. It will allow the Customers to buy new books from anywhere they want thereby reducing the manual work for searching the books. It will allow the administrator to keep track of the books that are sold and the delivery details of the customer when the customer places an order.

**NoSQL Database Used :** MongoDB

**Other Technologies Used :** HTML,CSS,javascript,Handlebars,Node.js,express.js

In my Application there are 3 collections namely registers, carts and orders. The collection registers is used to store the user details, collection carts is used to store the item present in the cart to the database and orders is used to store the delivery details when a user places an order.



The screenshot shows the MongoDB Compass interface. On the left, the 'Local' sidebar lists the host 'localhost:27017', cluster 'Standalone', and edition 'MongoDB 5.0.3 Community'. Under the 'Registration' database, three collections are listed: 'carts', 'orders', and 'registers'. The main panel displays the 'Registers' collection with a table of 3 documents. The table has columns for Collection Name, Documents, Avg. Document Size, Total Document Size, Num. Indexes, Total Index Size, and Properties.

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
carts	9	168.2 B	1.5 KB	1	36.0 KB	
orders	3	871.7 B	2.6 KB	3	108.0 KB	
registers	3	173.7 B	521.0 B	3	108.0 KB	

Whenever a new user visits the store he/she needs to register himself/herself by filling the correct details in registration form. Each field entered by the user is validated properly. All the fields are required. Name Field should only contain alphabets . Phone number should be of length 10 . Password should contain atleast 1 capital letter,1 small letter,1 digit, a special character and should have atleast 8 characters. Password and Confirm Password should match . If all the details are entered correctly then the user will get a message as “Registration Successful” else the user will get an alert and the user will not be registered.

localhost:3000/register

Book Store

HOME REGISTER

login or Create Account

localhost:3000 says  
Registration Successful

OK

Please fill in this form to create account.

Neha

Pandey

7387007168

nehpandey174@gmail.com

\*\*\*\*\*

\*\*\*\*\*

Register

Already Signed Up?  
Login Here !

As soon as the user registers successfully the database will be updated.

MongoDB Compass - localhost:27017

Connect View Collection Help

Local

4 DBS 4 COLLECTIONS

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.3 Community

Filter your data

Registration

cars

orders

registers

admin

config

local

Registration registers

Documents

Documents 2 TOTAL SIZE 341B AVG. SIZE 171B INDEXES 3 TOTAL SIZE 108.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 3 of 3

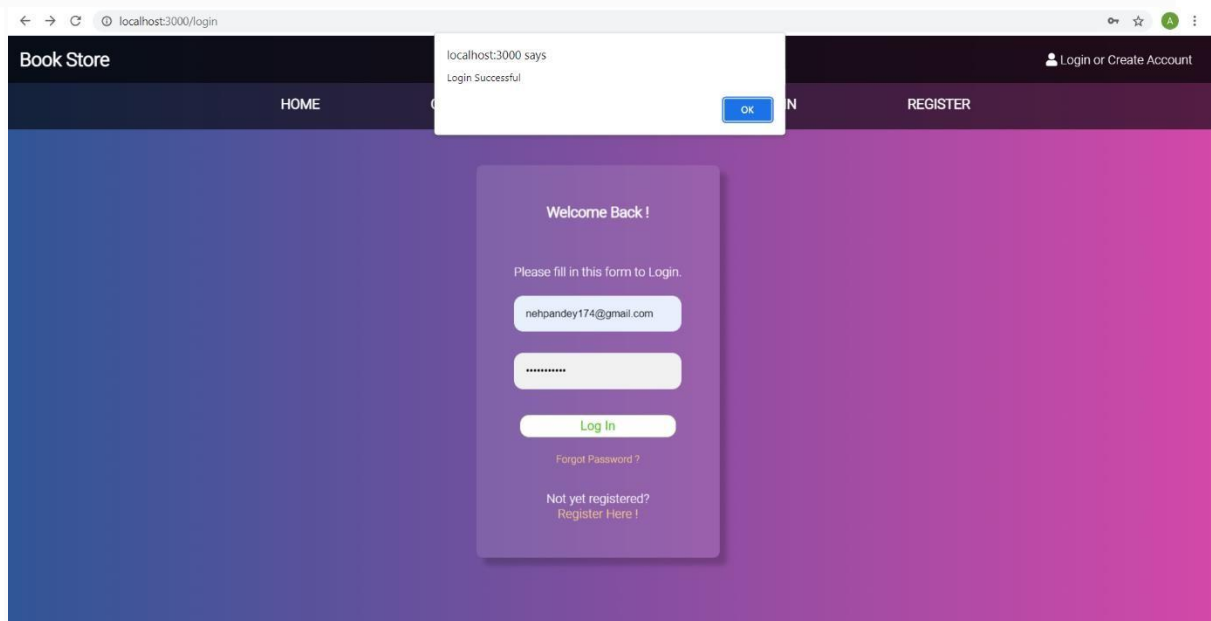
REFRESH

```
{
  "_id": "616c368aa1afb1c7e995d9ff",
  "firstname": "Neha",
  "lastname": "Pandey",
  "email": "nehpandey174@gmail.com",
  "phone": "7387007168",
  "password": "Grapes@1234",
  "confirmpassword": "Grapes@1234",
  "_v": 0
}
```

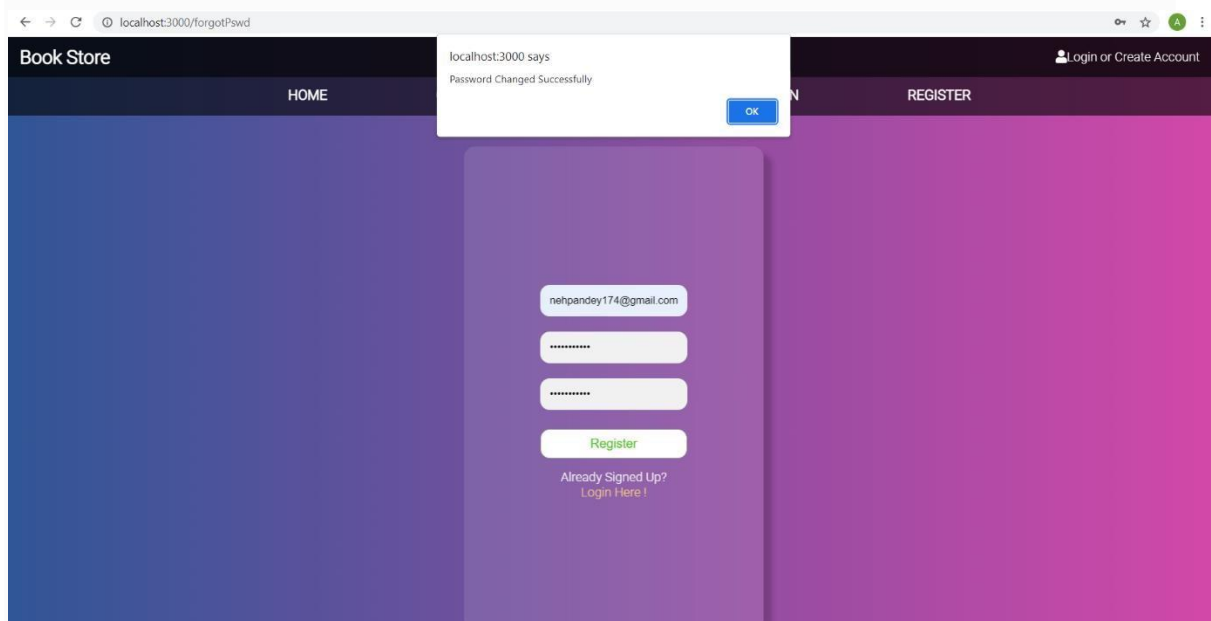
```
{
  "_id": {"$oid": "616c368aa1afb1c7e995d9ff"},
  "firstname": "Neha",
  "lastname": "Pandey",
  "email": "nehpandey174@gmail.com",
  "phone": "7387007168",
  "password": "Grapes@1234",
  "confirmpassword": "Grapes@1234",
  "_v": 0
}
```

A user who has already registered can directly login by entering the correct details in the login form . The data entered by the user is validated properly as it was validated at the time of registration. The email entered by the user is fetched from the form and matched with the email present in the database which was stored at the time of registration. If it is matched, then likewise password is

fetches and matches. If all the login details match to the details stored in the database then login is successful else the user will get Invalid login details as message.



In case if user has forgot the password he/she can click on the Forgot Password link and can update the password.



As soon as the user will get the message of "Password Changed Successfully". The database will find the email in the database and will update the password corresponding to it.

```
_id: ObjectId("616c368aa1afb1c7e995d9ff")
firstname: "Neha"
lastname: "Pandey"
email: "nehpandey174@gmail.com"
phone: 7387007168
password: "Grapes@1234"
confirmpassword: "Grapes@1234"
__v: 0
```

Password has changed from Apple@4321 to Grapes@1234.

---

## Registers

### use Registration

switched to db Registration

### show collections

carts

orders

registers

```
db.registers.find({email:'nehpandey174@gmail.com'})
```

```
{
  "_id" : ObjectId("616c368aa1afb1c7e995d9ff"),
  "firstname" : "Neha",
  "lastname" : "Pandey",
  "email" : "nehpandey174@gmail.com",
  "phone" : 7387007168,
  "password" : "Grapes@1234",
  "confirmpassword" : "Grapes@1234",
  "_v" : 0
}
```

```
db.registers.find({email:'anny45@gmail.com'})
```

```

{
  "_id" : ObjectId("616aecaf23cc197e145a0853"),
  "firstname" : "Anny",
  "lastname" : "Bond",
  "email" : "anny45@gmail.com",
  "phone" : 8790211456,
  "password" : "Bond%9876",
  "confirmpassword" : "Bond%9876",
  "__v" : 0
}

db.registers.find({email:'watsonle@gmail.com'})

{
  "_id" : ObjectId("61521ea6069dcce46452c2ba"),
  "firstname" : "Leo",
  "lastname" : "Watson",
  "email" : "watsonle@gmail.com",
  "phone" : 9875542384,
  "password" : " King&Leo98",
  "confirmpassword" : "King&Leo98",
  "__v" : 0
}

db.registers.updateOne( {
  'email' : 'watsonle@gmail.com'},
  {$set: {'password': 'Mango*9876'}}
)

db.registers.save( { 'email' : 'watsonle@gmail.com'})

db.registers.updateOne(

```

```

{'email' : 'watsonle@gmail.com'},

{$set: {'confirmpassword': 'Mango*9876'}}

})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

db.registers.save( { 'email' : 'watsonle@gmail.com'})

db.registers.find({email:'watsonle@gmail.com'})

{

"_id" : ObjectId("61521ea6069dcce46452c2ba"),

"firstname" : "Leo",

"lastname" : "Watson",

"email" : "watsonle@gmail.com",

"phone" : 9875542384,

"password" : "Mango*9876",

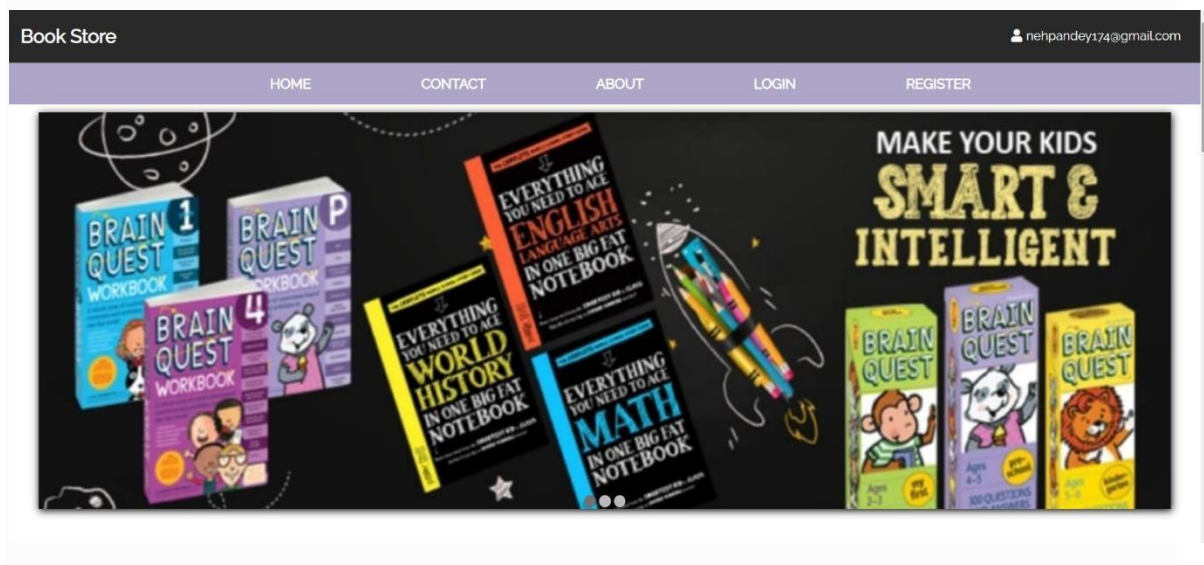
"confirmpassword" : "Mango*9876",

"_v" : 0 }

```

---

After successful Registration/Login user will be redirected to store.



## BIOGRAPHIES AND AUTOGRAPHIES

### Wings Of Fire

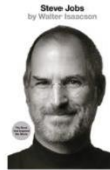


Wings of Fire: An Autobiography of Abdul Kalam  
by Arun Tiwari and A. P. J. Abdul Kalam

Rs.192

[ADD TO CART](#)

### Steve Jobs



Steve Jobs: The Exclusive Biography  
by Walter Isaacson

Rs.440

[ADD TO CART](#)

### A Promised Land



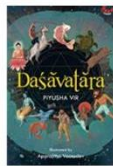
A Promised Land  
Barack Obama

Rs.1374

[ADD TO CART](#)

## CHILDREN AND TEENS

### Dasavatara



Dasavatara  
Piyusha Vir

Rs.196

[ADD TO CART](#)

### The Nameless God



The Nameless God  
Savie Karnel

Rs.199

[ADD TO CART](#)

### The Crocodile



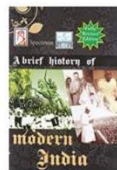
The Crocodile Who Ate Butter Chicken for Breakfast...  
Khyrunnisa A.

Rs.196

[ADD TO CART](#)

## HISTORY

### Modern India

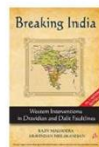


A Brief History of Modern India (2019-2020 Edition)...

Rs.219

[ADD TO CART](#)

### Breaking India



Breaking India: Western Interventions in Dravidian...  
Rajiv Malhotra, Aravindan Neelakandan

Rs.610

[ADD TO CART](#)

### An Era Of Darkness



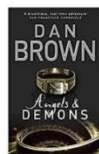
An Era of Darkness: The British Empire in India  
Shashi Tharoor

Rs.646

[ADD TO CART](#)

## MYSTERY, CRIME AND THRILLER

### Angels and Demons

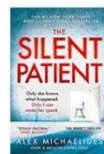


Angels and Demons: (Robert Langdon Book 1)  
Dan Brown

Rs.259

[ADD TO CART](#)

### The Silent Patient



The Silent Patient: The record-breaking, multiplatform...  
Alex Michaelides

Rs.279

[ADD TO CART](#)

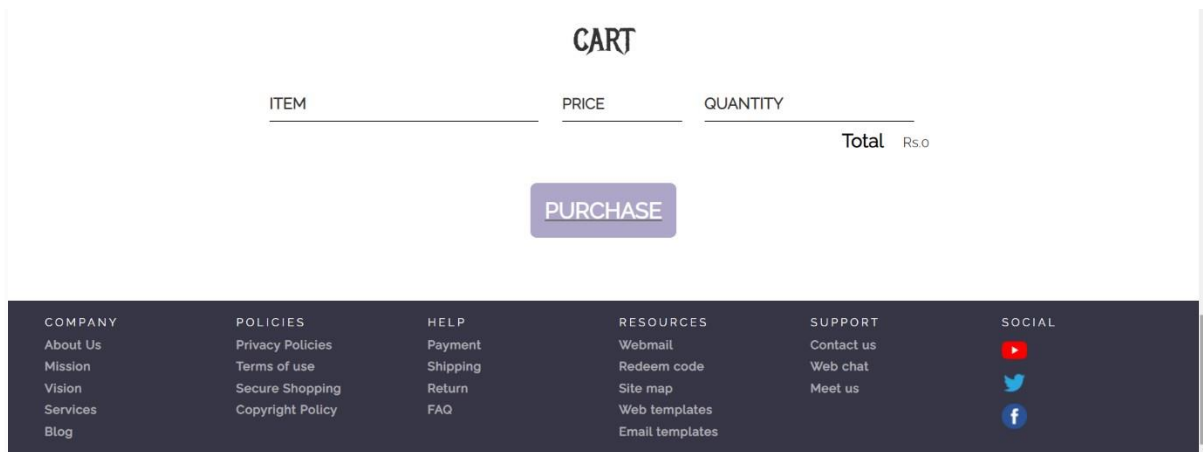
### Sherlock Holmes



The Complete Novels of Sherlock Holmes (Deluxe Edition)...  
Arthur Conan Doyle

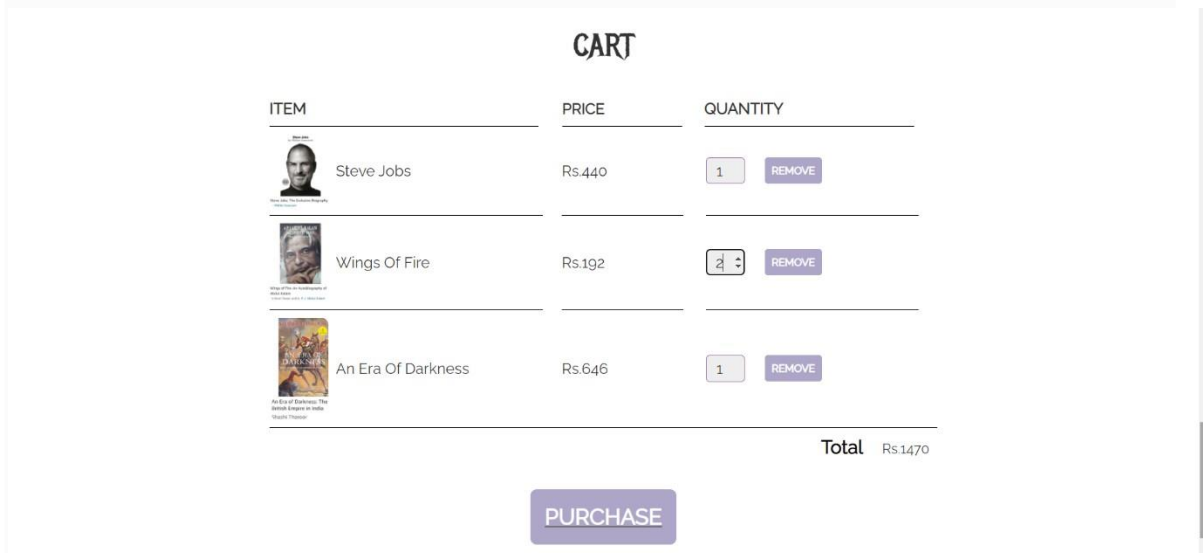
Rs.499

[ADD TO CART](#)



Initially, When the user enters there will be no books in the cart.

When user will select add to cart button then items will be stored in the cart and the database will be updated. User can then increase the quantity of books or remove the books present in the cart. Accordingly the database will be updated.



Books in the cart are stored in the database automatically when user clicks add to cart button. If later user changes the quantity of book or remove any book the database will be updated accordingly.



```
_id: ObjectId("616c49aa5b54b8375c0a24a8")
useremail: "nehpandey174@gmail.com"
imagePath: "http://localhost:3000/Images/2.png"
title: "Steve Jobs"
price: 440
quantity: 1
__v: 0
```

```
_id: ObjectId("616c49ad5b54b8375c0a24ab")
useremail: "nehpandey174@gmail.com"
imagePath: "http://localhost:3000/Images/1.png"
title: "Wings Of Fire"
price: 192
quantity: 2
__v: 0
```

```
_id: ObjectId("616c49af5b54b8375c0a24ae")
useremail: "nehpandey174@gmail.com"
imagePath: "http://localhost:3000/Images/16.png"
title: "An Era Of Darkness"
price: 646
quantity: 1
__v: 0
```

---

## Carts

```
db.carts.find({title: 'A Promised Land'})
{
  "_id" : ObjectId("616b2f514057621c2c6bfb85"),
  "useremail" : "anny45@gmail.com",
  "imagePath" : "http://localhost:3000/Images/4.png",
  "title" : "A Promised Land",
  "price" : 1374,
  "quantity" : 1,
```

```

    "__v" : 0
  }

  db.carts.find({title:'The Nameless God'})

  {
    "_id" : ObjectId("616b312f24e06f973da2083a"),
    "useremail" : "anny45@gmail.com",
    "imagePath" : "http://localhost:3000/Images/6.png",
    "title" : "The Nameless God",
    "price" : 199,
    "quantity" : 1,
    "__v" : 0
  }

  db.carts.updateOne(
    { 'title' : 'The Nameless God'},
    {$set:{'quantity': 10}}
  )

  { "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

  db.carts.save( { 'title' : 'The Nameless God'})

  WriteResult({ "nInserted" : 1 })

  //After Update

  db.carts.find({title:'The Nameless God'})

  {
    "_id" : ObjectId("616b312f24e06f973da2083a"),
    "useremail" : "anny45@gmail.com",
    "imagePath" : "http://localhost:3000/Images/6.png",
    "title" : "The Nameless God",
    "price" : 199,

```

```
"quantity" : 10,  
  
  "__v" : 0  
}
```

---

If a user tries to add a book which is already in cart he/she will get a message product already in cart.

When the user clicks on purchase button he/she is redirected to order page, where the user enters the delivery details. In this form also all the fields are validated as it was validated in earlier forms.

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/order'. A white notification box at the top center says 'localhost:3000 says Order Successful' with an 'OK' button. Below the notification is a form with a purple background. The form contains several input fields with the following values: '7387007168', '440123', 'Platinum Green, Shivaji Nagar, M', 'Mumbai', 'Maharashtra', and 'India'. At the bottom of the form is a green 'Order' button. The footer of the page has links for 'COMPANY', 'POLICIES', 'HELP', 'RESOURCES', 'SUPPORT', and 'SOCIAL'.

If all the details entered by the user is correct then he/she will get the message as "Order Successful" and the details in the form are fetched and are saved in the database in Collection named order else the user will have to re-enter the correct details. All the products in the cart collection are removed and now added in order collection.

The orders collection has fields name, email, phone, books(where all the books ordered by the user is stored),total price, pincode ,address, city, state and country. All the fields are necessary.

istration.orders

Registration.orders Documents

Registration.orders

DOCUMENTS 3 TOTAL SIZE 2.5KB AVG. SIZE 868B INDEXES 3 TOTAL SIZE 108.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: "value" }

ADD DATA VIEW

Displaying documents 1 - 3 of 3 REFRESH

```
{
  "_id": ObjectId("616c49d35b54b8375c0a24b2"),
  "name": "Neha Pandey",
  "email": "nehpandey174@gmail.com",
  "phone": 7387007168,
  "books": Array
    - 0: Object
      "_id": ObjectId("616c49aa5b54b8375c0a24a8")
      "useremail": "nehpandey174@gmail.com"
      "imagePath": "http://localhost:3000/Images/2.png"
      "title": "Steve Jobs"
      "price": 440
      "quantity": 1
      "__v": 0
    - 1: Object
      "_id": ObjectId("616c49ad5b54b8375c0a24ab")
      "useremail": "nehpandey174@gmail.com"
      "imagePath": "http://localhost:3000/Images/1.png"
      "title": "Wings Of Fire"
      "price": 192
      "quantity": 2
      "__v": 0
    - 2: Object
      "_id": ObjectId("616c49af5b54b8375c0a24ae")
      "useremail": "nehpandey174@gmail.com"
      "imagePath": "http://localhost:3000/Images/16.png"
      "title": "An Era Of Darkness"
      "price": 646
      "quantity": 1
      "__v": 0
  "totalprice": 1470
  "pincode": 440123
  "address": "Platinum Green, Shivaji Nagar, Mumbai"
  "city": "Mumbai"
  "state": "Maharashtra"
}
```

orders

```
db.orders.find({email: 'nehpandey174@gmail.com'})
```

```
{
```

```
  "_id" : ObjectId("616c49d35b54b8375c0a24b2"),
```

```
  "name" : "Neha Pandey",
```

```
  "email" : "nehpandey174@gmail.com",
```

```
  "phone" : 7387007168,
```

```
  "books" : [ {
```

```
    "_id" : ObjectId("616c49aa5b54b8375c0a24a8"),
```

```
    "useremail" : "nehpandey174@gmail.com",
```

```
    "imagePath" : "http://localhost:3000/Images/2.png",
```

```
    "title" : "Steve Jobs",
```

```
    "price" : 440,
```

```
    "quantity" : 1,
```

```
    "__v" : 0 },
```

```
{
  "_id" :
  ObjectId("616c49ad5b54b8375c0a24ab"),
  "useremail" : "nehpandey174@gmail.com",
  "imagePath" : "http://localhost:3000/Images/1.png",
  "title" : "Wings Of Fire",
  "price" : 192,
  "quantity" : 2,
  "__v" : 0
},
{
  "_id" : ObjectId("616c49af5b54b8375c0a24ae"),
  "useremail" : "nehpandey174@gmail.com",
  "imagePath" : "http://localhost:3000/Images/16.png",
  "title" : "An EraOf Darkness",
  "price" : 646, "quantity" : 1, "__v" : 0
} ],
"totalprice" : 1470,
"pincode" : 440123,
"address" : "Platinum Green,
Shivaji Nagar, Mumbai",
"city" : "Mumbai",
"state" : "Maharashtra",
"country" : "India",
  "__v" : 0
}
```

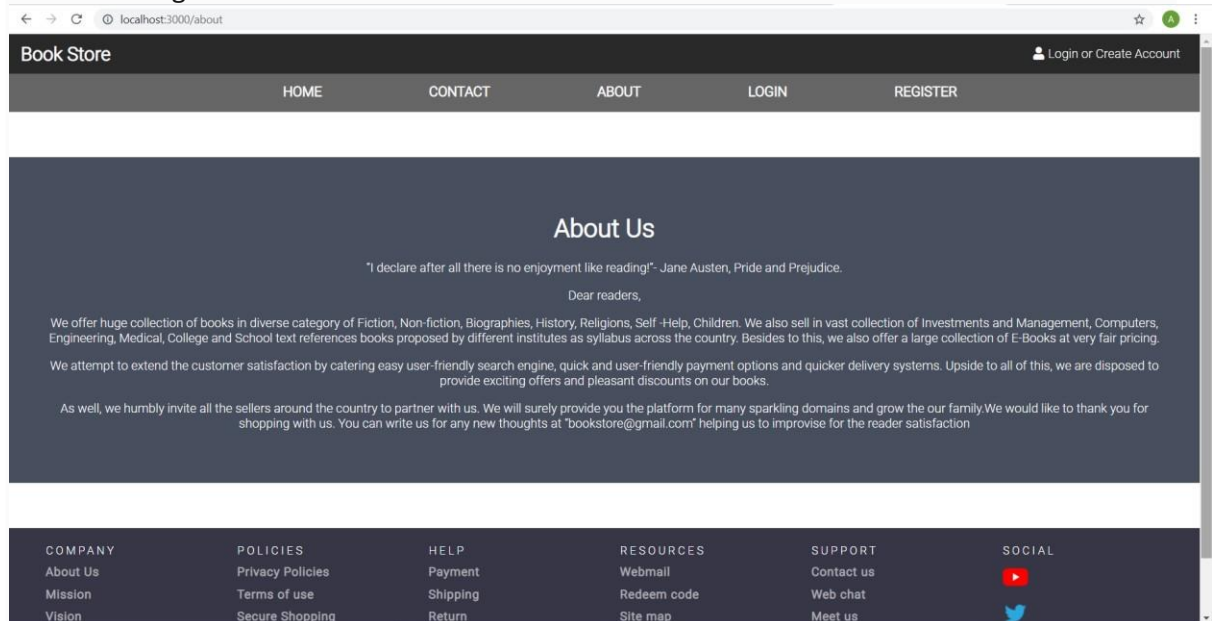
```
db.orders.deleteOne({email:'nehpandey174@gmail.com'})

{ "acknowledged" : true, "deletedCount" : 1 }

db.orders.find({email:'nehpandey174@gmail.com'})

//No Output Because the document is deleted
```

## About Us Page



## Contact Us Pag

