E-COMMERCE APP DEVELOPMENT

PHASE 5: PROJECT DOCUMENTATION AND SUBMISSION

In this phase we are going to document the final look of our e-commerce app. This includes the outline of the e-commerce app's objective, design thinking, development phases (1 and 2), the platforms layout, features we have added till now and technical implementation details.

As a part of this documentation, we are adding details about the updated frontend and backend.

As instructed, we have used html and CSS for frontend, flask for backend and SQLite for database purposes.

PROBLEM AND OBJECTIVE:

Crafts have been an integral part of daily life in villages, towns, courts, and religious establishments. There are approximately 70 lakh handicraft artisans in the country. As a largely unorganized sector, handicrafts face problems such as a paucity of professional infrastructure such as work sheds, storage space, shipping, and packing facilities. Due to their low

education, artisans often cannot identify potential new markets for their products, nor do they understand the requirements for interacting with these markets.

This project is to build an artisanal e-commerce platform. The goal is to connect skilled artisans with global audience, showcasing their handmade products and providing features like secure shopping carts, payment gateways and an intuitive checkout process. This involves designing of e-commerce platform, implementing necessary features, and ensuring a seamless user experience.

The foremost objective is to provide local artisans with an easy-to-use online platform wherein they can register and sell/promote their products.

PROJECT OVERVIEW

PROJECT NAME:

ARHA - CONNECTING ARTISANS

PROJECT DESCRIPTION:

ARHA - CONNECTING ARTISANS is an e-commerce platform that connects artisans and customers. It provides artisans with a platform to showcase and sell their handcrafted products while allowing customers to discover unique and authentic artisanal items.

DESIGN THINKING:

♣ Platform layout:

The layout consists of product categories, individual product pages, shopping cart, payment, and a smooth checkout process.

- ♠ The homepage has products, product categories, sign in, log in, shopping cart etc.…
- ♠ The product categories page has various categories of handmade products. For example, it has categories such as bamboo works, rattan works, pottery, etc.…
- ♠ Individual product pages are the detailed pages about a particular product which includes description, pictures, price, seller info and many more.
- ♠ Shopping cart has all the products that have been added by the added by the customer. It has the image, name, product details, quantity, total price and buying options. The user can also delete a product from the cart.
- ♠ Checkout processes begin when customers place an order in the online store and proceed to payment.

- ♠ Payment can be made after the customer has ordered the product. The user has the option to pay with card.

♣ Product Showcase:

We have database tables for registration, user, customer order, categories, products, brands.
We have used SQLite3 for database.

- ♠ The registration table has details such as user name, email, password, country, city, contact, address, zip code, profile, account creation date.
- ♠ The customer order table has invoice number, status of the order, customer id., date ordered, other orders made by the customer.
- ♠ The add products table is made for sellers. This has the name, price, discount, stock, colors, description, category, and images of the products.
- ♠ There are separate tables for brands and category too.

♣ User Authentication:

Flask_login is a flask extension that enables user authentication. Flask_login uses cookie-based authentication. When a client login via his credentials, Flask creates a session containing a user

id and then sends the session id to the user via a cookie, using which he/she can log in and out as and when required.

♣ Shopping cart:

The shopping cart enables the users to add products to the cart, update quantities and remove items, calculate the total cost, proceed to checkout.

♣ Checkout process:

The checkout process helps the user to place the order, review order details, get the pdf of the invoice and enables the user to pay with a card.

♣ Payment Integration:

To facilitate transactions, the app integrates with secure payment gateway stripe.

♣ User Experience:

HTML, CSS has been used to create a visually appealing user interface. It offers a user-friendly navigation and search features.

LOGO OF THE APP:

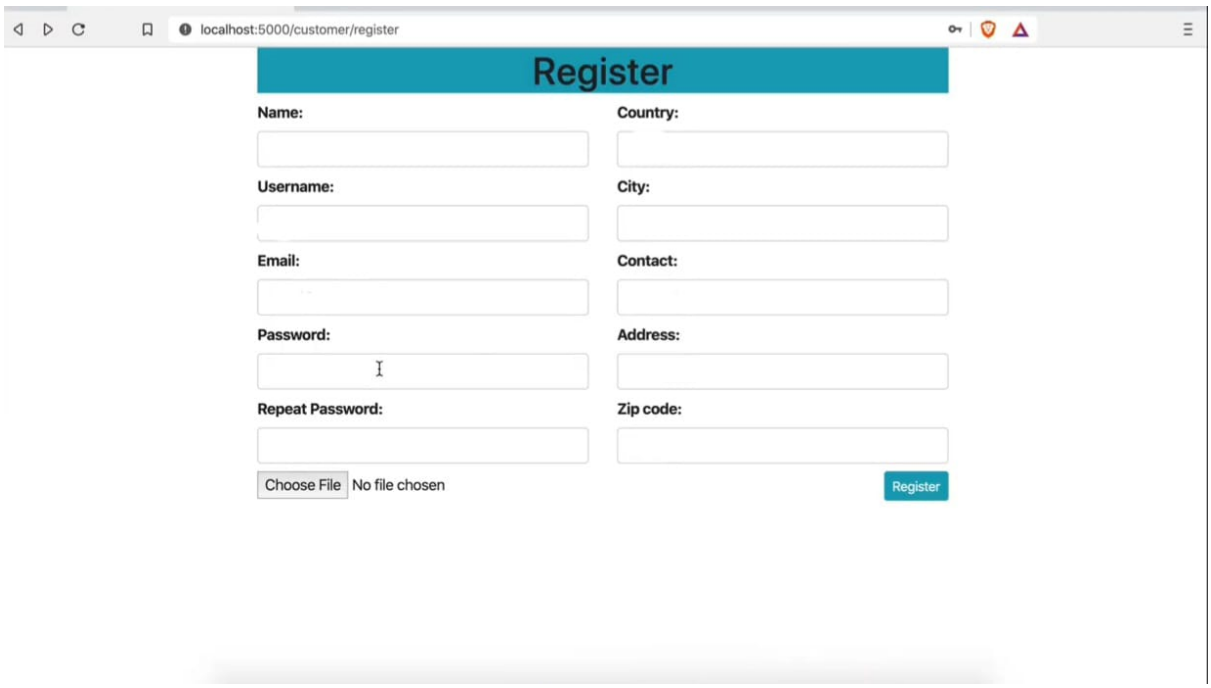
ARHA

CONNECTING ARTISANS

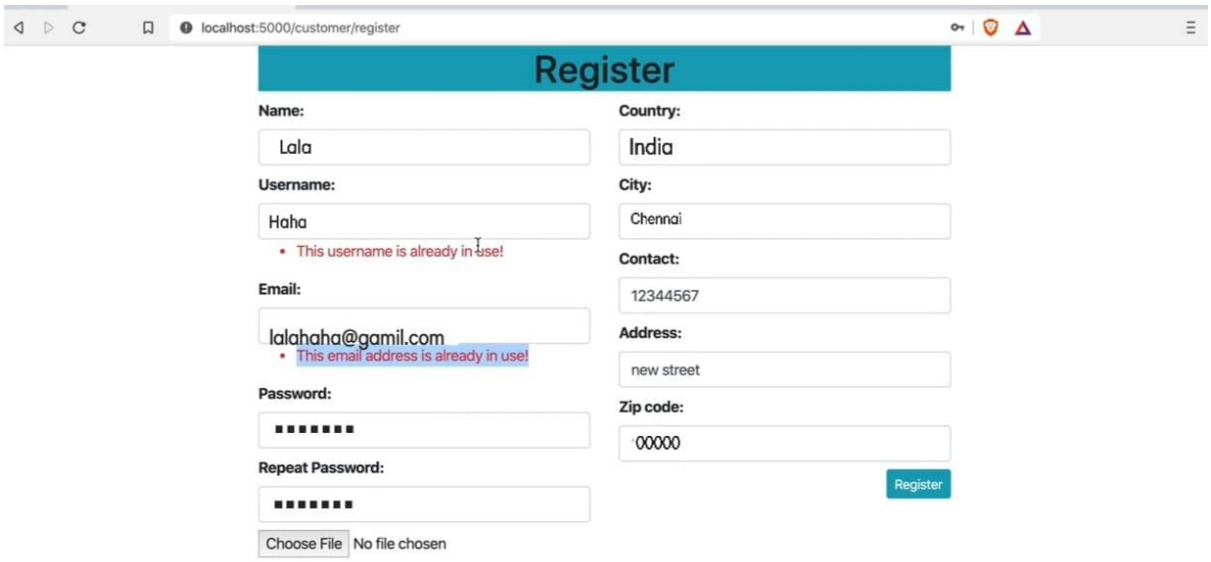
FRONTEND:

The Front end of our app has been designed using html and CSS. The various pages available in our app are showcased below:

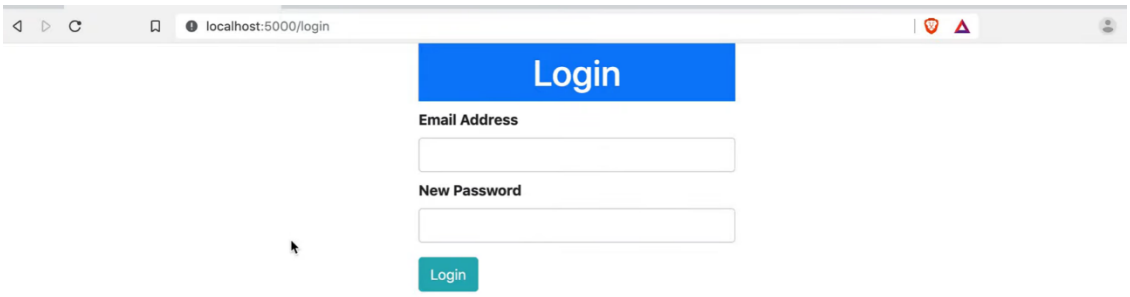
As the app gets opened the user will be prompted to register. This is basic process as all apps do in the beginning. The registration page of our app is:



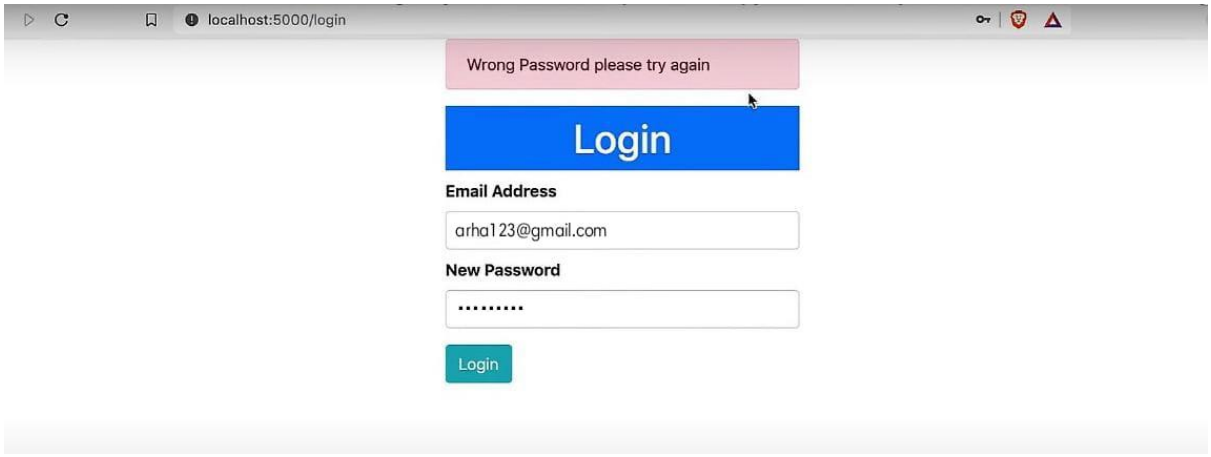
If the customer uses an email or username which is already taken then he/she will be warned of that error.



Once the user has registered in our app, he/she will be prompted to login. The login page is:

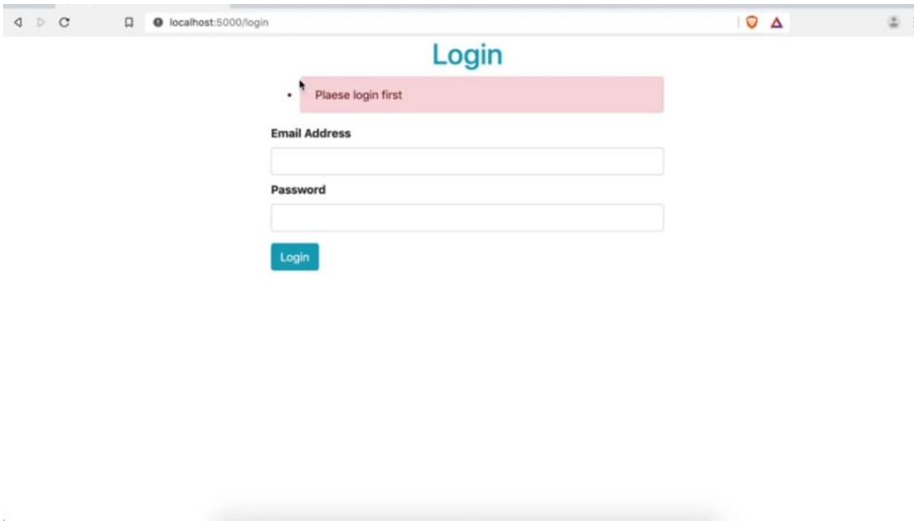


If the user logs in properly then the user will be redirected to the home page else the user will be shown an error.

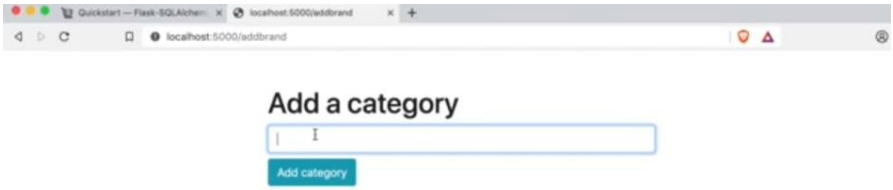


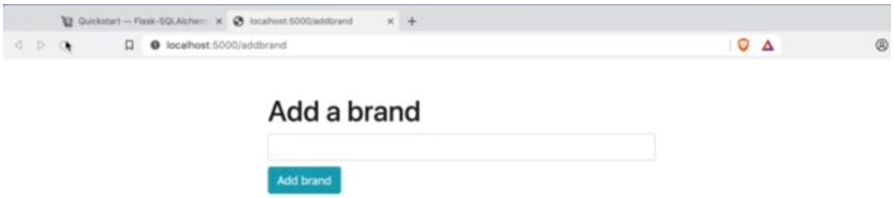
Once the user has completed a successful login the user will be redirected to the home page.

If the user is a seller/Merchant then he/she has to login as an admin. This enables the seller to add brands, categories, products to the platform.



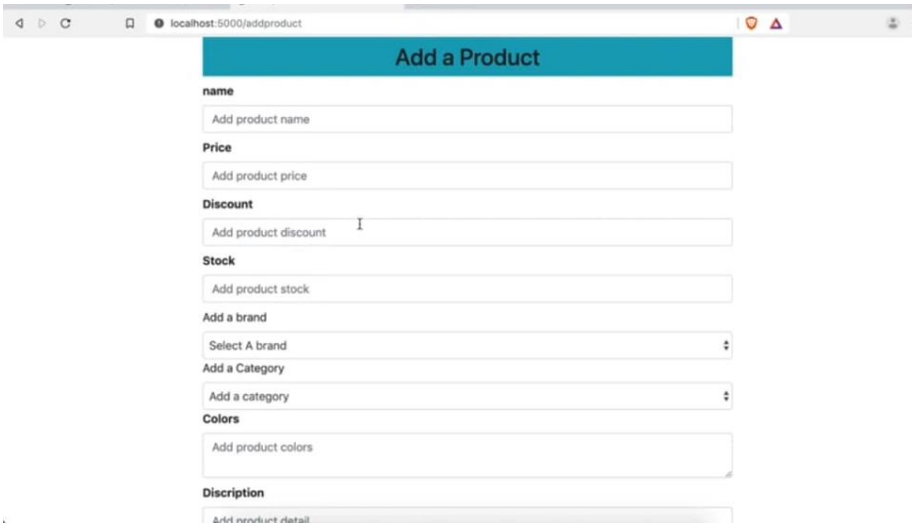
Once the seller has logged in, he can add brands, categories, products.

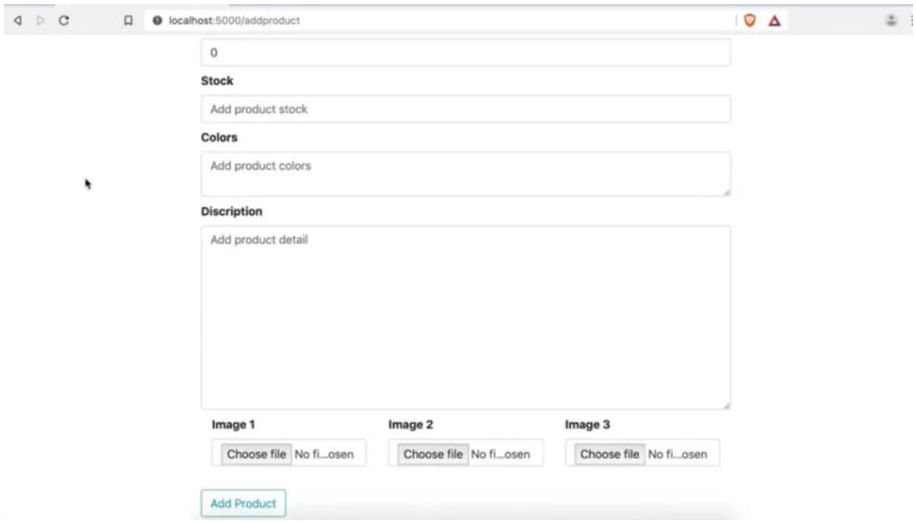




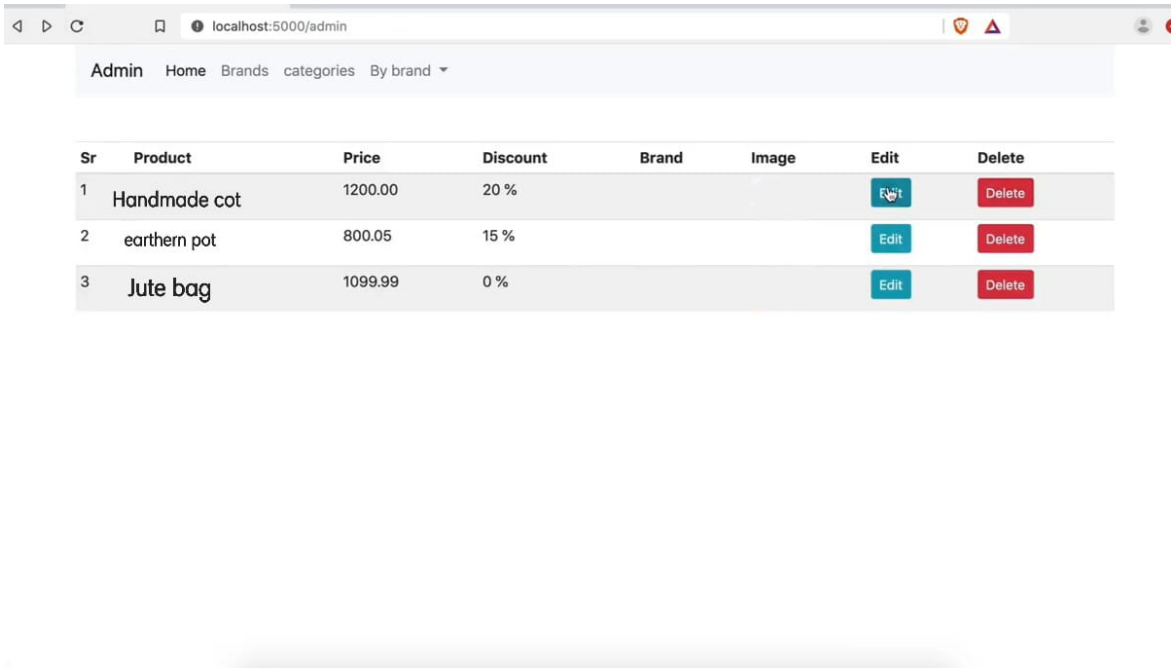
Once the seller adds an item to the brands/categories it will be added to database.

The seller can add the products by just filling up the add products form.

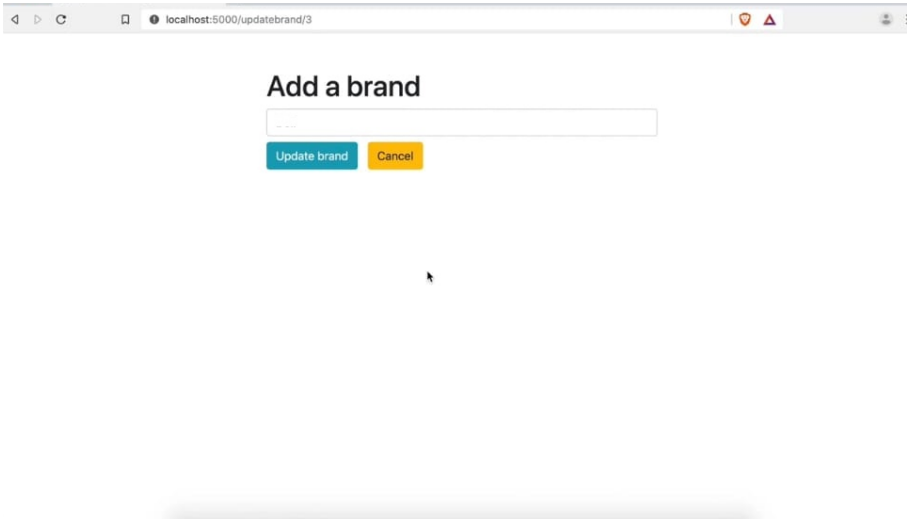




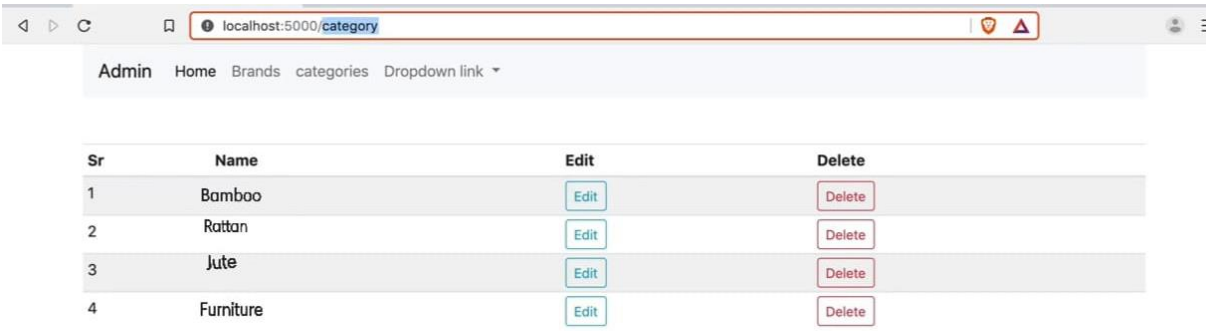
The seller can also edit/delete the product, brands, categories added by him/her. All these activities can be carried out in the admin page itself.



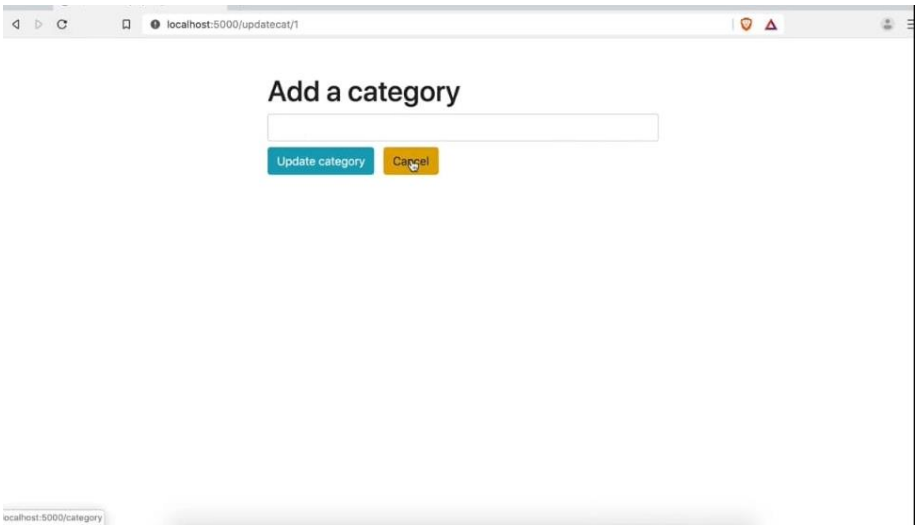
This page shows brands maintained by admin. These brands can be updated by just pressing the edit button.



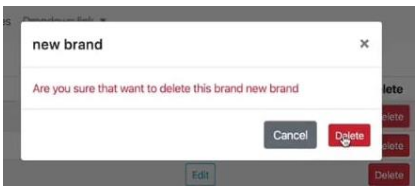
The categories page maintained by the admin(seller) is shown below:



The admin can upgrade a category by just pressing edit button.



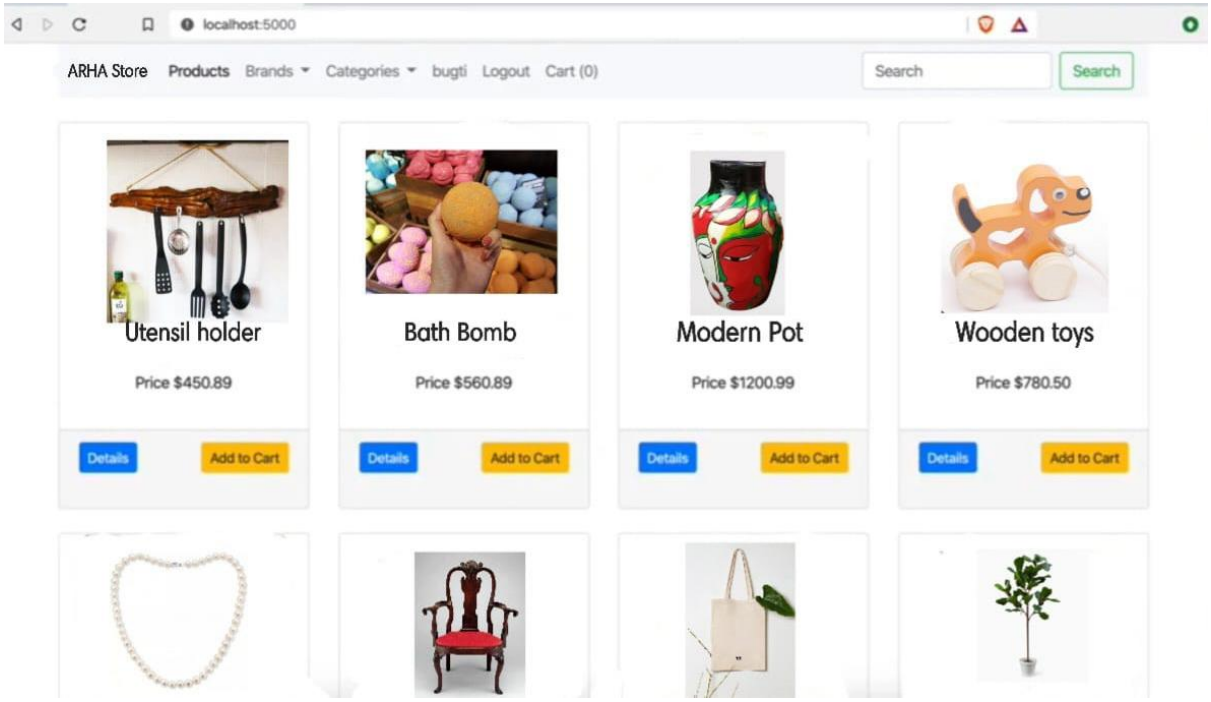
If the admin wishes to delete a category or brand, he/she can use the delete button in the respective pages.



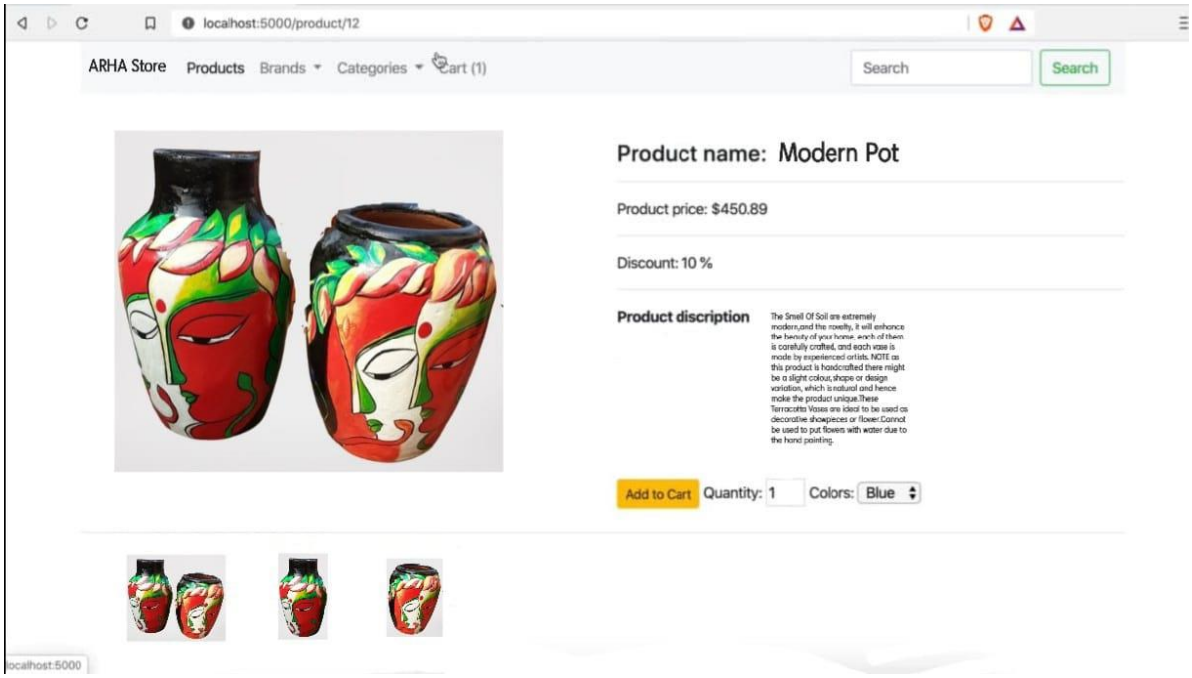
These are user end features that a seller can make use of for adding, editing, or deleting a product/brand/category.

Next, let us see the user features such as home page, products page, individual products page, shopping cart, payment page, checkout page.

The Home page of the app has navigation bar to product, categories, and so on. The home page looks like this:

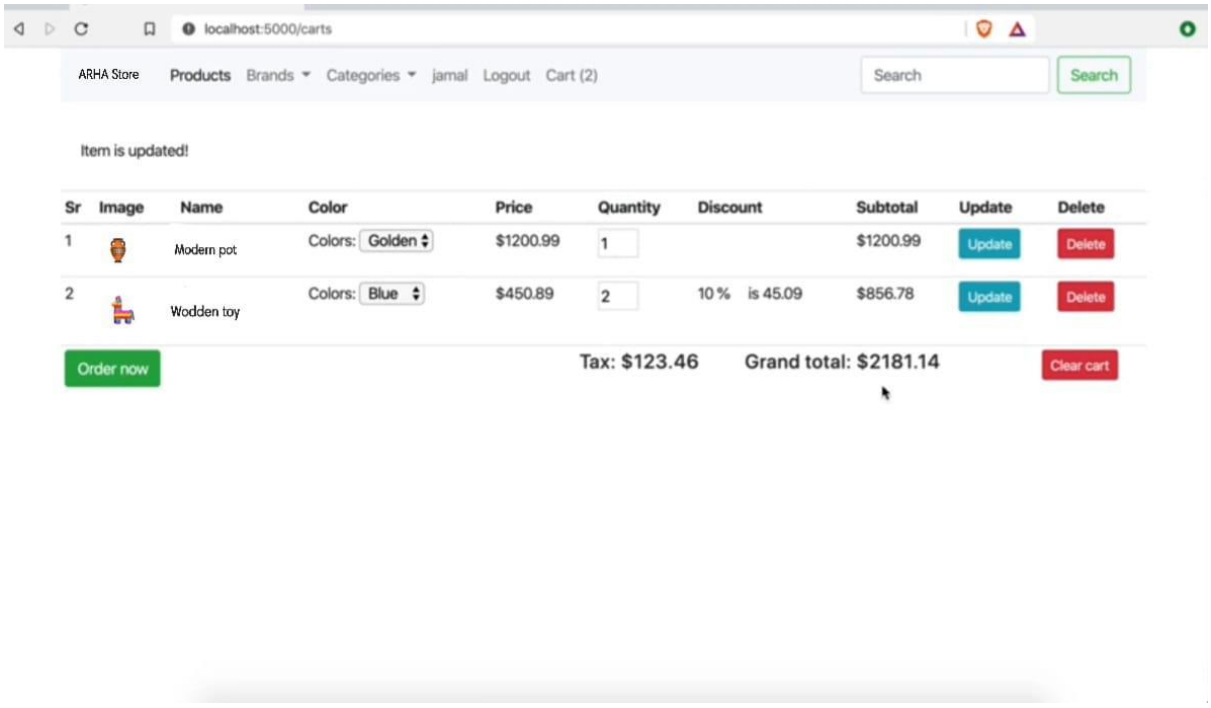


The individual product page has add to cart option and a summarized description about the product.



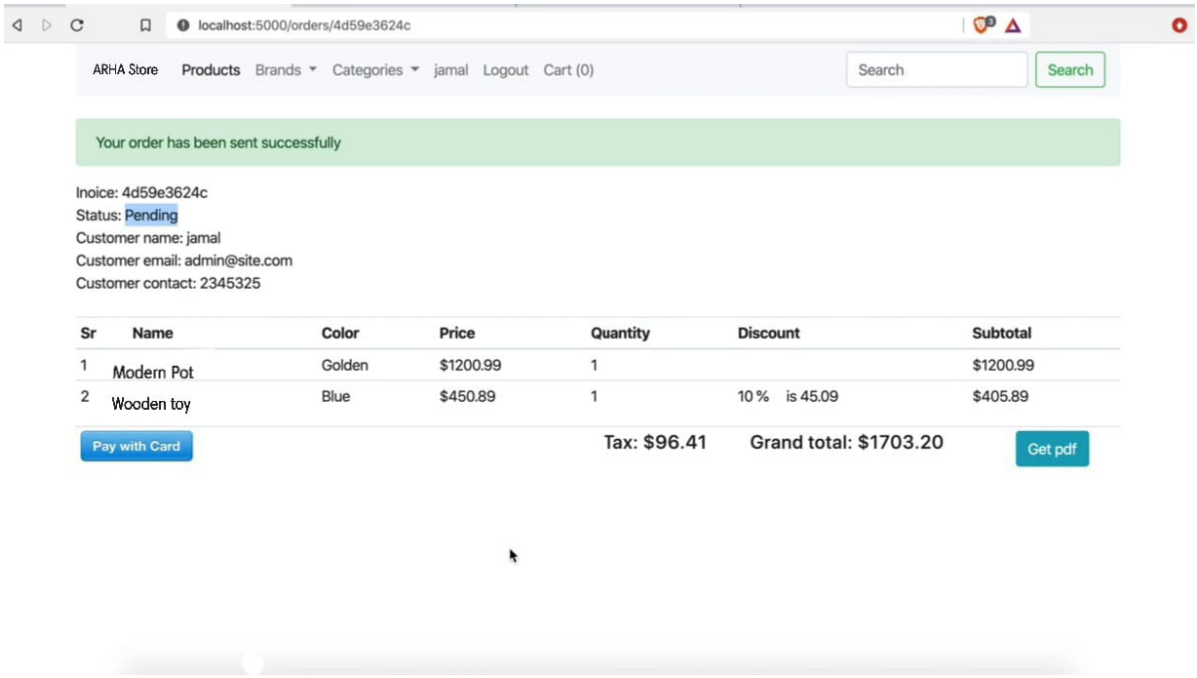
The user can add items to the cart by pressing add to cart button and can choose the colors, quantity.

The shopping cart page looks like this:



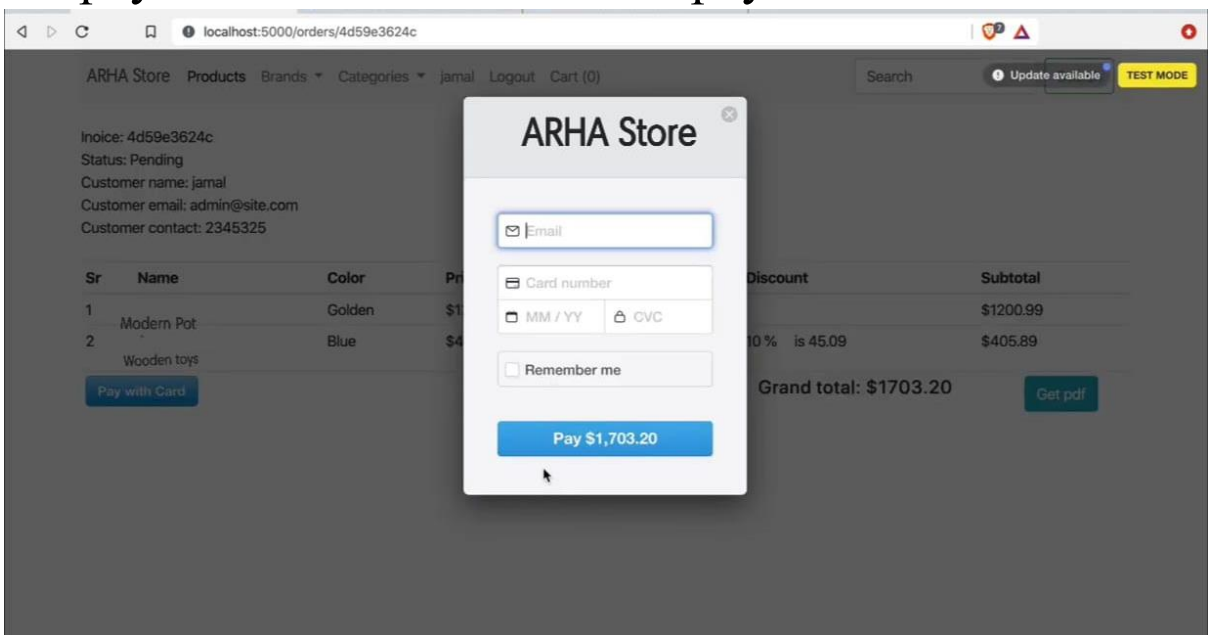
When the user touches the order now button, the user proceeds to the payment and checkout process.

The orders page will be like this:

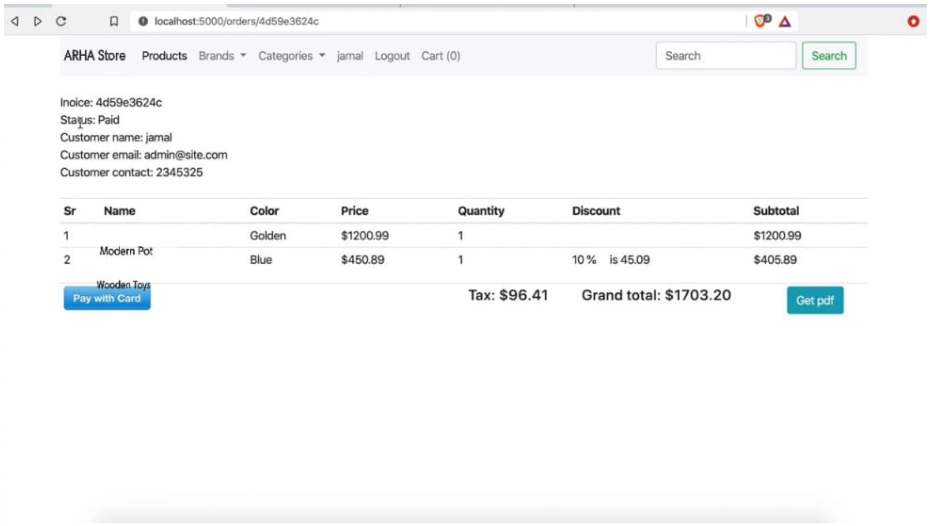


The user has placed an order but still the status is pending because the user has not yet paid. To proceed

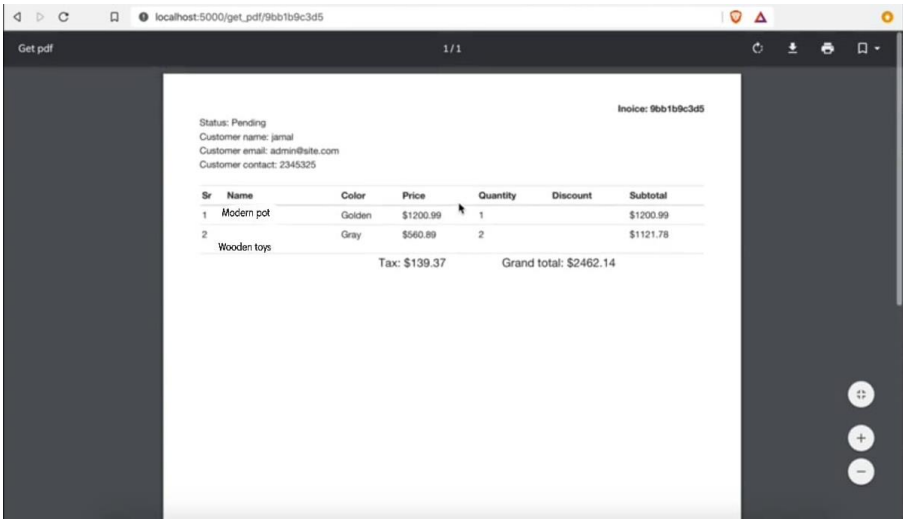
for payment the user must click pay with card.



Then the user must fill the details to continue the payment. After payment the status is paid.



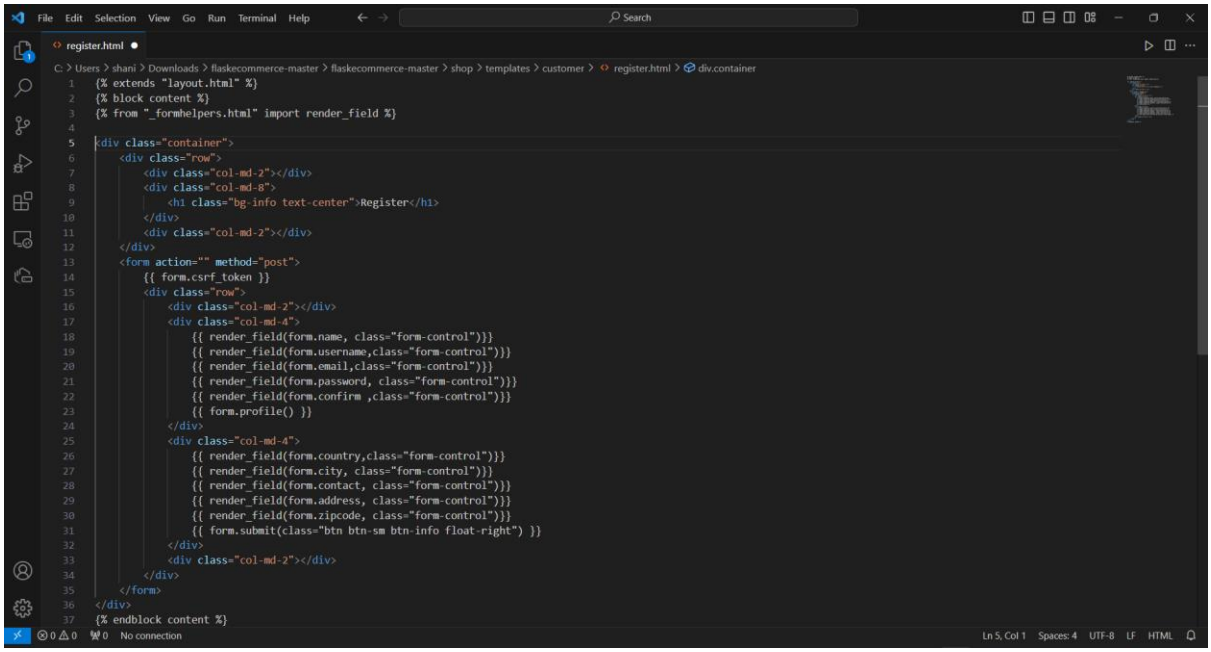
The user can get the pdf of the invoice generated by clicking get pdf.



Thus, the app comes out with a smooth payment and checkout process.

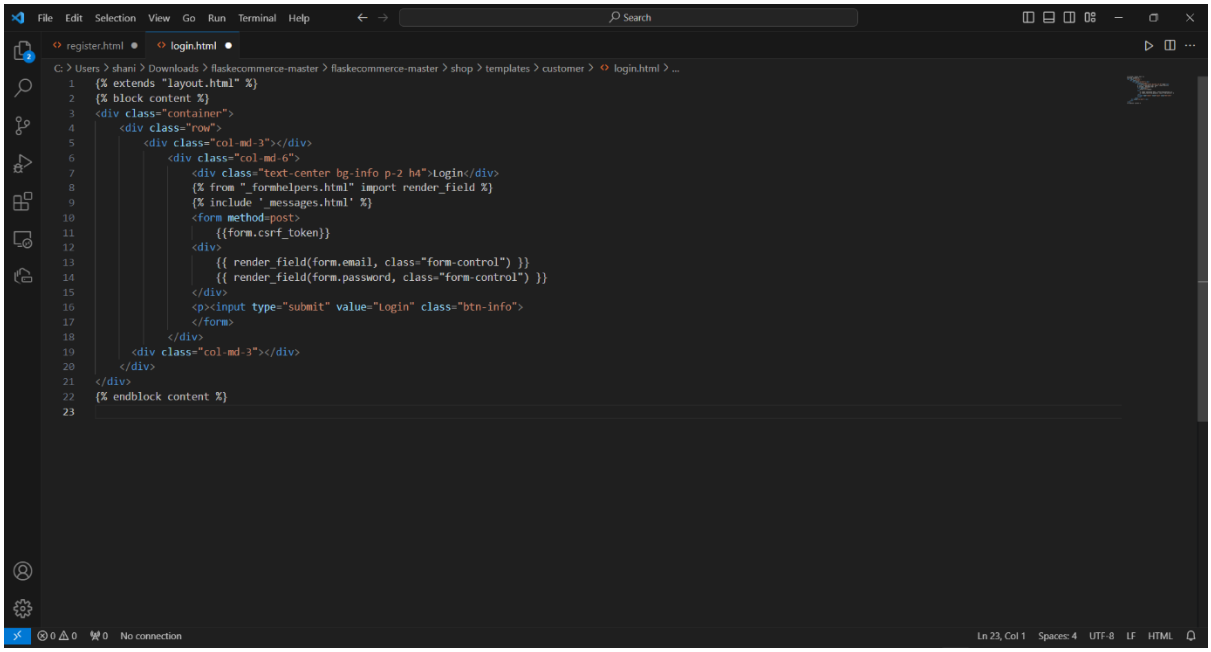
The frontend coding using html and CSS are shown below:

Register.html



CSRF tokens are used to validate the user's registration requests. This prevents the same mail id or user name to be registered again.

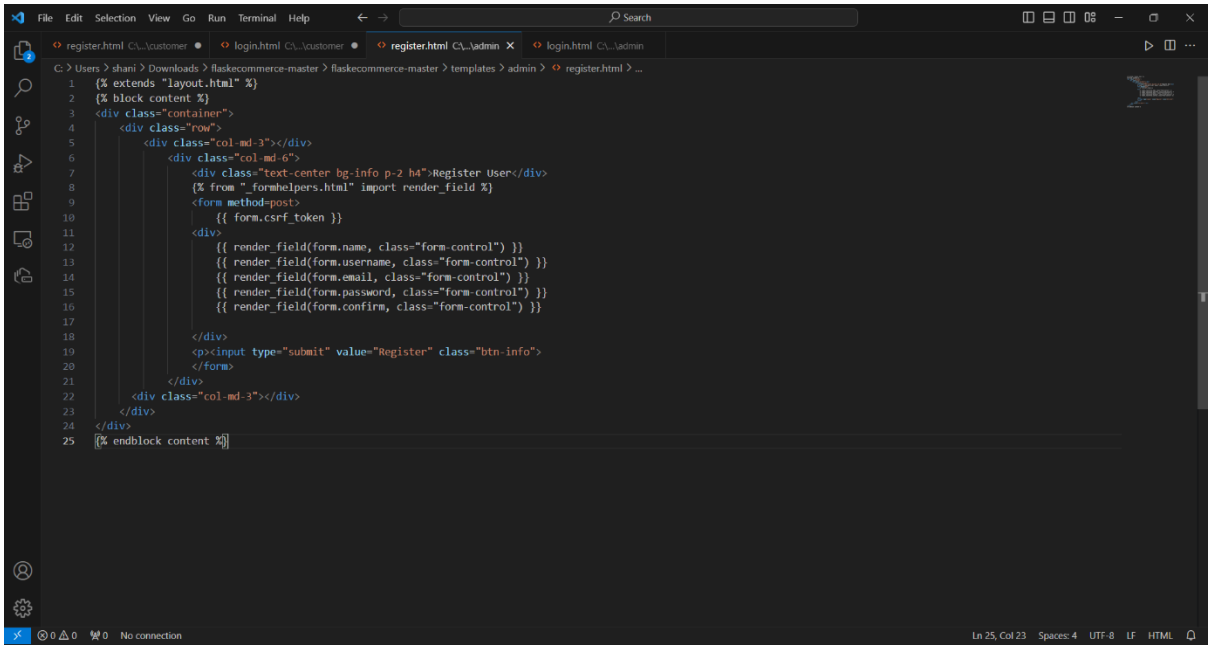
Login.html



The login page is also designed to use CSRF token. This prevents the user from logging in using an unregistered mail or a wrong password.

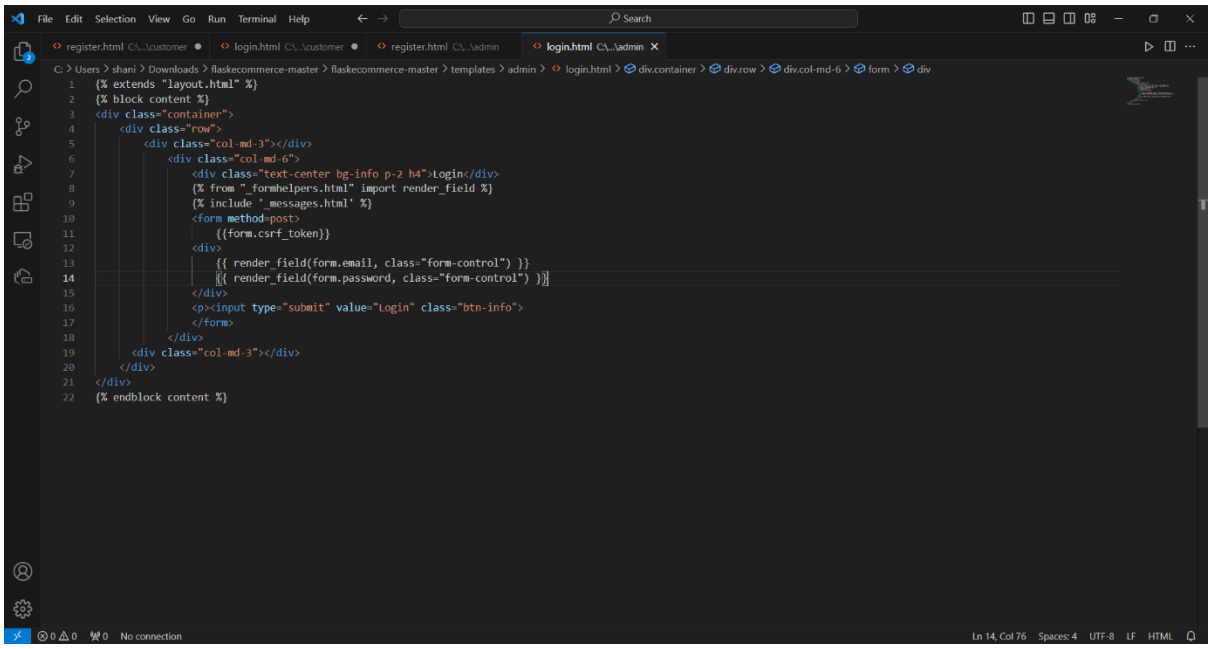
ADMIN:

Registration.html



```
{% extends "layout.html" %}
{% block content %}
<div class="container">
    <div class="row">
        <div class="col-md-3"></div>
            <div class="col-md-6">
                <div class="text-center bg-info p-2 h4">Register User</div>
                {% from "_formhelpers.html" import render_field %}
                <form method=post>
                    {{ form.csrf_token }}
                <div>
                    {{ render_field(form.name, class="form-control") }}
                    {{ render_field(form.username, class="form-control") }}
                    {{ render_field(form.email, class="form-control") }}
                    {{ render_field(form.password, class="form-control") }}
                    {{ render_field(form.confirm, class="form-control") }}

                </div>
                <p><input type="submit" value="Register" class="btn-info">
                </form>
            </div>
        <div class="col-md-3"></div>
    </div>
</div>
{% endblock content %}
```

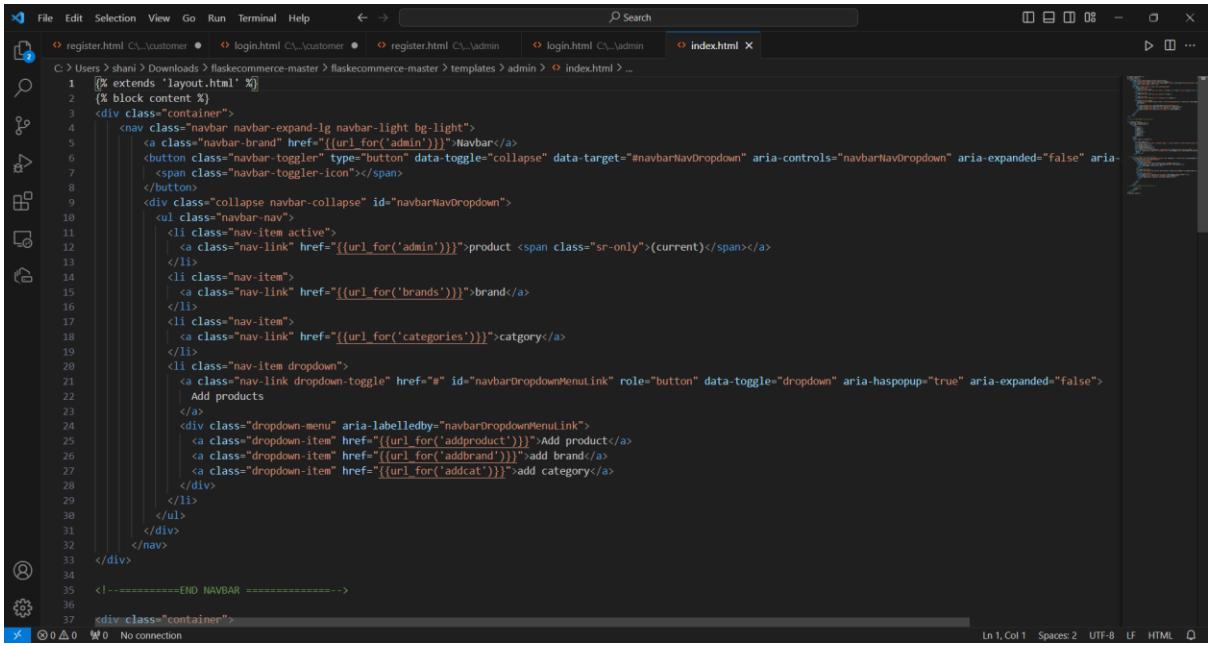
Login.html



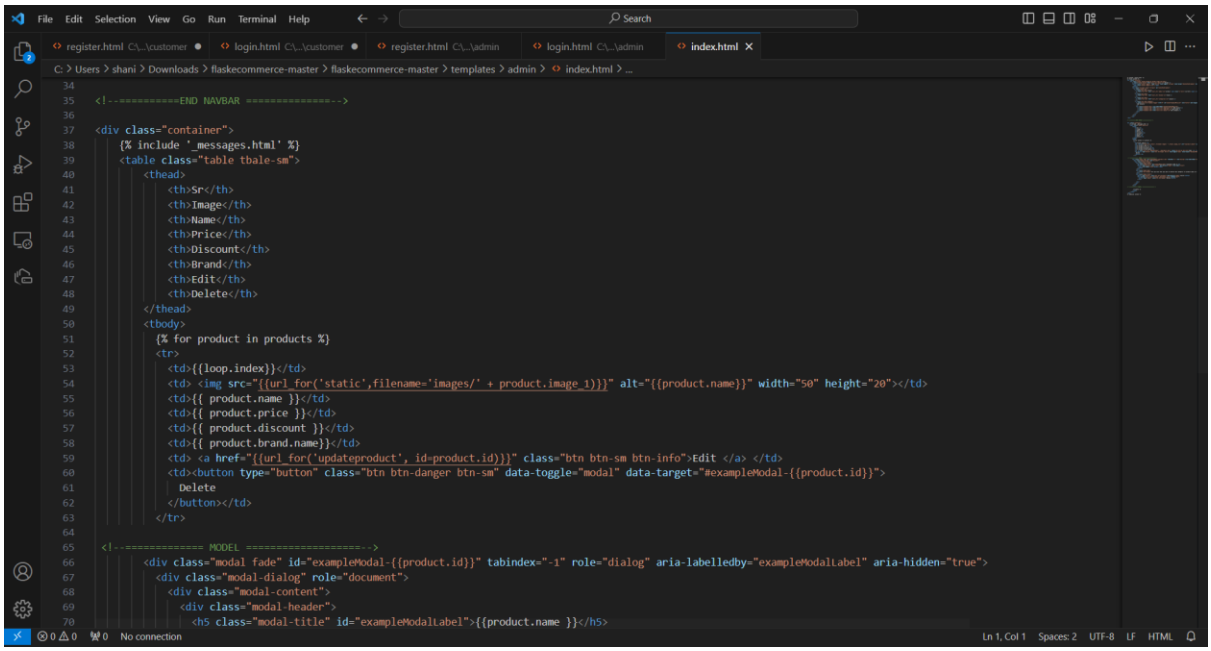
```
{% extends "layout.html" %}
{% block content %}
<div class="container">
    <div class="row">
        <div class="col-md-3"></div>
            <div class="col-md-6">
                <div class="text-center bg-info p-2 h4">Login</div>
                {% from "_formhelpers.html" import render_field %}
                {% include '_messages.html' %}
                <form method=post>
                    {{form.csrf_token}}
                <div>
                    {{ render_field(form.email, class="form-control") }}
                    {{ render_field(form.password, class="form-control") }}
                </div>
                <p><input type="submit" value="Login" class="btn-info">
                </form>
            </div>
        <div class="col-md-3"></div>
    </div>
</div>
{% endblock content %}
```

The registration and login page of admin also uses the CSRF tokens to avoid CSRF attacks as the app uses cookies to store the user details.

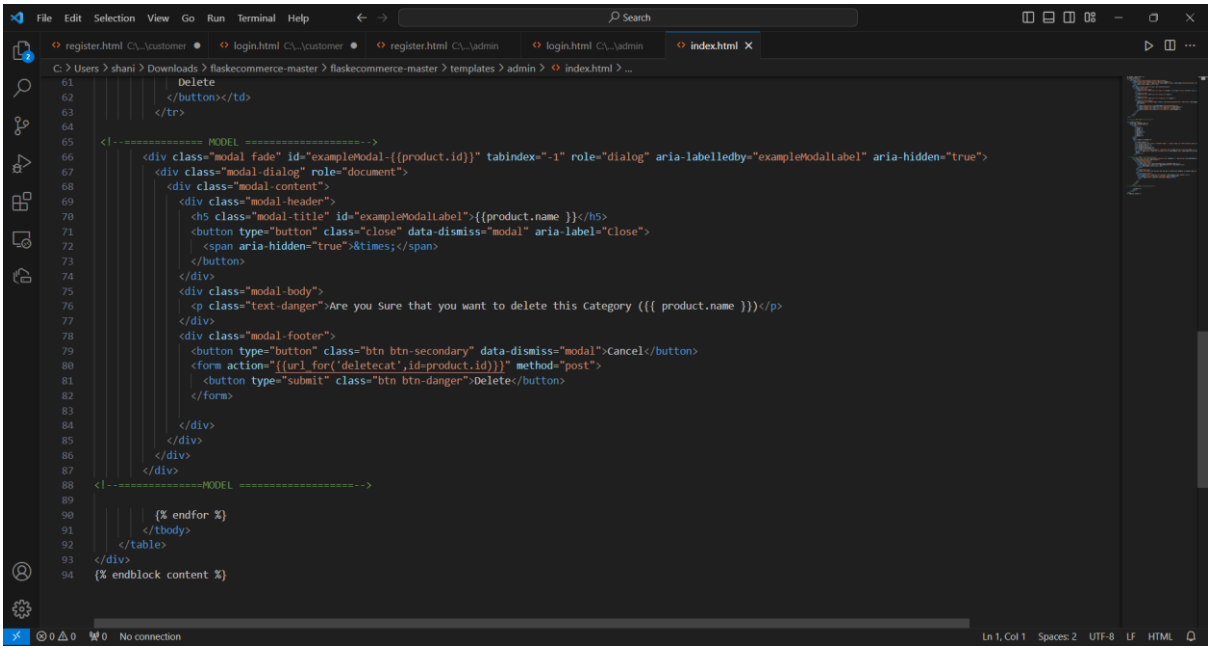
Index.html



```
{% extends 'layout.html' %}
{% block content %}
<div class="container">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="{{url_for('admin')}}">Navbar</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavDropdown">
          <ul class="navbar-nav">
            <li class="nav-item active">
              <a class="nav-link" href="{{url_for('admin')}}">product <span class="sr-only">(current)</span></a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{url_for('brands')}}">brand</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{url_for('categories')}}">catgory</a>
            </li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                Add products
              </a>
              <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                <a class="dropdown-item" href="{{url_for('addproduct')}}">Add product</a>
                <a class="dropdown-item" href="{{url_for('addbrand')}}">add brand</a>
                <a class="dropdown-item" href="{{url_for('addcat')}}">add category</a>
              </div>
            </li>
          </ul>
        </div>
      </nav>
</div>

<!--==========END NAVBAR ==============-->

<div class="container">
```



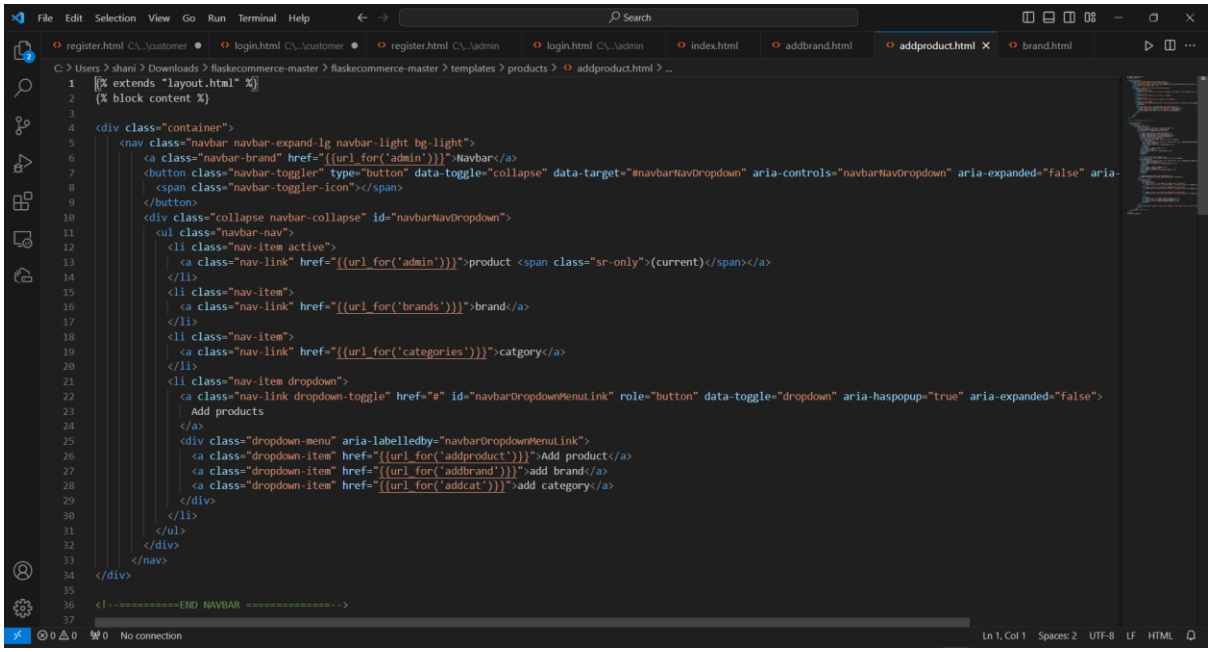
```
<!--==========END NAVBAR ==============-->

<div class="container">
    {% include '_messages.html' %}
    <table class="table tbale-sm">
        <thead>
            <th>Sr</th>
            <th>Image</th>
            <th>Name</th>
            <th>Price</th>
            <th>Discount</th>
            <th>Brand</th>
            <th>Edit</th>
            <th>Delete</th>
        </thead>
        <tbody>
          {% for product in products %}
          <tr>
            <td>{{loop.index}}</td>
            <td> <img src="{{url_for('static',filename='images/' + product.image_1)}}" alt="{{product.name}}" width="50" height="20"></td>
            <td>{{ product.name }}</td>
            <td>{{ product.price }}</td>
            <td>{{ product.discount }}</td>
            <td>{{ product.brand.name}}</td>
            <td> <a href="{{url_for('updateproduct', id=product.id)}}" class="btn btn-sm btn-info">Edit </a> </td>
            <td><button type="button" class="btn btn-danger btn-sm" data-toggle="modal" data-target="#exampleModal-{{product.id}}">
              Delete
            </button></td>
          </tr>

 <!--============= MODEL ==================-->
        <div class="modal fade" id="exampleModal-{{product.id}}" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
          <div class="modal-dialog" role="document">
            <div class="modal-content">
              <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">{{product.name }}</h5>
```



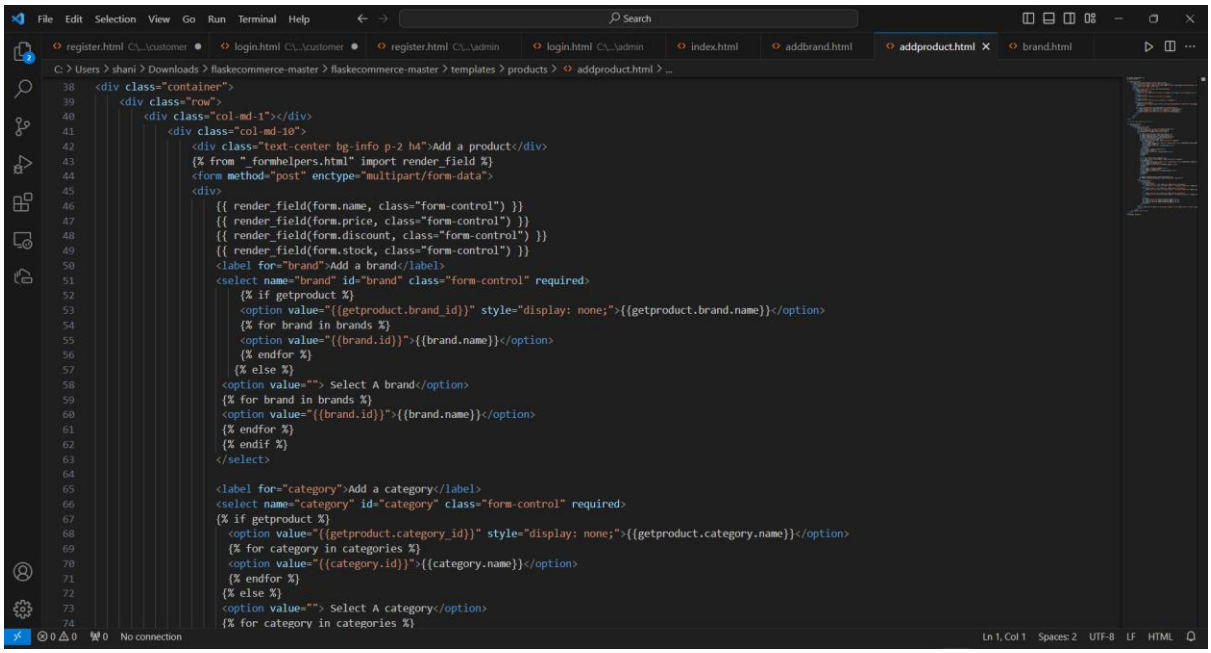
This is the coding for admin page from which the admin can navigate to pages with features of add/edit/delete products/categories and perform these activities.

Products:

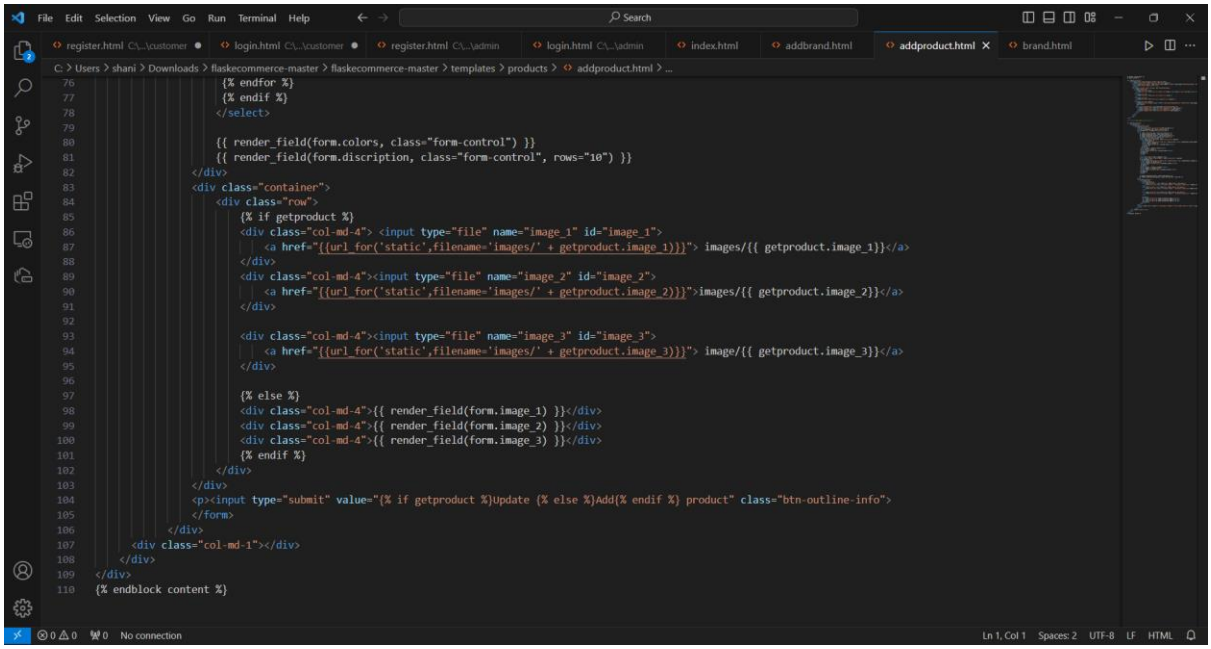
Add product.html



```
{% extends "layout.html" %}
{% block content %}

<div class="container">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="{{url_for('admin')}}">Navbar</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavDropdown">
          <ul class="navbar-nav">
            <li class="nav-item active">
              <a class="nav-link" href="{{url_for('admin')}}">product <span class="sr-only">(current)</span></a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{url_for('brands')}}">brand</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{url_for('categories')}}">catgory</a>
            </li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                Add products
              </a>
              <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                <a class="dropdown-item" href="{{url_for('addproduct')}}">Add product</a>
                <a class="dropdown-item" href="{{url_for('addbrand')}}">add brand</a>
                <a class="dropdown-item" href="{{url_for('addcat')}}">add category</a>
              </div>
            </li>
          </ul>
        </div>
      </nav>
</div>

<!--==========END NAVBAR ==============-->
```



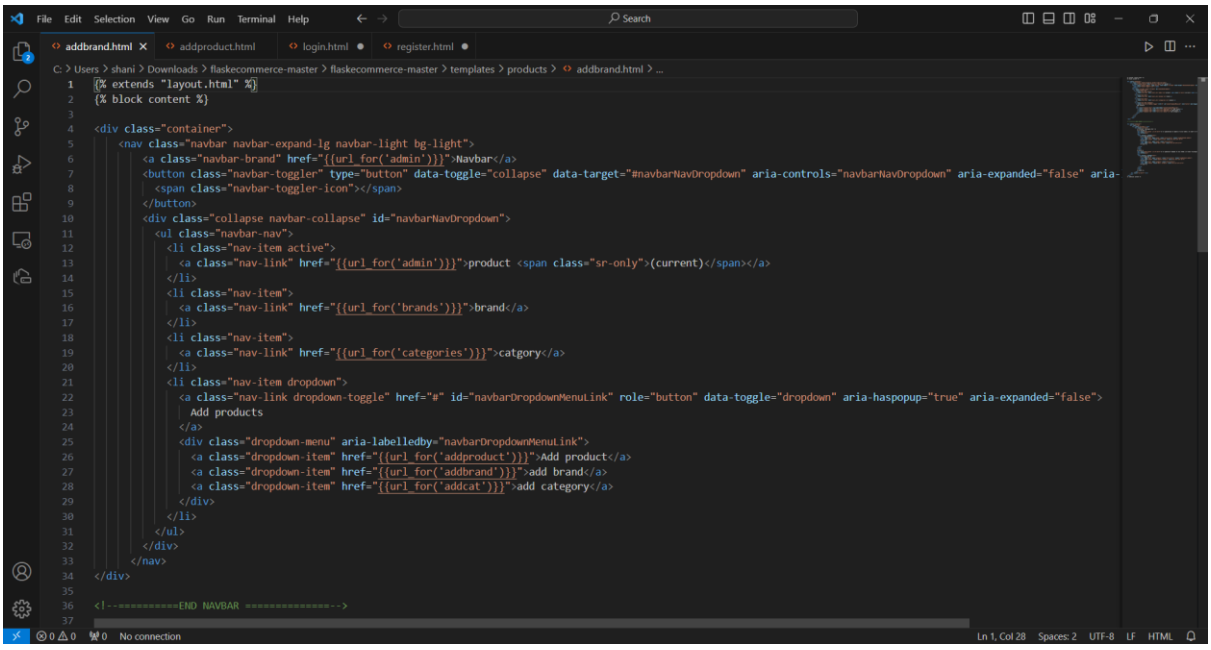
```
<div class="container">
    <div class="row">
        <div class="col-md-1"></div>
            <div class="col-md-10">
                <div class="text-center bg-info p-2 h4">Add a product</div>
                {% from "_formhelpers.html" import render_field %}
                <form method="post" enctype="multipart/form-data">
                <div>
                    {{ render_field(form.name, class="form-control") }}
                    {{ render_field(form.price, class="form-control") }}
                    {{ render_field(form.discount, class="form-control") }}
                    {{ render_field(form.stock, class="form-control") }}
                    <label for="brand">Add a brand</label>
                    <select name="brand" id="brand" class="form-control" required>
                        {% if getproduct %}
                        <option value="{{getproduct.brand_id}}" style="display: none;">{{getproduct.brand.name}}</option>
                        {% for brand in brands %}
                        <option value="{{brand.id}}">{{brand.name}}</option>
                        {% endfor %}
                       {% else %}
                      <option value=""> Select A brand</option>
                      {% for brand in brands %}
                      <option value="{{brand.id}}">{{brand.name}}</option>
                      {% endfor %}
                      {% endif %}
                      </select>

                      <label for="category">Add a category</label>
                      <select name="category" id="category" class="form-control" required>
                      {% if getproduct %}
                        <option value="{{getproduct.category_id}}" style="display: none;">{{getproduct.category.name}}</option>
                        {% for category in categories %}
                        <option value="{{category.id}}">{{category.name}}</option>
                        {% endfor %}
                       {% else %}
                       <option value=""> Select A category</option>
                       {% for category in categories %}
```



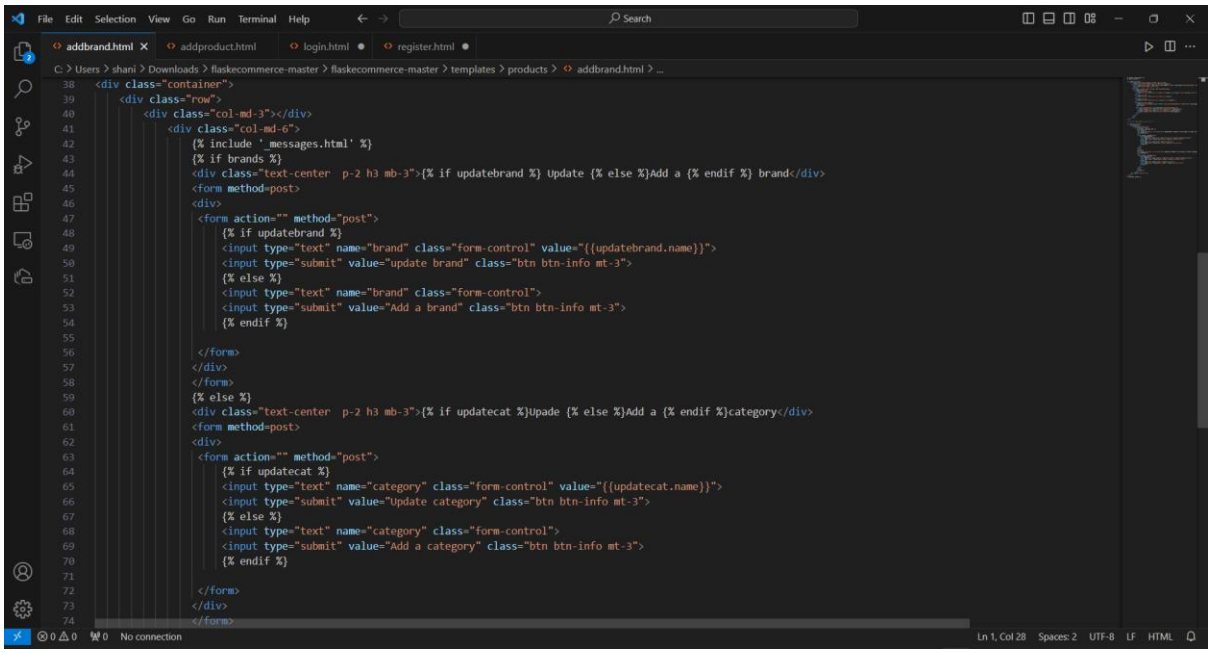
```
                        {% endfor %}
                        {% endif %}
                        </select>

                        {{ render_field(form.colors, class="form-control") }}
                        {{ render_field(form.discription, class="form-control", rows="10") }}
                    </div>
                    <div class="container">
                        <div class="row">
                            {% if getproduct %}
                            <div class="col-md-4"> <input type="file" name="image_1" id="image_1">
                                <a href="{{url_for('static',filename='images/' + getproduct.image_1)}}"> images/{{ getproduct.image_1}}</a>
                            </div>
                            <div class="col-md-4"><input type="file" name="image_2" id="image_2">
                                <a href="{{url_for('static',filename='images/' + getproduct.image_2)}}">images/{{ getproduct.image_2}}</a>
                            </div>

                            <div class="col-md-4"><input type="file" name="image_3" id="image_3">
                                <a href="{{url_for('static',filename='images/' + getproduct.image_3)}}"> image/{{ getproduct.image_3}}</a>
                            </div>

                            {% else %}
                            <div class="col-md-4">{{ render_field(form.image_1) }}</div>
                            <div class="col-md-4">{{ render_field(form.image_2) }}</div>
                            <div class="col-md-4">{{ render_field(form.image_3) }}</div>
                            {% endif %}
                        </div>
                    </div>
                    <p><input type="submit" value="{% if getproduct %}Update {% else %}Add{% endif %} product" class="btn-outline-info">
                    </form>
                </div>
            <div class="col-md-1"></div>
        </div>
    </div>
    {% endblock content %}
```

This is the coding for adding products.

Add brands.html



```
{% extends "layout.html" %}
{% block content %}

<div class="container">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="{{url_for('admin')}}">Navbar</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavDropdown">
          <ul class="navbar-nav">
            <li class="nav-item active">
              <a class="nav-link" href="{{url_for('admin')}}">product <span class="sr-only">(current)</span></a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{url_for('brands')}}">brand</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="{{url_for('categories')}}">catgory</a>
            </li>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                Add products
              </a>
              <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                <a class="dropdown-item" href="{{url_for('addproduct')}}">Add product</a>
                <a class="dropdown-item" href="{{url_for('addbrand')}}">add brand</a>
                <a class="dropdown-item" href="{{url_for('addcat')}}">add category</a>
              </div>
            </li>
          </ul>
        </div>
      </nav>
</div>

<!--==========END NAVBAR ==============-->
```


File Edit Selection View Go Run Terminal Help
Search
addbrand.html
addproduct.html
login.html
register.html
C: > Users > shani > Downloads > flaskecommerce-master > flaskecommerce-master > templates > products > addbrand.html > ...

{% include '_messages.html' %}
{% if brands %}
{% if updatebrand %} Update {% else %}Add a {% endif %} brand
<form method=post>

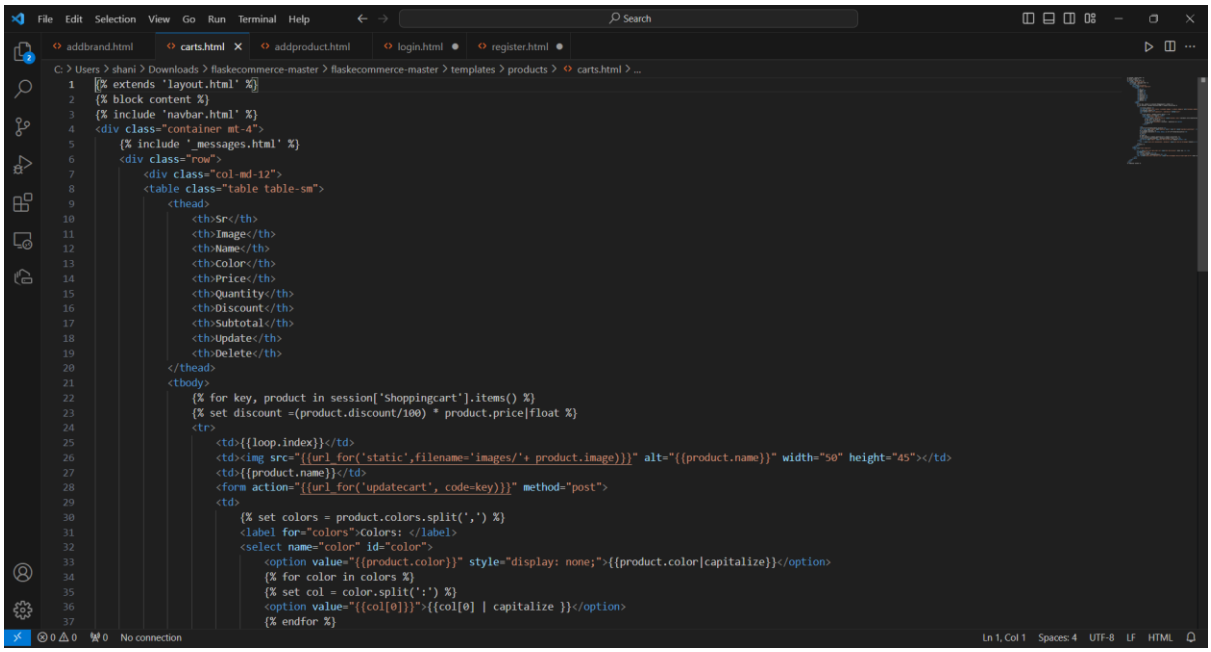
<form action="" method="post">
{% if updatebrand %}
<input type="text" name="brand" class="form-control" value="{{updatebrand.name}}">
<input type="submit" value="update brand" class="btn btn-info mt-3">
{% else %}
<input type="text" name="brand" class="form-control">
<input type="submit" value="Add a brand" class="btn btn-info mt-3">
{% endif %}
</form>

</form>
{% else %}
{% if updatecat %}Upade {% else %}Add a {% endif %}category
<form method=post>

<form action="" method="post">
{% if updatecat %}
<input type="text" name="category" class="form-control" value="{{updatecat.name}}">
<input type="submit" value="Update category" class="btn btn-info mt-3">
{% else %}
<input type="text" name="category" class="form-control">
<input type="submit" value="Add a category" class="btn btn-info mt-3">
{% endif %}
</form>

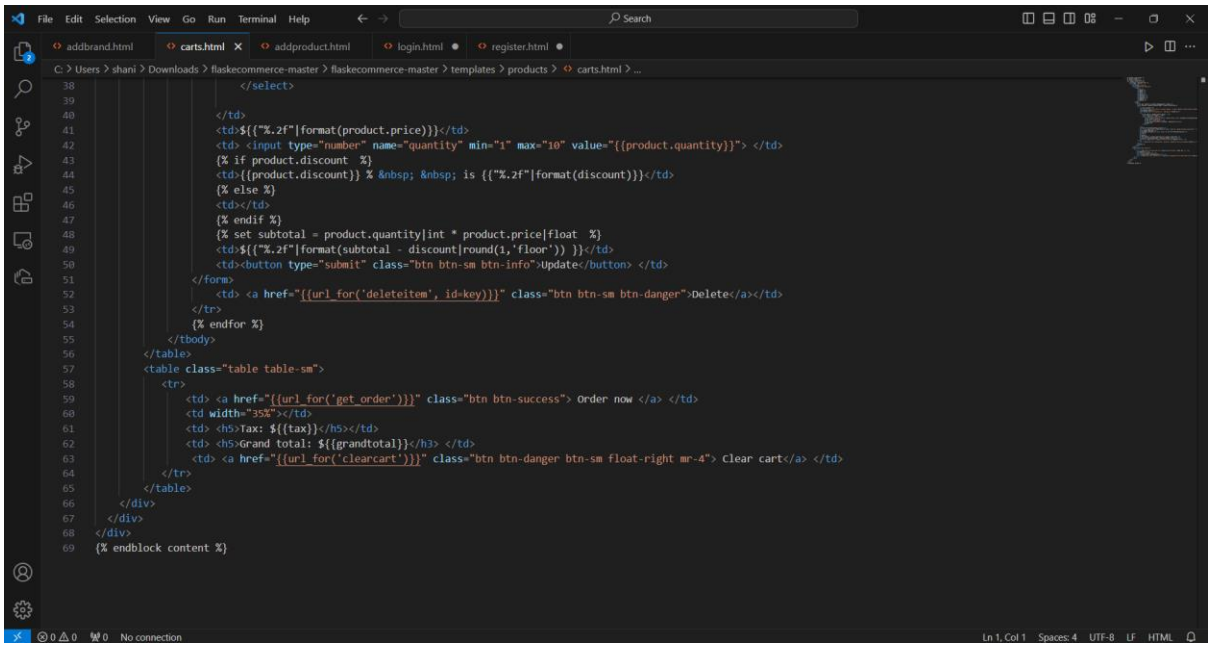
</form>
No connection
Ln 1, Col 28 Spaces: 2 UTF-8 LF HTML

Carts.html


File Edit Selection View Go Run Terminal Help
Search
addbrand.html
carts.html
addproduct.html
login.html
register.html
C: > Users > shani > Downloads > flaskecommerce-master > flaskecommerce-master > templates > products > carts.html > ...
{% extends 'layout.html' %}
{% block content %}
{% include 'navbar.html' %}

{% include '_messages.html' %}

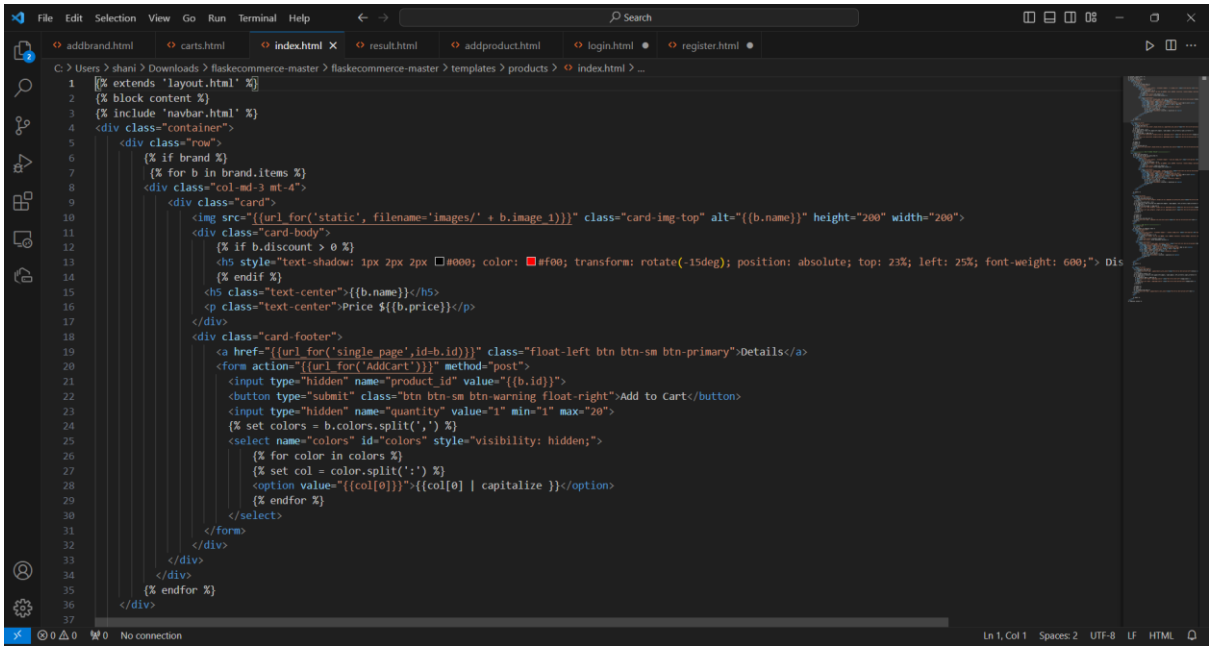
<table class="table table-sm">
<thead>
<th>Sr</th>
<th>Image</th>
<th>Name</th>
<th>Color</th>
<th>Price</th>
<th>Quantity</th>
<th>Discount</th>
<th>Subtotal</th>
<th>Update</th>
<th>Delete</th>
</thead>
<tbody>
{% for key, product in session['Shoppingcart'].items() %}
{% set discount =(product.discount/100) * product.price|float %}
<tr>
<td>{{loop.index}}</td>
<td><img src="{{url_for('static',filename='images/'+ product.image)}}" alt="{{product.name}}" width="50" height="45"></td>
<td>{{product.name}}</td>
<form action="{{url_for('updatecart', code=key)}}" method="post">
<td>
{% set colors = product.colors.split(',') %}
<label for="colors">Colors: </label>
<select name="color" id="color">
<option value="{{product.color}}" style="display: none;">{{product.color|capitalize}}</option>
{% for color in colors %}
{% set col = color.split(':') %}
<option value="{{col[0]}}">{{col[0] | capitalize }}</option>
{% endfor %}
No connection
Ln 1, Col 1 Spaces: 4 UTF-8 LF HTML



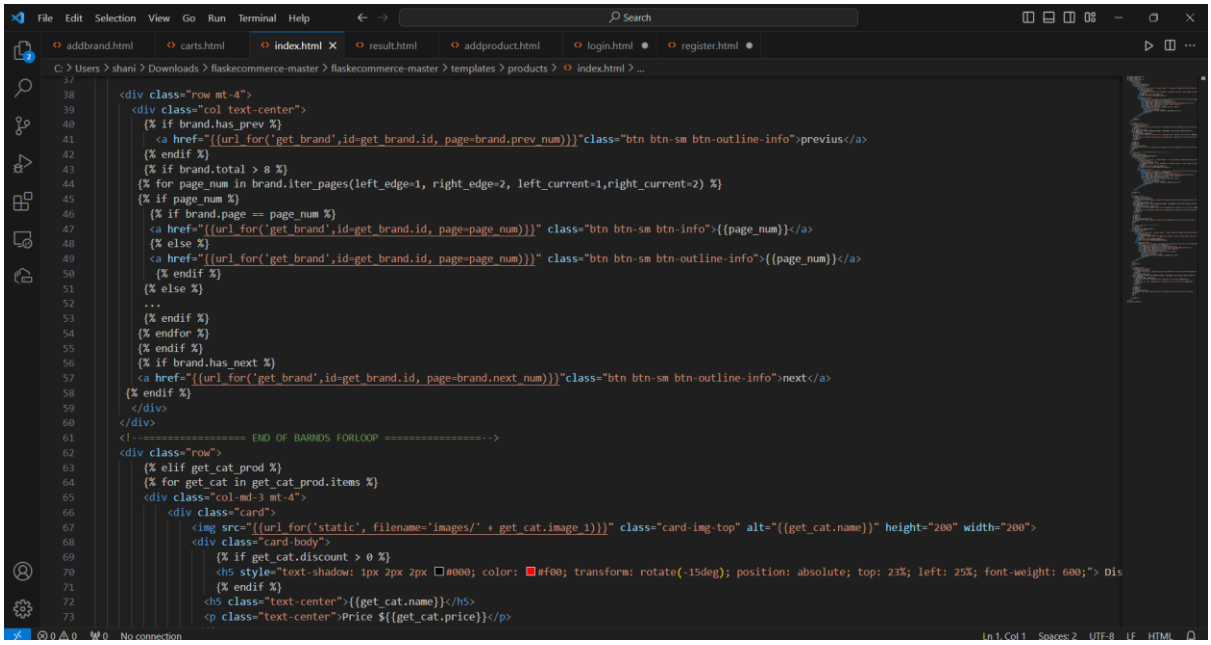
```
                                </select>

                            </td>
                            <td>${{"%.2f"|format(product.price)}}</td>
                            <td> <input type="number" name="quantity" min="1" max="10" value="{{product.quantity}}"> </td>
                            {% if product.discount  %}
                            <td>{{product.discount}} %     is {{"%.2f"|format(discount)}}</td>
                            {% else %}
                            <td></td>
                            {% endif %}
                            {% set subtotal = product.quantity|int * product.price|float  %}
                            <td>${{"%.2f"|format(subtotal - discount|round(1,'floor')) }}</td>
                            <td><button type="submit" class="btn btn-sm btn-info">Update</button> </td>
                        </form>
                            <td> <a href="{{url_for('deleteitem', id=key)}}" class="btn btn-sm btn-danger">Delete</a></td>
                        </tr>
                        {% endfor %}
                    </tbody>
                </table>
                <table class="table table-sm">
                    <tr>
                        <td> <a href="{{url_for('get_order')}}" class="btn btn-success"> Order now </a> </td>
                        <td width="35%"></td>
                        <td> <h5>Tax: ${{tax}}</h5></td>
                        <td> <h5>Grand total: ${{grandtotal}}</h3> </td>
                         <td> <a href="{{url_for('clearcart')}}" class="btn btn-danger btn-sm float-right mr-4"> Clear cart</a> </td>
                    </tr>
                </table>
            </div>
        </div>
    </div>
    {% endblock content %}
```

This is the coding to display cart. This page has the items added to cart by the customers. From this page the customer can clear the cart items or can move to the payment and checkout process by ordering the items.

Index.html



```
{% extends 'layout.html' %}
{% block content %}
{% include 'navbar.html' %}
<div class="container">
    <div class="row">
        {% if brand %}
        {% for b in brand.items %}
        <div class="col-md-3 mt-4">
            <div class="card">
                <img src="{{url_for('static', filename='images/' + b.image_1)}}" class="card-img-top" alt="{{b.name}}" height="200" width="200">
                <div class="card-body">
                    {% if b.discount > 0 %}
                    <h5 style="text-shadow: 1px 2px 2px #000; color: #f00; transform: rotate(-15deg); position: absolute; top: 23%; left: 25%; font-weight: 600;"> Dis
                    {% endif %}
                  <h5 class="text-center">{{b.name}}</h5>
                  <p class="text-center">Price ${{b.price}}</p>
                </div>
                <div class="card-footer">
                    <a href="{{url_for('single_page',id=b.id)}}" class="float-left btn btn-sm btn-primary">Details</a>
                    <form action="{{url_for('AddCart')}}" method="post">
                      <input type="hidden" name="product_id" value="{{b.id}}">
                      <button type="submit" class="btn btn-sm btn-warning float-right">Add to Cart</button>
                      <input type="hidden" name="quantity" value="1" min="1" max="20">
                      {% set colors = b.colors.split(',') %}
                      <select name="colors" id="colors" style="visibility: hidden;">
                          {% for color in colors %}
                          {% set col = color.split(':') %}
                          <option value="{{col[0]}}">{{col[0] | capitalize }}</option>
                          {% endfor %}
                      </select>
                    </form>
                </div>
            </div>
        </div>
        {% endfor %}
    </div>
```


File Edit Selection View Go Run Terminal Help
Search
addbrand.html
carts.html
index.html
result.html
addproduct.html
login.html
register.html
C: > Users > shani > Downloads > flaskecommerce-master > flaskecommerce-master > templates > products > index.html > ...

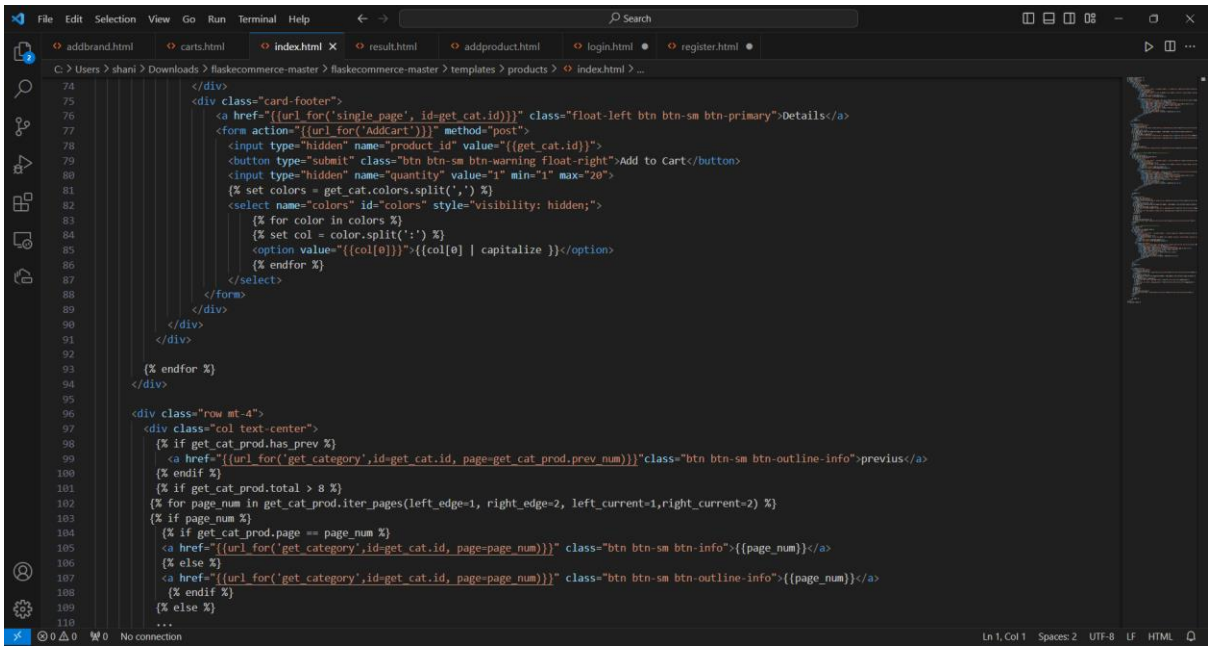
{% if brand.has_prev %}
previus
{% endif %}
{% if brand.total > 8 %}
{% for page_num in brand.iter_pages(left_edge=1, right_edge=2, left_current=1,right_current=2) %}
{% if page_num %}
{% if brand.page == page_num %}
{{page_num}}
{% else %}
{{page_num}}
{% endif %}
{% else %}
...
{% endif %}
{% endfor %}
{% endif %}
{% if brand.has_next %}
next
{% endif %}

<!--================ END OF BARNDS FORLOOP ================-->

{% elif get_cat_prod %}
{% for get_cat in get_cat_prod.items %}

<img src="{{url_for('static', filename='images/' + get_cat.image_1)}}" class="card-img-top" alt="{{get_cat.name}}" height="200" width="200">

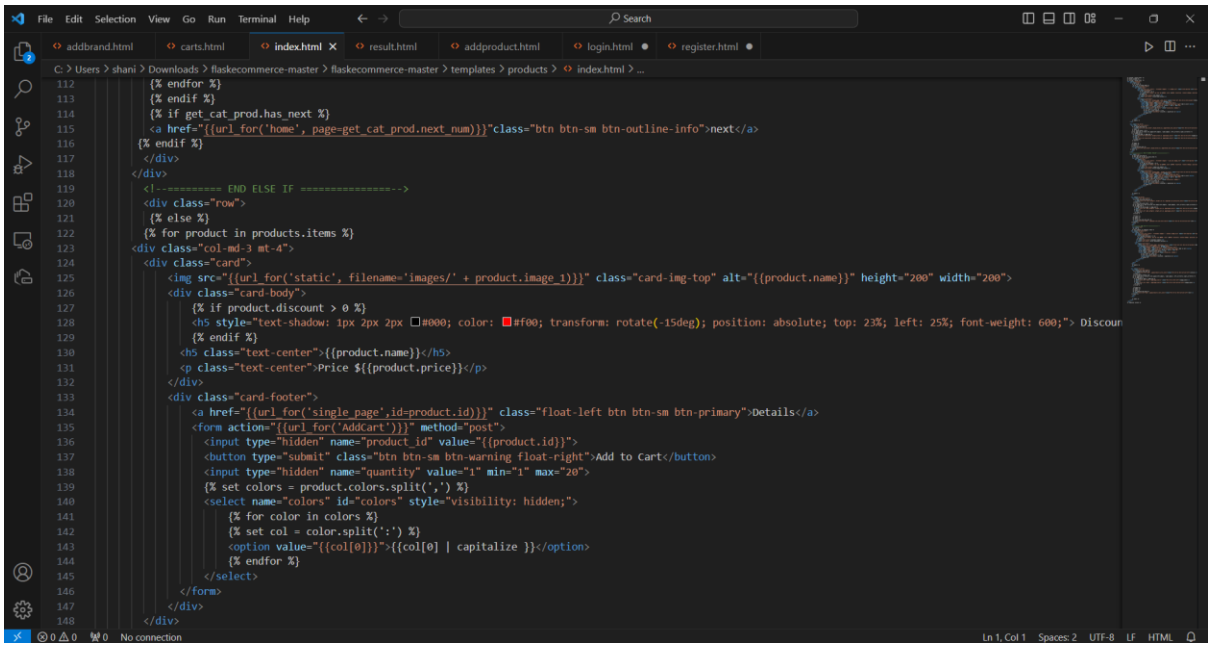
{% if get_cat.discount > 0 %}
<h5 style="text-shadow: 1px 2px 2px #000; color: #f00; transform: rotate(-15deg); position: absolute; top: 23%; left: 25%; font-weight: 600;"> Dis
{% endif %}
<h5 class="text-center">{{get_cat.name}}</h5>
Price ${{get_cat.price}}
No connection
Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML


File Edit Selection View Go Run Terminal Help
Search
addbrand.html
carts.html
index.html
result.html
addproduct.html
login.html
register.html
C: > Users > shani > Downloads > flaskecommerce-master > flaskecommerce-master > templates > products > index.html > ...

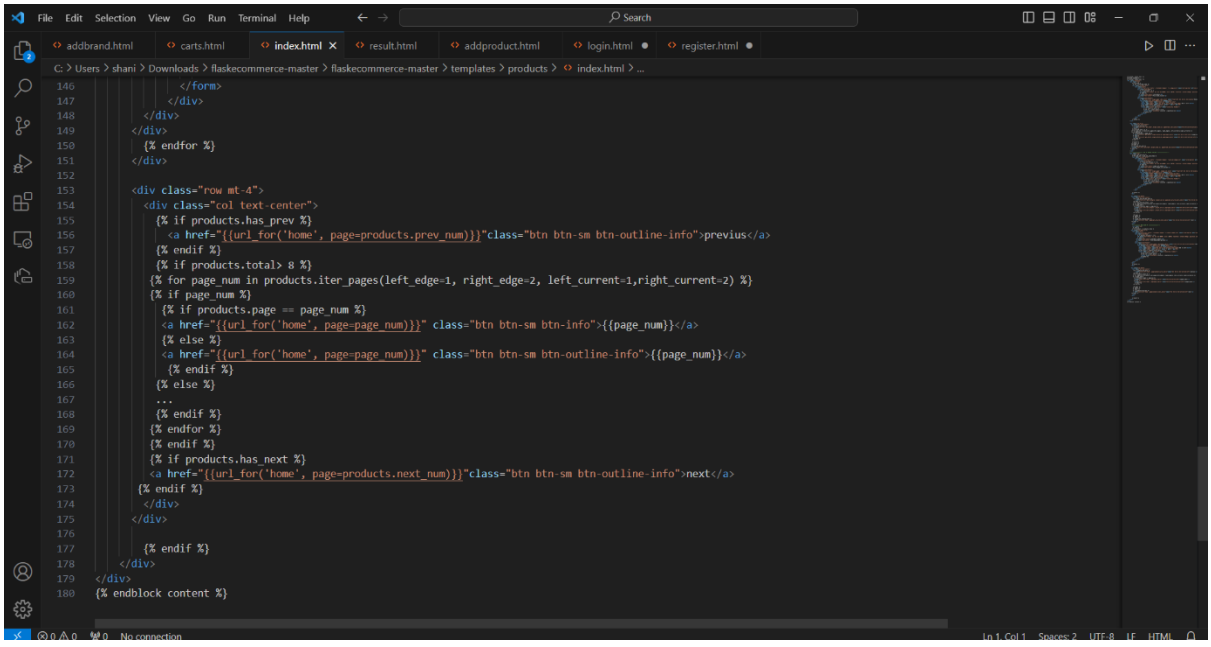
Details
<form action="{{url_for('AddCart')}}" method="post">
<input type="hidden" name="product_id" value="{{get_cat.id}}">
<button type="submit" class="btn btn-sm btn-warning float-right">Add to Cart</button>
<input type="hidden" name="quantity" value="1" min="1" max="20">
{% set colors = get_cat.colors.split(',') %}
<select name="colors" id="colors" style="visibility: hidden;">
{% for color in colors %}
{% set col = color.split(':') %}
<option value="{{col[0]}}">{{col[0] | capitalize }}</option>
{% endfor %}
</select>
</form>

{% endfor %}

{% if get_cat_prod.has_prev %}
previus
{% endif %}
{% if get_cat_prod.total > 8 %}
{% for page_num in get_cat_prod.iter_pages(left_edge=1, right_edge=2, left_current=1,right_current=2) %}
{% if page_num %}
{% if get_cat_prod.page == page_num %}
{{page_num}}
{% else %}
{{page_num}}
{% endif %}
{% else %}
...
No connection
Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML



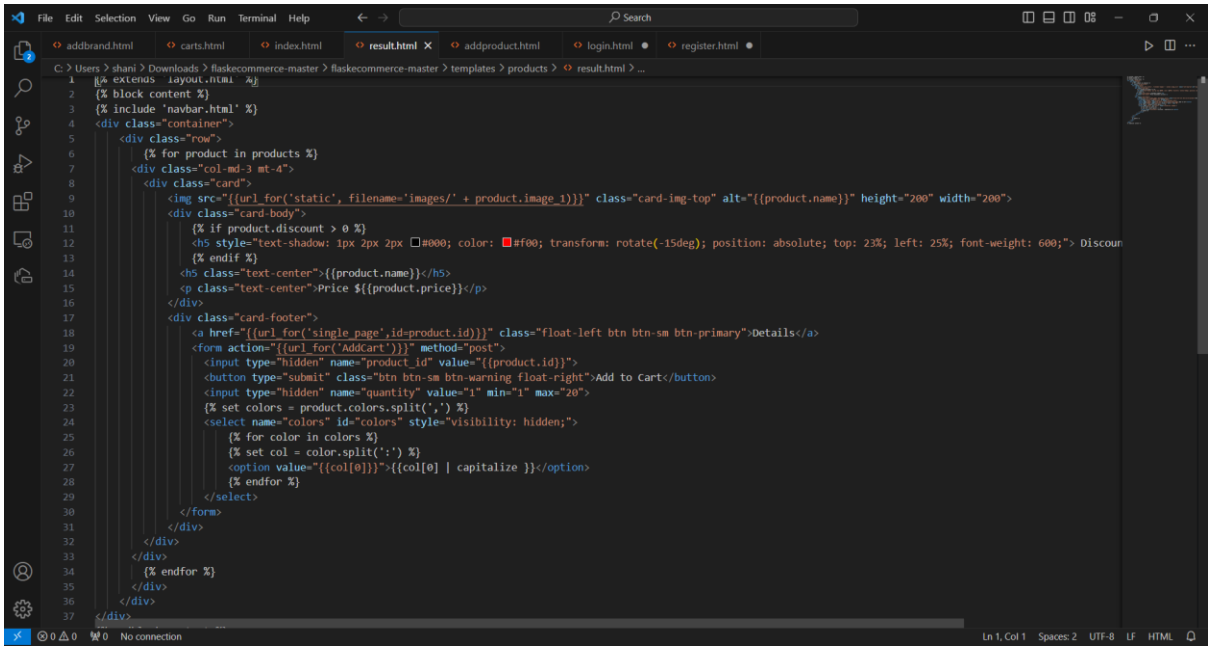
```
{% endfor %}
{% endif %}
{% if get_cat_prod.has_next %}
<a href="{{url_for('home', page=get_cat_prod.next_num)}}"class="btn btn-sm btn-outline-info">next</a>
{% endif %}
</div>
</div>
<!--========= END ELSE IF ==============-->
<div class="row">
{% else %}
{% for product in products.items %}
<div class="col-md-3 mt-4">
<div class="card">
<img src="{{url_for('static', filename='images/' + product.image_1)}}" class="card-img-top" alt="{{product.name}}" height="200" width="200">
<div class="card-body">
{% if product.discount > 0 %}
<h5 style="text-shadow: 1px 2px 2px #000; color: #f00; transform: rotate(-15deg); position: absolute; top: 23%; left: 25%; font-weight: 600;"> Discoun
{% endif %}
<h5 class="text-center">{{product.name}}</h5>
<p class="text-center">Price ${{product.price}}</p>
</div>
<div class="card-footer">
<a href="{{url_for('single_page',id=product.id)}}" class="float-left btn btn-sm btn-primary">Details</a>
<form action="{{url_for('AddCart')}}" method="post">
<input type="hidden" name="product_id" value="{{product.id}}">
<button type="submit" class="btn btn-sm btn-warning float-right">Add to Cart</button>
<input type="hidden" name="quantity" value="1" min="1" max="20">
{% set colors = product.colors.split(',') %}
<select name="colors" id="colors" style="visibility: hidden;">
{% for color in colors %}
{% set col = color.split(':') %}
<option value="{{col[0]}}">{{col[0] | capitalize }}</option>
{% endfor %}
</select>
</form>
</div>
</div>
```



```
</form>
</div>
</div>
</div>
{% endfor %}
</div>

<div class="row mt-4">
<div class="col text-center">
{% if products.has_prev %}
<a href="{{url_for('home', page=products.prev_num)}}"class="btn btn-sm btn-outline-info">previus</a>
{% endif %}
{% if products.total> 8 %}
{% for page_num in products.iter_pages(left_edge=1, right_edge=2, left_current=1,right_current=2) %}
{% if page_num %}
{% if products.page == page_num %}
<a href="{{url_for('home', page=page_num)}}" class="btn btn-sm btn-info">{{page_num}}</a>
{% else %}
<a href="{{url_for('home', page=page_num)}}" class="btn btn-sm btn-outline-info">{{page_num}}</a>
{% endif %}
{% else %}
...
{% endif %}
{% endfor %}
{% endif %}
{% if products.has_next %}
<a href="{{url_for('home', page=products.next_num)}}"class="btn btn-sm btn-outline-info">next</a>
{% endif %}
</div>
</div>

{% endif %}
</div>
</div>
{% endblock content %}
```

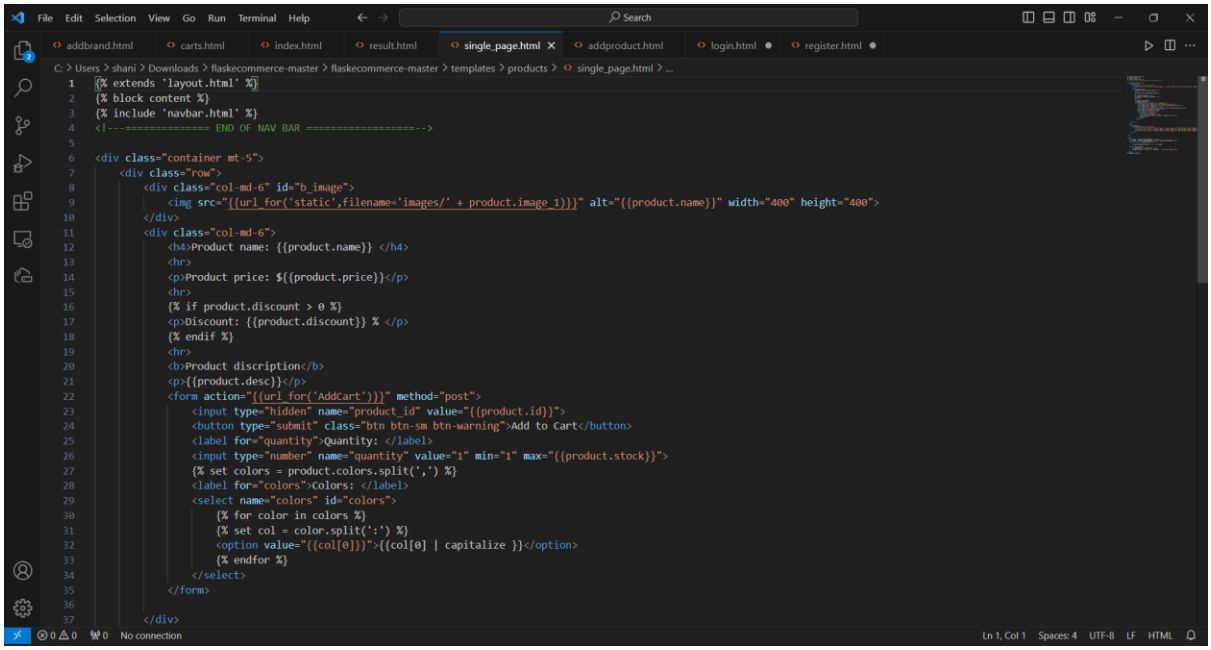
This is the coding for all products page. The customer can view all the available products in this page. If the customer clicks any product, then the user will be redirected to the individual products page.

Result.html

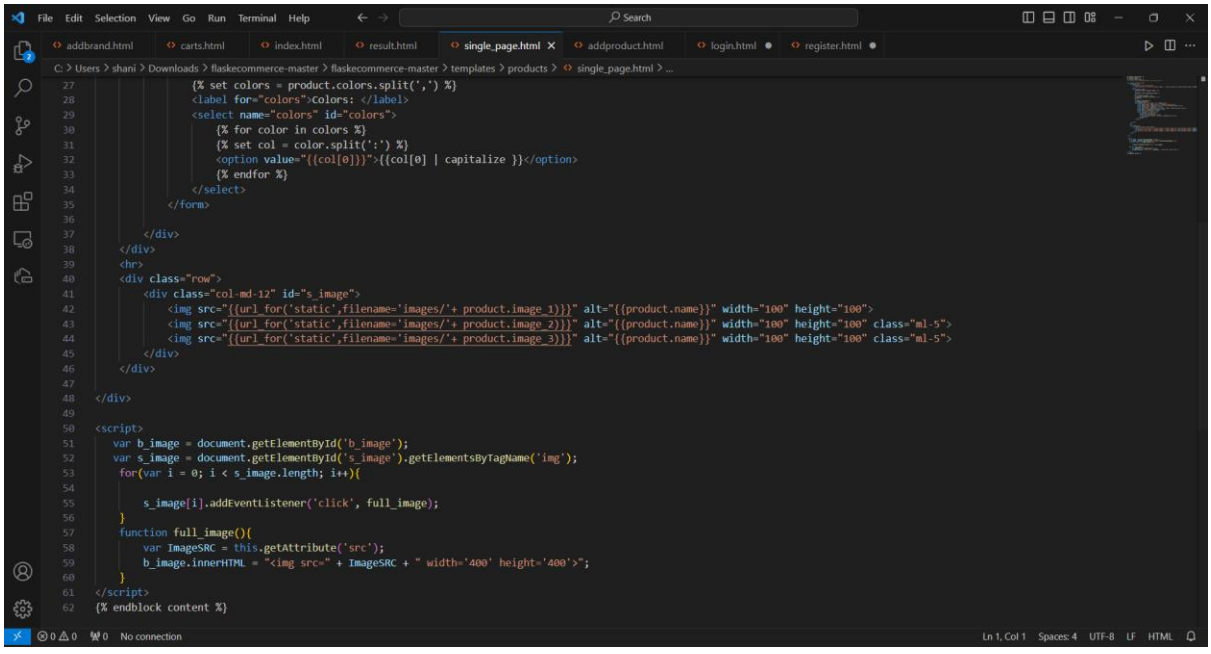


This is the coding for individual product page. This has the details about a particular product the user to add the product to the cart by clicking add to cart button provided there.

Single result.html



```
{% extends 'layout.html' %}
{% block content %}
{% include 'navbar.html' %}
<!---------------- END OF NAV BAR ------------------->

<div class="container mt-5">
    <div class="row">
        <div class="col-md-6" id="b_image">
            <img src="{{url_for('static',filename='images/' + product.image_1)}}" alt="{{product.name}}" width="400" height="400">
        </div>
        <div class="col-md-6">
            <h4>Product name: {{product.name}} </h4>
            <hr>
            <p>Product price: ${{product.price}}</p>
            <hr>
            {% if product.discount > 0 %}
            <p>Discount: {{product.discount}} % </p>
            {% endif %}
            <hr>
            <b>Product discription</b>
            <p>{{product.desc}}</p>
            <form action="{{url_for('AddCart')}}" method="post">
                <input type="hidden" name="product_id" value="{{product.id}}">
                <button type="submit" class="btn btn-sm btn-warning">Add to Cart</button>
                <label for="quantity">Quantity: </label>
                <input type="number" name="quantity" value="1" min="1" max="{{product.stock}}">
                {% set colors = product.colors.split(',') %}
                <label for="colors">Colors: </label>
                <select name="colors" id="colors">
                    {% for color in colors %}
                    {% set col = color.split(':') %}
                    <option value="{{col[0]}}">{{col[0] | capitalize }}</option>
                    {% endfor %}
                </select>
            </form>

        </div>
```

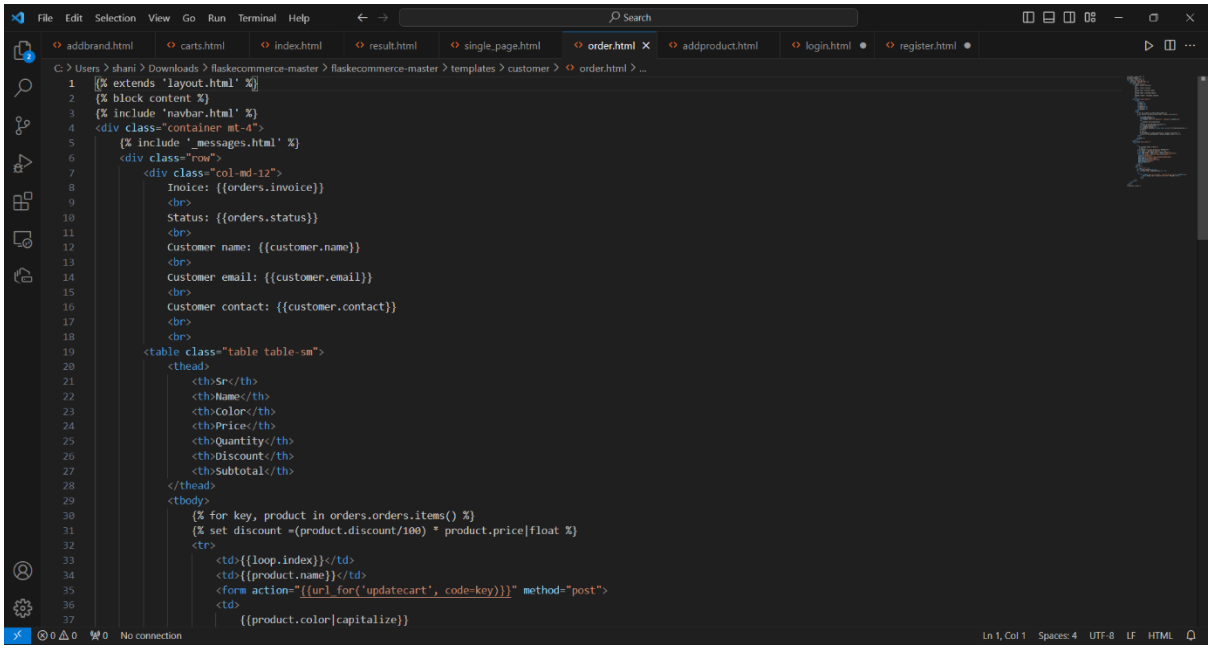


```
                {% set colors = product.colors.split(',') %}
                <label for="colors">Colors: </label>
                <select name="colors" id="colors">
                    {% for color in colors %}
                    {% set col = color.split(':') %}
                    <option value="{{col[0]}}">{{col[0] | capitalize }}</option>
                    {% endfor %}
                </select>
            </form>

        </div>
    </div>
    <hr>
    <div class="row">
        <div class="col-md-12" id="s_image">
            <img src="{{url_for('static',filename='images/'+ product.image_1)}}" alt="{{product.name}}" width="100" height="100">
            <img src="{{url_for('static',filename='images/'+ product.image_2)}}" alt="{{product.name}}" width="100" height="100" class="ml-5">
            <img src="{{url_for('static',filename='images/'+ product.image_3)}}" alt="{{product.name}}" width="100" height="100" class="ml-5">
        </div>
    </div>

</div>

<script>
    var b_image = document.getElementById('b_image');
    var s_image = document.getElementById('s_image').getElementsByTagName('img');
    for(var i = 0; i < s_image.length; i++){

        s_image[i].addEventListener('click', full_image);
    }
    function full_image(){
        var ImageSRC = this.getAttribute('src');
        b_image.innerHTML = "<img src=" + ImageSRC + " width='400' height='400'>";
    }
</script>
{% endblock content %}
```

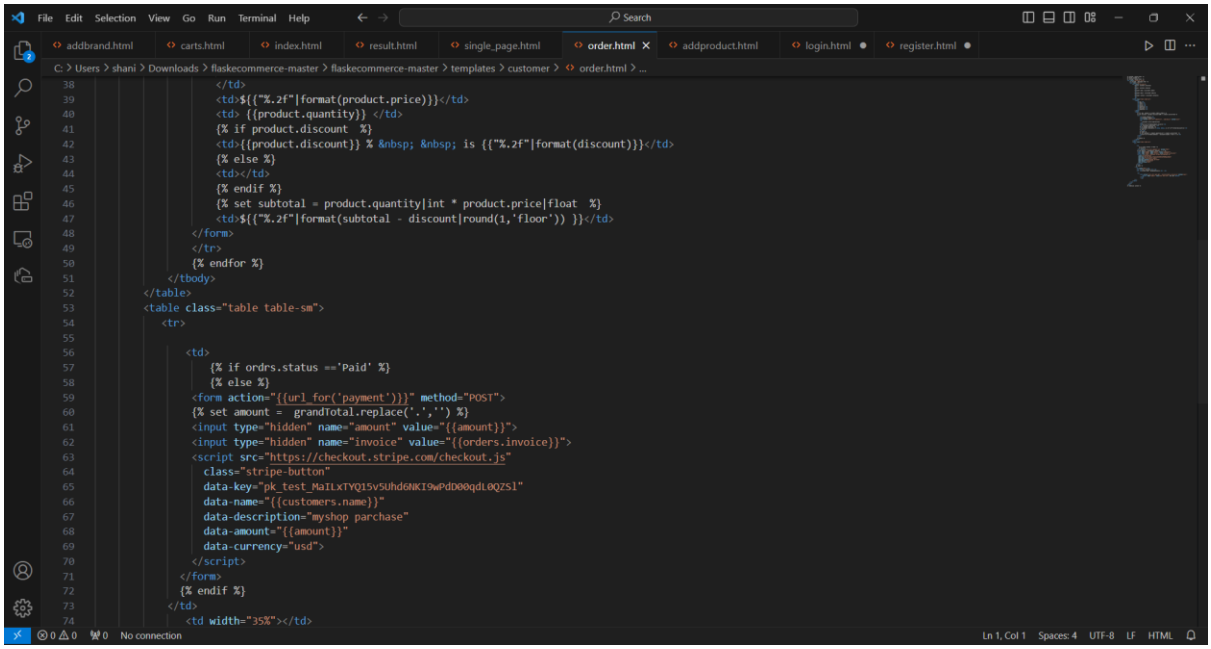
This is the individual page with description and other necessary details. Once the user clicks a particular

product he will navigate to the individual products page. The page has a summarized detail about the product. To view a detailed description of the product this page has been designed.

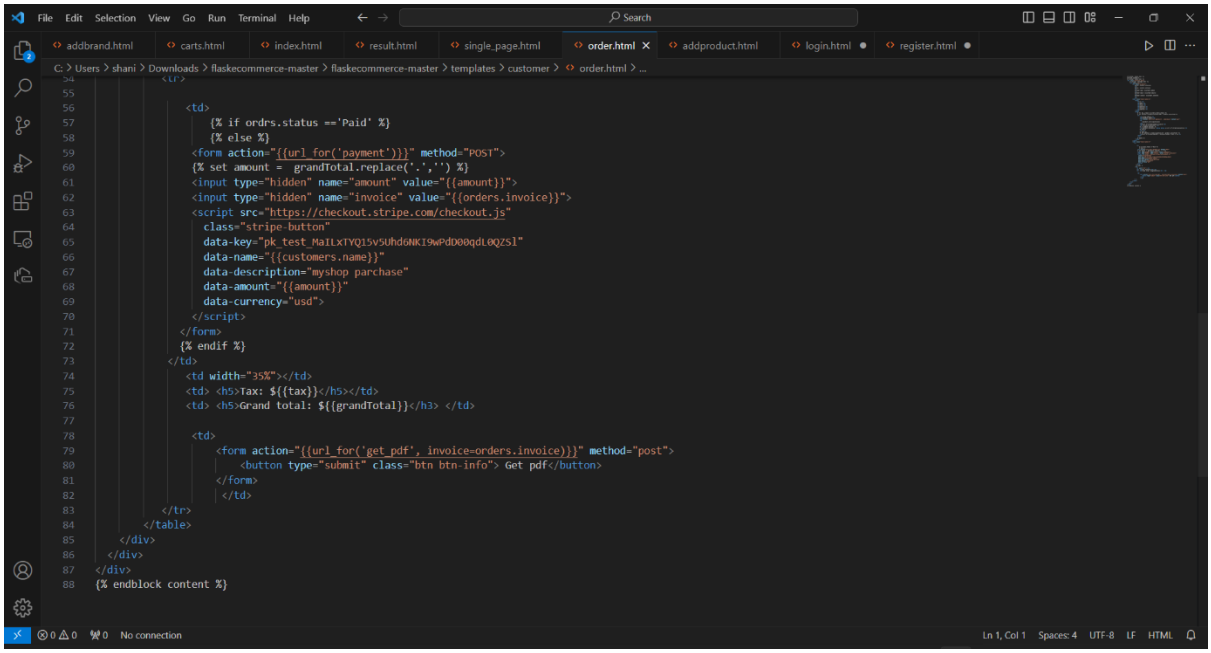
Order.html



```
{% extends 'layout.html' %}
{% block content %}
{% include 'navbar.html' %}
<div class="container mt-4">
    {% include '_messages.html' %}
    <div class="row">
        <div class="col-md-12">
            Inoice: {{orders.invoice}}
            <br>
            Status: {{orders.status}}
            <br>
            Customer name: {{customer.name}}
            <br>
            Customer email: {{customer.email}}
            <br>
            Customer contact: {{customer.contact}}
            <br>
            <br>
        <table class="table table-sm">
            <thead>
                <th>Sr</th>
                <th>Name</th>
                <th>Color</th>
                <th>Price</th>
                <th>Quantity</th>
                <th>Discount</th>
                <th>Subtotal</th>
            </thead>
            <tbody>
                {% for key, product in orders.orders.items() %}
                {% set discount =(product.discount/100) * product.price|float %}
                <tr>
                    <td>{{loop.index}}</td>
                    <td>{{product.name}}</td>
                    <form action="{{url_for('updatecart', code=key)}}" method="post">
                    <td>
                        {{product.color|capitalize}}
```



```
                    </td>
                    <td>${{"%.2f"|format(product.price)}}</td>
                    <td> {{product.quantity}} </td>
                    {% if product.discount  %}
                    <td>{{product.discount}} %     is {{"%.2f"|format(discount)}}</td>
                    {% else %}
                    <td></td>
                    {% endif %}
                    {% set subtotal = product.quantity|int * product.price|float  %}
                    <td>${{"%.2f"|format(subtotal - discount|round(1,'floor')) }}</td>
                </form>
                </tr>
                {% endfor %}
            </tbody>
        </table>
        <table class="table table-sm">
          <tr>

            <td>
                {% if ordrs.status =='Paid' %}
                {% else %}
              <form action="{{url_for('payment')}}" method="POST">
              {% set amount =  grandTotal.replace('.','') %}
              <input type="hidden" name="amount" value="{{amount}}">
              <input type="hidden" name="invoice" value="{{orders.invoice}}">
              <script src="https://checkout.stripe.com/checkout.js"
                class="stripe-button"
                data-key="pk_test_MaILxTYQ15v5Uhd6NKI9wPdD00qdL0QZS1"
                data-name="{{customers.name}}"
                data-description="myshop parchase"
                data-amount="{{amount}}"
                data-currency="usd">
              </script>
            </form>
            {% endif %}
          </td>
            <td width="35%"></td>
```

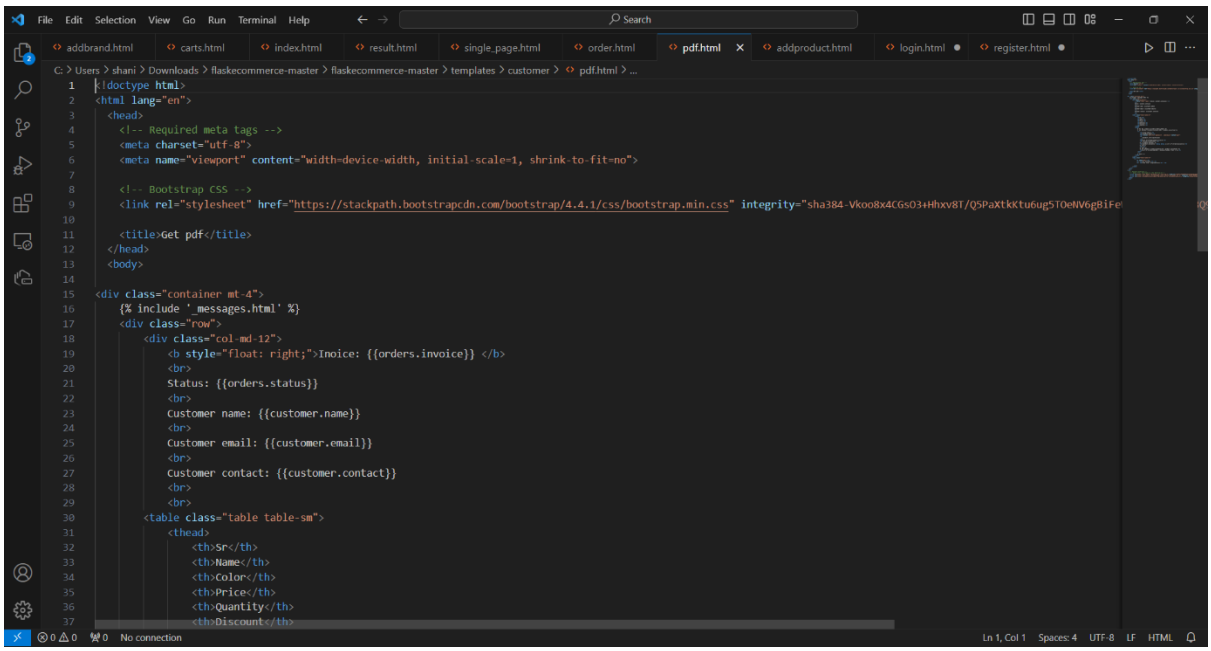


```
<tr>
    <td>
        {% if ordrs.status =='Paid' %}
        {% else %}
      <form action="{{url_for('payment')}}" method="POST">
      {% set amount =  grandTotal.replace('.','') %}
      <input type="hidden" name="amount" value="{{amount}}">
      <input type="hidden" name="invoice" value="{{orders.invoice}}">
      <script src="https://checkout.stripe.com/checkout.js"
        class="stripe-button"
        data-key="pk_test_MaILxTYQ15v5Uhd6NKI9wPdD00qdL0QZ5l"
        data-name="{{customers.name}}"
        data-description="myshop parchase"
        data-amount="{{amount}}"
        data-currency="usd">
      </script>
    </form>
    {% endif %}
  </td>
    <td width="35%"></td>
    <td> <h5>Tax: ${{tax}}</h5></td>
    <td> <h5>Grand total: ${{grandTotal}}</h3> </td>

    <td>
        <form action="{{url_for('get_pdf', invoice=orders.invoice)}}" method="post">
            <button type="submit" class="btn btn-info"> Get pdf</button>
        </form>
        </td>
</tr>
</table>
</div>
</div>
</div>
{% endblock content %}
```

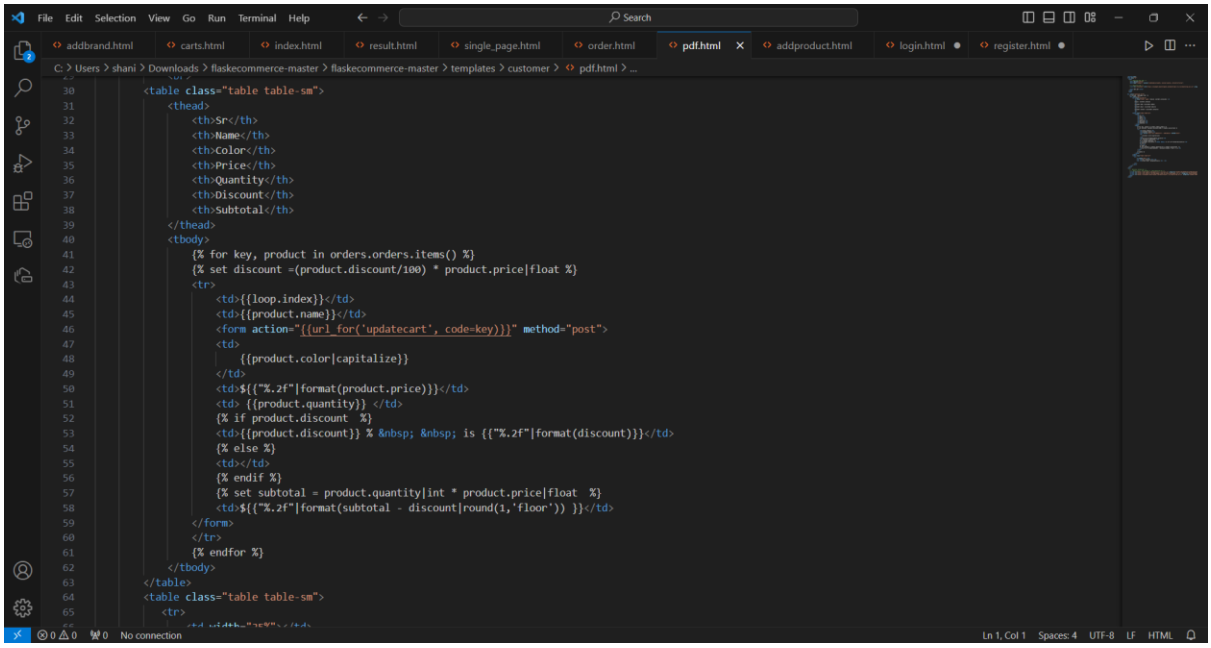
This is the orders page coding. Once the customer has ordered the items in the cart he will proceed to the payment and checkout process.

Pdf.html

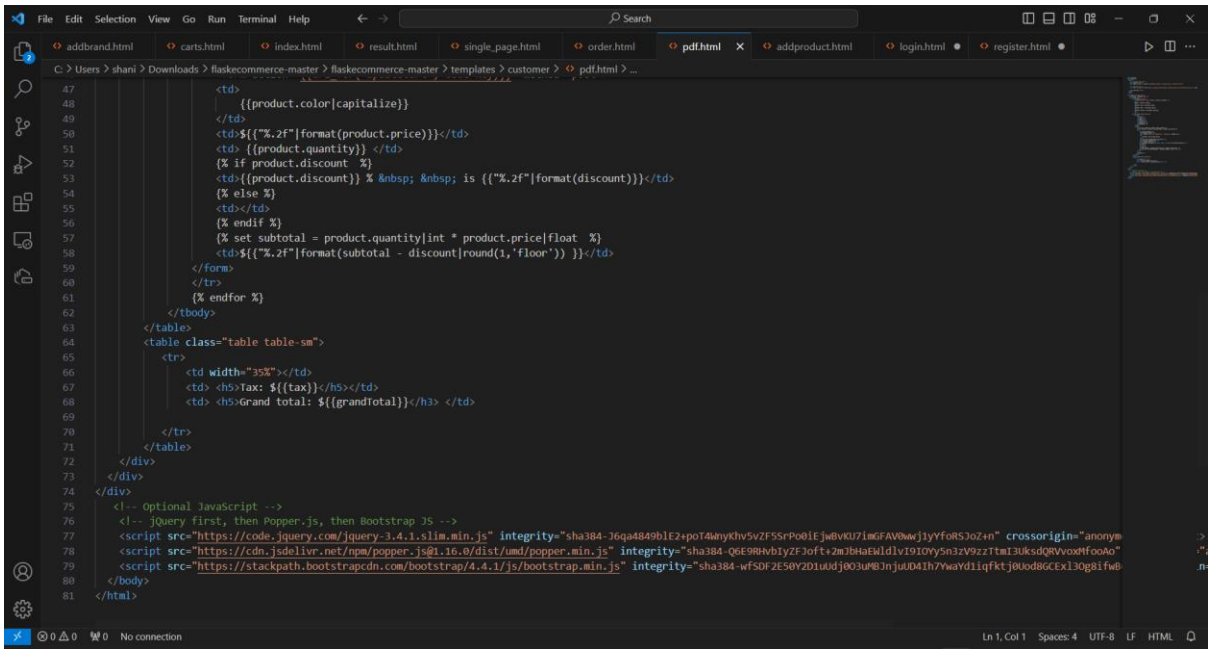
To receive the invoice that is the bill of the items we have the pdf facility.



```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFe

    <title>Get pdf</title>
  </head>
  <body>

<div class="container mt-4">
    {% include '_messages.html' %}
    <div class="row">
        <div class="col-md-12">
            <b style="float: right;">Inoice: {{orders.invoice}} </b>
            <br>
            Status: {{orders.status}}
            <br>
            Customer name: {{customer.name}}
            <br>
            Customer email: {{customer.email}}
            <br>
            Customer contact: {{customer.contact}}
            <br>
            <br>
        <table class="table table-sm">
            <thead>
                <th>Sr</th>
                <th>Name</th>
                <th>Color</th>
                <th>Price</th>
                <th>Quantity</th>
                <th>Discount</th>
```



```
<table class="table table-sm">
    <thead>
        <th>Sr</th>
        <th>Name</th>
        <th>Color</th>
        <th>Price</th>
        <th>Quantity</th>
        <th>Discount</th>
        <th>Subtotal</th>
    </thead>
    <tbody>
        {% for key, product in orders.orders.items() %}
        {% set discount =(product.discount/100) * product.price|float %}
        <tr>
            <td>{{loop.index}}</td>
            <td>{{product.name}}</td>
            <form action="{{url_for('updatecart', code=key)}}" method="post">
            <td>
                {{product.color|capitalize}}
            </td>
            <td>${{"%.2f"|format(product.price)}}</td>
            <td> {{product.quantity}} </td>
            {% if product.discount  %}
            <td>{{product.discount}} %     is {{"%.2f"|format(discount)}}</td>
            {% else %}
            <td></td>
            {% endif %}
            {% set subtotal = product.quantity|int * product.price|float  %}
            <td>${{"%.2f"|format(subtotal - discount|round(1,'floor')) }}</td>
        </form>
        </tr>
        {% endfor %}
    </tbody>
</table>
<table class="table table-sm">
  <tr>
```



```
                    <td>
                        {{product.color|capitalize}}
                    </td>
                    <td>${{"%.2f"|format(product.price)}}</td>
                    <td> {{product.quantity}} </td>
                    {% if product.discount  %}
                    <td>{{product.discount}} %     is {{"%.2f"|format(discount)}}</td>
                    {% else %}
                    <td></td>
                    {% endif %}
                    {% set subtotal = product.quantity|int * product.price|float  %}
                    <td>${{"%.2f"|format(subtotal - discount|round(1,'floor')) }}</td>
                </form>
                </tr>
                {% endfor %}
            </tbody>
        </table>
        <table class="table table-sm">
            <tr>
                <td width="35%"></td>
                <td> <h5>Tax: ${{tax}}</h5></td>
                <td> <h5>Grand total: ${{grandTotal}}</h3> </td>

            </tr>
        </table>
    </div>
  </div>
</div>
  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonym
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB
</body>
</html>
```

This is the coding to format and style the pdf generated.

Thankyou.html

<h1>Thank you shopping with us! </h1>

This is the coding to generate a thank you message. The output is this.

Thank you shopping with us!

Backend:

Flask program:

Init.py

*__init__.py - Notepad

File Edit View

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_bcrypt import Bcrypt
from flask_uploads import UploadSet, configure_uploads, IMAGES, patch_request_class
import os
from flask_msearch import Search
from flask_login import LoginManager
from flask_migrate import Migrate
basedir = os.path.abspath(os.path.dirname(__file__))
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///test.db'
app.config['SECRET_KEY']='hfouewhfoiwefoquw'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
app.config['UPLOADED_PHOTOS_DEST'] = os.path.join(basedir, 'static/images')
photos = UploadSet('photos', IMAGES)
configure_uploads(app, photos)
patch_request_class(app)
db = SQLAlchemy(app)
bcrypt = Bcrypt(app)
search = Search()
search.init_app(app)
migrate = Migrate(app, db)
with app.app_context():
    if db.engine.url.drivername == "sqlite":
        migrate.init_app(app, db, render_as_batch=True)
    else:
        migrate.init_app(app, db)
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view='customerLogin'
login_manager.needs_refresh_message_category='danger'
login_manager.login_message = u"Please login first"
from shop.products import routes
from shop.admin import routes
from shop.carts import carts
from shop.customers import routes
```

Ln 33, Col 1 100% Unix (LF) UTF-8

Run.py

```
from shop import app
if __name__ =="__main__":
    app.run(debug=True)
```

We have created a folder named shop which has all the coding related to the flask program so that the flask coding can be made smaller.

Shop:

Admin:

Forms.py:

forms.py - Notepad

File Edit View

```
from wtforms import BooleanField, StringField, PasswordField, validators , ValidationError
from flask_wtf import FlaskForm, Form
from .models import User


class RegistrationForm(FlaskForm):
    name = StringField('Name', [validators.Length(min=4, max=25)])
    username = StringField('Username', [validators.Length(min=4, max=25)])
    email = StringField('Email Address', [validators.Length(min=6, max=35)])
    password = PasswordField('Password', [
        validators.DataRequired(),
        validators.EqualTo('confirm', message='Passwords must match')
    ])
    confirm = PasswordField('Repeat Password')

    def validate_username(self, field):
        if User.query.filter_by(username=field.data).first():
            raise ValidationError('Username already in use.')


    def validate_email(self, field):
        if User.query.filter_by(email=field.data).first():
            raise ValidationError('Email already registered.')


class LoginForm(FlaskForm):
    email = StringField('Email Address', [validators.Length(min=6, max=35)])
    password = PasswordField('New Password', [validators.DataRequired()])
```

Models.py

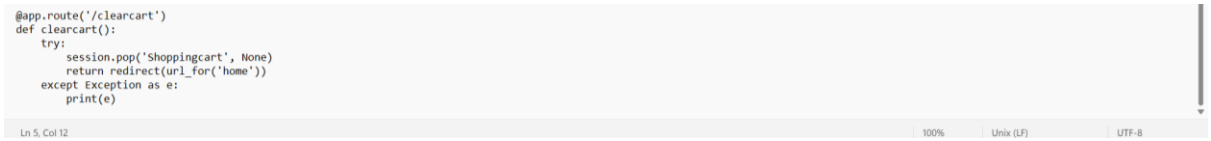

models.py - Notepad
File Edit View
from shop import db
from datetime import datetime
class User(db.Model):
id = db.Column(db.Integer, primary_key=True)
name = db.Column(db.String(50),unique=False, nullable=False)
username = db.Column(db.String(80), unique=True, nullable=False)
email = db.Column(db.String(120), unique=True, nullable=False)
password = db.Column(db.String(180),unique=False, nullable=False)
profile = db.Column(db.String(180), unique=False, nullable=False,default='profile.jpg')
def __repr__(self):
return '<User %r>' % self.username
db.create_all()
Ln 1, Col 1
100%
Unix (LF)
UTF-8

Routes.py


*routes.py - Notepad
File Edit View
from flask import render_template,session, request,redirect,url_for,flash
from shop import app,db,bcrypt
from .forms import RegistrationForm,LoginForm
from .models import User
from shop.products.models import Addproduct,Category,Brand
@app.route('/admin')
def admin():
products = Addproduct.query.all()
return render_template('admin/index.html', title='Admin page',products=products)
@app.route('/brands')
def brands():
brands = Brand.query.order_by(Brand.id.desc()).all()
return render_template('admin/brand.html', title='brands',brands=brands)
@app.route('/categories')
def categories():
categories = Category.query.order_by(Category.id.desc()).all()
return render_template('admin/brand.html', title='categories',categories=categories)
@app.route('/register', methods=['GET', 'POST'])
def register():
form = RegistrationForm()
if form.validate_on_submit():
hash_password = bcrypt.generate_password_hash(form.password.data)
user = User(name=form.name.data,username=form.username.data, email=form.email.data,
password=hash_password)
db.session.add(user)
flash(f'welcome {form.name.data} Thanks for registering','success')
db.session.commit()
return redirect(url_for('login'))
return render_template('admin/register.html',title='Register user', form=form)
@app.route('/login', methods=['GET','POST'])
def login():
form = LoginForm()
if form.validate_on_submit():
user = User.query.filter_by(email=form.email.data).first()
Ln 35, Col 1
100%
Unix (LF)
UTF-8

Carts:

Carts.py:



Customers:

Forms.py

Models.py

*model.py - Notepad

File Edit View

```
from shop import db, login_manager
from datetime import datetime
from flask_login import UserMixin
import json

@login_manager.user_loader
def user_loader(user_id):
    return Register.query.get(user_id)

class Register(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key= True)
    name = db.Column(db.String(50), unique= False)
    username = db.Column(db.String(50), unique= True)
    email = db.Column(db.String(50), unique= True)
    password = db.Column(db.String(200), unique= False)
    country = db.Column(db.String(50), unique= False)
    # state = db.Column(db.String(50), unique= False)
    city = db.Column(db.String(50), unique= False)
    contact = db.Column(db.String(50), unique= False)
    address = db.Column(db.String(50), unique= False)
    zipcode = db.Column(db.String(50), unique= False)
    profile = db.Column(db.String(200), unique= False , default='profile.jpg')
    date_created = db.Column(db.DateTime, nullable=False, default=datetime.utcnow)

    def __repr__(self):
        return '<Register %r>' % self.name

class JsonEcodedDict(db.TypeDecorator):
    impl = db.Text
    def process_bind_param(self, value, dialect):
        if value is None:
            return '{}'
        else:
            return json.dumps(value)
    def process_result_value(self, value, dialect):
        if value is None:
            return {}
        else:
            return json.loads(value)
```

Ln 51, Col 1 100% Unix (LF) UTF-8

```
            return json.dumps(value)
    def process_result_value(self, value, dialect):
        if value is None:
            return {}
        else:
            return json.loads(value)

class CustomerOrder(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    invoice = db.Column(db.String(20), unique=True, nullable=False)
    status = db.Column(db.String(20), default='Pending', nullable=False)
    customer_id = db.Column(db.Integer, unique=False, nullable=False)
    date_created = db.Column(db.DateTime, default=datetime.utcnow, nullable=False)
    orders = db.Column(JsonEcodedDict)

    def __repr__(self):
        return'<CustomerOrder %r>' % self.invoice
db.create_all()
```

Routes.py

routes.py - Notepad

File Edit View

```
from flask import render_template,session, request,redirect,url_for,flash,current_app,make_response
from flask_login import login_required, current_user, logout_user, login_user
from shop import app,db,photos, search,bcrypt,login_manager
from .forms import CustomerRegisterForm, CustomerLoginFrom
from .model import Register,CustomerOrder
import secrets
import os
import json
import pdfkit
import stripe

buplishable_key ='pk_test_MaILxTYQ15v5Uhd6NKI9wPdD00qdL0QZ5l'
stripe.api_key ='sk_test_9JlhVB6qwjcRdYzizjdwgIo0Dt00N55uxbWy'

@app.route('/payment',methods=['POST'])
def payment():
    invoice = request.get('invoice')
    amount = request.form.get('amount')
    customer = stripe.Customer.create(
      email=request.form['stripeEmail'],
      source=request.form['stripeToken'],
    )
    charge = stripe.Charge.create(
      customer=customer.id,
      description='Myshop',
      amount=amount,
      currency='usd',
    )
    orders =  CustomerOrder.query.filter_by(customer_id = current_user.id,invoice=invoice).order_by(CustomerOrder.id.desc()).first()
    orders.status = 'Paid'
    db.session.commit()
    return redirect(url_for('thanks'))

@app.route('/thanks')
def thanks():
    return render_template('customer/thank.html')


@app.route('/customer/register', methods=['GET','POST'])
def customer_register():
```

Ln 1, Col 1 100% Unix (LF) UTF-8

routes.py - Notepad

File Edit View

```
@app.route('/customer/register', methods=['GET','POST'])
def customer_register():
    form = CustomerRegisterForm()
    if form.validate_on_submit():
        hash_password = bcrypt.generate_password_hash(form.password.data)
        register = Register(name=form.name.data, username=form.username.data, email=form.email.data,password=hash_password,country=form.country.data,
city=form.city.data,contact=form.contact.data, address=form.address.data, zipcode=form.zipcode.data)
        db.session.add(register)
        flash(f'Welcome {form.name.data} Thank you for registering', 'success')
        db.session.commit()
        return redirect(url_for('login'))
    return render_template('customer/register.html', form=form)


@app.route('/customer/login', methods=['GET','POST'])
def customerLogin():
    form = CustomerLoginFrom()
    if form.validate_on_submit():
        user = Register.query.filter_by(email=form.email.data).first()
        if user and bcrypt.check_password_hash(user.password, form.password.data):
            login_user(user)
            flash('You are login now!', 'success')
            next = request.args.get('next')
            return redirect(next or url_for('home'))
        flash('Incorrect email and password','danger')
        return redirect(url_for('customerLogin'))

    return render_template('customer/login.html', form=form)


@app.route('/customer/logout')
def customer_logout():
    logout_user()
    return redirect(url_for('home'))

def updateshoppingcart():
    for key, shopping in session['Shoppingcart'].items():
        session.modified = True
        del shopping['image']
        del shopping['colors']
```

Ln 1, Col 1 100% Unix (LF) UTF-8

routes.py - Notepad

File Edit View

```
        del shopping['image']
        del shopping['colors']
    return updateshoppingcart

@app.route('/getorder')
@login_required
def get_order():
    if current_user.is_authenticated:
        customer_id = current_user.id
        invoice = secrets.token_hex(5)
        updateshoppingcart
        try:
            order = CustomerOrder(invoice=invoice,customer_id=customer_id,orders=session['Shoppingcart'])
            db.session.add(order)
            db.session.commit()
            session.pop('Shoppingcart')
            flash('Your order has been sent successfully','success')
            return redirect(url_for('orders',invoice=invoice))
        except Exception as e:
            print(e)
            flash('Some thing went wrong while get order', 'danger')
            return redirect(url_for('getCart'))



@app.route('/orders/<invoice>')
@login_required
def orders(invoice):
    if current_user.is_authenticated:
        grandTotal = 0
        subTotal = 0
        customer_id = current_user.id
        customer = Register.query.filter_by(id=customer_id).first()
        orders = CustomerOrder.query.filter_by(customer_id=customer_id, invoice=invoice).order_by(CustomerOrder.id.desc()).first()
        for _key, product in orders.orders.items():
            discount = (product['discount']/100) * float(product['price'])
            subTotal += float(product['price']) * int(product['quantity'])
            subTotal -= discount
            tax = ("%.2f" % (.06 * float(subTotal)))
            grandTotal = ("%.2f" % (1.06 * float(subTotal)))
```

Ln 1, Col 1 100% Unix (LF) UTF-8

routes.py - Notepad

File Edit View

```
            subTotal += float(product['price']) * int(product['quantity'])
            subTotal -= discount
            tax = ("%.2f" % (.06 * float(subTotal)))
            grandTotal = ("%.2f" % (1.06 * float(subTotal)))

    else:
        return redirect(url_for('customerLogin'))
    return render_template('customer/order.html', invoice=invoice, tax=tax,subTotal=subTotal,grandTotal=grandTotal,customer=customer,orders=orders)



@app.route('/get_pdf/<invoice>', methods=['POST'])
@login_required
def get_pdf(invoice):
    if current_user.is_authenticated:
        grandTotal = 0
        subTotal = 0
        customer_id = current_user.id
        if request.method =="POST":
            customer = Register.query.filter_by(id=customer_id).first()
            orders = CustomerOrder.query.filter_by(customer_id=customer_id, invoice=invoice).order_by(CustomerOrder.id.desc()).first()
            for _key, product in orders.orders.items():
                discount = (product['discount']/100) * float(product['price'])
                subTotal += float(product['price']) * int(product['quantity'])
                subTotal -= discount
                tax = ("%.2f" % (.06 * float(subTotal)))
                grandTotal = float("%.2f" % (1.06 * subTotal))

            rendered =  render_template('customer/pdf.html', invoice=invoice, tax=tax,grandTotal=grandTotal,customer=customer,orders=orders)
            pdf = pdfkit.from_string(rendered, False)
            response = make_response(pdf)
            response.headers['content-Type'] ='application/pdf'
            response.headers['content-Disposition'] ='inline; filename='+invoice+'.pdf'
            return response
    return request(url_for('orders'))
```

Products:

Forms.py

forms.py - Notepad

File Edit View

```
from wtforms import Form, SubmitField,IntegerField,FloatField,StringField,TextAreaField,validators
from flask_wtf.file import FileField,FileRequired,FileAllowed

class Addproducts(Form):
    name = StringField('Name', [validators.DataRequired()])
    price = FloatField('Price', [validators.DataRequired()])
    discount = IntegerField('Discount', default=0)
    stock = IntegerField('Stock', [validators.DataRequired()])
    colors = StringField('Colors', [validators.DataRequired()])
    discription = TextAreaField('Discription', [validators.DataRequired()])

    image_1 = FileField('Image 1', validators=[FileRequired(), FileAllowed(['jpg','png','gif','jpeg']), 'Images only please'])
    image_2 = FileField('Image 2', validators=[FileRequired(), FileAllowed(['jpg','png','gif','jpeg']), 'Images only please'])
    image_3 = FileField('Image 3', validators=[FileRequired(), FileAllowed(['jpg','png','gif','jpeg']), 'Images only please'])
```

Models.py

*models.py - Notepad

File Edit View

```
from shop import db
from datetime import datetime
class Addproduct(db.Model):
    __seachbale__ = ['name','desc']
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), nullable=False)
    price = db.Column(db.Numeric(10,2), nullable=False)
    discount = db.Column(db.Integer, default=0)
    stock = db.Column(db.Integer, nullable=False)
    colors = db.Column(db.Text, nullable=False)
    desc = db.Column(db.Text, nullable=False)
    pub_date = db.Column(db.DateTime, nullable=False,default=datetime.utcnow)

    category_id = db.Column(db.Integer, db.ForeignKey('category.id'),nullable=False)
    category = db.relationship('Category',backref=db.backref('categories', lazy=True))

    brand_id = db.Column(db.Integer, db.ForeignKey('brand.id'),nullable=False)
    brand = db.relationship('Brand',backref=db.backref('brands', lazy=True))

    image_1 = db.Column(db.String(150), nullable=False, default='image1.jpg')
    image_2 = db.Column(db.String(150), nullable=False, default='image2.jpg')
    image_3 = db.Column(db.String(150), nullable=False, default='image3.jpg')

    def __repr__(self):
        return '<Post %r>' % self.name
class Brand(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(30), unique=True, nullable=False)

    def __repr__(self):
        return '<Brand %r>' % self.name
class Category(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(30), unique=True, nullable=False)

    def __repr__(self):
        return '<Catgory %r>' % self.name
db.create_all()
```

Routes.py

routes.py - Notepad

File Edit View

```
from flask import render_template,session, request,redirect,url_for,flash,current_app
from shop import app,db,photos, search
from .models import Category,Brand,Addproduct
from .forms import Addproducts
import secrets
import os


def brands():
    brands = Brand.query.join(Addproduct, (Brand.id == Addproduct.brand_id)).all()
    return brands

def categories():
    categories = Category.query.join(Addproduct,(Category.id == Addproduct.category_id)).all()
    return categories


@app.route('/')
def home():
    page = request.args.get('page',1, type=int)
    products = Addproduct.query.filter(Addproduct.stock > 0).order_by(Addproduct.id.desc()).paginate(page=page, per_page=8)
    return render_template('products/index.html', products=products,brands=brands(),categories=categories())

@app.route('/result')
def result():
    searchword = request.args.get('q')
    products = Addproduct.query.msearch(searchword, fields=['name','desc'] , limit=6)
    return render_template('products/result.html',products=products,brands=brands(),categories=categories())

@app.route('/product/<int:id>')
def single_page(id):
    product = Addproduct.query.get_or_404(id)
    return render_template('products/single_page.html',product=product,brands=brands(),categories=categories())



@app.route('/brand/<int:id>')
def get_brand(id):
```

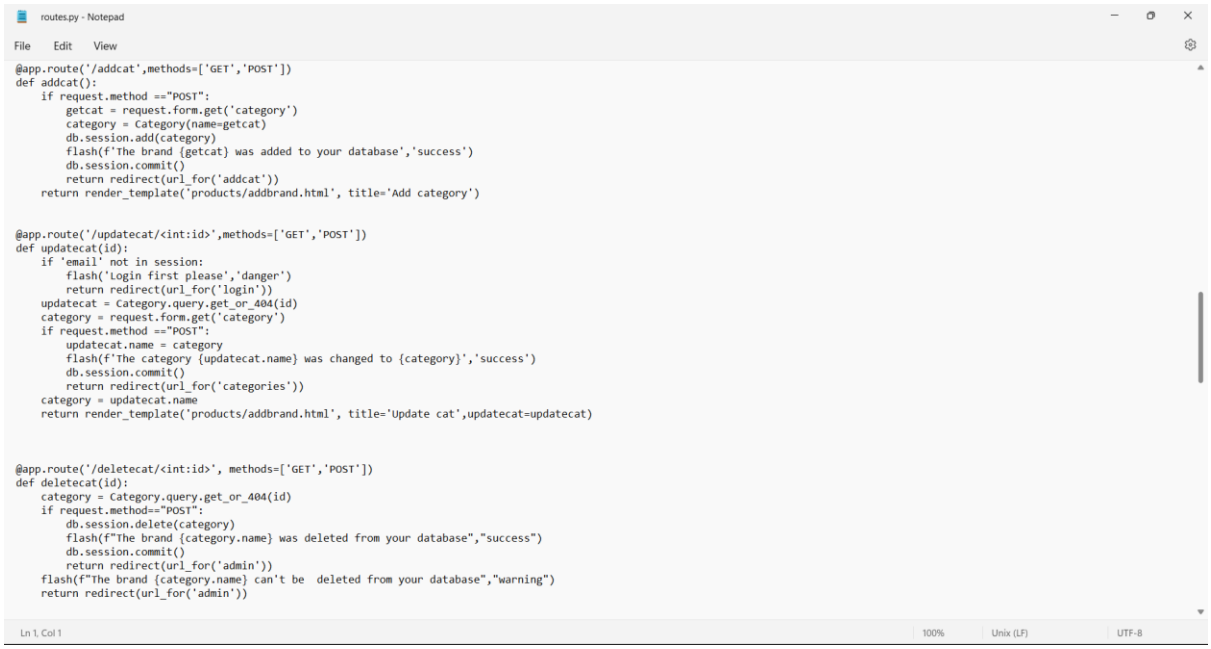
Ln 1, Col 1 100% Unix (LF) UTF-8

routes.py - Notepad

File Edit View

```
@app.route('/brand/<int:id>')
def get_brand(id):
    page = request.args.get('page',1, type=int)
    get_brand = Brand.query.filter_by(id=id).first_or_404()
    brand = Addproduct.query.filter_by(brand=get_brand).paginate(page=page, per_page=8)
    return render_template('products/index.html',brand=brand,brands=brands(),categories=categories(),get_brand=get_brand)


@app.route('/categories/<int:id>')
def get_category(id):
    page = request.args.get('page',1, type=int)
    get_cat = Category.query.filter_by(id=id).first_or_404()
    get_cat_prod = Addproduct.query.filter_by(category=get_cat).paginate(page=page, per_page=8)
    return render_template('products/index.html',get_cat_prod=get_cat_prod,brands=brands(),categories=categories(),get_cat=get_cat)


@app.route('/addbrand',methods=['GET','POST'])
def addbrand():
    if request.method =="POST":
        getbrand = request.form.get('brand')
        brand = Brand(name=getbrand)
        db.session.add(brand)
        flash(f'The brand {getbrand} was added to your database','success')
        db.session.commit()
        return redirect(url_for('addbrand'))
    return render_template('products/addbrand.html', title='Add brand',brands='brands')

@app.route('/updatebrand/<int:id>',methods=['GET','POST'])
def updatebrand(id):
    if 'email' not in session:
        flash('Login first please','danger')
        return redirect(url_for('login'))
    updatebrand = Brand.query.get_or_404(id)
    brand = request.form.get('brand')
    if request.method =="POST":
        updatebrand.name = brand
        flash(f'The brand {updatebrand.name} was changed to {brand}','success')
        db.session.commit()
```

Ln 1, Col 1 100% Unix (LF) UTF-8


routes.py - Notepad

File Edit View

```
@app.route('/addcat',methods=['GET','POST'])
def addcat():
    if request.method =="POST":
        getcat = request.form.get('category')
        category = Category(name=getcat)
        db.session.add(category)
        flash(f'The brand {getcat} was added to your database','success')
        db.session.commit()
        return redirect(url_for('addcat'))
    return render_template('products/addbrand.html', title='Add category')


@app.route('/updatecat/<int:id>',methods=['GET','POST'])
def updatecat(id):
    if 'email' not in session:
        flash('Login first please','danger')
        return redirect(url_for('login'))
    updatecat = Category.query.get_or_404(id)
    category = request.form.get('category')
    if request.method =="POST":
        updatecat.name = category
        flash(f'The category {updatecat.name} was changed to {category}','success')
        db.session.commit()
        return redirect(url_for('categories'))
    category = updatecat.name
    return render_template('products/addbrand.html', title='Update cat',updatecat=updatecat)


@app.route('/deletecat/<int:id>', methods=['GET','POST'])
def deletecat(id):
    category = Category.query.get_or_404(id)
    if request.method=="POST":
        db.session.delete(category)
        flash(f"The brand {category.name} was deleted from your database","success")
        db.session.commit()
        return redirect(url_for('admin'))
    flash(f"The brand {category.name} can't be  deleted from your database","warning")
    return redirect(url_for('admin'))
```

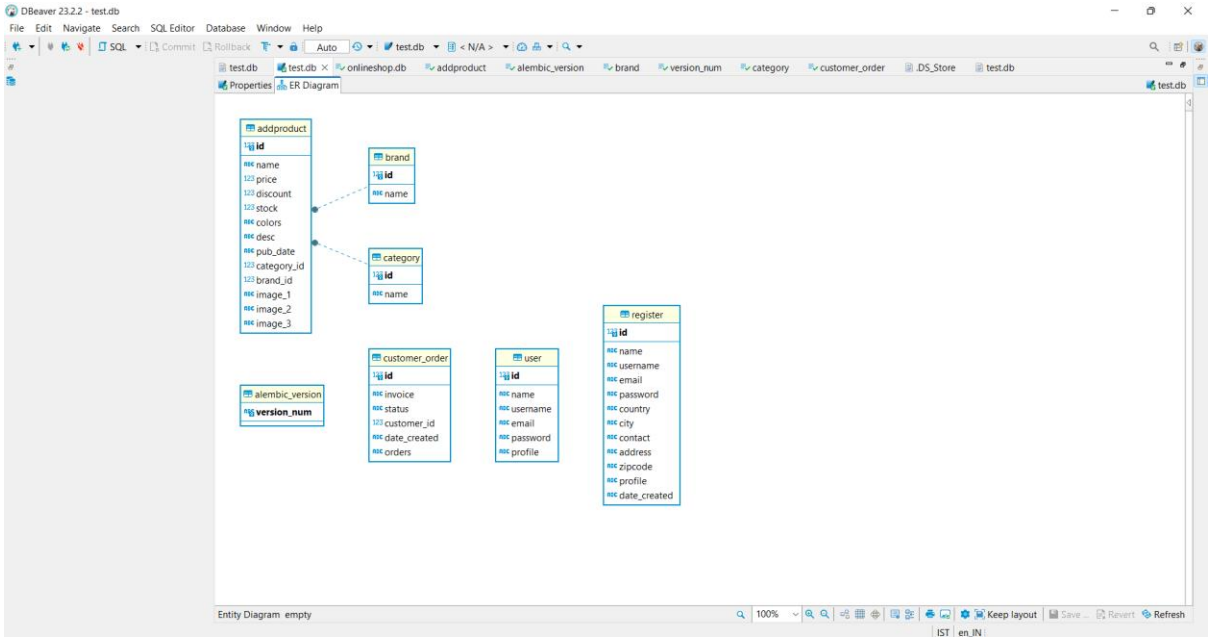
Ln 1, Col 1 100% Unix (LF) UTF-8

These are the coding in the shop folder.

Database:

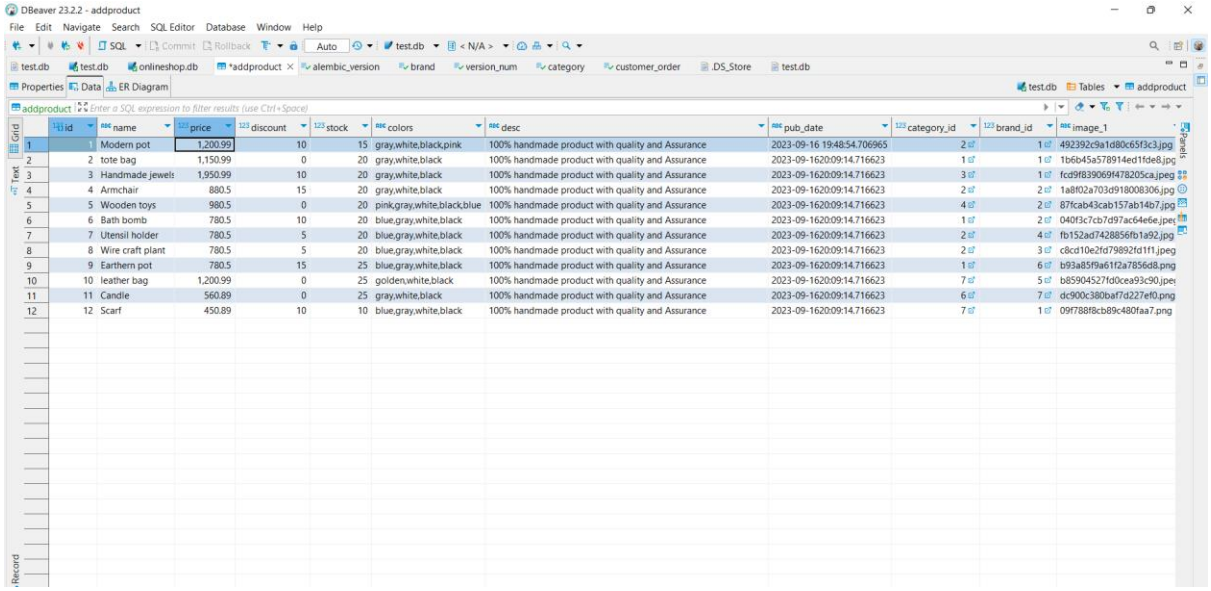
As said, we have used SQLite for database which we have named test.db.

The entity relationship diagram is shown below

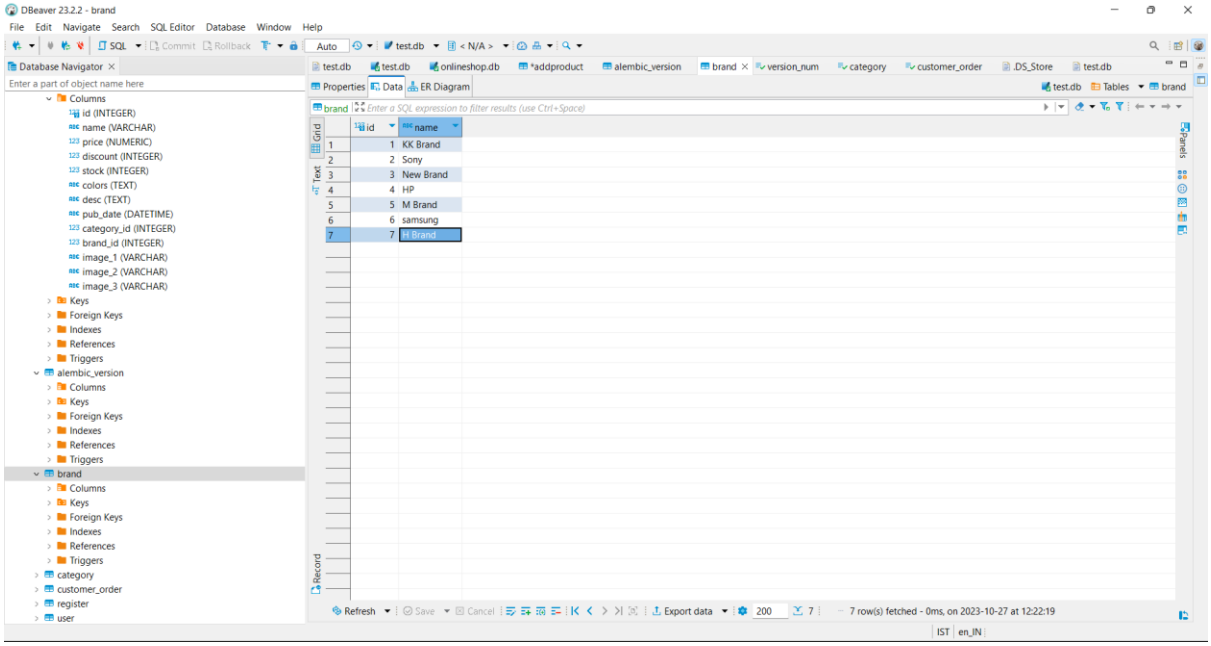


The various databases handled are shown below:

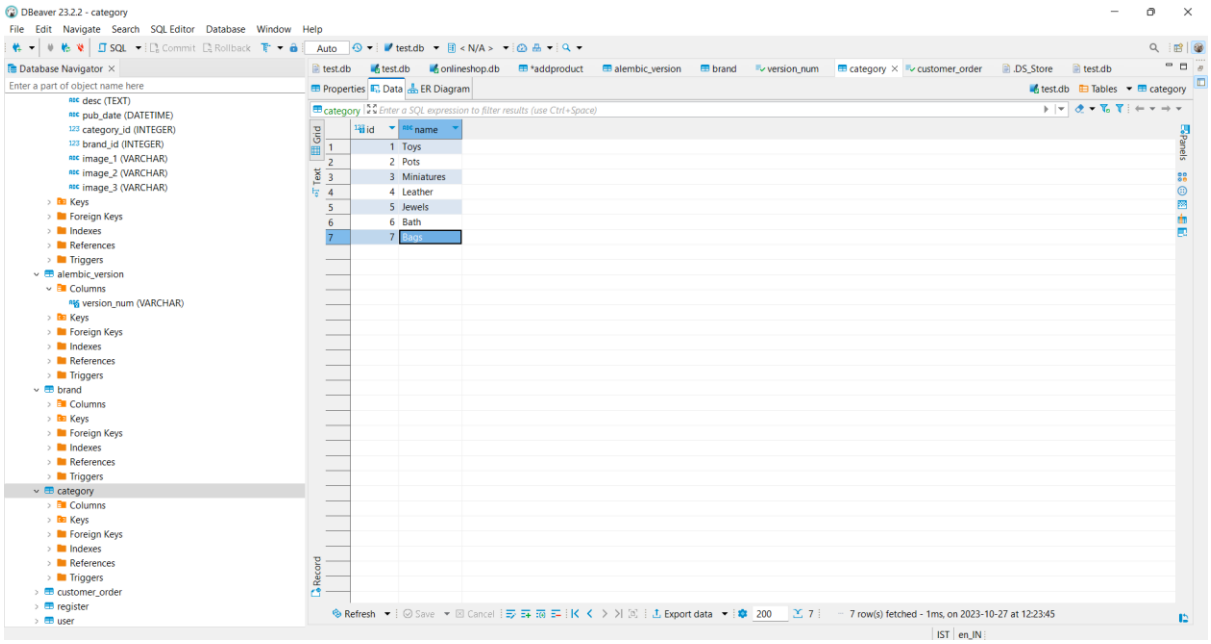
Database for products:



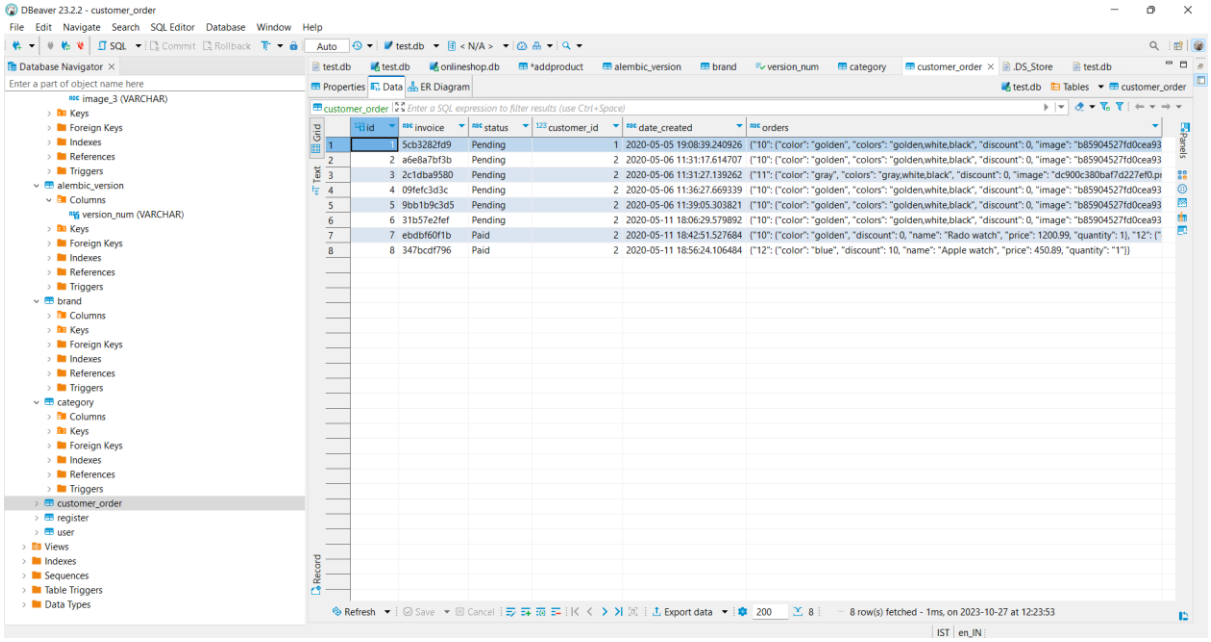
Database for brands:



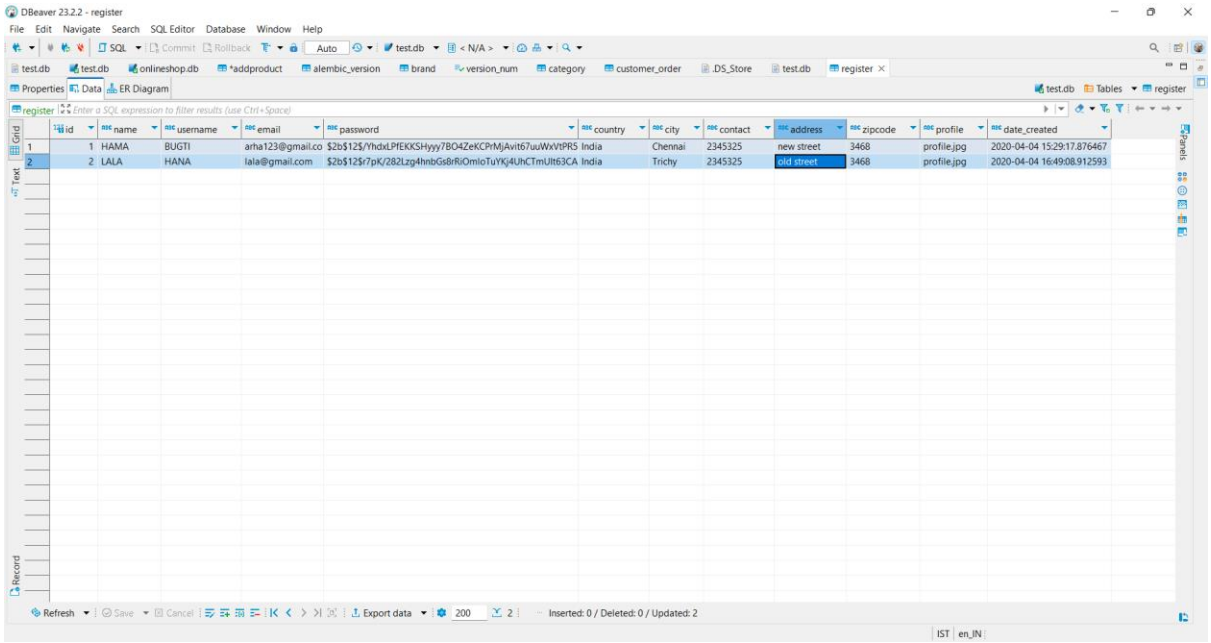
Database for categories:

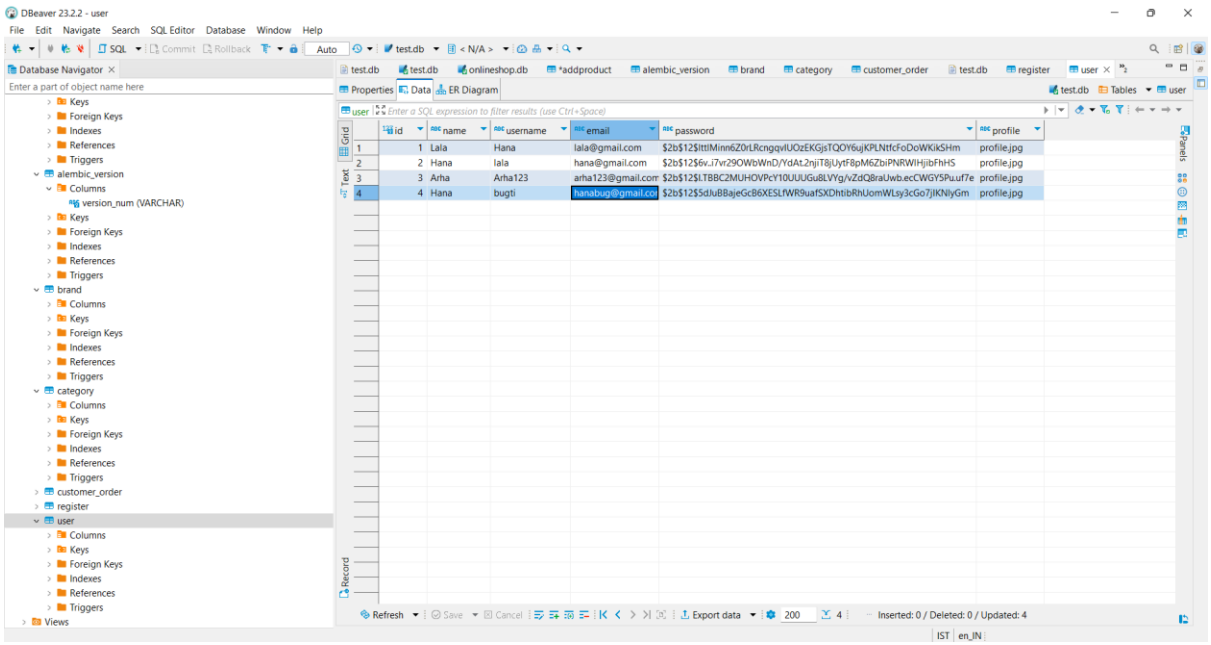


Database for customer orders:



Database for registration and login:





Cloud deployment:

The flask app developed can be deployed in cloud by following these steps:

1. Build a app by using Python Flask.
2. Build and test a Docker container (Postman is an alternative to Docker).
3. Push the container to the IBM Cloud Registry.
4. Deploy and manage the application by using Code Engine.
5. Be enabled for OpenAPI 3.0.

This is the entire final design, development, and deployment of our simple and user friendly E-Commerce app ARHA.