

Project Title: E-Commerce Store

Website Name: E-Comm Store

Checkout our project at <http://wpl-ecomm-store.web.app/>

1. PROJECT DESCRIPTION

Welcome to E-Comm Store! This website has been designed to make purchase of various items belonging to category such as electronics, clothing, books etc. convenient for the customers with a user-friendly user interface.

Although this website has not implemented the digital payment functionalities yet, but it does offer other features which, we hope, the users will surely enjoy. Some of the major functionalities of this website are provided below.

- Users can browse the store, add items to the cart and checkout as Registered Users by creating a user account
- Email and password fields are validated during user sign-in/signup process to ensure criteria are met .
- User passwords are kept secured by storing only the password hashes into the database
- Users can view the product, it's availability, it's price.
- This website supports pagination for products catalog and each page displays up to 100 products at a time
- Users have the option to search and filter products
 - This website supports both partial and full search.
 - Users can filter products by various available categories.
- Users can add/remove items from their cart and can edit purchase quantities
- Users can fill out their shipping details on the checkout page. Users can also maintain multiple shipping details.
- Once an order is placed, the order gets stored in the database and the users can retrieve their order history from the website. Users are also sent an email confirmation for each order.
- Users can view order details and the current status for each order
- Admin users have the additional privileges to add new products to the store by providing details such as product name, price, product category, product weight, unit, rate, quantity, buffer quantity and uploading product images.
- Admin users have the privileges to add a new category.

- Admin users have the privileges to go through the order details and have the option to filter the orders by its status. Admins can also update order status to alert users about the progress on their order.
- Admin users is also provided with Search Functionality to search products and have the option to edit, delete(soft delete) and update products by going to the product management tab and also have the option to view the deleted products.

2. DATABASE DESIGN

In this Project, there are 4 Data Models that we use to store Data in MongoDB.

Those models are:

1. **User Model:** For Storing the User Details (If the User has signed up !).
2. **Product Model:** For storing Product Data with all its attributes.
3. **Order Model:** For storing the Order details including some attributes of User and the Products ordered.
4. **Cart Model:** For storing the Product Data that are added to the cart by user with all its attributes.
5. **Address Model:** For storing the Address Data that is selected while placing the order
6. **Product Category Model :** For Storing the Product categories while creating different categories for products.
7. **Authenticate Model:** For storing the User ID & Password of the users.

Below are the attributes for each of the respective Models mentioned above:

User Model-

```
/ USER MODEL
/ This is User model it contains Firstname, Lastname, Email, Password, Role,
/ If the user is active & verified
/ All attributes are mandatory

{
  firstName: {
    type: String,
    required: true,
    trim: true,
  },
  lastName: {
    type: String,
    required: true,
    trim: true,
  },
  email: {
    type: String,
    unique: true,
    required: true,
    trim: true,
  },
  password: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "Authentication",
  },
  role: {
    type: Number,
    enum: [0, 1, 2],
    default: 0,
  },
  isActive: {
    type: Boolean,
    default: true,
  },
  isVerified: {
    type: Boolean,
    default: false,
  },
},
{
  timestamps: {
    createdAt: "createdAt",
    updatedAt: "updatedAt",
  },
}
```

Product Model-

```
/ PRODUCT MODEL
/ This is Product model it contains Product name, weight, unit, rate, category to which it belongs,
/ imageurl that is it's path, quantity, bufferquantity & whether the product is there or not
```

```
{
  name: {
    type: String,
  },
  weight: {
    type: Number,
  },
  unit: {
    type: String,
  },
  rate: {
    type: Number,
  },
  category: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "ProductCategories",
  },
  imageURL: {
    type: String,
  },
  quantity: {
    type: Number,
  },
  bufferQuantity: {
    type: Number,
  },
  isDeleted: {
    type: Boolean,
    default: false,
  },
},
{
  timestamps: {
    createdAt: "createdAt",
    updatedAt: "updatedAt",
  },
}
```

Order Model-

```

/ ORDER MODEL
/ This is the Order model it contains the ordertotal that contains total number of products ordered,
/ Delivery Address, UserID, Status of Order & Products data.

```

```

{
  orderTotal: {
    type: Number,
    required: true,
  },
  deliveryAddress: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "Addresses",
  },
  user: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "User",
  },
  status: {
    type: String,
    default: "NEW",
    required: true,
    enum: ["CANCELLED", "NEW", "PROCESSING", "DISPATCHED", "COMPLETED"],
  },
  productData: {
    type: [
      {
        product: {
          type: mongoose.Schema.Types.ObjectId,
          ref: "Products",
        },
        quantity: {
          type: Number,
        },
      },
    ],
    default: [],
  },
}
{
  timestamps: {
    createdAt: "createdAt",
    updatedAt: "updatedAt",
  },
}

```

Cart Model-

```
// CART MODEL
// This is Adding to cart model it contains UserID, Data of products that are there in the cart

{
  userId: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "User",
  },
  productData: {
    type: [
      {
        product: {
          type: mongoose.Schema.Types.ObjectId,
          ref: "Products",
        },
        quantity: {
          type: Number,
        },
      },
    ],
    default: [],
  },
},
{
  timestamps: {
    createdAt: "createdAt",
    updatedAt: "updatedAt",
  },
}
```

Address Model-

```

//ADDRESS MODEL
//This is Address model it contains the Name of person, Mobile No,
//Address where the product needs to be delivered.

{
  name: {
    type: String,
    required: true,
  },
  mobile: {
    type: String,
    required: true,
  },
  addressLine1: {
    type: String,
    required: true,
  },
  addressLine2: {
    type: String,
  },
  apartment: {
    type: String,
  },
  pincode: {
    type: String,
    required: true,
  },
  deliveryInstructions: {
    type: String,
  },
  isDeleted: {
    type: Boolean,
    default: false,
  },
  user: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "User",
  },
},
{
  timestamps: {
    createdAt: "createdAt",
    updatedAt: "updatedAt",
  },
}

```

Authenticate Model-

```
//AUTHENTICATE MODEL
// This contains userId & Password
{
  userId: {
    type: String,
    unique: true,
  },
  passwordHash: {
    type: String,
  },
},
{
  timestamps: {
    createdAt: "createdAt",
    updatedAt: "updatedAt",
  },
},
}
```

Encrypted Password in User Collection-

```
> _id: ObjectId("6195b86a732e836a45ec097e")
  firstName: "Aishwarya"
  lastName: "Singh"
  email: "aishwaryasingh1122@gmail.com"
  role: 2
  isActive: true
  isVerified: true
  createdAt: 2021-11-18T02:20:26.211+00:00
  updatedAt: 2021-11-18T02:20:26.296+00:00
  __v: 0
  password: ObjectId("6195b86a732e836a45ec0980")
```

```
_id: ObjectId("61a80cbeadd6eecbda741da")
firstName: "Sakshi"
lastName: "Laddha"
email: "sal200000@utdallas.edu"
role: 1
isActive: true
isVerified: true
createdAt: 2021-12-02T00:01:02.773+00:00
updatedAt: 2021-12-02T00:18:38.846+00:00
__v: 0
password: ObjectId("61a80cbeadd6eecbda741dc")
```

```
_id: ObjectId("61a8e24f8d9a91d37e2f0133")
firstName: "Soumya"
lastName: "Mohapatra"
email: "b113057@iiit-bh.ac.in"
role: 0
isActive: true
isVerified: true
createdAt: 2021-12-02T15:12:15.201+00:00
updatedAt: 2021-12-07T03:22:30.773+00:00
__v: 0
password: ObjectId("61a8e24f8d9a91d37e2f0135")
```


Product category Model

```
//PRODUCT CATEGORY MODEL
// This contains category title
{
  title: {
    type: String,
    unique: true,
  },
  isDeleted: {
    type: Boolean,
    default: false,
  },
},
{
  timestamps: {
    createdAt: "createdAt",
    updatedAt: "updatedAt",
  },
}
```

3. LANGUAGES/Framework USED

Our entire application has been hosted on cloud.

Frontend and Image Storage - Firebase

Server - Heroku

Database - Mongo Cloud (Atlas)

Technologies/Frameworks:

Node.js, Express.js, Angular, Handlebars, Bootstrap and MongoDB are used. Object modeling is done using Mongoose. We have implemented OAuth using JWT to handle user authentication and authorization.

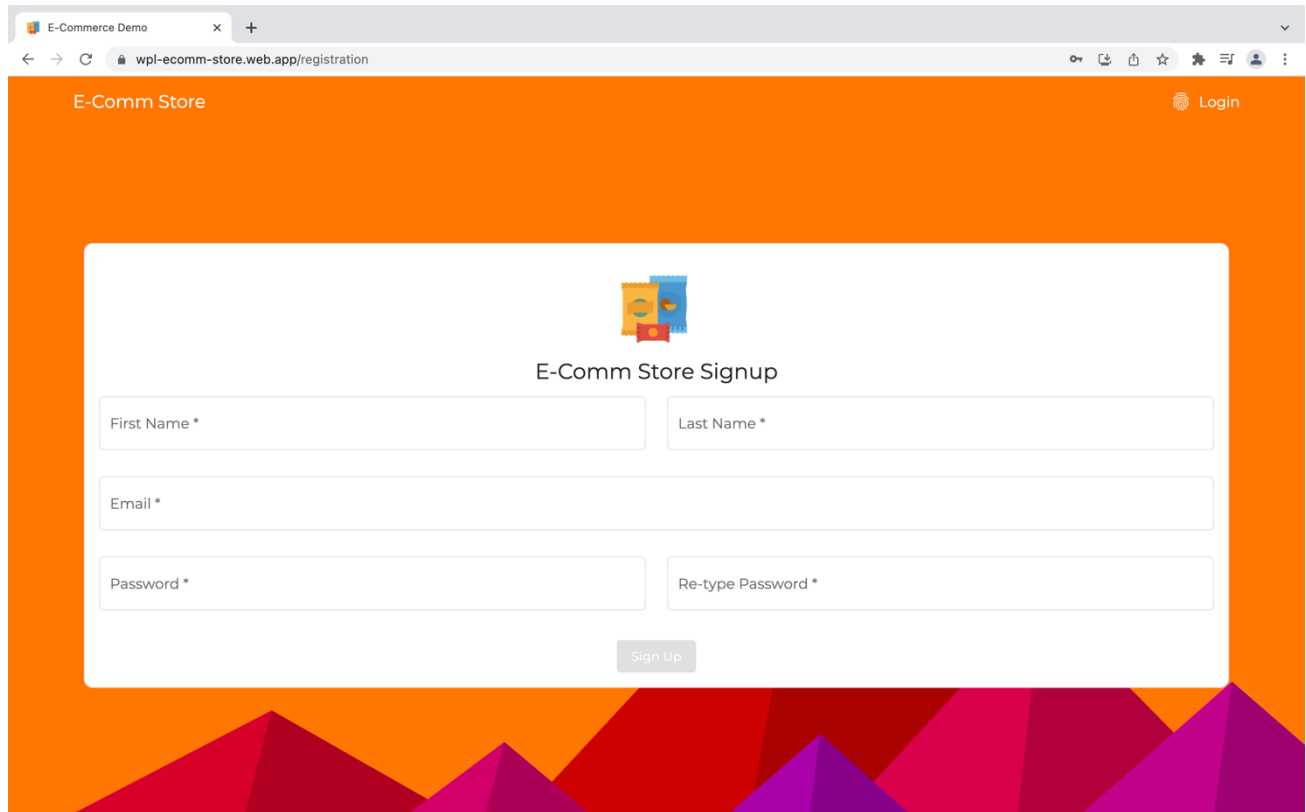
GitHub is used as source code repository.

<https://github.com/aishwaryasingh1122/wpl-ecomm-server>

<https://github.com/aishwaryasingh1122/wpl-ecomm-store>


4. SCREENSHOTS

USER REGISTRATION PAGE:



The screenshot displays a web browser window with the address bar showing "wpl-ecomm-store.web.app/registration". The page has an orange header with "E-Comm Store" on the left and a "Login" link on the right. The main content area features a white registration form titled "E-Comm Store Signup" with a logo of three shopping bags. The form includes input fields for "First Name *", "Last Name *", "Email *", "Password *", and "Re-type Password *", followed by a "Sign Up" button. The background is orange with a colorful geometric pattern at the bottom.

E-Comm Store [Login](#)



E-Comm Store Signup

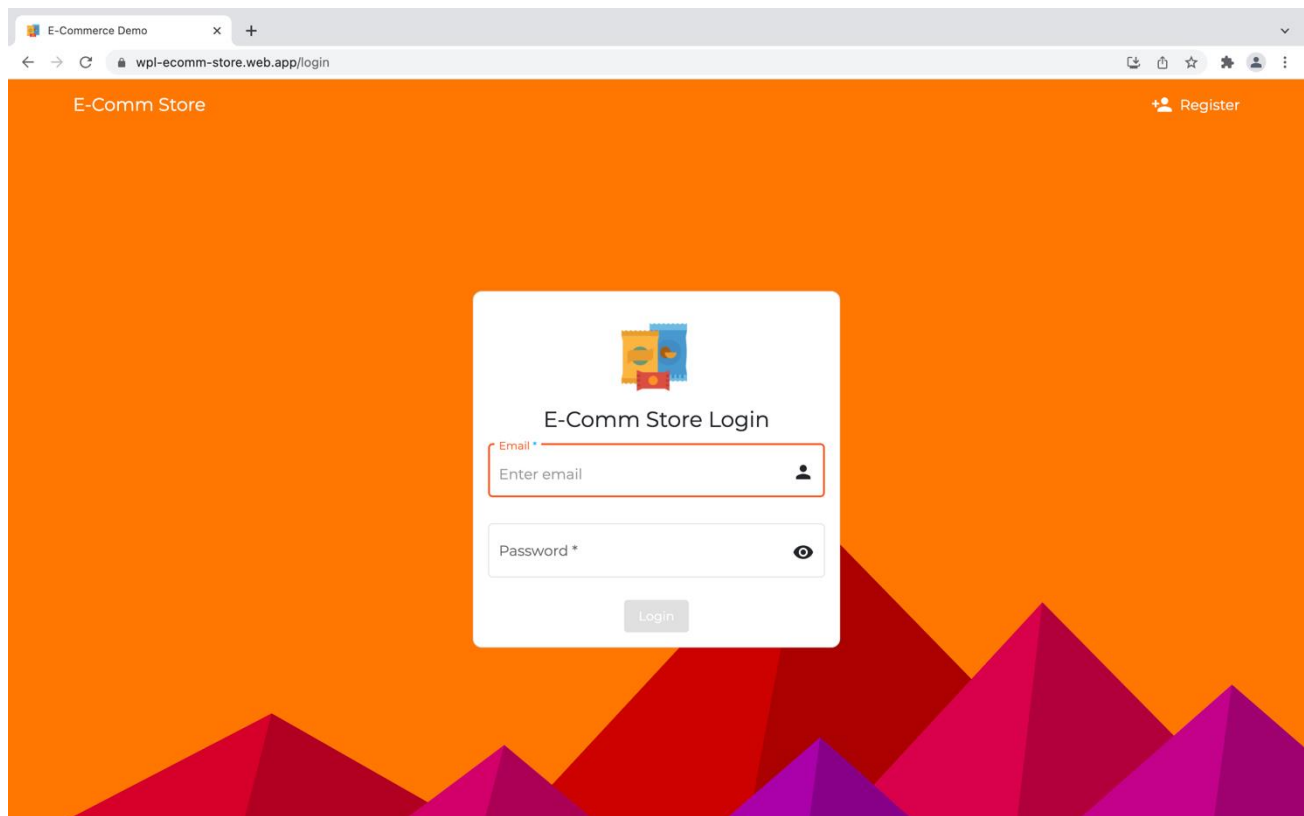
First Name * Last Name *

Email *

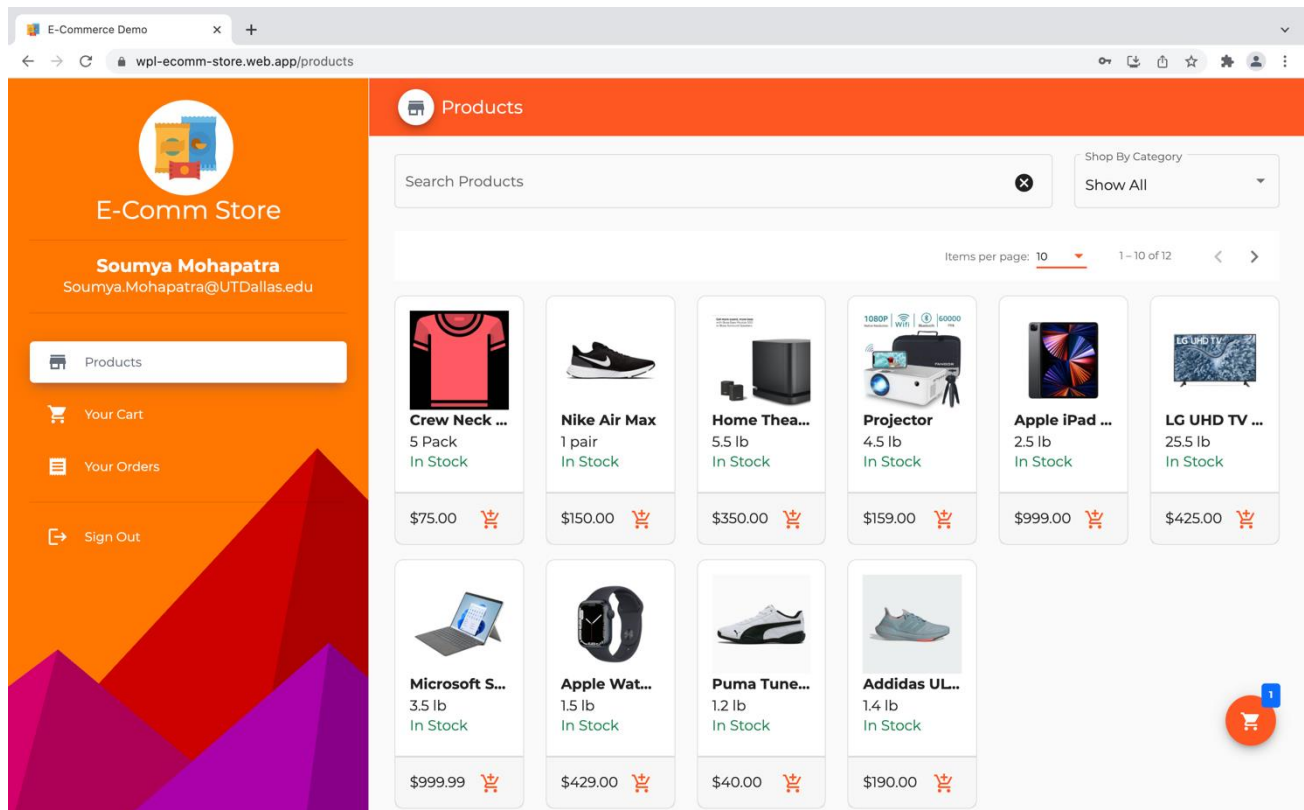
Password * Re-type Password *

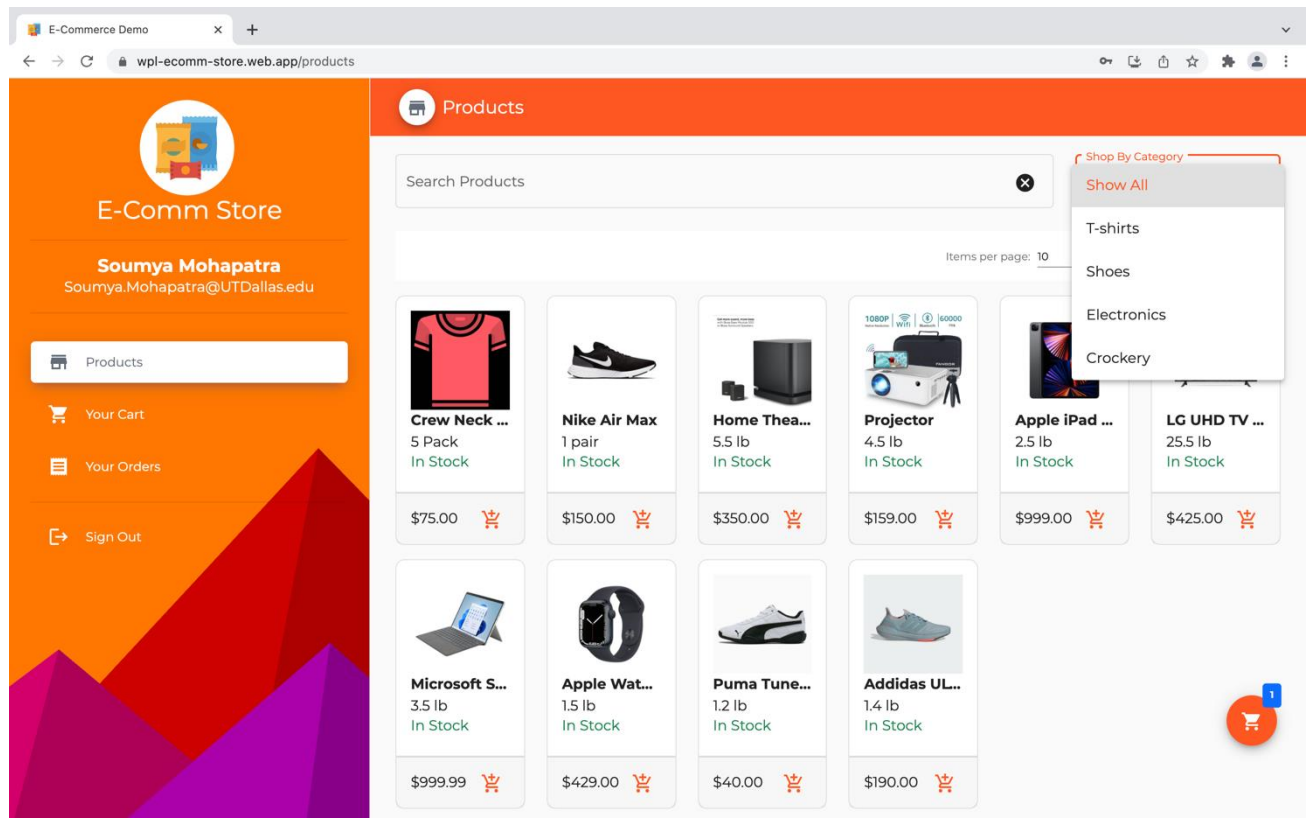
Sign Up

LOGIN PAGE:

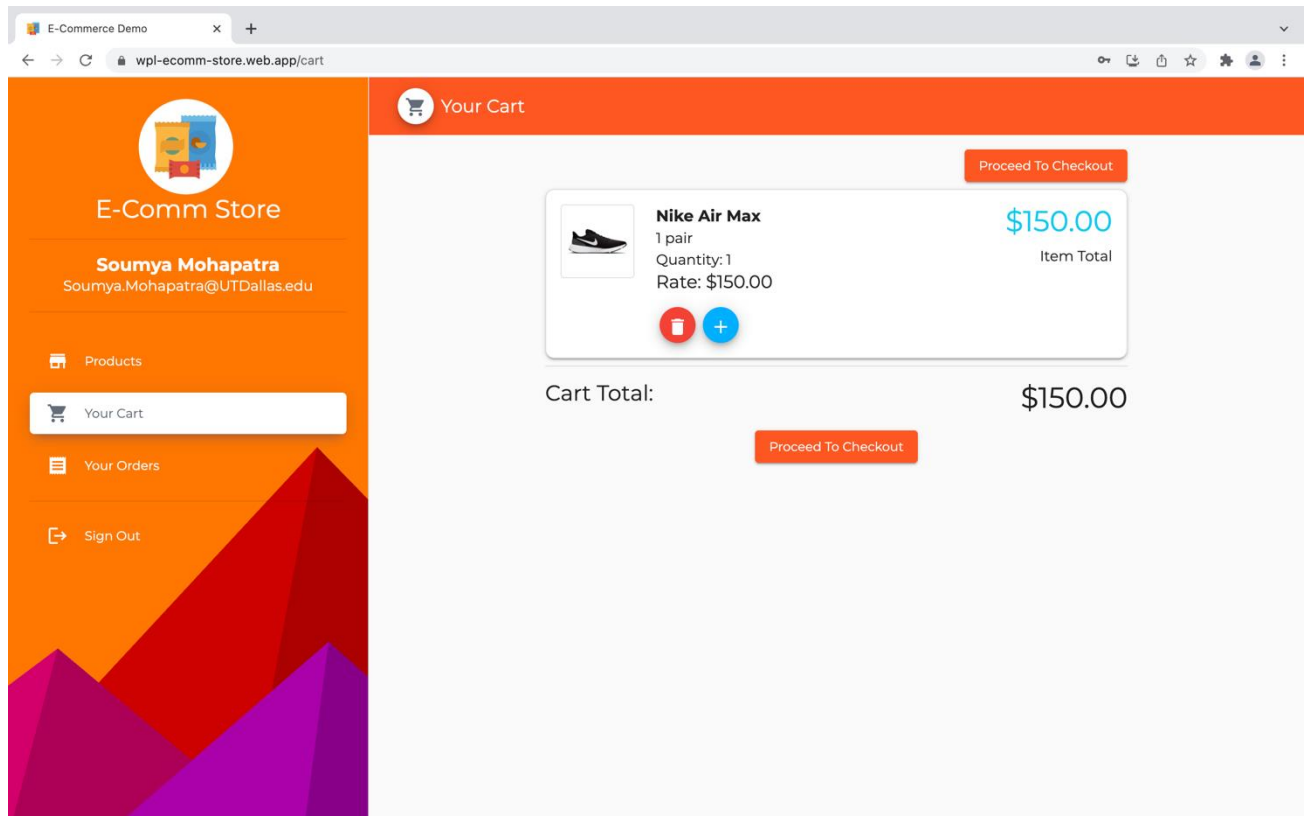


MAIN PAGE LISTING ALL THE PRODUCT DETAILS ALONGWITH THE PAGINATION:

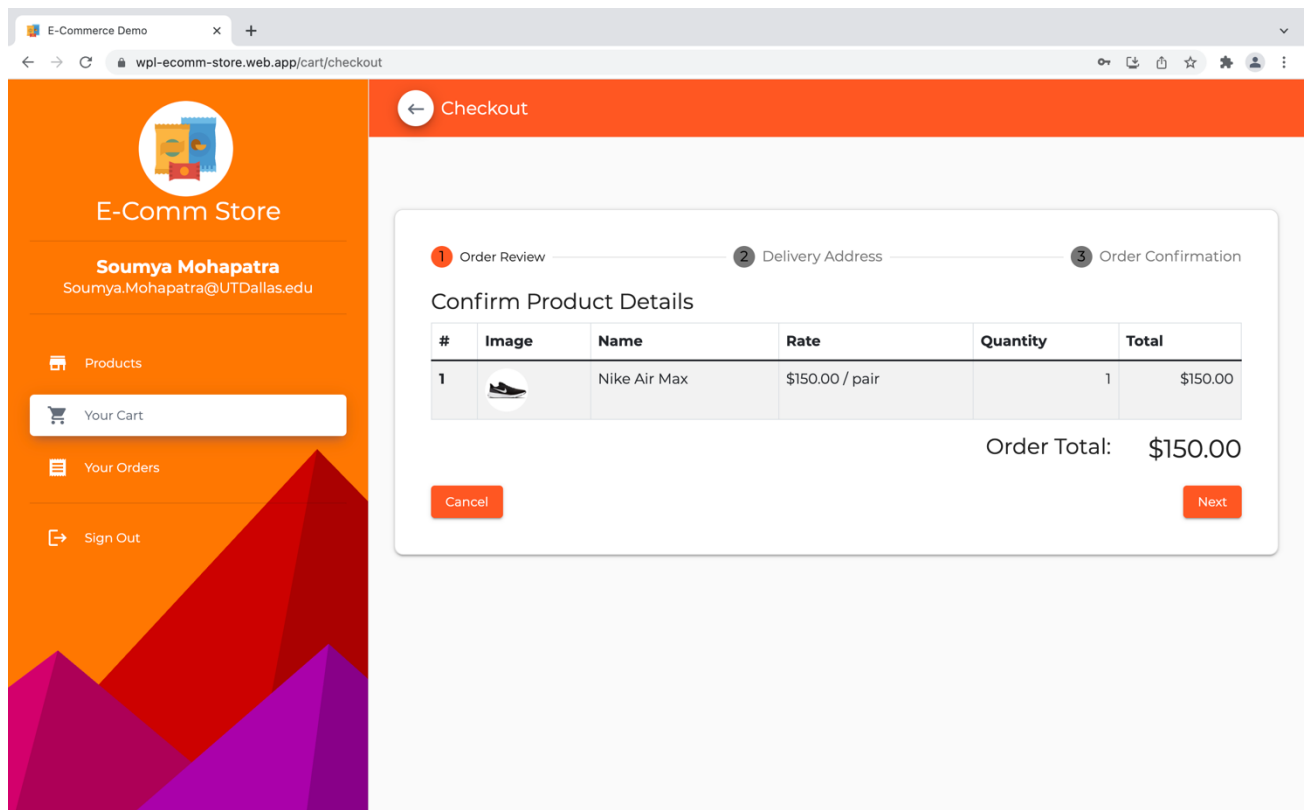


CATEGORY FILTER:

CART PAGE:



CHECKOUT PAGE:



E-Commerce Demo x +

wpl-ecomm-store.web.app/cart/checkout

← Checkout

Add New Address

Name * Mobile Number *

Apartment *

Address Line 1 *

Address Line 2

Pincode *

Delivery Instructions

Cancel Add Address

Order Confirmation

Add New Address

Next

E-Comm Store

Soumya Mohapatra
Soumya.Mohapatra@UTDallas.edu

Products

Your Cart

Your Orders

Sign Out

E-Commerce Demo x +

wpl-ecomm-store.web.app/cart/checkout

← Checkout

Order Review Delivery Address Order Confirmation

Confirm And Place Order

Order Total: \$150.00

Deliver to:

Soumya
4999434790

2629
Estates on Frankford
Pincode: 75252
Delivery Instructions:

Payment Method: **Pay On Delivery**

You can pay by Cash, Credit, Debit, Venmo or Paypal when you receive your products at your doorstep!

Previous Place Order

ORDER DETAILS PAGE:

E-Comm Store
Soumya Mohapatra
Soumya.Mohapatra@UTDallas.edu

- Products
- Your Cart
- Your Orders
- Sign Out

Order Details

Delivery Address
Soumya
Estates on Frankford
2629
Pincode: 75252
Contact: 4999434790

Order Id: E03F88A
Order Status: **NEW**
Date: Dec 06, 2021 10:47:31 pm
Updated: Dec 06, 2021 10:47:31 pm

#	Name	Rate	Quantity	Total
1.	Nike Air Max	\$150.00 / 1 pair	1	\$150.00

Total: \$150.0

ADMIN PAGE FUNCTIONALITIES:

I. CATEGORY MANAGEMENT:

E-Comm Store
Soumya Mohapatra
Soumya.Mohapatra@UTDallas.edu
[Admin Account](#)

- Products
- Your Cart
- Your Orders
- User Management
- Categories Management**
- Products Management
- Orders Management
- Sign Out

Categories Management

Show deleted categories

Product Categories

#	Title	Added On	Action
1	T-shirts	Nov-30-2021	
2	Shoes	Dec-01-2021	

Add new product category

Product Category *

Enter product category

Cancel Submit

II. PRODUCT MANAGEMENT:

The screenshot shows the 'Products Management' interface of the 'E-Comm Store'. The left sidebar contains navigation links: Products, Your Cart, Your Orders, User Management, Categories Management, Products Management (selected), Orders Management, and Sign Out. The main area displays a table of products with columns: #, Image, Name, Weight, Rate, Category, Status, Quantity, Buffer Quantity, and Date Added. A modal titled 'Update quantities for Crew Neck Tees' is open, showing input fields for 'Product Quantity' (set to 100) and 'Buffer Quantity' (set to 10). The modal has 'Cancel' and 'Update' buttons. The background table lists various products including Crew Neck Tees, Lg Uhd Tv 55', Microsoft Surface Pro 8 Laptop, Apple Watch Series 7 Gps, and Puma Tune Cat 3 Shoes Jr.

#	Image	Name	Weight	Rate	Category	Status	Quantity	Buffer Quantity	Date Added
1		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	100	10	Nov-30-2021
2		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	10	10	Dec-01-2021
3		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	25	10	Dec-06-2021
4		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	50	17	Dec-06-2021
5		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	45	20	Dec-06-2021
6		Lg Uhd Tv 55'	25.5 lb	\$425.00	Electronics	Available	15	2	Dec-06-2021
7		Microsoft Surface Pro 8 Laptop	3.5 lb	\$999.99	Electronics	Available	16	4	Dec-06-2021
8		Apple Watch Series 7 Gps	1.5 lb	\$429.00	Electronics	Available	400	20	Dec-06-2021
9		Puma Tune Cat 3 Shoes Jr	1.2 lb	\$40.00	Shoes	Available	200	50	Dec-06-2021

The screenshot shows the 'Products Management' interface of the 'E-Comm Store'. The left sidebar is the same as the previous screenshot. The main area displays the same table of products. A modal titled 'Confirm remove product' is open, asking 'Are you sure you want to remove Crew Neck Tees?'. The modal has a yellow warning icon and 'Cancel' and 'Remove' buttons. The background table lists various products including Crew Neck Tees, Lg Uhd Tv 55', Microsoft Surface Pro 8 Laptop, Apple Watch Series 7 Gps, and Puma Tune Cat 3 Shoes Jr.

#	Image	Name	Weight	Rate	Category	Status	Quantity	Buffer Quantity	Date Added
1		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	100	10	Nov-30-2021
2		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	10	10	Dec-01-2021
3		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	25	10	Dec-06-2021
4		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	50	17	Dec-06-2021
5		Crew Neck Tees	5 Pack	\$75.00	T-shirts	Available	45	20	Dec-06-2021
6		Lg Uhd Tv 55'	25.5 lb	\$425.00	Electronics	Available	15	2	Dec-06-2021
7		Microsoft Surface Pro 8 Laptop	3.5 lb	\$999.99	Electronics	Available	16	4	Dec-06-2021
8		Apple Watch Series 7 Gps	1.5 lb	\$429.00	Electronics	Available	400	20	Dec-06-2021
9		Puma Tune Cat 3 Shoes Jr	1.2 lb	\$40.00	Shoes	Available	200	50	Dec-06-2021

III. ORDER MANAGEMENT:

E-Comm Store
Soumya Mohapatra
Soumya.Mohapatra@UTDallas.edu
[Admin Account](#)

- Products
- Your Cart
- Your Orders
- User Management
- Categories Management
- Products Management
- Orders Management**
- Sign Out

Orders Management

Filter orders by status

Show All

NEW

PROCESSING

DISPATCHED

COMPLETED

Order Id	Date Created	Order Total	Actions
1F6	Dec-06-2021 09:54 pm	\$1,158.00	View Details
43EC1BC	Dec-06-2021 04:56 pm	\$300.00	View Details
43EC00D	Dec-06-2021 04:44 pm	\$570.00	View Details
43EBFAE	Dec-06-2021 04:43 pm	\$750.00	View Details
8E1DE35	Dec-06-2021 03:32 pm	\$900.00	View Details
340BBD4	Dec-06-2021 03:25 pm	\$420.00	View Details
60A9DC8	Dec-06-2021 08:09 am	\$1,540.00	View Details
60A9981	Dec-06-2021 07:03 am	\$1,675.00	View Details
F286ABB	Dec-06-2021 05:38 am	\$420.00	View Details
8BE2D79	Dec-05-2021 09:10 pm	\$480.00	View Details
6501348	Dec-05-2021 08:58 pm	\$790.00	View Details

IV. USER MANAGEMENT:

E-Comm Store
Soumya Mohapatra
Soumya.Mohapatra@UTDallas.edu
[Admin Account](#)

- Products
- Your Cart
- Your Orders
- User Management**
- Categories Management
- Products Management
- Orders Management
- Sign Out

User Management

[Refresh](#)

Users List

#	Name	Email	Role	Account Verification	Account Status
1	Jainam Sheth	j4inam@gmail.com	Admin	Completed	Active
2	Demo User	demouser@gmail.com	User	Completed	Active
3	Aishwarya Singh	aishwaryasingh1122@gmail.com	Super User	Completed	Active
4	Sakshi Laddha	sal200000@utdallas.edu	Admin	Completed	Active
5	Soumya Mohapatra	b113057@iiit-bh.ac.in	User	Completed	Active
6	Sakshi Laddha	laddha.sakshi@yahoo.com	User	Completed	Active
7	Soumya Mohapatra	Soumya.Mohapatra@UTDallas.edu	Admin	Completed	Active
8	Aishwarya Singh	axs200109@utdallas.edu	User	Pending	Active

5. TEAM MEMBERS

- Sakshi Laddha – SAL200000 - 6314.002
- Aishwarya Singh – AXS200109 - 6314.002
- Soumya Mohapatra – SXM200273 -6314.002

6. WORK CONTRIBUTIONS

Aishwarya Singh

- Handled cart management, checkout and orders for users.
- Worked on the entire tech stack to write APIs as well as to implement features on the front end by consuming those APIs
- Designed the MongoDB schema for cart, orders, delivery addresses and also designed the user data flow for related features.
- Implemented the checkout process to allow the user to place an order from the checkout page and have the order history stored to the database
- Implemented admin features for updating order status based on progress. Admin can mark any user created order as: Processing, Dispatched, Completed or Cancelled.
- Implemented the mailing infrastructure to allow users to be notified via email whenever they create a new order
- Implemented filter based on order status for filtering user orders in admin accounts.
- Implemented the logic to calculate total quantities and total price of items
- Handled inventory for products and designed logic to mark a product as: Available, Low Stock or Out of Stock and accordingly restrict users from making purchases that exceed available stocks.
- Defined various express routes for cart, order and addresses
- Files worked on:

Server Side (Node App):

services/cart/index.js, services/orders/index.js, services/addresses/index.js, schema/Cart.js, schema/Address.js, schema/Order.js, routes.js, models.js, utils/mail/index.js, templates/order-confirmation.hbs, utils/cart/index.js

Client Side (Angular App):

All files inside:

modules/cart, modules/orders and

models/address.ts, models/cart.ts, models/order.ts

services/cart.service.ts, services/addresses.service.ts, services/orders.services.ts

Sakshi Laddha

- Handled user management, authentication and authorization for APIs and pages on front end app
- Implemented user sign-up/sign-in process
- Worked on password hashing
- Implemented the navigation bar
- Defined various routes for login ,register, verify user email.
- Implemented middlewares on server to check for authorization headers on each incoming request and to validate if the client has a role of user or admin.
- Designed the User Model and Auth Model .
- Implemented the logic of displaying Option in Nav Bar according to type of User signedin.
- Files worked on:

Server Side(Node App):

services/auth/index.js, services/users/index.js, routes.js, models.js

Client Side(Anglar App):

All files inside

modules/auth, modules/user,

services/auth.service.ts, services/user.service.ts,

Soumya Mohapatra

- Handled products and product category management
- Implemented CRUD functionalities for admin including Soft Delete.
- Implemented adding and editing product details along with image upload for products.
- Implemented search and filter functionalities for products.
- Implemented the logic of changing/adding Product image by Admin.
- Designed product catalog, product management, category management pages on front end app.
- Modeled products and category data.
- Defined various express routes for products and category management.
- Files worked on:

Server Side (Node App):

services/products/index.js,

services/categories/index.js,

schema/Product.js, schema/ProductCategory.js

Client Side (Angular App):

All files inside

modules/product, modules/product-management,

modules/product-categories-management,

services/products.service.ts, services/categories.service.ts

7. Link for Recording of Project walkthrough-

<https://cometmail->

my.sharepoint.com/:v:/g/personal/axs200109_utdallas_edu/EYZ7lvT8xf5Gtzf2EbKjEiYBg0Y1V8D

[PKODnLaFbKnQerQ?e=BNQTUt](https://my.sharepoint.com/:v:/g/personal/axs200109_utdallas_edu/EYZ7lvT8xf5Gtzf2EbKjEiYBg0Y1V8D?e=BNQTUt)

