

CS 6314.0W1 - Web Programming Languages - S21

Project Title - Beer & Wine Distribution E-Commerce Website

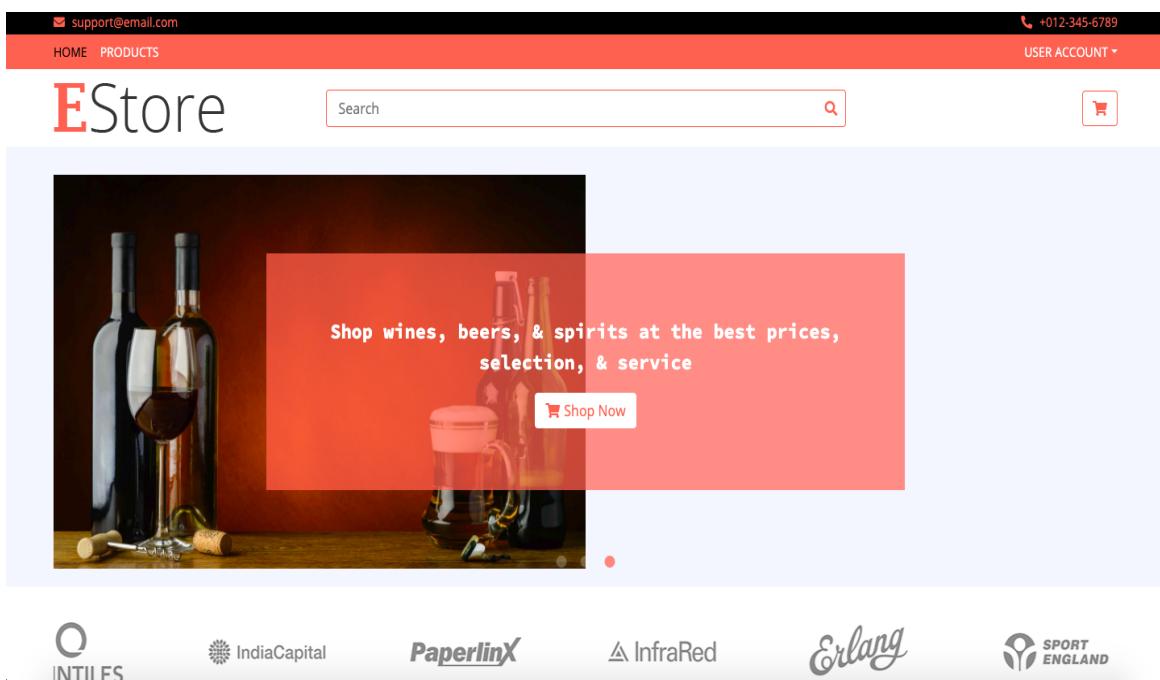
Team 27

Ashwani Kumar Kashyap - axk190033

Ruchi Singh - rxs180057

Soumil Mandal - sxm180159

Website Name - Beer & Wine Distribution E-Commerce



Contents

1. Project Description
2. Architecture
3. Technologies Used
 - Front End
 - Backend
 - Storage System
4. Database Design
5. Main Functionalities
 - User Signup
 - User SignIn
 - Updating User Details
 - View All Products
 - Searching & Filtering Products
 - Cart Operations (Add, Update & Delete Items)
 - Place Order & Checkout
 - View Ordered Products History
 - Admin View/Add/Update/Delete
 - Soft Deletion of Products
6. Work Division

Project Description

We have created an ecommerce website for selling liquors such as wine and beer. Our project has a strong user-interface which gives the users a myriad of options while buying. In the backend, we have implemented several security measures such as password policy, password hashing and validating phone numbers and email addresses. We also check uniqueness of credentials while a new user is signing up. For products, we have textual search, as well as filtering based on categories and price range with paging incorporated.

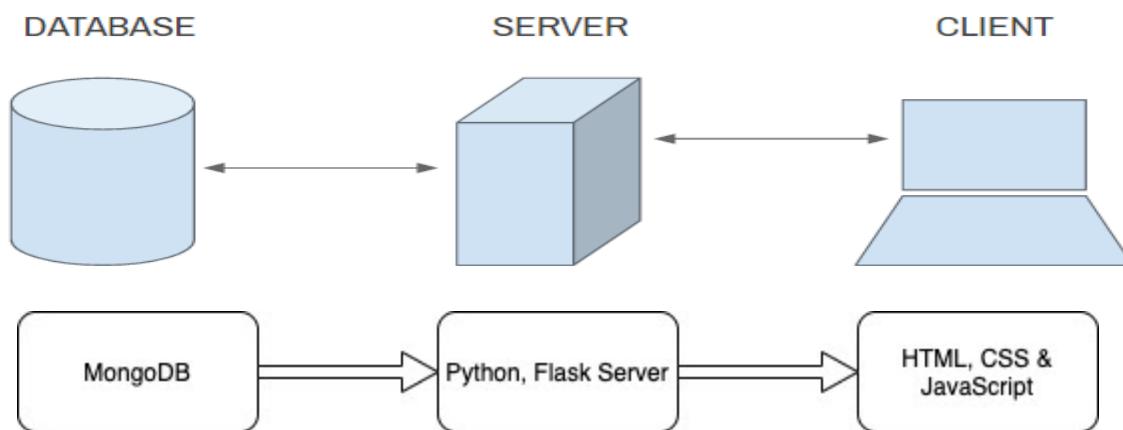
Important Components of the System -

The two main important components of the system are - Admin and Users, which form the very basic foundation of the website.

Admin - A person who has additional access to features of the ecommerce portal and able to list all the products in the ecommerce website, add new products, delete products and update products. Only the admin can add/change the picture of the product added in the ecommerce portal.

User - A person who can register himself/herself and then login into the website anytime. User can view all the available products, search for specific products, filter products based on the category and price range. After that, users can add products into the cart and place the order after providing the shipping address. Users can see their ordered products history as well.

Architecture



Technologies Used

Front-End

1. HTML, CSS JavaScript
2. Bootstrap <https://icons.getbootstrap.com/>
3. Jquery <https://jquery.com/>
4. Font Awesome fonts <https://fontawesome.com/>

Back-End

1. Programming Language - Python 3.x
2. Web-Framework - Flask (<https://pypi.org/project/Flask/>)
3. MongoDB Interaction - PyMongo (<https://pypi.org/project/pymongo/>)
4. Password Hashing - bcrypt (<https://pypi.org/project/bcrypt/>)
5. Password Policy - password-strength (<https://pypi.org/project/password-strength/>)
6. WSGI Utilities - Werkzeug (<https://pypi.org/project/Werkzeug/>)
7. String Validations - Regex (<https://docs.python.org/3/library/re.html>)

Database

1. MongoDB Atlas which provides a shared cluster and allows users to create the database on cloud.

Database Design

For our database, we took a NoSQL approach using MongoDB. Our database beer_wine_website consists of 4 collections, namely - product_details, customer_details, cart and order. A brief description of all is given below

1. product_details - Collection used for storing details of all products present in inventory
 - a. _id (ObjectId) - unique id generated by MongoDB
 - b. name (String) - name of the product
 - c. price (Float) - price of the product
 - d. description (String) - description of the product
 - e. images (Array) - store locations of product images
 - f. category (String) - to store whether product is of type wine or beer
 - g. stock (Integer) - number of items present in inventory
 - h. brand (String) - brand name of the product
 - i. deleted (Boolean) - to store flag for soft-deletion

```

_id: ObjectId("6090f2a97abd2e3115054af1")
name: "DINO"
brand: "Pinot Gregio"
description: "DINO Pinot Gregio White Wine"
stock: 100
category: "wine"
price: 75
deleted: false
images: Array
  0: ".../static/prod_images/6090f2a97abd2e3115054af1-photo-1601303981356-528..."

```

Product_details table

2. customer_details - Collection used for storing details of all registered users

- _id (ObjectId) - unique id generated by MongoDB
- user_name (String) - customer username for logging into account
- full_name (String) - customer full name for credentials
- email_id (String) - customer email address
- password (Binary) - hashed password of user
- shipping_address (Object) - contains address object with details like city, state, zip code, etc
- contact_no (String) - customer contact number
- govt_id (String) - customer government id number
- isAdmin (Boolean) - flag to store if user has admin rights

```

_id: ObjectId("6091976ea9d68b5cd754f12b")
user_name: "soumandal"
full_name: "Soumil Mandal"
email_id: "smandal@gmail.com"
password: Binary('JDJiJDEyJHVESGdDYnpiTnJleE00a15BQWZUUXV2M0hSdmka2pMTlFyUXdLdjZsL0lBNFvSkZudXht', 0)
shipping_address: Object
  first_name: "Soumil"
  last_name: "Mandal"
  email: "smandal@gmail.com"
  phone: "4692239133"
  address: "800 W Renner Rd"
  city: "Richardson"
  state: "TX"
  zip_code: "75080"
  country: "USA"
contact_no: "4692260399"
govt_id: "1942"
isAdmin: true

```

3. cart - Collection used for storing required cart information based on a certain customer

- _id (ObjectId) - unique id generated by MongoDB
- customer_id (String) - customer id of person to whom this cart belongs, this refers to _id (ObjectId) attribute of customer_details table
- product_ids (Array) - contains objects with product_id and quantity data
- total_price (Float) - computed total price of the products in the cart

```

_id: ObjectId("6091ce8b15d2661ecfce1573")
customer_id: "6091cdc415d2661ecfce1572"
product_ids: Array
  0: Object
    product_id: "6091bba75b306694f09826ca"
    quantity: 1
  1: Object
    product_id: "6091bbee5b306694f09826cb"
    quantity: 1
  2: Object
    product_id: "6091bb205b306694f09826c9"
    quantity: 2
total_price: 84

```

cart table

4. order - Collection used for strong all orders made with respect to a certain customer
 - a. _id (ObjectId) - unique id generated by MongoDB
 - b. customer_id (String) - customer id of person to whom this order belongs, this refers to _id (ObjectId) attribute of customer_details table
 - c. shipping_address (Object) - shipping address of customer
 - d. order_date (Date) - store date and time when order was made
 - e. total_price_post_charges (Float) - total price of cart + additional costs
 - f. order_status (String) - store status of the order, like order received/shipped /delivered, etc
 - g. customer_cart (Object) - this is the whole cart object pulled from the cart table based on a certain customer_id

```

_id: ObjectId("6091dbb940006673d4450467")
customer_id: "6091db5e40006673d4450465"
shipping_address: Object
  first_name: "Ashwani"
  last_name: "Kashyap"
  email: "ashwanikumarkashyap22@gmail.com"
  phone: "4692309274"
  address: "7575 Frankford Rd"
  country: "United States"
  city: "Dallas"
  state: "Texas"
  zip_code: "75252"
order_date: 2021-05-04T18:41:45.496+00:00
customer_cart: Object
  _id: ObjectId("6091db8d40006673d4450466")
  customer_id: "6091db5e40006673d4450465"
  product_ids: Array
    0: Object
      product_id: "608496314fc85d619057f291"
      quantity: 5
      product_details: Object
        _id: ObjectId("608496314fc85d619057f291")
        name: "Jacob's Pater"
        images: Array
          price: 120
          description: "Pater Red Wine"
          deleted: false
          category: "wine"
          stock: 95
          brand: "Red Wine"
        total_price: 600
      total_price_post_charges: 612
      order_status: "order received"

```

order table

Main Functionalities

1. User SignUp

Any new user can SignUp or register by providing valid details. All the input fields are mandatory. The inputs are validated both in the front-end and back-end and error messages are provided for any invalid entry. We have also incorporated strong password checks and provide the error message if the user doesn't provide the password according to the constraint.

The screenshot shows the EStore website's user registration page. At the top, there is a navigation bar with links for support@mailto:, +012-345-6789, and USER ACCOUNT. Below the navigation bar, the EStore logo is displayed, along with a search bar and a shopping cart icon. The main content area shows two forms side-by-side. The left form is for user registration, with fields for Username (virat), Full Name (Virat Kohli), E-mail (virat.kohli@gmail.com), Mobile No (4692341933), Password, Retype Password, and Government ID Number (5557). The right form is for logging in, with fields for Username and Password, a checked checkbox for 'Log in as admin', and a 'Submit' button. The URL in the browser's address bar is /Products/LogInRegister.

Get in Touch

📍 123 E Store, Los Angeles, USA
✉️ email@example.com

Follow Us



Company Info

➤ About Us
➤ Privacy Policy

Purchase Info

➤ Payment Policy
➤ Shipping Policy

After successful SignUp, we can see that an entry of the new user has been created in our customer_details collection. Also, the password is stored in a hashed form and is not human readable. If the user tries to register again with the same username or email, our system throws an error that the user already exists. Also, by default, the isAdmin flag is set to False.

```
_id: ObjectId("6092031fc170a962a88be3a6")
user_name: "virat"
full_name: "Virat Kohli"
email_id: "virat.kohli@gmail.com"
password: Binary('JDjiJDDEyJG1jR2owdmNZMmx2S1dy0HYvcHJaYy5ZdU12b1NwRG1Da0dETUw1MGNyVnhkL05sbVdGY0Zt', 0)
> shipping_address: Object
contact_no: "4692341933"
govt_id: "5557"
isAdmin: false
```

2. User SignIn

Any registered user can sign in using valid credentials. For admins, they can check the “Login as admin” check-box, and if credentials are successfully validated, the admin view will open instead of the user view.

The screenshot shows the EStore login page. At the top, there's a navigation bar with links for support@email.com, +012-345-6789, and USER ACCOUNT. Below the navigation is the EStore logo and a search bar. The main content area has two side-by-side forms. The left form is for regular users, requiring fields like Username, Full Name, E-mail, Mobile No, Password, Retype Password, and Government ID Number. The right form is for admins, requiring Username, Password, and a checked "Log in as admin" checkbox. Both forms have a "Submit" button at the bottom.

Get in Touch

123 E Store, Los Angeles, USA
email@example.com

Follow Us



Company Info

About Us
Privacy Policy

Purchase Info

Payment Policy
Shipping Policy

3. Updating User Details

After logging in, an user can provide/update his/her shipping address details and other contact information. This information is populated automatically when an user tries to place the order. If not, then the user has to provide

The screenshot shows the EStore address update page. At the top, there's a navigation bar with links for support@email.com, +012-345-6789, and MY ACCOUNT. Below the navigation is the EStore logo and a search bar. The main content area has a sidebar with links for Orders, Address (which is highlighted in red), and Logout. The main form is for updating an address, with fields for First Name (Virat), Last Name (Kumar Kohli), E-mail (virat@yahoo.com), Mobile No (4892743991), Address (850 N Renner Rd), Country (USA), City (Plano), State (TX), and ZIP Code (75047). A "Update Address" button is at the bottom of the form.

shipping address at the time of order placement

After clicking on “Update Address”, the changes are reflected in MongoDB.

```
_id: ObjectId("6092031fc170a962a88be3a6")
user_name: "virat"
full_name: "Virat Kohli"
email_id: "virat.kohli@gmail.com"
password: Binary('JDJiJDEyJG1jR2owdmNZMmx2S1dy0HYvcHJaYy5zdUl2b1NwRG1Da0dETUw1MGNyVnhkL05sbVdGY0Zt', 0)
shipping_address: Object
  first_name: "Virat"
  last_name: "Kumar Kohli"
  email: "virat@yahoo.com"
  phone: "4892743991"
  address: "850 N Renner Rd"
  city: "Plano"
  state: "TX"
  zip_code: "75047"
  country: "USA"
contact_no: "4692341933"
govt_id: "557"
isAdmin: false
```

4. View All Products

Clicking on the “PRODUCTS” button on the Navigation pane, the user can view all the products available in the inventory with the price and the available stock.

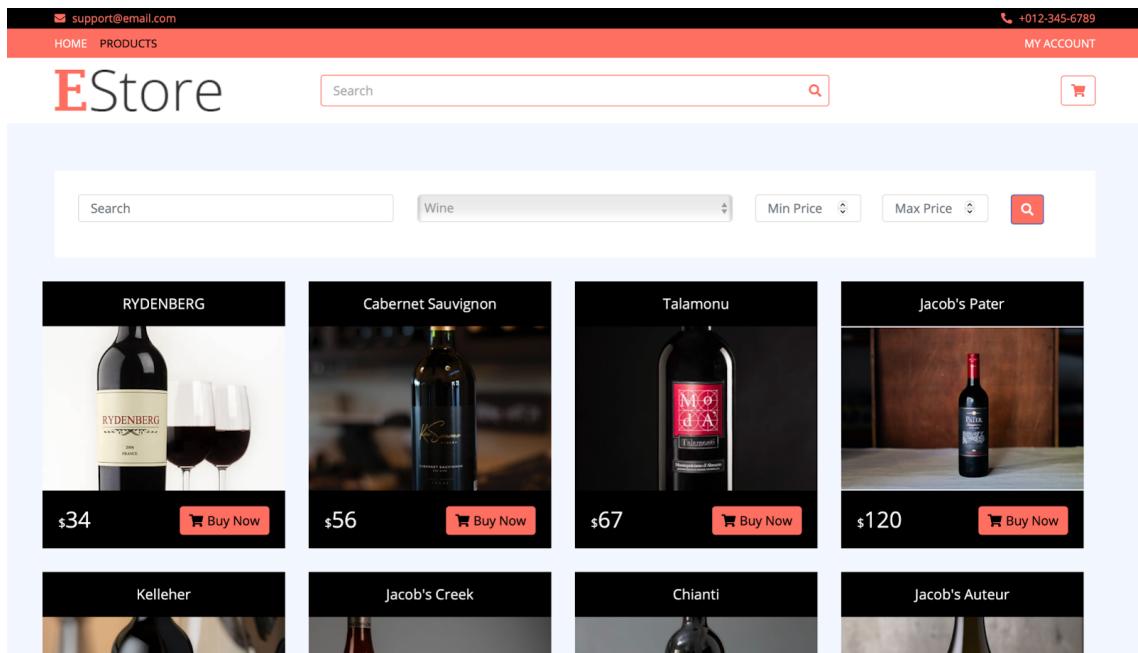
The screenshot shows the EStore website interface. At the top, there is a navigation bar with links for support@email.com, +012-345-6789, HOME, PRODUCTS, and MY ACCOUNT. Below the navigation bar, the EStore logo is displayed next to a search bar with a magnifying glass icon and a shopping cart icon. The main content area features a grid of wine products. Each product card includes the wine name, a small image of the bottle, the price, and a 'Buy Now' button. The products shown are RYDENBERG (price \$34), Cabernet Sauvignon (price \$56), Talamonu (price \$67), Jacob's Pater (price \$120), Kelleher, Jacob's Creek, Chianti, and Jacob's Auteur.

Product Name	Price	Action
RYDENBERG	\$34	Buy Now
Cabernet Sauvignon	\$56	Buy Now
Talamonu	\$67	Buy Now
Jacob's Pater	\$120	Buy Now
Kelleher		
Jacob's Creek		
Chianti		
Jacob's Auteur		

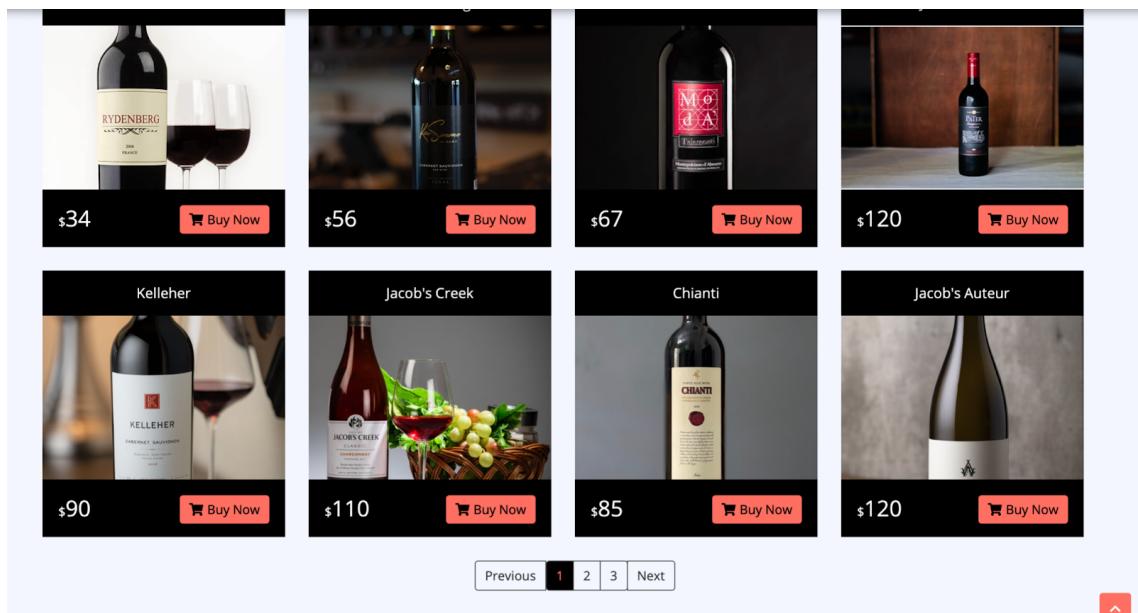
5. Searching & Filtering Products

View Products By Category

The user has the option to filter by category. For example, selecting the category as wine and clicking on the search button shows all the wines available in the inventory.

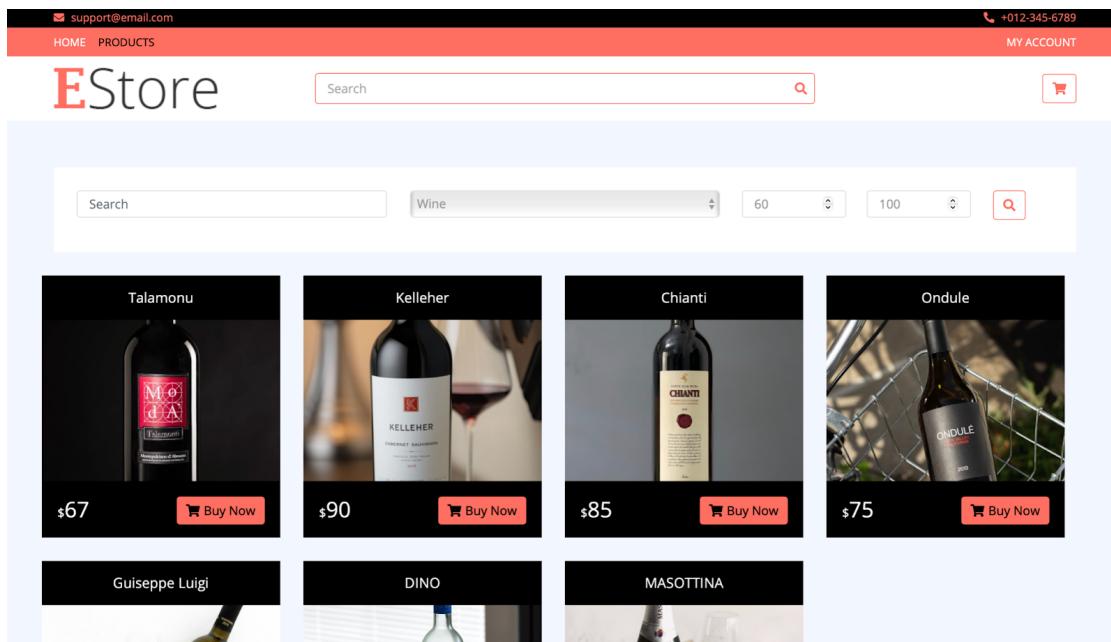


On each page, we provide pagination as well. Users can go to different pages to see other products.



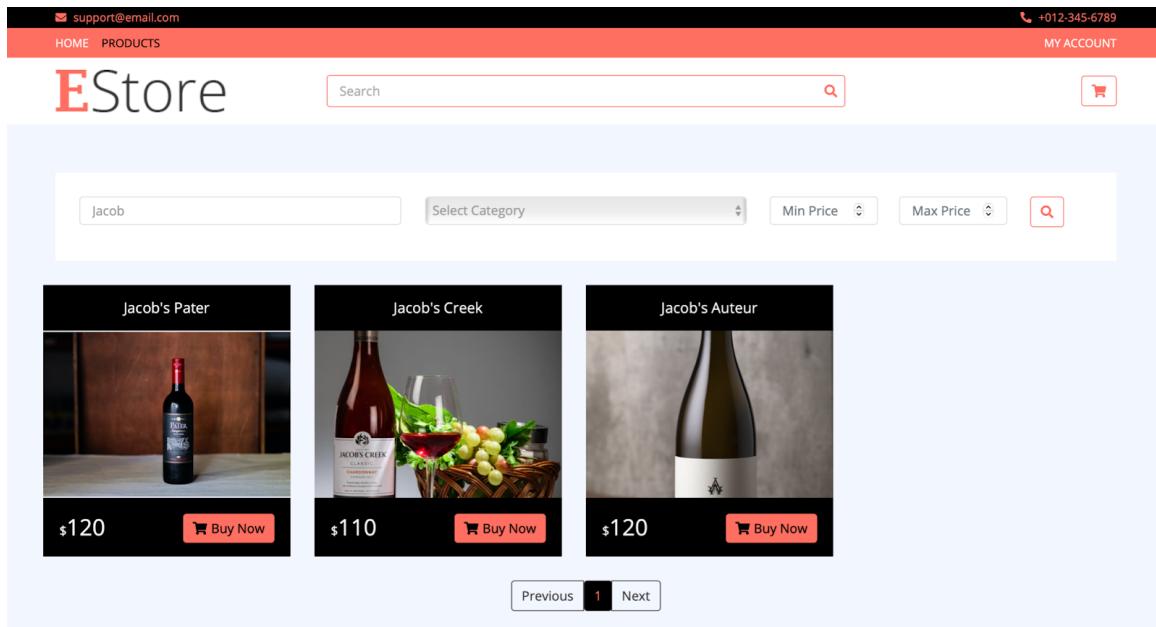
View Products By Price Range

The user also has the option of adding price ranges for filtering products. For example, here setting the price range between 60 and 100 will show the products in that range. Paging functionality has also been incorporated in filtering functionality.



Product Search By Name

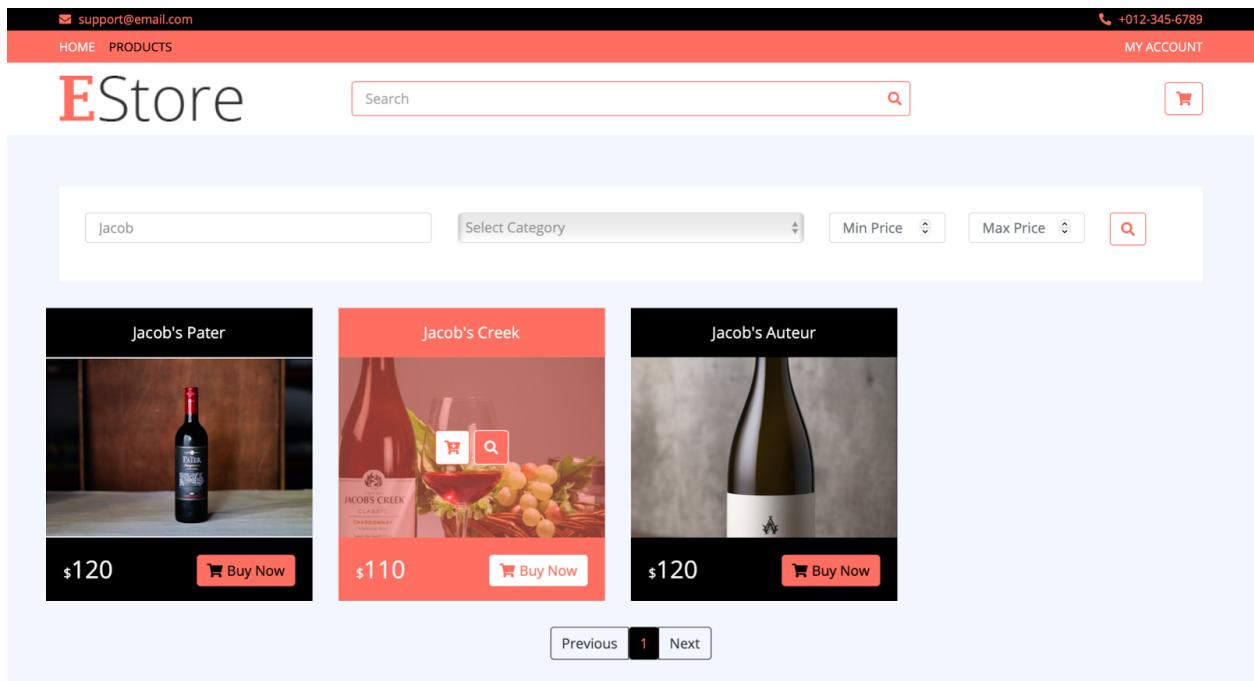
In addition to the above filtering methods, an user also has the luxury of textual search. In such cases, the backend will return all the products having that string in the product title. For example, here searching using the phrase “Jacob” displays all the products with that phrase in their title. Paging functionality has also been incorporated in Searching functionality as well.



6. Cart

Adding Product To Cart

An user can hover over the item and the cart button will appear. Clicking on this button will add the product to the users Cart.



To view all the items an user has in their Cart, they can simply click on the cart button which appears on the upper right hand corner of the screen.

Product	Price	Quantity	Total	Remove
Jacob's Creek	\$110	- 1 +	\$110	

Cart Summary

Sub Total	\$110
Shipping Cost	\$12
Grand Total	\$122

Update Cart **Checkout**

Get in Touch

123 E Store, Los Angeles, USA
email@example.com
+123-456-7890

Follow Us



Company Info

About Us
Privacy Policy
Terms & Condition

Purchase Info

Payment Policy
Shipping Policy
Return Policy

Adding a product to the Cart is also reflected on the database as shown below.

```
_id: ObjectId("60920822c170a962a88be3a7")
customer_id: "6092031fc170a962a88be3a6"
product_ids: Array
  0: Object
    product_id: "6084966e4fc85d619057f293"
    quantity: 1
    total_price: 110
```

Modify Product Quantity In Cart

The website also gives the user the functionality of modifying the Cart. One can easily change the quantity of the product they have added by clicking on the +/- buttons.

The screenshot shows a web application for an e-commerce store named 'EStore'. At the top, there is a navigation bar with links for 'support@email.com', '+012-345-6789', 'HOME', 'PRODUCTS', 'MY ACCOUNT', and a search bar. Below the navigation bar, the store logo 'EStore' is displayed next to a search input field and a magnifying glass icon. The main content area features a table for managing the shopping cart. The table has columns for 'Product', 'Price', 'Quantity', 'Total', and 'Remove'. A row in the table shows a product named 'Jacob's Creek' with a price of '\$110', a quantity of '3' (with minus and plus buttons), a total of '330', and a 'Remove' button. To the right of the cart table is a 'Cart Summary' box containing the following information:

Cart Summary	
Sub Total	\$330
Shipping Cost	\$12
Grand Total	\$342

At the bottom of the summary box are two buttons: 'Update Cart' and 'Checkout'.

Get in Touch

📍 123 E Store, Los Angeles, USA
✉️ email@example.com
📞 +123-456-7890

Follow Us



Company Info

➤ About Us
➤ Privacy Policy
➤ Terms & Condition

Purchase Info

➤ Payment Policy
➤ Shipping Policy
➤ Return Policy

We Accept:

Secured By:

The database is updated as well once the user changes the values and clicks the “Update Cart” button.

```

_id: ObjectId("60920822c170a962a88be3a7")
customer_id: "6092031fc170a962a88be3a6"
product_ids: Array
  0: Object
    product_id: "6084966e4fc85d619057f293"
    quantity: 3
  total_price: 330

```

Delete Item From Cart

Clicking on the trashcan icon that appears on the left side of each product removed the item from the users Cart.

Product	Price	Quantity	Total	Remove

Cart Summary

Sub Total	\$0
Shipping Cost	\$0
Grand Total	\$0

[Update Cart](#) [Checkout](#)

Get in Touch

📍 123 E Store, Los Angeles, USA
✉️ email@example.com
📞 +123-456-7890

Follow Us



Company Info

➤ About Us
➤ Privacy Policy
➤ Terms & Condition

Purchase Info

➤ Payment Policy
➤ Shipping Policy
➤ Return Policy

We Accept:

Secured By:

Like quantity update, deletion is also reflected on the database. Here, we can see that post deletion, there are no items in the product_ids Array.

```

_id: ObjectId("60920822c170a962a88be3a7")
customer_id: "6092031fc170a962a88be3a6"
product_ids: Array
  total_price: 0

```

7. Place An Order & Checkout

Once satisfied with the Cart items, an user can click on the “Checkout” button for proceeding with the order.

The screenshot shows the EStore website's shopping cart page. At the top, there is a navigation bar with links for support@email.com, HOME, PRODUCTS, +012-345-6789, and MY ACCOUNT. The main header says "EStore". Below the header is a search bar and a shopping cart icon. The central area displays a table of items in the cart:

Product	Price	Quantity	Total	Remove
RYDENBERG	\$34	- 2 +	\$68	
BRAND Goods	\$10	- 1 +	\$10	
CLAUSTHALER	\$20	- 1 +	\$20	

To the right, a "Cart Summary" box shows the following details:

Cart Summary	
Sub Total	\$98
Shipping Cost	\$12
Grand Total	\$110

At the bottom of the cart summary are "Update Cart" and "Checkout" buttons. Below the cart area, there are four sections: "Get in Touch", "Follow Us", "Company Info", and "Purchase Info", each with links to About Us, Privacy Policy, and Terms & Condition. There are also icons for payment methods (VISA, MasterCard, PayPal, American Express, Discover) and security (SECURED, SECURE, SSL SECURE).

The user will next have to provide a shipping address (or can stick with the default address) and has to choose a method of payment.

The screenshot shows the EStore website's checkout page. On the left, there is a "Shipping Address" form with fields for First Name (Virat), Last Name (Kumar Kohli), E-mail (virat@yahoo.com), Mobile No (4892743991), Address (850 N Renner Rd), Country (USA), City (Plano), State (TX), and ZIP Code (75047). Below the form is a "Set as default Address" button. On the right, there are two columns: "Cart Total" and "Payment Methods".

Cart Total

Item	Quantity	Total
RYDENBERG	x (2)	\$68
BRAND Goods	x (1)	\$10
CLAUSTHALER	x (1)	\$20
Sub Total		\$98
Shipping Cost		\$12
Grand Total		\$110

Payment Methods

- Paypal
- Direct Bank Transfer
- Credit/Debit Card

At the bottom right is a large "Place Order" button.

8. View Ordered Products History

To view all the orders/order history, an user can navigate to Orders by clicking “MY ACCOUNT” > “ORDERS”.

The screenshot shows the EStore website's order history section. At the top, there is a navigation bar with links for support@email.com, +012-345-6789, HOME, PRODUCTS, MY ACCOUNT, and a search bar. Below the navigation is the EStore logo and a shopping cart icon. On the left, a sidebar menu includes 'Orders', 'Address', and 'Logout'. The main content area displays a table of order details:

No	Delivery Address	Date	Price	Status	Action
1	850 N Renner Rd Plano, USA	Tue May 04 2021	\$110	order received	View

Below the table, there are four sections: 'Get in Touch' (with address, email, and phone), 'Follow Us' (social media icons), 'Company Info' (links to About Us, Privacy Policy, and Terms & Condition), and 'Purchase Info' (links to Payment Policy, Shipping Policy, and Return Policy). At the bottom, it says 'We Accept:' followed by payment method icons (Visa, MasterCard, PayPal, American Express, and Visa Electron) and 'Secured By:' followed by security logos (Secure, Verified, and SSL Secure). A copyright notice at the very bottom reads "Copyright © WPL Project. All Rights Reserved".

To view order summary, the user can click on the “View” button that appears on the left side of each order.

This screenshot shows a modal window titled 'Order Detail' overlaid on the EStore website. The modal contains a table of product details:

Product	Price	Quantity	Total
RYDENBERG	\$34	2	\$68
BRAND Goods	\$10	1	\$10
CLAUSTHALER	\$20	1	\$20

On the right side of the modal, there is a separate 'Order Summary' section with the following bullet points:

- Sub Total: \$98
- Shipping Cost: \$12
- Grand Total: \$110

At the bottom of the modal, there is a status column and an 'Action' column, with the first row showing 'order received' and a 'View' button.

The background of the website shows the same layout as the previous screenshot, including the 'Get in Touch', 'Follow Us', 'Company Info', 'Purchase Info', payment acceptance icons, and security seals.

After successfully placing an order, a new entry is created in the orders collection.

```
_id: ObjectId("60920af8c170a962a88be3a8")
customer_id: "6092031fc170a962a88be3a6"
shipping_address: Object
order_date: 2021-05-04T22:03:20.563+00:00
customer_cart: Object
  _id: ObjectId("60920822c170a962a88be3a7")
  customer_id: "6092031fc170a962a88be3a6"
  product_ids: Array
    0: Object
      product_id: "6084953f4fc85d619057f28e"
      quantity: 2
      product_details: Object
    1: Object
      product_id: "6090fdad314f64277bf203ef"
      quantity: 1
      product_details: Object
    2: Object
      product_id: "6090fd57314f64277bf203ee"
      quantity: 1
      product_details: Object
    total_price: 98
  total_price_post_charges: 110
order_status: "order received"
```

Here, we can see that the whole Cart object is taken from the Cart collection and is added. Once added, the Cart entry is deleted from the Cart collection as it will no longer be needed and the user will start from an empty Cart from next time. We also have some new fields like order_date, total_price_post_charges and order_status.

9. Admin View

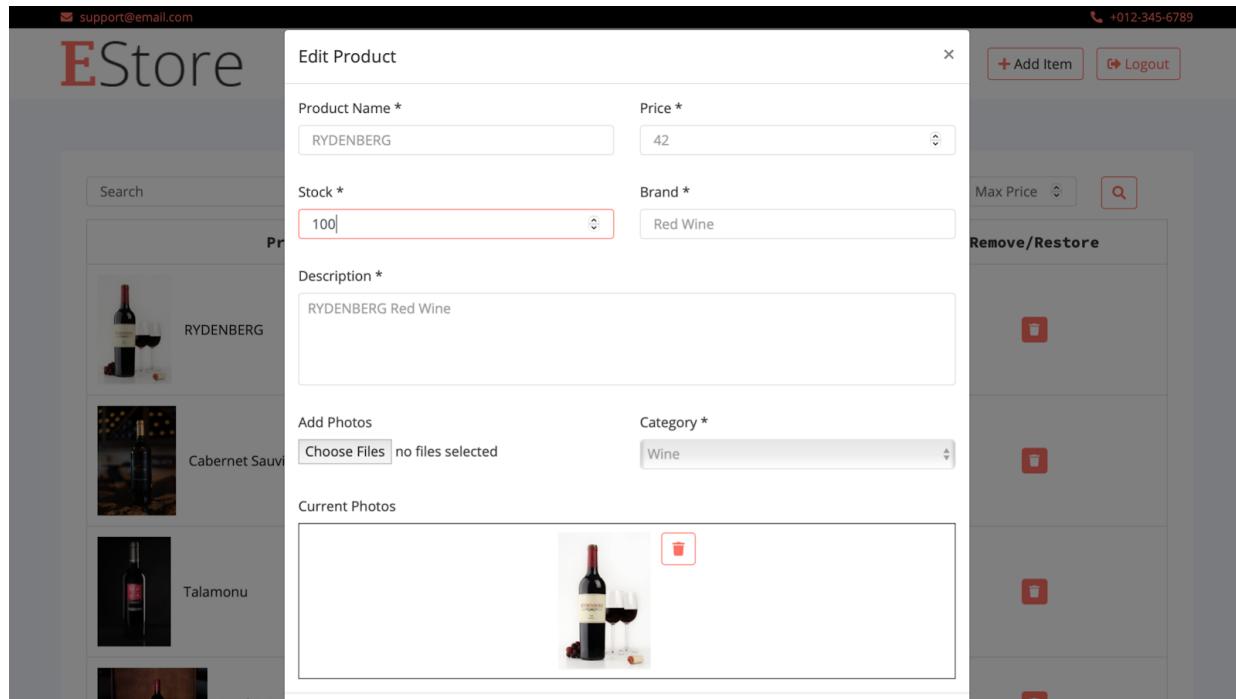
By logging as admin, an admin can view all the products in the inventory. He/She will be access to add/delete/update any product's image, description, quantity etc.

The screenshot shows the EStore Admin View. At the top, there is a header bar with a support email link, a phone number, and a search bar. Below the header is the EStore logo. To the right of the logo are buttons for 'Add Item' and 'Logout'. The main area displays a table of products:

Product	Price	Quantity	Edit	Remove/Restore
RYDENBERG	34	46		
Cabernet Sauvignon	56	60		
Talamonu	67	100		
Larch's Pater	120	75		

Add/Delete/Update Product Details

The Admin can change and modify all these fields, which are product name, price, stock, brand, description, add and delete photos, and change the category of the product as well. For images, PNG and JPEG formats are allowed.



The updates are instantly reflected on the website and database.

Product	Price	Quantity	Edit	Remove/Restore
 RYDENBERG	42	100		

```
_id: ObjectId("6084953f4fc85d619057f28e")
name: "RYDENBERG"
> images: Array
price: 42
description: "RYDENBERG Red Wine"
deleted: false
category: "wine"
stock: 100
brand: "Red Wine"
```

10. Soft Deletion of Products

Admin can soft delete a product by clicking on the trashcan icon that appears on the left side of each product. Once clicked, the button is greyed out confirming the deletion.

Product	Price	Quantity	Edit	Remove/Restore
 RYDENBERG	42	0		
 Cabernet Sauvignon	56	60		

The changes are reflected in the product_details collection as well. Here we see that the stock has been changed to 0 and the boolean field “deleted” has been changed to true.

```
_id: ObjectId("6084953f4fc85d619057f28e")
name: "RYDENBERG"
> images: Array
price: 42
description: "RYDENBERG Red Wine"
deleted: true
category: "wine"
stock: 0
brand: "Red Wine"
```

Work Division

All of the team members collaborated together in designing the flow of the website and database schema. Debugging and testing of the website with regular team discussions are conducted throughout the development process together.

Ashwani Kumar Kashyap -

- 1. User signup:**
 - Developed the signup page
 - Register a new user to the system in the mongo Database.
 - Check the new username or email address if it's already registered to the system using AJAX and maintaining the DB.
 - Form validation: Check if all mandatory fields are filled out. Check if the password is strong enough. Define the rules of having a strong password. User passwords are hashed and hashed versions of passwords are stored in the database.
- 2. User login and validations**
 - Developed the sign in page
 - Maintained user sessions
 - Checked for input validations
 - Raised appropriate errors in case of an invalid request.
- 3. User account page**
 - Developed the user account page.
 - Maintained the history of purchases for a user in the database with the details of each individual product.
 - Displayed the history of purchases in the frontend and order details of each product

Ruchi Singh -

- 1. List available products/services**
 - Added functionality to list available products in the system and view the product details about each product.
 - Implemented the backend part for the feature and also showed the products list at the front-end side.
 - Also taken care of the stored items in MongoDB.
- 2. Searching & Filtering with Pagination**
 - Users are able to filter the results based on category and price range. Users are also able to "Search" for a specific item. Implemented a global search as well as a local search with filtering on products.
 - Implemented the backend part for searching and filtering products by using the regex feature of querying any item in MongoDB. Also created the searchbar and all filtering sections at the front-end side.
 - Handled pagination at the backend and rendered the pages at front-end side as well by keeping track of total pages.
- 3. Adding items to the Cart**
 - Implemented the functionality of adding products to the cart. Users can add as many products to the cart. Also Users can see the available quantity and increase/decrease the quantity in the cart.
 - Implemented the backend part of the feature and stored the added items to the cart in the cart collection in MongoDB.
 - Implemented the feature (show added items to the cart) at the front-end side as well.

Soumil Mandal -

1. Implemented functionalities for the Admin side

- List All Items - Created the item listing view page and created buttons for each of the functionalities
- Add New Item - Created a template for adding a new item which consisted functionalities like adding name, brand, description, stock, and also uploading multiple images.
- Delete Item - To delete an item from the inventory. Here soft-delete is implemented on the backend and on deletion, the boolean flag delete is changed to True and the stock is changed to 0 in our database. This prevents the users from viewing this product.
- Update Item - To update product details, such as name, brand, description, images (option to add multiple images and delete them), stock, etc.

2. Inventory / Stock Management

- Stock Consistency - Whenever products are bought, the quantity is accordingly reduced from the stock in the product_details collection.
- Stock Handling - Whenever the quantity of a stock becomes 0, it is soft-deleted. Later, when the product again comes back to stock, the soft-deletion flag is again changed to False.
- Stock Checking - While checking out, the product quantities in the users cart is checked against the product quantity in the inventory to maintain consistency.

3. Implemented Checkout Functionality

- Address Validation - While checkout, shipping address is validated and for any invalid inputs, an informative error is shown to the user to correct it. By default, the user address stored in the database is shown.
- Order Summary - For viewing summary of all the individual orders a user has made.