**Shopping cart web development:**

To create a full stack web application by having Angular as the front end and back end Java with Spring boot and JPA. Backed by a mysql database with a schema name called shopping\_cart

Design planning was done by understanding the requirements completely.

Requirements:

Each product in database should have the following information:

* Name
* Price
* Domestic/Imported
* Category

Every item in the database should fit into one of the following categories:

* Books
* Food
* Medical
* Music
* Luxury Items
* Clothes

Rules:

1. Sales tax of 10% applies on all goods except Books, Food, Medical Supplies, because they are exempt
2. Import duty is an additional tax applied to all import items at a rate of 5%, no exemptions
3. The users should be able to search/filter goods based on name or product category
4. Users need to be able to add or remove items in various quantities to and from their shopping cart
5. When the user clicks "Purchase", they should receive a receipt which includes the names of their items and their prices, including tax. The receipt should also show the total cost of all items, and total amount of sales tax paid
6. Rounding rules: A tax of n% on a product that costs p dollars is calculated as n \* p / 100. This tax should be rounded up to the nearest 0.05 dollars, or 5 cents, for each item. For example, if a domestic CD cost $15.30, the tax would be 10%. 10 \* 15.30 / 100 = 1.53, which we round up to the nearest .05, so the tax for this item is $1.55

Development went in 2 phases.

1. Back end
2. Front end

Back end:

1. Created the table – Product
2. Created the controller for creating, updating (put & Patch), Deleting & reading a product
3. Following api routes are used for crud operations
4. **POST**: [http://localhost:8080/product](http://localhost:8081/product)

{

"name": "Charlie",

"price": 512.0,

"category": "Luxury Itesms",

"domesticOrImported": "domestic",

"quantity": 2,

"quantityToBuy": 0,

"imageUrl": "https://images.pexels.com/photos/2895148/pexels-photo-2895148.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"

}

1. **GET**: [http://localhost:8080/product](http://localhost:8081/product)
2. **PUT**: [http://localhost:8080/product](http://localhost:8081/product)

Request body: {

"id":12,

"name": "Charlie",

"price": 25.0,

"category": "Luxury Items",

"domesticOrImported": "domestic",

"quantity": 2,

"quantityToBuy": 0,

"imageUrl": "https://images.pexels.com/photos/2895148/pexels-photo-2895148.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"

}

1. **PATCH**: [http://localhost:8080/product/12](http://localhost:8081/product/12)

{

"price": 55.0

}

1. SQL file attached in the same folder which contains the insert statements for adding the items.
2. Created the repository and service layer for handling the crud operations
3. To generate a transaction / invoice – created the dto for transaction. It has the list of products that is from the shopping cart. Each product has a field called quantityToBuy which denotes the number of items that Customer wants.
4. Controller for Transaction was created and the below **post** operation can be used

**POST**: [**http://localhost:8080/purchase**](http://localhost:8081/purchase)

[

{

"id": 1,

"name": "Camera",

"price": 512.0,

"category": "Luxury Items",

"domesticOrImported": "domestic",

"quantity": 2,

"quantityToBuy": 1,

"imageUrl": "https://images.pexels.com/photos/2895148/pexels-photo-2895148.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"

},

{

"id": 2,

"name": "Gold bar",

"price": 50.5,

"category": "Luxury Items",

"domesticOrImported": "imported",

"quantity": 3,

"quantityToBuy": 1,

"imageUrl": "https://images.pexels.com/photos/1342251/pexels-photo-1342251.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"

},

{

"id": 3,

"name": "Fruit salads",

"price": 20.3,

"category": "Food",

"domesticOrImported": "domestic",

"quantity": 34,

"quantityToBuy": 1,

"imageUrl": "https://images.pexels.com/photos/2736370/pexels-photo-2736370.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"

},

{

"id": 4,

"name": "Tablets",

"price": 124.2,

"category": "Medical",

"domesticOrImported": "domestic",

"quantity": 12,

"quantityToBuy": 1,

"imageUrl": "https://images.pexels.com/photos/1389104/pexels-photo-1389104.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500"

}

]

1. All testing scenarios have been considered and written for controllers, service and dao’s

Front End:

1. Created the files for product, invoice class
2. Created the components – navbar, footer, admin, product list, product detail, front page, page not found, shopping-cart.
3. The components names are self explanatory
4. Created the service- product, shopping-cart
5. Product service handles adding, deleting, updating, getting the product(s) by contacting the server through api route
6. Shopping cart service adds a product, updates the quantity of a product that customer wants to buy, removes a product and also generates the invoice by contacting the server.
7. **Shopping cart service** also messages the **navbar** to display the **number** of the **cart** items
8. There is admin portion where products can be added, removed / edited in the Inventory
9. In Shopping cart, products can be removed , quantity can be updated .
10. Finally, after the products are added to the cart, purchase button can clicked to generate the transaction which lists the product price + tax(based on non-exempt category and imported tax), total sales tax and Grand total

**SQL Inserts:**

insert into product(category,domestic\_or\_imported,image\_url,name,price,quantity,quantity\_to\_buy)

values

("Luxury Items","domestic","https://images.pexels.com/photos/2895148/pexels-photo-2895148.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Camera",512,2,0),

("Luxury Items","imported","https://images.pexels.com/photos/1342251/pexels-photo-1342251.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Gold bar",50.5,3,0),

("Food","domestic","https://images.pexels.com/photos/2736370/pexels-photo-2736370.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Fruit salads",20.3,34,0),

("Medical","domestic","https://images.pexels.com/photos/1389104/pexels-photo-1389104.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Tablets",124.2,12,0),

("Food","imported","https://images.pexels.com/photos/257840/pexels-photo-257840.jpeg?auto=compress&cs=tinysrgb&dpr=3&h=750&w=1260","Second apple",1.1,12,0),

("food","imported","https://images.pexels.com/photos/2294137/pexels-photo-2294137.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Parfait",10.2,12,0),

("Food","domestic","https://images.pexels.com/photos/1071878/pexels-photo-1071878.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Pineapple",3.3,23,0),

("Food","imported","https://images.pexels.com/photos/2067488/pexels-photo-2067488.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Chocolate bar",0.85,10,0),

("Books","domestic","https://cdn.pixabay.com/photo/2017/09/28/15/12/book-2795850\_\_340.png","Sherlock Holmes",12.49,5,0),

("Music","domestic","https://cdn.pixabay.com/photo/2017/08/16/08/08/blogs-music-2646784\_\_340.png","Liv Concert",14.99,7,0),

("Luxury Items","domestic","https://images.pexels.com/photos/122400/pexels-photo-122400.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500","Camera",1000.5,2,0)