ADD PRODUCT TO SHOPING CART

Cart.js

```
static save(product) {
    if(cart) { //cart not null
        const existingproductIndex = cart.products.findIndex(p => p.id == product.id);
        if(existingproductIndex>0) {
            const existingProduct = cart.products[existingproductIndex];
            existingProduct.qty+= 1;
            cart.totalPrice+=product.price;
        } else { //no exist
            product.qty = 1;
            cart.products.push(product);
            cart.totalPrice = product.price;
        }
    } else { //cart null
        cart = {products: [], totalPrice: 0};
        product.qty = 1;
        cart.products.push(product);
        cart.totalPrice = product.price;
    }
}
```

Index.ejs

```
<form action="/addToCart" method="POST">
                        <input type="hidden" name="id" value="<%=p.id%>"/>
                        <button type="submit" class="btn btn-primary">Add to
cart</button>
                    </form>
```

GET SHOPING CART

Cart.ejs

```
<div class="col-sm-12">
    <% if(cart.products.length == 0) {%>
    <h2>Empty</h2>
    <%} else {%>
       <h2>Shopign cart detail</h2>
       <table class="table">
         <thead>
          <tr>
            <th scope="col">#</th>
            <th scope="col">Title</th>
            <th scope="col">Price</th>
            <th scope="col">Quantity</th>
            <th scope="col">Quantity</th>
          </tr>
         </thead>
         <tbody>
           <% for(let i = 0; i< cart.products.length; i++){
               let product = cart.products[i]%>
          <tr>
            <th scope="row"><%= i+1 %></th>
            <td><%= product.title%></td>
            <td><%= product.price%></td>
            <td><%= product.qty%></td>
          </tr>
          <%}%>
          <tr><td>Total price</td><td><%=
cart.totalPrice%></td></tr>
         </tbody>
       </table>
    <%}%>
    <br>
</div>
```

Cart.js

```
static save(product) {

    console.log('the product is ' +product);

    if(cart) { //cart not null

        const existingproductIndex = cart.products.findIndex(p => p.id == product.id); //try
check product is existing in cart


        if(existingproductIndex>0) {

            const existingProduct = cart.products[existingproductIndex];

            existingProduct.qty+= 1;

            cart.totalPrice+=product.price;

        } else { //no exist

            product.qty = 1;

            cart.products.push(product);

            cart.totalPrice += product.price;

        }

    } else { //cart null

        cart = {products: [], totalPrice: 0};

        product.qty = 1;

        cart.products.push(product);

        cart.totalPrice = product.price;

    }

}
```

Navigation.js

```
<li class="nav-item <%= path==='/cart' ? 'active' : "%>">

        <a class="nav-link" href="/cart">Cart</a>

    </li>
```

Shop.js (routes)

```
router.get('/cart', shopController.getCart)
```

Shop.js (controller)

```
exports.getCart = (req, res, next) => {

    res.render('cart', {cart: cart.getCart(), pageTitle: 'Shoping cart detail', path: '/cart', name:
'Jimmy'});

}
```

DELETE PRODUCTS FROM SHOPING CART

Cart.ejs

```
<tr>
                                        <th scope="row"><%= i+1 %></th>

                                        <td><%= product.title%></td>

                                        <td><%= product.price%></td>

                                        <td><%= product.qty%></td>

                                        <td>

                                          <form action="/delete-cart" method="POST">

                                            <input type="hidden" name="productid" value="<%=
product.id%>"/>

                                            <button type="submit" class="btn btn-
primary">Delete</button>


                                          </form>

                                        </td>

                                      </tr>
```

Cart.js

```
static delete(productid) {

    const products = cart.products;

    const isExisting = products.findIndex(p => p.id == productid);

    if(isExisting > 0) {

        products.splice(isExisting, 1);

    }

  }
```

Shop.js (routes)

router.post('/delete-cart', shopController.deleteInCart)

Shop.js (controller)

```
exports.deleteInCart = (req, res, next) => {
  cart.delete(req.body.productid);
  res.redirect('/cart');
}
```