

ShopEasy - Shopping Management System

- Harish Sivaprakash

Key Functionalities

- ▶ 1. Shopping cart for customers grouped by category and seller
- ▶ 2. Each shopping item will have image, description, name, stock quantity and price
- ▶ 3. Sign Up/ Sign In for customers
 - ▶ • Sign Up (Name, Email, Username, Password, Card Number(Optional))
 - ▶ • Sign In (Username, Password)
 - ▶ • Forget Password feature
 - ▶ • Password to be sent to the User Email Address, Password Recovery
 - ▶ • Password Show / Hide feature
 - ▶ • Username Validation (Check if username already exists in the DB, should be unique)

- ▶ 4. Product Information
- ▶ 5. Adding to Cart feature with quantity input from customer
- ▶ 6. Validation for Quantity input (display message if quantity input > stock quantity)
- ▶ 7. Cart View should have a table with these columns
 - ▶ • Product Name
 - ▶ • Model
 - ▶ • Price
 - ▶ • Quantity
- ▶ Total Bill in the cart and a Buy feature.
- ▶ 8. Search feature for the customers to search for products
- ▶ 9. On Buy from Cart View, Enter delivery address modal window
- ▶ 10. Account Information of a user
- ▶ 11. Sign Out Feature
- ▶ The application will be based on Spring MVC architecture and Hibernate.

- ▶ **Front End:** React JS based Views for the store.
- ▶ **Middle Tier:** Spring MVC Java Controller using annotations and Hibernate , Relational Mappings, Interceptors for logging purposes, Email confirmation for user
- ▶ **Data Tier:** Relational Database - MSSQL

Shop
Easy

Search products 🔍

Welcome Harish
Sivaprakash
226 Parker Hill Avenue
North Carolina

Logout

Orders

State tax : 6%
No. of items : 2
Sub Total : 3300
Tax : 198
Delivery : 66



Electronics

Laptop

Mobiles



FinePix Pro2 3D
Camera

\$ 1500

Available quantity: 100

− 1 +



EXP Portable Hard
Drive

\$ 800

Available quantity: 47

− 3 +



Luxury Ultra thin Wrist
Watch

\$ 300

Available quantity: 64

− 3 +



FinePix Pro2 3Fire TV
Stick 4K

\$ 50

Available quantity: 43

− 1 +

arish

1

Hill Avenue
lina



tra thin Wrist
atch

300

antity: 64

3



TO CART

Logout

Orders

State tax : 6%
No. of items : 2



EXP Portable Hard Drive

3 Nos.

\$ 800

\$ 2400



Luxury Ultra thin Wrist
Watch

3 Nos.

\$ 300

**\$
900**



PROCEED TO CHECKOUT

Shop
Easy



FinePix Pro2 3D
Camera

\$ 1500

Available quantity: 100

− 1 +

ADD TO CART

Checkout



FIRSTNAME Harish
LASTNAME Sivaprakash
HOMEADDRESS226 Parker Hill Avenue
EMAIL sivaprakash.h@husky.neu.edu
STATE North Carolina
State Tax 6
Sub Amount3300
Total Items 2
Tax 198
Delivery 66
Grand Total 3564
Credit Card

Close

Checkout

out

Orders



EXP Portable

\$ 800



Luxury Ultra t
Watch

\$ 300

PROCEED

Orders

| orderid | customerid | placeDate | totalPrice | creditcard | View Cart |
|---------|------------|-----------------------|------------|------------|----------------------|
| 3010 | 1025 | 2020-04-19 14:52:37.0 | 864 | 1788 | <div>View Cart</div> |
| 3012 | 1025 | 2020-04-19 16:54:59.0 | 1728 | 19876 | <div>View Cart</div> |

Cart Items

| cartitemid | customerid | quantity | code | orderid |
|------------|------------|----------|---------|---------|
| 3033 | 1025 | 1 | LPN4556 | 3010 |

Close

Cart Items

×

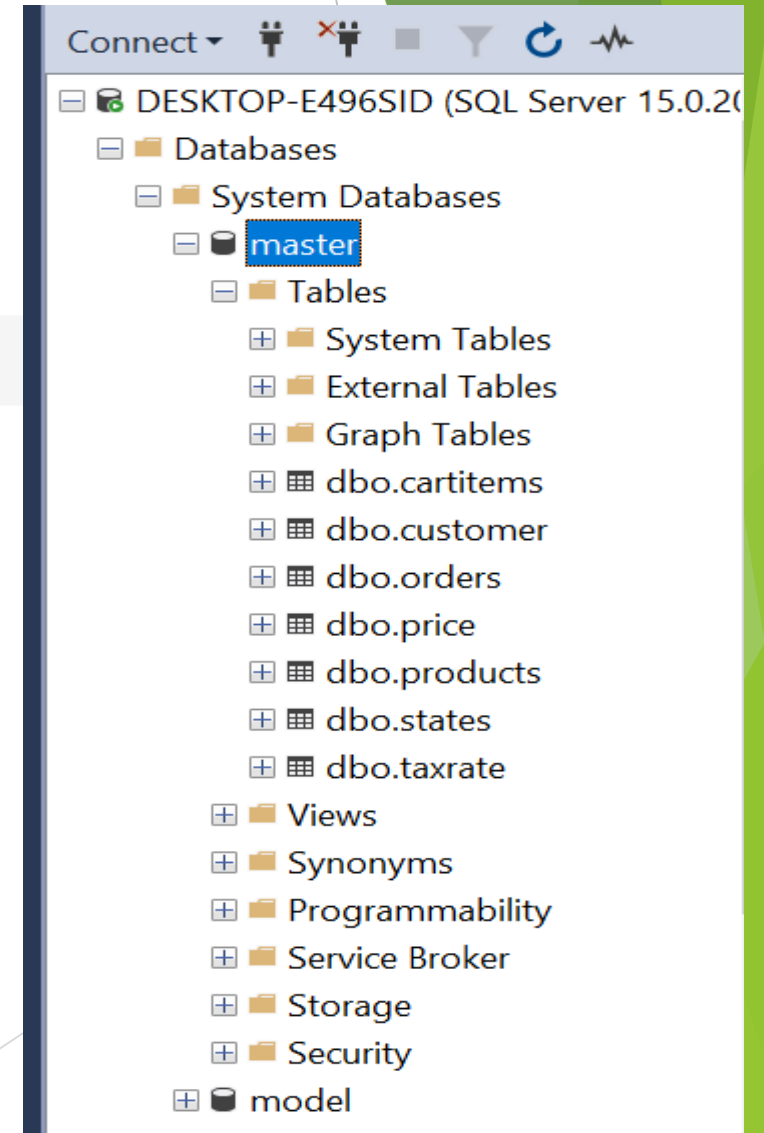
| cartitemid | customerid | quantity | code | orderid |
|------------|------------|----------|---------|---------|
| 3033 | 1025 | 1 | LPN4556 | 3010 |

Close

Backend APIs/Database

| ShoppingCart 17 requests | |
|-----------------------------|----------------------------|
| POST | Register customer |
| POST | Login customer |
| GET | List all products |
| GET | Get State and Abbreviation |
| POST | Add to cart |
| PUT | Update cart quantity |
| DEL | Delete cart item |
| GET | Get Cart Items |
| GET | Add product |
| PUT | Update Product |
| DEL | Delete product |
| POST | Place order |
| - | |

| | |
|------|----------------------|
| POST | Place order |
| GET | Get orders |
| GET | Confirm Email |
| GET | Get Customer by id |
| GET | Get State tax |
| GET | Get Order Cart Items |



Controller

```
package shoppingcart.controller;

@CrossOrigin(origins = "*", allowedHeaders = "*")
@RestController
@RequestMapping(value = "/cart")
public class CartController {

    @Autowired
    CartService cartService;

    @RequestMapping(value = "/add/{customerid}", method = RequestMethod.POST)
    public String addCart(@PathVariable int customerid, @RequestBody CartItem cartitem) {
        return cartService.addToCart(customerid, cartitem);
    }
}
```

```
@RequestMapping(value = "/getCartItems/{customerid}", method = RequestMethod.GET)
```

```
public List<CartItem> getCartItems(@PathVariable int customerid) {
```

```
    return cartService.getCartItems(customerid);
```

```
}
```

```
@RequestMapping(value = "/update/{customerid}", method = RequestMethod.PUT)
```

```
public void updateCart(@PathVariable int customerid, @RequestParam(value = "quantity")  
int quantity,
```

```
    @RequestParam(value = "code") String code) {
```

```
    cartService.updateCart(customerid, quantity, code);
```

```
}
```

```
@RequestMapping(value = "/delete/{customerid}", method = RequestMethod.DELETE)
```

```
public void deleteItemInCart(@PathVariable int customerid, @RequestParam(value = "code")  
String code) {
```

```
    cartService.deleteItemInCart(customerid, code);
```

```
}
```

```
}
```

```
@CrossOrigin(origins = "*", allowedHeaders = "*")
@RestController
@RequestMapping(value = "/order")
public class OrderController {
    @Autowired
    OrderService orderService;

    @RequestMapping(value = "/place/{customerid}", method = RequestMethod.POST)
    public String placeOrder(@PathVariable int customerid, @RequestBody Price price) {
        return orderService.placeOrder(customerid, price);
    }

    @RequestMapping(value = "/getOrders/{customerid}", method = RequestMethod.GET)
    public List<Order> GetOrders(@PathVariable int customerid) {
        return orderService.getOrders(customerid);
    }

    @RequestMapping(value = "/cartitems/{orderid}", method = RequestMethod.GET)
    public List<CartItem> GetCartItems(@PathVariable int orderid) {
        return orderService.getOrderCartItems(orderid);
    }
}
```

```
@CrossOrigin(origins = "*", allowedHeaders = "*")
@RestController
@RequestMapping(value = "/pricing")
public class PricingController {
    @Autowired
    PricingService pricingService;
    // gets all the states and corresponding abbreviations in US
    @RequestMapping(value = "/region", method = RequestMethod.GET)
    public HashMap<String, String> getStates() {
        HashMap<String, String> states = pricingService.getStates();
        return states;
    }

    // gets all the state taxes and corresponding abbreviations in US
    @RequestMapping(value = "/tax", method = RequestMethod.GET)
    public HashMap<String, Integer> getStateTax() {
        HashMap<String, Integer> tax = pricingService.getStateTax();
        return tax;
    }
}
```

```
@CrossOrigin(origins = "*", allowedHeaders = "*")
@RestController
@RequestMapping(value = "/product")
public class ProductController {
    @Autowired
    ProductService productService;

    @RequestMapping(value = "/list", method = RequestMethod.GET)
    public Object[] listAllProducts() {
        Object[] products = productService.listAllProducts();
        return products;
    }

    @RequestMapping(value = "/add", method = RequestMethod.POST)
    public void addProduct(@RequestBody Product product) {
        productService.addProduct(product);
    }
}
```

```
@RequestMapping(value = "/delete", method = RequestMethod.DELETE)
public void deleteProduct(@RequestParam(value = "code") String code) {
    productService.deleteProduct(code);
}
```

```
@RequestMapping(value = "/update", method = RequestMethod.PUT)
public void updateProduct(@RequestParam(value = "code") String code, @RequestBody
Product product) {
    productService.updateProduct(code, product);
}
}
```

```
@CrossOrigin(origins = "*", allowedHeaders = "*")
```

```
@RestController
```

```
@RequestMapping(value = "/user")
```

```
public class UserController {
```

```
    @Autowired
```

```
    UserService userService;
```

```
    @RequestMapping(value = "/registerCustomer", method = RequestMethod.POST)
```

```
    public String registerCustomer(@RequestBody Customer customer) {
```

```
        return userService.registerCustomer(customer);
```

```
    }
```

```
    @RequestMapping(value = "/loginCustomer", method = RequestMethod.POST)
```

```
    public ResponseEntity<Object[]> loginCustomer(@RequestBody Customer customer) {
```

```
        return userService.loginCustomer(customer);
```

```
    }
```



```
@RequestMapping(value = "/confirmEmail/{customerid}", method =  
RequestMethod.GET)
```

```
public String confirmEmail(@PathVariable int customerid) {  
    return userService.confirmEmail(customerid);  
}
```

```
@RequestMapping(value = "/getCustomer/{customerid}", method =  
RequestMethod.GET)
```

```
public ResponseEntity<Object[]> getCustomer(@PathVariable int customerid) {  
    return userService.getCustomer(customerid);  
}
```

```
}
```

Thank You!