controllers:

1. HomeController:this is the initial controller where the application starts 1.@RequestMapping(value = "/", method = RequestMethod.GET)

getHome(Model model) return home page

2.@RequestMapping(value = "/signUp", method = RequestMethod.GET) getSignUpPage(Model model)

return signup page

3.@RequestMapping(value = "/signInCreateAccount", method = RequestMethod.POST) createAccount(@Validated custCredModel ccm, Model model)

return to createdMsg file by creating the user 4.@RequestMapping(value = "/signIn", method = RequestMethod.GET)

getSignInPage(Model model) return signin page

5.@RequestMapping(value = "/signOk", method = RequestMethod.GET) getHomeFinal(@RequestParam("em") String email,

@RequestParam("ps") String psd, Model model,HttpSession session)

return to home page by validating the signed in user 6.@RequestMapping(value = "/profilePage")

sendProfilePage(Model model, HttpSession session) return to profile page of the logged in user

7.@RequestMapping(value = "/updateProfile", method = RequestMethod.POST) userupdate(@ModelAttribute("Customer") custCredModel cust,

Model model, HttpSession session)

return to profile with updation made

8. @RequestMapping(value = "/logout")

userlogout(Model model, HttpSession session) return home page by logging out the logged in user

1. ForgotPasswordController

1.@RequestMapping(value = "/forgotPassword", method = RequestMethod.GET) forgotPswd(Model model)

return forgotPage 2.@PostMapping("/otpAction")

sendOTP(@RequestParam("email") String email) return the otp generated back to ajax call

3.@PostMapping("/validateOTP") validateOTP(@RequestParam("otp12") String otp) return the string by validating the otp

4.@RequestMapping(value = "/updatepwd") usersignup(@RequestParam("psd2") String p2, Model model) return signin page by updating the user password

1. ProductController
   1. @GetMapping("/retrieveCategories") displayCategories(Model model)

return the list of categories to product listing page

* 1. @GetMapping("/getProducts")

showProducts(@RequestParam(value = "category", required = false) String category, Model model)

Return the list of products available according to the category selected to home page

// returns the list of products to product listing page

* 1. @RequestMapping(value = "/allProducts", method = RequestMethod.get) showAllProducts(Model model)

Return the list of all products to home page 4)CartController

* + 1. @RequestMapping(value = "/addToCart", method = RequestMethod.post) addToCart(@RequestParam(value = "productId", required = true)

int productId, Model model)

return to product listing page by adding product to home page and displaying the remove from cart option

* + 1. @GetMapping("/cartItems")

userCartItems(@RequestParam(value = "userId", required = true) int cust\_id, Model model)

returns the list of products to home view that are in the cart

* + 1. @GetMapping("/removeFromCart") removeFromCart(@RequestParam(value = "productId", required = true)

int productId, Model model) deletes the product from the cart

1. WishlistController
   1. @RequestMapping(value = "/addToWishlist", method = RequestMethod.post) addToWishlist(@RequestParam(value = "productId", required = true)

int productId, Model model)

return to product listing page by adding product to wishlist by displaying the remove from wishlist option

* 1. @GetMapping("/wishlistItems")

userWishlistItems(@RequestParam(value = "userId", required = true)

int cust\_id, Model model)

returns the list of products to wishlist view that are in the wishlist

* 1. @GetMapping("/removeFromWishlist") removeFromWishlist(@RequestParam(value = "productId", required = true)

int productId, Model model) deletes the product from the wishlist

1. CustomerController
   1. @RequestMapping(value = "/buy", method = RequestMethod.get) processBuy(Model model)

return to checkout page

* 1. @GetMapping("/processPayment") payment(Model model)

return createInvoice

3.@RequestMapping(value = "/createInvoice", method = RequestMethod.post) createInvoice(orderid,customerid)

return invoice view

4.@RequestMapping(value = "/addToEcash", method = RequestMethod.get) addToEcash(customerid)

add ecash to customer ecash

5.@RequestMapping(value = "/useEcash", method = RequestMethod.get) deductEcash(customerid)

deducts ecash from customer ecash 7)OrderController

1.@RequestMapping(value = "/ordersList", method = RequestMethod.post)

displayUserOrders(@RequestParam(value = "userId", required = true) int cust\_id, Model model)

return the list of orders to the user order history view 2.@RequestMapping(value = "/cancelOrder", method = RequestMethod.post)

cancelOrder(custid,orderid)

return to order view by canceling the order 3.@RequestMapping(value = "/invoice", method = RequestMethod.post)

showInvoice(@RequestParam(value = "orderid", required = true) int order\_id, Model model)

return invoice view

8.AdminController

1.@RequestMapping(value = "/admin", method = RequestMethod.GET) admin(Model model)

return admin adminLogin view

2.@RequestMapping(value = "/adminSignOk", method = RequestMethod.GET) getHomeFinal(@RequestParam("em") String email,

@RequestParam("ps") String psd, Model model,HttpSession session)

return to dashboard page by validating the signed in admin 3.@RequestMapping(value = "/adminShipments",

method = RequestMethod.GET) adminShipments(Model model)

return all shipments done through ajax call 4.@RequestMapping(value = "/adminPayments",

method = RequestMethod.GET)

adminPayments(Model model)

return all payments done through ajax call 5.@RequestMapping(value = "/adminStock",

method = RequestMethod.GET) adminStocks(Model model)

return all available stock details through ajax call 6.@RequestMapping(value = "/productStock",

method = RequestMethod.post) productStock(product,Model model)

return all avilable stock of particular product through ajax call 7.@RequestMapping(value = "/adminReport",

method = RequestMethod.GET) showReports(Model model)

return the options list to display the report through ajax call 8.@RequestMapping(value = "/selectionReport",

method = RequestMethod.GET) showselectedReport(string,Model model)

return the report based on the selected option through ajax call