INTRODUCTION

ShopForHome is a popular Store in the market for shopping the home décor stuff. Due to Covid 19 all the offline shopping stopped. So, the store wants to move to the online platforms and wants their own web application.

So we made a website for the store using different technologies such as angular , springboot , TypeScript , mySQL to help them move their business to online platform

INTRODUCTION

The store wants its website to have 2 type of users namely:

- Admin
- User

As an admin I should be able to:

- 1. Login, Logout and Register into the application.
- 2. As an Admin I should be able to perform CRUD on Users.
- 3. As an Admin I should be able to Perform CRUD on the products.
- 4. As an Admin I should be able to get bulk upload option to upload a csv for products details
- 5. As an Admin I should be able to get the stocks.
- 6. As an Admin I should be able to mail if any stock is less than 10.
- 7. As an Admin I should be able to get the sales report of a specific duration.
- 8. As an Admin I should be able to set the discount coupons for the specific set of users

INTRODUCTION

As an I should be able to:

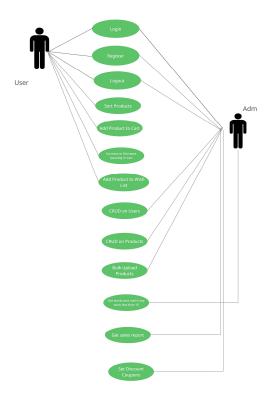
- 1. login, Logout and Register into the application.
- 2. As a user I should be able to see the products in different categories.
- 3. As a user I should be able to sort the products.
- 4. As a user I should be able to add the products into the shopping cart.
- 5. As a user I should be able to increase or decrease the quantity added in the cart.
- 6. As a user I should be able to add "n" number of products in the cart.
- 7. As a user I should be able to get the Wishlist option where I can add those products which I want but don't want to order now
- 8. As a user I should get different discount coupons.

TECHNOLOGIES USED

To make the website for ShopForHome store we used the following technologies:

- HTML
- CSS
- BOOTSTRAP
- TYPESCRIPT
- JAVA SPRINGBOOT
- MYSQL
- ANGULAR
- TOMCAT

USECASE DIAGRAM



CLASS DIAGRAM FOR SERVICE

<<Java Class>>

⊕CartServiceImp

com.capstone.shopforhome.service

a cartRepo: CartRepository

- CartServiceImp()
- @ listCartItems(User):List<CartItem>
- addProductToCart(Product, User, int):Integer
- ø deleteByUserAndProduct(User,Product):Integer
- @ deleteByld(Long):void
- @ getCartCount():Integer

<<Java Class>> **⊕**CSVService

com.capstone.shopforhome.service

△ repo: ProductRepository

CSVService()

save(MultipartFile):void

<<Java Class>>

@EmailServiceImp

com.capstone.shopforhome.service

- △ javaMailSender: JavaMailSender △ repo: ProductRepository
- sendEmail(Long):String

<<Java Class>>

⊕OrderServiceImpl

com.capstone.shopforhome.service

- repo: OrderRepo
- productRepo; ProductRepository
- productService: IProductService
- o OrderServiceImpl()
- addOrder(ArrayList<CartItem>): String
- getOrder():List<Orders>
- getOrdersByProduct(Product):List<Orders>
- getByDate(String,String):List<Orders>
- getByDateAndProduct(String,String,Long):List<Orders>

<<Java Class>> **⊕**UserServiceImp

com.capstone.shopforhome.service

- △ userRepository: UserRepository
- o isLoggedin: boolean
- *UserServiceImp()
- registerUser(UserDTO):String
- o loginUser(LoginForm):User
- logoutUser(String):String
- @ getAllUsers():List<User>
- updateUser(User):User removeUser(Long):void
- getUserByld(long):User
- o getLoggedinStatus():boolean

<<Java Class>>

ProductServiceImp

com.capstone.shopforhome.service

repo: ProductRepository

ProductServiceImp()

- addProduct(ProductDTO):Product
- deleteProduct(Long):void
- updateProduct(ProductDTO,Long):Product
- getProduct():List<Product>
- getProductById(Long):Product
- getAllIds():List<Long>

<<Java Class>>

⊕ WishlistServiceImp

com.capstone.shopforhome.service

wishlistRepo: WishlistRepository

WishlistServiceImp()

- listWishlistItems(User):List<Wishlist>
- addProductToWishlist(Product,User):Integer
- ø deleteByUserAndProduct(User,Product):Integer
- getWishlistCount():Integer

<<Java Class>>

@CouponServiceImpl

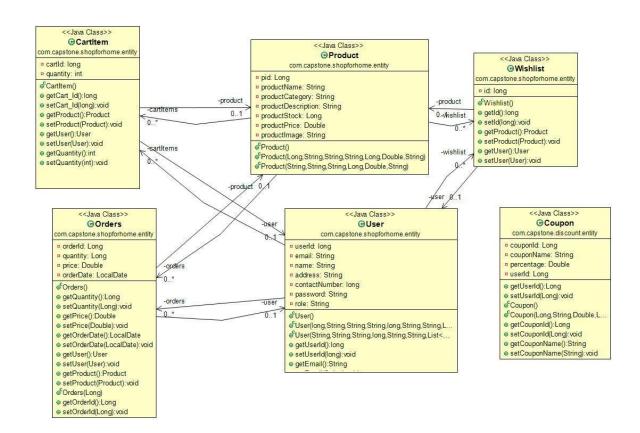
com.capstone.discount.service

a repo: CouponRepo

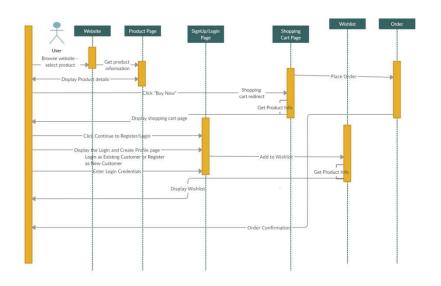
CouponServiceImpl()

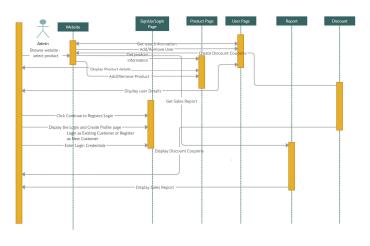
- addCoupon(CouponDTO):Coupon
- ø deleteCoupon(Long):void
- updateCoupon(CouponDTO,Long):Coupon
- o getCoupon():List<Coupon>
- getByuser(Long):Coupon
- ø deleteByuser(Long):void

CLASS DIAGRAM FOR ENTITY

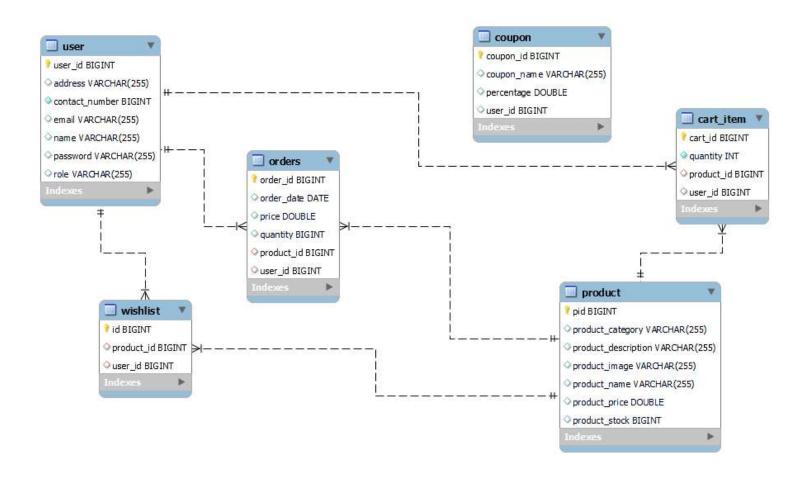


SEQUENCE DIAGRAM

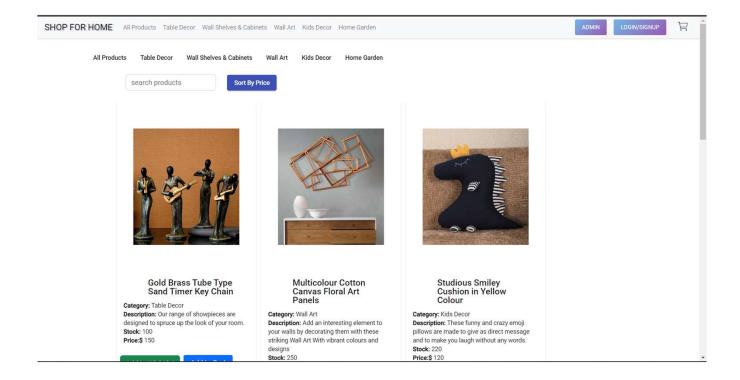




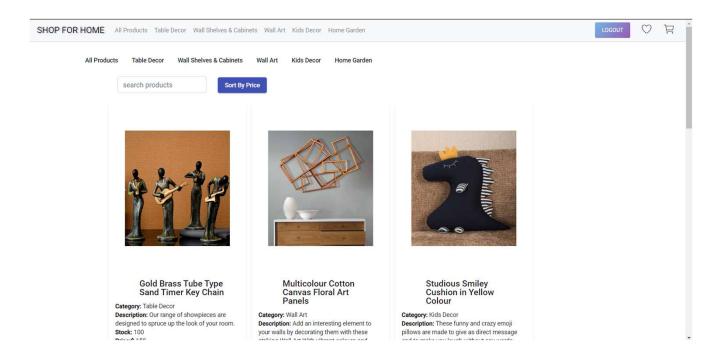
ER DIAGRAM



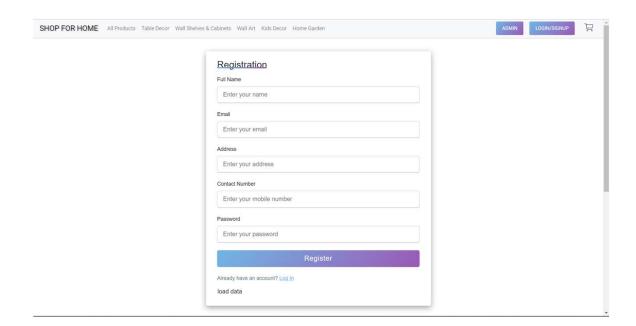
HOME PAGE



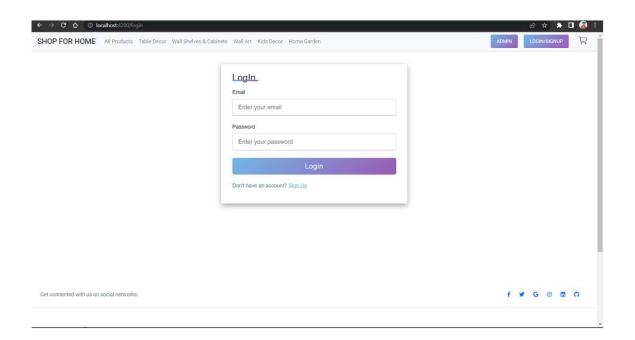
USER DASHBOARD



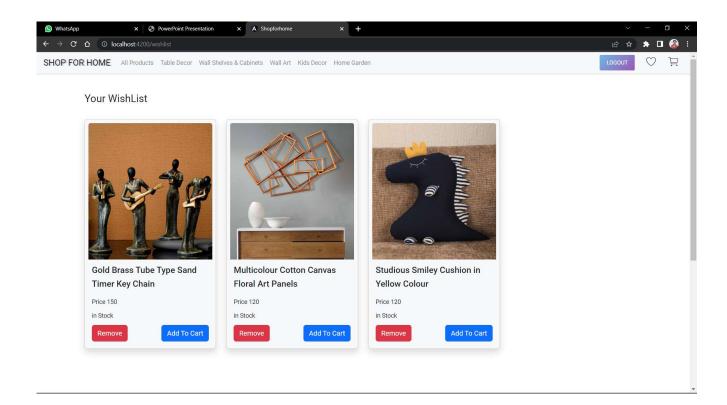
REGISTRATION PAGE



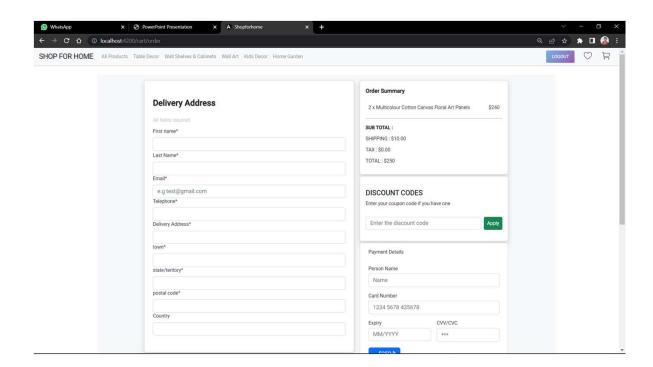
LOGIN PAGE



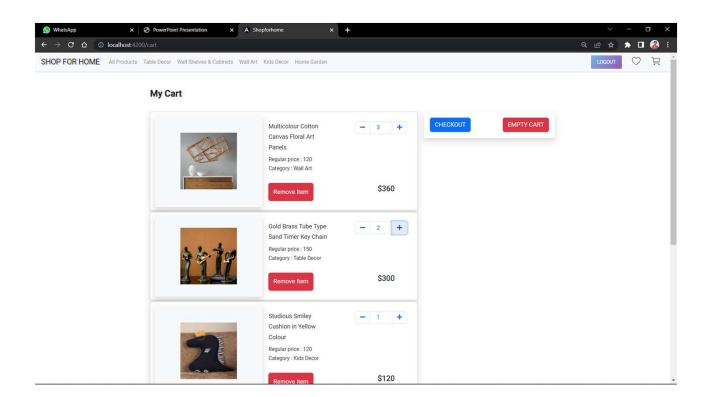
WISHLIST



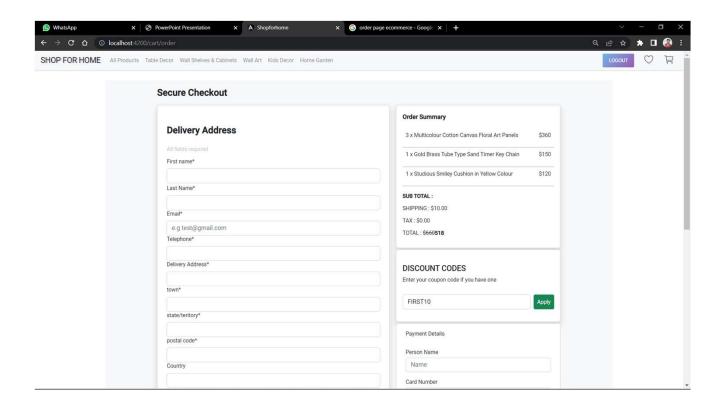
CHECKOUT PAGE



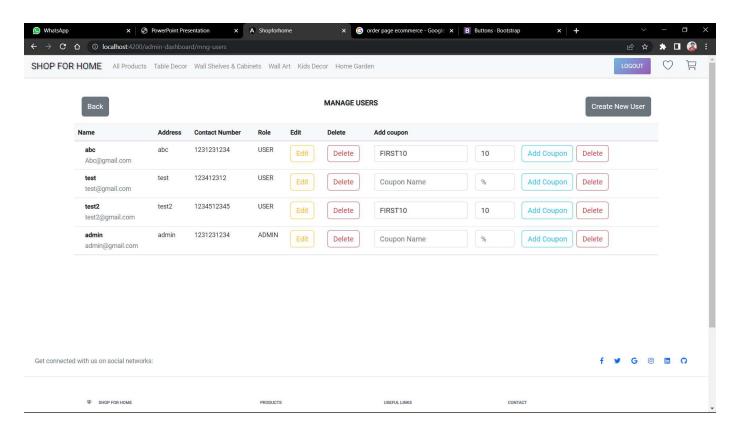
CART



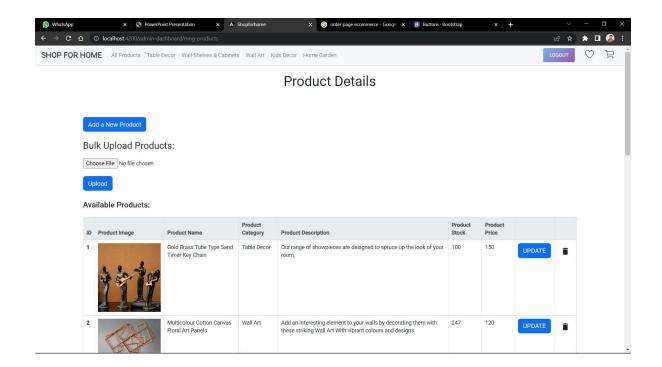
DISCOUNT



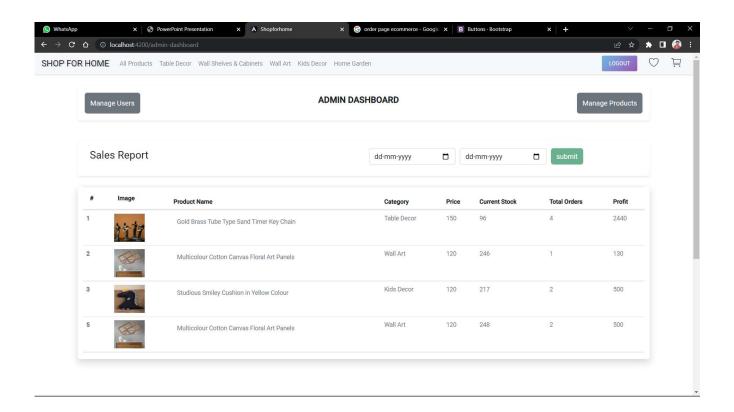
ADMIN DASHBOARD



BULK UPLOAD



SALES REPORT



THANK YOU!