Technologies used:

- ReactJS
- jQuery
- Java
- JDBC
- Java Servlets
- Spring & Spring Boot
- Hibernate
- JPA
- JavaScript
- SQL
- jUnit

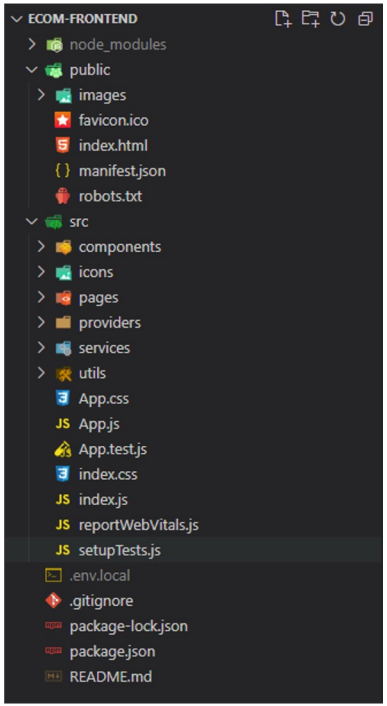
Project Overview:

Design and develop an e-commerce application with following criteria:

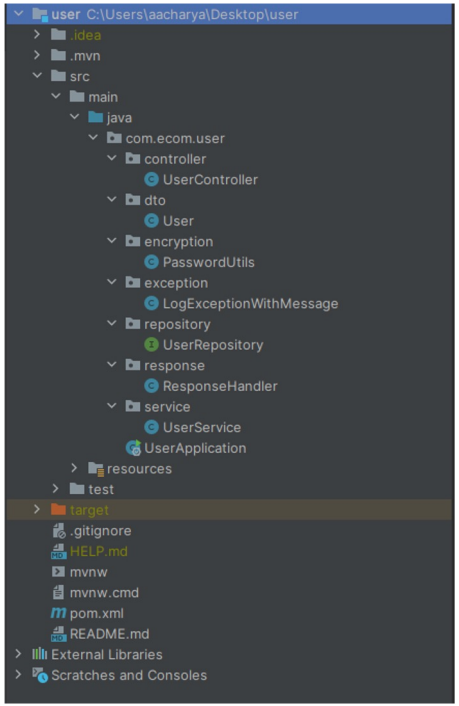
- ReactJS as front-end
- Implement components, pages, providers, services, utils
- Use axios for API calls.
- Backend developed using microservice architecture using Java, Spring Boot, Hibernate JPA, MySQL, MongoDB.
- Testing for application using jUnit

Project Structure:

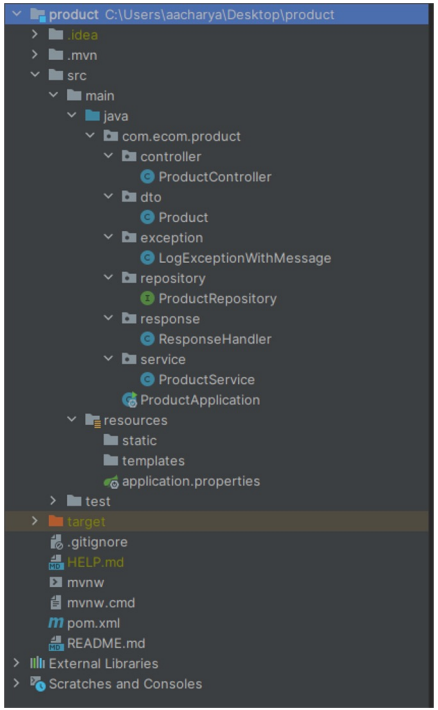
Front-end



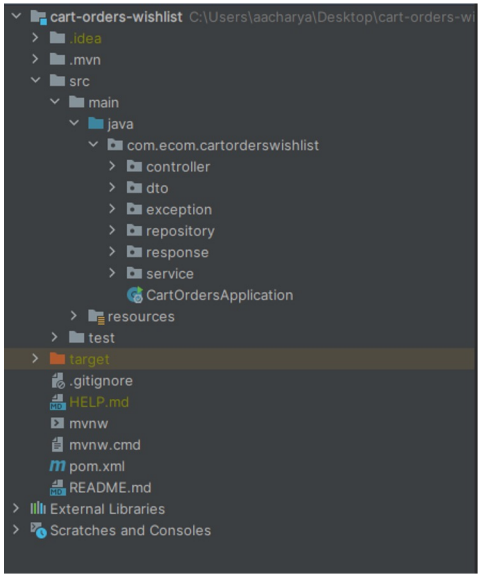
User Microservice



Product Microservice



Cart-order-wishlist Microservice



Project Architecture:

User microservice

User Microservice
Endpoints:

- **GET:**
  /user/all - Returns all the users
  /user/{userName} - Return user by userName
- **POST:**
  /user - Add a user
  /user/verify-password/{userName} - Verify the password for the given userName
- **PUT:**
  /user/{userName} - Update user details
- **DELETE:**
  /user/{userName} - Delete a user by userName

Project Architecture:

Product microservice

Product Microservice
Endpoints:

- **GET:**
  /product/all - Returns all the products
  /product/{sku} - Return product by SKU
  /product?sortBy={sortByField} - Return products sorted by field
  /product/product-suggestions/{sku} - Return products suggestions based on SKU
  /product/category/{category} - Return products for this category
- **POST:**
  /product - Add a product
- **PUT:**
  /product/{sku} - Update product details
- **DELETE:**
  /product/{sku} - Delete a product by SKU

Project Architecture:

Cart-orders microservice

Cart-Orders Microservice
Endpoints:

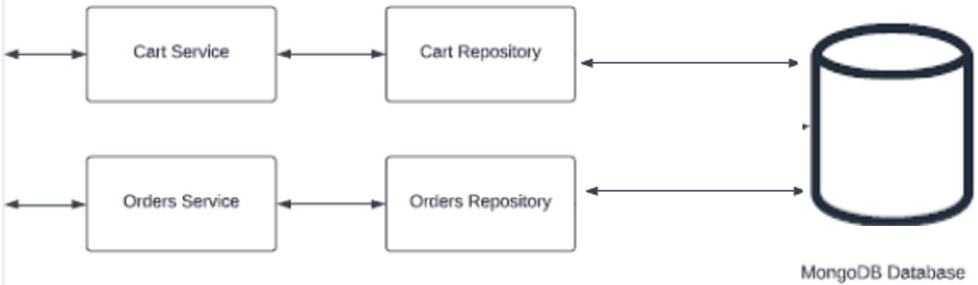
- **GET:**
  /cart/{userName} - Return cart details for user by userName

  /orders/{userName} - Return orders for userName
  /orders/sort-by-time/{userName}/{ascOrDesc} - Return orders sorted by time in ascending or descending order for userName

- **POST:**
  /cart/initialize-cart/{userName} - Initialize cart details for userName
  /cart/add?userName={userName}&sku={productSku}&unitPrice={unitPrice} - Add product details to cart for userName
  /cart/remove?userName={userName}&sku={productSku} - Remove product details from cart for userName

  /orders/{userName} - Add order for userName

- **DELETE:**
  /cart/place-order/{userName} - Place order and empty cart for userName



Project Architecture:

Wishlist microservice

Wishlist Microservice
Endpoints:

- **GET:**
  /wishlist/{userName} - Return wishlist for user by userName
- **POST:**
  /wishlist/add/{userName}/{productSku} - Add productSku to userNames's wishlist
  /wishlist/remove/{userName}/{productSku} - Remove productSku from userNames's wishlist

Features of E-commerce Application (Watch World):

Admin:

- Can view all products, edit product or delete product
- Can view all users or delete any user
- Can view all subscriptions for newsletter as well as remove a subscription

User:

- Can create profile to browse, wishlist or order products
- Update profile details as well as password

Future Scope:

In future following features can be implemented:

- User authentication security can be improved using tokens to be stored in front-end as well as backend
- The application can be deployed to cloud
- The front-end components can be divided more into smaller levels

Conclusion:

As part of the team, I worked on Jira tickets during sprints which covered both frontend as well as backend aspects of web development.

- Worked on writing JUnit test cases to cover classes at service and controller layers. Increased the coverage to 95% method coverage and 80% method coverage respectively
- Worked on HTTP status code and Not Found page to be returned for not available resources.
- Worked on frontend analysis for app pages.

The technologies and tech stack I worked with included Java, Spring Boot, JUnit test cases, ReactJs, Restful APIs, etc.

References:

Fastenal

React – A JavaScript library for building user interfaces (reactjs.org)

Spring | Home

Spring Data JPA

Hibernate Example using JPA and MySQL – GeeksforGeeks

Log4j2 with XML Configuration - Java Example – Studytonight

Stack Overflow - Where Developers Learn, Share, & Build Careers

Java Tutorial | Learn Java Programming – javatpoint

https://youtu.be/w7ejDZ8SWv8

https://youtu.be/OuBUUkQfBYM

https://youtu.be/35EQXmHKZY

Thank you!