**Subscriptions based E-Commerce Web application**

**Project Scope**

This project is a web application that represents subscription-based e-commerce application.

It should allow customers to order and receive food on a recurring basis.

Customers can sign up for various subscription plans.to receive food items at a discounted rate compared to one-time purchases.

The system offers wide range of food products, including organic fruits, vegetables, snacks, ready-to-eat meals, and more.

It utilizes the replenishment (or "auto-ship") subscription model, which allows customers to receive essential food items and goods regularly.

This model saves customers the trouble of making a purchase every time they need fresh food.

The subscription service ensures consistent delivery of products, with a flexible schedule that meets the subscribers' needs, whether weekly, bi-weekly, or monthly.

**Description**

Customers can browse and subscribe to various food products.

Subscription options include weekly, bi-weekly, monthly, and customizable plans with

specific start and end dates.

Admin defines subscription types.

Admin manages products including adding, updating, and deleting products.

Admin should be able to deactivate and activate subscriptions.

Each product may have different subscription options, and admin can define these options.

Admin can view all active and inactive subscriptions, order history, and an order dashboard.

Customers can place orders to subscribe to products

Admins can view daily delivery lists based on active subscriptions.

Whenever required customer should be able to cancel subscriptions.

**User Interface Design**

The application should have following screens/pages to interact with various users of the application:

1. Application should have single entrypoint with home page which contains navigation to following actions:

**1) User registration:** This page should allow user to register with the system with required fields. Apply validation for all the fields. Admin needs not to register.

**2) Login:** This page should allow user to login to the system with valid credentials.

2. Main Page (**accessible to Admin**): This page should be displayed to admin after login. It should have links to perform all actions that are associated with admin.

3. Main Page (**accessible to Customer**): This page should be displayed to customer after login. It should have links to perform all actions that are associated with customer.

**Test Cases**

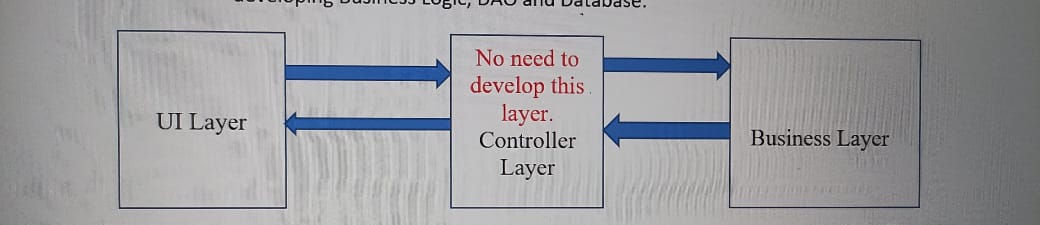
Write Junit test cases for all scenarios present in the application.

**Guidelines**

1. Divide your team into two sub teams.

**UI Team** - The Ul Team will be responsible to design and develop the User Interface with the help of Ul Technologies like HTML, CSS, Java Script. All required pages develop with the dummy data.

**Back End Team**-The Back End Team will be responsible for designing and developing Business Logic, DAO and Database.



2. Use layered architecture with loose coupling.

3. Validate all user inputs with proper error handling.

4. The UI should use a layout with a header, footer and sidebar (with navigation links), which should be maintained on all pages.

5. Optional: Explore the various CSS frameworks for designing the UI, The Ul should look elegant.