D'Baesics Whole Software Development Lifecycle (SDLC)

1. Requirements Gathering

Business Requirement:
- Have a website to market and showcase D'Baesics products
- be able to handle orders and shipping
- Integrate a payment system
- manage inventory
- user interaction

Key Features:
- Login/Signup of a user.
- Add to cart
- Shopping Cart
- Checkout and payment (Stripe Integration)
- Inventory Management (Admin dashboard)

User Stories (For the User):

Story	As a shopper, I need to be able to add items to my shopping cart so I can purchase them.
Acceptance Criteria	Select item, click button to add item to cart. Item is in cart.
Size	**SMALL**

Story	As a shopper, I need to be able to view the items in my shopping cart so I can decide if I want to purchase them.
Acceptance Criteria	Select item, add to cart, select link to "view cart" and item is present.

Size	**MEDIUM**

Story	As a shopper, I want to be able to update the quantity of items in my shopping cart so I can adjust my purchase quantity.
Acceptance Criteria	In the cart view, I can change the quantity of a selected item, and the total price is updated accordingly.
Size	**SMALL**

Story	As a shopper, I need the ability to remove items from my shopping cart in case I change my mind about a purchase.
Acceptance Criteria	In the cart view, I can click a button to remove a selected item, and the cart is updated without that item.
Size	**SMALL**

Story	As a shopper, I want to see the total cost of the items in my shopping cart, including taxes and any applied discounts.
Acceptance Criteria	The cart view displays the total cost with a detailed breakdown of individual item costs, taxes, and discounts.

Size	**MEDIUM**

Story	As a shopper, I need the option to apply a promotional code or voucher to my shopping cart to avail discounts.
Acceptance Criteria	During the checkout process, there is a field to enter a promotional code, and applying it reflects the discounted price.
Size	**MEDIUM**

Story	As a shopper, I want to proceed to the checkout page with a single click to save time, using Stripe API.
Acceptance Criteria	A prominent and easily accessible "Checkout" button is available in the cart view, leading directly to the checkout page.
Size	**SMALL**

Story	As a shopper, I need to enter my shipping address during checkout to ensure accurate delivery.
Acceptance Criteria	The checkout process includes a step for entering and validating the shipping address.

Size	MEDIUM

Story	As a shopper, I want to select my preferred payment method during checkout for a seamless payment experience.
Acceptance Criteria	The checkout process includes a step for choosing a payment method, such as credit card or PayPal.
Size	MEDIUM

Story	As a shopper, I want to receive an order confirmation email after completing my purchase.
Acceptance Criteria	An email is sent to the shopper's registered email address confirming the order details.
Size	SMALL

Story	As a shopper, I need the option to save multiple shipping addresses for future orders.
Acceptance Criteria	The user profile includes a feature to add and manage multiple shipping addresses.

Size	**MEDIUM**

Story	As a shopper, I want to see real-time shipping updates and tracking information for my order.
Acceptance Criteria	After completing the purchase, the shopper can track the order status and receive timely updates on shipping progress.
Size	**LARGE**

Story	As a business owner, I want to see real-time shipping updates and tracking information for my order.
Acceptance Criteria	After completing the purchase, the shopper can track the order status and receive timely updates on shipping progress.
Size	**LARGE**

Managing Inventory:

Story	As a business owner, I want to efficiently manage the inventory of products in my store.
Acceptance Criteria	1. I can add new products to the inventory, specifying details such as name, category, price, and quantity. 2. I can update product information, including price and quantity available. 3. I can easily mark products as out of stock or remove them from the inventory. 4. The system should provide real-time updates on the available quantity of each product.
Size	**MEDIUM**

Sales Report:

Story	As a business owner, I want to generate comprehensive sales reports to track the performance of my store.
Acceptance Criteria	1. I can generate a sales report for a specific time period, showing total revenue, number of orders, and average order value.

	2. The report should break down sales by product, indicating the best-selling items. 3. The system should allow me to export the sales report in a downloadable format (e.g., CSV, PDF).
Size	**LARGE**

Customer Management:

Story	As a business owner, I want to manage customer information and track their purchasing history.
Acceptance Criteria	1. I can view a list of registered customers, including their names and contact details. 2. For each customer, I can see a history of their past orders, including dates and purchased items. 3. I can add notes or tags to individual customer profiles for personalized service.
Size	**MEDIUM**

Discount Management:

Story	As a business owner, I want the ability to create and manage discounts for promotional purposes.
Acceptance Criteria	1. I can create discount codes with specified conditions (e.g., percentage off, fixed amount). 2. The system should apply discounts automatically during checkout when a valid code is entered. 3. I can set expiration dates for discount codes. 4. I can track the usage and effectiveness of each discount code.
Size	**MEDIUM**

Order Tracking:

Story	As a business owner, I want to easily track and manage the status of customer orders.
Acceptance Criteria	1. The system should provide a dashboard displaying the current status of all orders. 2. I can filter orders by status (e.g., pending, shipped, delivered). 3. I receive timely notifications for critical order events (e.g., new order, order shipped).

Size	SMALL

2. Design:

System Architecture:

Backend (Java Spring Boot):
1. Controllers:
   - Create controllers to handle different functionalities (e.g., product management, cart, order processing).
   - Use annotations like @RestController for API endpoints.

2. Service Layer:
   - Implement services to encapsulate business logic.
   - Use @Service annotation for service classes.

3. Data Access Layer:
   - Use Spring Data JPA for database access.
   - Define repositories with @Repository annotation.

4. Models:
   - Create Java classes for entities like Product, User, Cart, Order, and Inventory.
   - Annotate classes with @Entity and define relationships.

5. Security:
   - Implement security features using Spring Security if user authentication and authorization are required.

Frontend (Thymeleaf with HTML/CSS/Bootstrap):
   - Have a Front-end for the end-users (HTML/CSS)
   - Have a front-end for the business owner (HTML/CSS)

Database Design:
Entity Relationship Diagram (ERD):
1. Product:

Fields: ID, Name, Description, Price, Quantity, Category, Image URL.
Relationships: One-to-Many with Inventory.

2.User:
Fields: ID, Username, Password (hashed), Email.
Relationships: One-to-One with Cart, One-to-Many with Order.

3.Cart:
Fields: ID, User ID.
Relationships: One-to-One with User, One-to-Many with CartItem.

4.Order:
Fields: ID, User ID, Order Date, Status (e.g., Pending, Shipped).
Relationships: Many-to-One with User, One-to-Many with OrderItem.

5.CartItem:
Fields: ID, Product ID, Quantity.
Relationships: Many-to-One with Cart, Many-to-One with Product.

6.Inventory:
Fields: ID, Product ID, Purchase Cost, Sale Price.
Relationships: Many-to-One with Product.

3. Development:

Milestones:

January 15, 2024:
- Able to navigate to the needed pages.
- Product table is created for the design of the products. However, there is no data yet. That can be done in the next sprint.

4. Testing:
5 Deployment:
6. Maintenance: