

TechTrek2022 Challenge Statement

30th September 2021

Introduction

DBS's commitment, 'Live more, Bank less', promises to enable you to live hassle-free with invisible banking, seize life's opportunities and live larger than yourselves. With this, DBS has evolved into more than just a bank with the launch of DBS digital marketplace, revamping the way we think about banking.

Challenge Statement

Your task is to create a simplified full-stack e-commerce site that could be part of the DBS marketplace. This **must** be a **web application**.

*For every Frontend task, there will be a matching Backend task as indicated by the numbering. E.g.

Basic Requirements (Frontend)	Basic Requirements (Backend)
User must be able to login [1]	Server must be able to authenticate a user's identity [1]

Basic Application Requirements (Frontend):

- You must render a login page
 - User must be able to login [1]
- You must render a product page
 - User must be able to view all products [2]
 - User must be able to add products and its quantity to their cart [3]
- You must render a shopping cart component
 - User must be able to view products added to their cart (You are to display the product's name and quantity) [4]
 - User must be able to remove products from their cart [5]
 - Users must be able to checkout their cart. (Validation of sufficient balance is not required to purchase the items in the cart. You may add it in as an advance feature.) [6]

Basic Application Requirements (Backend):

- You must set up a valid authentication API
 - Server must be able to authenticate a user's identity [1]
- You must have an API to return a list of all products from **Products** and **Category** table [2]
- You must set up the respective shopping cart API with the following functionalities:
 - Insert products added from frontend cart into database [3]
 - Return a list of all products from the **OrderItem** table [4]
 - Delete from the **OrderItem** table [5]
 - Update the **Orders**, **Products**, and **OrderItem** table [6]

Advanced Requirements (Optional, in any order, but good to have):

- Data visualization through charts and graph
- Applying coupon codes
- Features that enhance the customer journey

Optional Technical Depth (showcase your skill set):

- AWS/GCP to deploy your application to the cloud

Data provided

You will be given:

- Data in JSON and SQL format
 - The entries provided are not exhaustive and you can add more to suit the needs of your application

Entity Relationship Diagram (ERD)

- The following ER diagram is given as a reference. We will be providing data sets for the **Customer**, **Product** and **Category** entities.
- The attributes in Customer, Product & Category are **MANDATORY**. You may not remove any of them. However, you are free to add new attributes to match any new features you wish to introduce.
- The **OrderItem** and **Order** entities are an **EXAMPLE** of how you can manage requirements [3-6] in the requirement section below. There are other ways to do this as well. Teams **MAY** make changes so long as tasks requirements are met.
- If you wish to use our modelling to implement your shopping cart requirements, here are the definitions for both entities.
- Order Entity Attributes:
 - **Id**: An incremental id to keep track of the latest order by any customer
 - **Customer_id**: The id of the customer that this purchase record belongs to.
 - **Status**: The status of the current purchase record
 - **Pending**: Items in cart but not yet checked out
 - **Purchased**: Items have been successfully checked out
 - **Created_at**: Timestamp of the creation of this record
- OrderItem Entity Attributes:
 - **Composite key of product_id & order_id**: Helps keep track of the specific products in each Order.
 - **Product_qty**: Quantity purchased for the product
 - **Total_price**: Product_qty * price
- Teams may also introduce new entities to accommodate any new features

ERD Diagram

