# Problem Statement

We are developing an e-commerce website for a tech retailer. The web app will have a login page for customers as well as admins, a landing page, product catalogue, shopping cart, order checkout.

**Features included will be:**

* **Home Page** – this will display all the products available.
* **Product Details Page –** this will display individual product details when product is opened.
* **Shipping Address Form** - for entering address other than permanent address when ordering.
* **Orders Page** – to see orders and order history.
* **Cart Page** – To see products that user selected to buy and place order.

# Functional Requirements

Mandatory Module to be developed in every project

* Admin - Module for maintaining master data
* User - Login
* Dashboard – to display all products
* Shopping cart - product quantity, items, price
* Checkout
* CRUD (Create, Read, Update, Delete) operations on products

**Login Module Details**

For login below details are to be captured:

* User email
* Password
* Authentication based on entered credentials

**Landing/Home Screen Module**

* Header - different technologies, desktop, tablets, laptops, account information, shopping cart
* Find a store (optional)
* Search products by name on homepage.

**Account information module**

* User details
* Edit user information

**Shopping cart module**

* Products in users shopping cart
* Image
* Price
* Product quantity
* Manipulating products quantity
* Adding Shipping address

**Orders module**

* Summary of user order
* Total price
* Shipping address

**Product Details Module**

* Description
* Image
* Price
* Add to shopping cart

**Admin module**

* Catalog of products
* CRUD, Create, Read, Update and Delete products
* Create discount codes and promotions on system

# Non-Functional Requirements

* Application should be hosted as HTTPS or HTTP
* The application UI should be supported on Google Chrome and Microsoft Edge
* The application should be responsive and should be accessible on Desktop and Mobile
* All the coding guidelines and code quality standards to be followed
* Proper database schema’s to be created
* Follow best practices for databases
* Unit Testing code coverage should be 80%
* Unit Testing for UI is optional
* The application should following proper naming conventions for both front-end as well as backend codes
* The application should be completed with 80% working functionalities to be accepted for final presentation

# Technology Stack Used

* Database : PostgreSQL
* Frontend UI : Angular Framework
* Backend Application : Spring Boot with Spring Data JPA and Spring REST API
* Rest Documentation: Swagger