1. Title of the Project

Online Shopping (Online Dukan)

2. Course Name

DAC (KP)

3.Names of the Project Team Members

Sl.No.	Name	Prn No.
1.	Nitin Ballari	200250120014
2.	Dawtie Grace Joseph	200250120023
3.	Amol Nagargoje	200250120055
4.	Pooja Upadhyay	200250120066
5.	Sarkar Soham	200250120086

Project Description:

1. The objective of this project is to develop a general-purpose e-commerce web application where any product can be bought from the comfort of home through the Internet.

2. Customer can browser the catalog and select products of interest. The selected products may be collected in a shopping cart.

3. To complete the transaction, the customer will be asked to fill details such as billing address, a shipping address, payment details etc.

4. We have used Angular in front-end and Spring Boot in back-end and will be communicating using REST API and in background we will have MySQL database.

5. JWT authentication was used for securing an exposed REST API.

Explain one scenario:
Products on site Explained-

Presentation layer
1. After successful login, user can see all kinds of items which are ready for sell. We are getting these items from backend through service calls (item.service.ts) and show it through item-list.component.html file.

2. We have used different routes in presentation layer to get different category of items. For example, if user is clicking on balls category, all balls available will be displayed.

3. Search bar allows user to search for products based on name. When user clicks on search button, text inside search bar is captured and a call to searchItems method in item.service.ts is invoked. In item.service.ts, GET request with corresponding URL and parameter is passed to backend server. Products are displayed if there is a match, else "No items found" message is shown.

Service Layer
In service layer, URL is mapped to repositories by spring data rest.
Spring Data REST identify ItemRepository interface that extends JPARepository and find item names containing the parameter in its name in case of search.

DAO Layer.

1. We have used JPA repository to use some of inbuilt query of hibernate ORM.On backend when productservice returning call to Jpa.Jpa have three query request.

2.Two written query are findbycategoryItemcategoryId and findbynameItemContaining to fetch items of selective category and for items of selective name category from databases.

Some scenarios where we got struck in the project and how did we overcome:

1. While searching item through textbox if item is not available in database it gives empty screen . In order to avoid empty screen we used safe navigation operator(?)

   a. In order to display a message we have written the following code:

```
<div *ngIf="items?.length == 0" class="alert alert-warning col-md-12"
role="alert">
    No items Found.
    </div>
```

   It Guards against null and undefined values in property paths.

2. CheckoutComponent will subscribe to events from CartService. However since CheckoutComponent is instantiated later in the application will miss out on previous message and it will give error in total price and quantity. Intially in CartService we were using Subject to publish events in code.
   a. Subject does not keep a buffer of previous events and subscribe only receives new events after they are subscribed.
   b. In order to solve this problem we have used BehaviorSubject

   It has a buffer of the last event and once subscribed, subscribe receives the latest event sent prior to subscribing.

```
totalPrice: Subject<number> = new BehaviorSubject<number>(0);
 totalQuantity: Subject<number> = new BehaviorSubject<number>(0);
```

3. After Order placed,we are getting still previous items list in our cart and page was not navigating to homepage

   We solved above problem by adding reset method in the calling rest api via checkoutservice function of checkout.component.ts

   And navigate back to the home page by adding this.router.navigateByUrl(“/items”).

Learnings during the project:

1. Spring security with JWT
2. Spring data rest
3. Angular Framework
4. Hibernate
5. JPA Query
6. Learnt the complete flow as to from where and how the data travels between layers.
7. Learnt cors origin concept
8. Learnt to work as a team across remote places