BlinkWorx

CODING ACTIVITY - Full Stack Developer NodeJS, React, PostgreSQL

Thank you for taking the time to take on the engineering challenge! We use this project as a way to understand your coding abilities as well as how you would work on a project. When you submit your code, we are not just looking for code that works but for code that is **production ready** and is able to be merged into our codebase.

Here are the criteria we look for when we review your code.
1.) Does your code work? Are all features in working condition?
2.) Code Readability.
3.) Code Reusability.
4.) Error Handling.
5.) Testability.
6.) JS/CSS/HTML fundamentals and best practices

This is code you would feel proud to submit to a real project. Please develop the assignment using **NodeJS, ReactJS (feel free to use any server and ui framework or libraries such as express, nest, next etc) and any RDBMS (preferably postgreSQL)**, and please submit your project as a GitHub or Gitlab repo with instructions on how to run your project. Please also include assumptions and tradeoffs you made while building out the component.

Delivery Expected:

1) please submit your project as a GitHub or Gitlab repo with instructions on how to run your project.(in case of unclear instructions or errors while compiling the code, assignment will be rejected)
2) Host your code on the vercel, netlify or any platform of your choice and share the link.

A detailed specification of this CODING ACTIVITY is available below.
1) Build a **ECOMMERCE** database in **PostgreSQL**

Create **ORDERS** table as below

Id	SERIAL	PK
orderDescription	VARCHAR(100)	NOT NULL
createdAt	TIMESTAMP	NOT NULL

Create **PRODUCTS** table as below

Id	INT	PK
productName	VARCHAR(100)	NOT NULL
productDescription	TEXT	

Create **OrderProductMap** table

Id	SERIAL	PK
orderId	INT	FK {ORDERS.Id} | NOT NULL
productId	INT	FK {PRODUCTS.Id} | NOT NULL

BlinkWorx

2) Insert below values in **PRODUCT** table

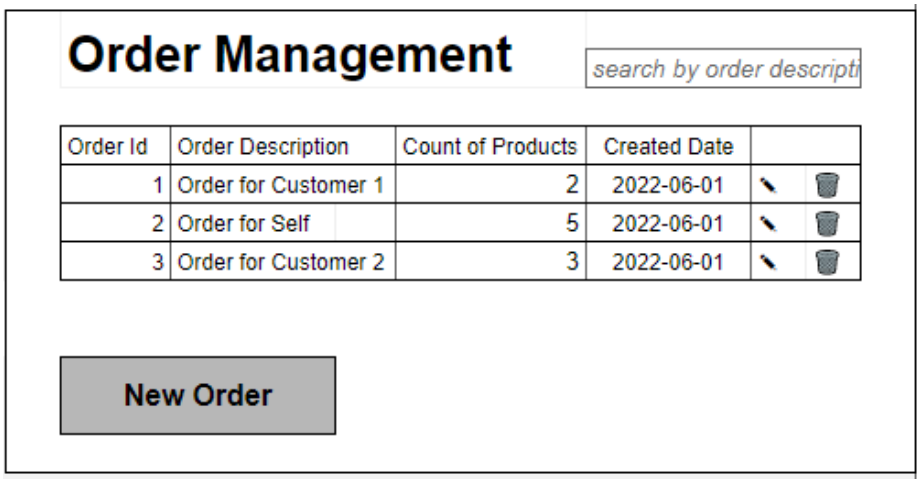
Id	productName	productDescription
1	HP laptop	This is HP laptop
2	lenovo laptop	This is lenovo
3	Car	This is Car
4	Bike	This is Bike

3) Build the following REST API

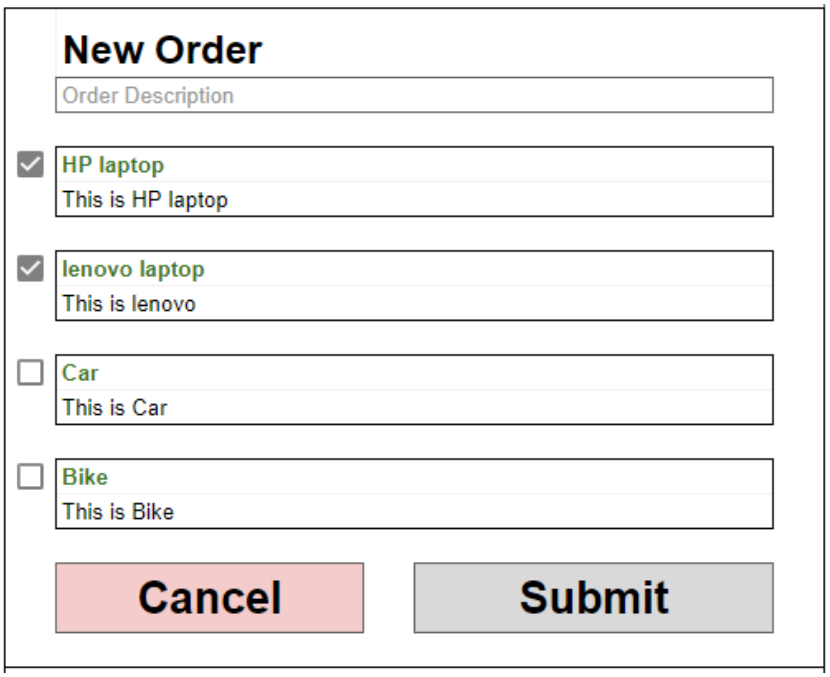
Methods	Urls	Actions
GET	api/order	get all Orders
GET	api/order/:id	get Order by id
POST	api/orders	add new Order
PUT	api/orders/:id	update Order by id
DELETE	api/orders/:id	remove Order by id

4) Build a Order management Screen (see wireframe below), below is just a wireframe use a good looking styling css and neat/clean UI

- Search should filter the table for matching “order id” or “Order Description”



BlinkWorx



- Cancel button should redirect back to Order Screen
- Add a cart icon to show “no of products” added
- “Book Order” button should add Order Entry and redirect back to Order Screen

ALL THE BEST!! 🙂