sezzle
The way forward.

Online Shopping Portal API: Golang & React

You're an owner of a newly-started e-commerce store, and you want to create a basic web service to cater to your customers. Since you're just starting up, you want to keep it simple, with the following schema:

User ->Cart of Items->Order with Items->Done!

You're planning to use gorm (https://github.com/jinzhu/gorm/) as the ORM, gin (https://github.com/gin-gonic/gin/) as the web framework and ginkgo (https://github.com/onsi/ginkgo) for any tests you want to add.

Stack Preferences: Golang & ReactJS

Assignment Submission: Code should be submitted through github with required access. Also host the assignment. Also, the assignment should be deployed to netlify or any publicly accessible hosting service provider with required access for the review.

You're keeping the following flow in mind for the user:

1. When a user will come to your platform and signup (/user/create API is called), a new User account gets created.

2. The user needs to be logged in order to create a cart. If the user already has an account, the user will login using /user/login API which will return a token for further requests in the user's session. A user can only be logged in from a single device at a time i.e. there'll always be a single token for the user.
3. When the user starts shopping and selects Items, the Items will get added to the Cart. This will be done by /cart/add API. A single user can have only a single cart, so you'll need to identify the Cart by the User's ID.
   Note: For sake of simplicity let's assume that we don't want to manage inventory yet i.e. we don't have to keep track of the number of items, whether it's in stock or out of stock, etc.
4. The cart will get converted into an order when the /cart/:cartId/complete API is called.
5. There should also be listing endpoints for User, Items, Carts and Orders.

Endpoints Summary as below:

Endpoint URL	Description
/user/create	Creates a new User
/user/login	Login for existing user based on username and password
/item/create	Creates an Item
/cart/add*	Adds Items to the cart
/cart/:cartId/complete*	Converted the cart into an order
/user/list	List all users
/item/list	List all items
/cart/list	List all carts
/order/list	List all orders

*The user's token must be present in the cart related endpoint request to identify which user the cart belongs to.

A general overview of the Entities that need to be used:

```
entity User {
      id integer
      name string
      username string
      password string
      token string // To identify the user in the requests
      cart_id ForeignKey(Cart) // One user can have only one cart
      created_at timestamp
}

entity Item {
      id integer
      name string
      created_at timestamp
}


entity Cart {
      id integer
      user_id ForeignKey(User)
      is_purchased boolean // Set this to true when the Cart gets converted
      into an Order
      created_at timestamp
}

// Many To Many relationship between Cart and Items.
// Cart -> has -> Items (one cart can have multiple items)
// Item -> belongsto -> Carts (one item can be in multiple carts)
Relationship CartItems {
      cart_id integer
      item_id integer
}

entity Order {
      id integer
      cart_id ForeignKey(Cart)
      user_id ForeignKey(User)
      created_at timestamp
}
```

Now that you've the backend ready, time to show it in the UI!

Create a React webapp with the following screens in order:

1. Create a basic User Login screen where the user can enter the username and password to List the Items in the shopping portal. On login failure show a window.alert() saying Invalid username/password. On successful login take the user to the List Items screen.
2. The List Items screen is where the user can see all the items. Clicking on an item in the items list should add that item to the cart. Show a Checkout button at the top of this screen which the user will use to place an order. Also next to the Checkout button show 2 buttons:
    1. Cart button to list all the added Items in the cart. Clicking on this button should show all the cart items (i.e. cart_id, item_id) in a toast or window.alert().
    2. Order History button to list all the placed orders for the user. Clicking on this button should show all the placed Order ids in a toast or window.alert().
3. On clicking the checkout button, the cart should get converted to an order. You can show the List Items screen again with a toast saying Order successful.

Good luck with your new store!