Team,

Please find below the complete use case : Write a **place-cart** function with following functionalities:

A] api gateway:
url: /place-cart
method: post
type of integration: lambda-proxy

B] lambda:
function: place-cart
flow:

1. take cart and other data mentioned below from request body
2. validate data
3. check if cart exist by scan/query
    - 3.1 if already exists, return cart exists
    - 3.2 if not proceed below
4. get discount info from s3 on basis promotion code,
    - 4.1 if failed, return cart failed
    - 4.2 if succeed, should proceed below
5. populate cart model and store in dynamodb with current timestamp
    - 5.1 if failed, return response ; cart failed
    - 5.2 if succeed, return response

code checks:

1. s3 url and db url should get from parameter store
2. all functions should be properly commented, variables naming convention should be camel case
3. error handling of each service call
4. mocha-chai unit testing as discussed

C] requestObject

```
1. cart {
        cartId: string
        address: string,
        userId: string,
        phoneNumber: number,
        emailId: string,
        promotionCode: string,
        items: [
                {
                        itemId: string,
                        quantity: number,
                        description: string
                }
         ]
        }
```

D] **Service response:**

1. http code: 200

```
{
    cartId: string,
    cartStatus: string{successfully placed cart}
}
```

2. other error http code: 500

```
{
    cartId: string,
    cartStatus: string{failed to place cart}

}
```

E] dynamodb data

1. make cartId as primary index, and userId and secondary

```
1. cart {
        cartId: string
        timesatmp: date,
        userId: string,
        phoneNumber: number,
        emailId: string,
        promotionCode: string,
        items: [
                {
                        itemId: string,
                        quantity: number,
                        description: string
                }
         ]
        }
```

F] s3

1. create a json file contain promotionId,

```
        {
                1111: {
                        description : string,
                        discount: number
                },
                1112: {
                        description : string,
                        discount: number
                }
        }
```

2. read file, get discount for promotionId [- 1111]