11

a. How to use your system

cd /folder address of project heroku login yangminp999@gmail.com Yqw520520.
heroku open
Then, can start doing something in the application address:
https://nwen304gropproject.herokuapp.com/

b. What the REST interface is

WEB SERVICE INTERFACE	Explanation

GET(/womens)
GET(/womens_most_popular)
GET(/womens_price_ascending)
GET(/womens_price_descending)
GET(/mens)
GET(/mens_most_popular)
GET(/mens_price_ascending)
GET(/mens_price_descending)
GET(/kids)
GET(/kids_most_popular)
GET(/kids_price_ascending)
GET(/kids_price_descending)
GET(/new)
GET(/new_most_popular)
GET(/new_price_ascending)
GET(/new_price_descending)
	https://minping304.herokuapp.com/
	All corresponding information on database will be shown on the corresponding position at webpage, once webpage was refreshed.

POST(/Login_account)
POST(/register)
POST(/search)
POST(/add)
POST(/view_cart)
POST(/recommendations)
POST(/get_orders)
POST(/create_order)
	click a corresponding button in webpage to login, register, search, add or view cart, then all information will be passed to corresponding POST function, used the information to make some corresponding changes in the database by using sql command.
	Recommendations uses order information to suggest items for the user.
	Get_orders gets the orders of the currently logged in user.
	Create_order creates an order using the items currently in the cart.

PUT (/reset_pwd)	click a reset password button after that filled the corresponding information of account. Firstly, PUT function will check if all information of account are correct. if it is, then update the password of corresponding account by using sql command.

DELETE(/delete_order)
DELETE(/delete_item)
	The /delete_order is used when the login email is the admin email and the admin deletes an order.
	The /delete_item is used when the user wants to delete an item in the cart.

c. What error handling has been implemented in your system

1. on every API function in server part, after client.query""(), I checked if it will query successfully. if it does, then it will send query result into corresponding callback function parameter value $.ajax().then(callback function). Otherwise, it will just pass a "'No data found'" string to callback function

2. use try{}catch(){} statement to catch connection error and print out the error on console, when the API function try to connect server.
3. When $.ajax called is rejected, then it will print out error on console. by using ERROR_LOG.

in $.ajax({}).then(my_nextfunction, ERROR_LOG); statement.

3. The test cases for frontend and the test scripts (e.g. a list of curl commands) for the server end of your web application.

GET(/womens)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/womens
```

GET(/womens_most_popular)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/womens_most_popular
```

GET(/womens_price_ascending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/womens_price_ascending
```

GET(/womens_price_descending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/womens_price_descending
```

GET(/mens)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/mens
```

GET(/mens_most_popular)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/mens_most_popular
```

GET(/mens_price_ascending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/mens_price_ascending
```

GET(/mens_price_descending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
mens_price_descending
```

GET(/kids)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
kids
```

GET(/kids_most_popular)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
kids_most_popular
```

GET(/kids_price_ascending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
kids_price_ascending
```

GET(/kids_price_descending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
kids_price_descending
```

GET(/new)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
new
```

GET(/new_most_popular)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
new_most_popular
```

GET(/new_price_ascending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
new_price_ascending
```

GET(/new_price_descending)

```
curl -H "Content-Type: application/json" -X GET https://nwen304gropproject.herokuapp.com/
new_price_descending
```

POST(/Login_account)

```
curl -H "Content-Type: application/json" -X POST -d
'{"pword":"321","emailadd":"1234@1.com"}' https://nwen304gropproject.herokuapp.com/
login_account
```

POST(/register)

```
curl -H "Content-Type: application/json" -X POST -d
'{"fname":"gg","lname":"dd","pword":"321","emailadd":"129934@1.com"}' https://
nwen304gropproject.herokuapp.com/register
```

POST(/search)

```
curl -H "Content-Type: application/json" -X POST -d '{"q":"Max"}' https://
nwen304gropproject.herokuapp.com/search
```

POST(/add)

```
curl -H "Content-Type: application/json" -X POST -d '{"add":"Air
Zoom","email":"1234@1.com"}' https://nwen304gropproject.herokuapp.com/add
```

POST(/view_cart)

```
curl -H "Content-Type: application/json" -X POST -d '{"email":"1234@1.com"}' https://
nwen304gropproject.herokuapp.com/view_cart
```

/delete_order

```
curl -H "Content-Type: application/json" -X DELETE -d '[{"order_id":2,
"email":"1234@1.com"}]' https://nwen304gropproject.herokuapp.com/delete_order
```

/delete_item

```
curl -H "Content-Type: application/json" -X DELETE -d '[{"item_name":"Gazelle",
"email":"hello@hello.com"}]' https://nwen304gropproject.herokuapp.com/delete_item
```

/recommendations

```
curl -H "Content-Type: application/json" -X POST -d '{"email":"hello@hello.com"}' https://
nwen304gropproject.herokuapp.com/recommendations
```

/get_orders

```
curl -H "Content-Type: application/json" -X POST -d '{"emailadd":"hello@hello.com"}'
https://nwen304gropproject.herokuapp.com/get_orders
```

/create_order

```
curl -H "Content-Type: application/json" -X POST -d '[{"item_name":"Gazelle", "price":"57",
"quantity":"2", "email":"hello@hello.com"}]' https://nwen304gropproject.herokuapp.com/
create_order
```

4Database Summary

For the database I created 6 tables for our web application. I tried to design the database in a way to minimize duplicate data.

The account_table contains uses user_id and email as the primary key. In the account_table there is a salt attribute used in conjuction with the plaintext password to get the hash password, stored as the attribute pwd in the account_table. The account_table is used for the login and registration API functions.

The in_cart table stores the item, user, and quantity. With these attributes we can locate and identify what items and quantity belongs to which user.

The items table stores all the information of the items as well as the image path. The items have 3 categories, mens, womens, kids, which corresponds to the category_id 1, 2, 3. The categories are stored in a separate category table.

The orders table is used to store an order for the user once the purchase items button is clicked in the view cart page. For each item in the cart, a new entry for the order_details is inserted which contains the item_id, quantity, and the order_id which refers to the order that was just created from the purchase button.