Go in Action Assignment Report

Title: Home Baker Application

Submitted by: Tho Yan Bo

Application Overview

Home Baker is a simple application that is used to manage the orders of a home based bakery for the upcoming week (Monday to Sunday).

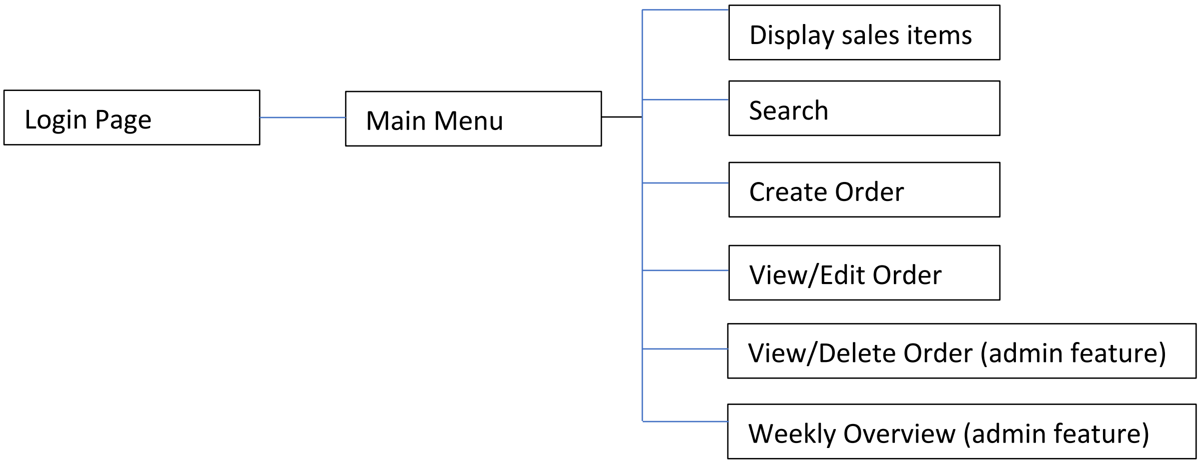
Below is an overview of the client and server structure for the home baker application.

Client	Server
Login Feature - New customer sign up - Existing customer login - Admin login	Login - Issue Cookie - Store session-user information - Store user information
Customer Page - Display all sales items - Search function - Create a new order - View or edit order	Sales item data - Sales item availability (derived from max weekly capacity and weekly order) - Item data are stored in array based list Delivery days data - Delivery day availability (based on pre-defined daily delivery cap) Booked orders data - Booked orders are stored into a linked list
Admin Page - Display sales items - Search function - Create a new order - View or edit order - Delete order (admin exclusive) - Order overview (admin exclusive)	Manages weekly overview by returning information to admin through backend handling of item and booking data: - Weekly Revenue - Daily Orders details - Item order details

For this assignment, the incorporation of the server and client is using a localhost at port 5221.

Application Features

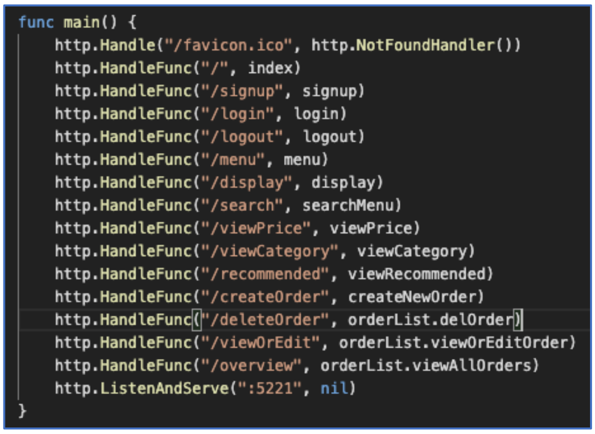
The Home Baker application consists of the following features.



Discussion of concepts discussed in Go in Action 1 and applied in this assignment.

1. ListenAndServe

ListenAndServe starts an HTTP server with a given address and handler. For this assignment, the given address is local port :5221. The handler is typically nil, which means to use DefaultServeMux. Handle and HandleFunc both add handlers to DefaultServeMux which is what is being used to set up the handlers for this application.



```go
func main() {
	http.Handle("/favicon.ico", http.NotFoundHandler())
	http.HandleFunc("/", index)
	http.HandleFunc("/signup", signup)
	http.HandleFunc("/login", login)
	http.HandleFunc("/logout", logout)
	http.HandleFunc("/menu", menu)
	http.HandleFunc("/display", display)
	http.HandleFunc("/search", searchMenu)
	http.HandleFunc("/viewPrice", viewPrice)
	http.HandleFunc("/viewCategory", viewCategory)
	http.HandleFunc("/recommended", viewRecommended)
	http.HandleFunc("/createOrder", createNewOrder)
	http.HandleFunc("/deleteOrder", orderList.delOrder)
	http.HandleFunc("/viewOrEdit", orderList.viewOrEditOrder)
	http.HandleFunc("/overview", orderList.viewAllOrders)
	http.ListenAndServe(":5221", nil)
}
```

2. Use of templates to write response to client.

The package html/templates implements data-driven templates for generating HTML output that is safe against code injection. It uses predefined pieces of information to create and merge data on HTML documents that allows personalized pages to be served to the users.

Throughout the application, templates with type .gohtml have been created with the intention to merge data from the server and subsequently display them to the users in the html pages. The pre-defined template datafiles were parsed into the program using template.Must and template.ParseGlob and would be executed as a written response to the user using the template.ExecuteTemplate(res,templateName,Data) function. The use of "." (dot) is where the current data that is passed in from the ExecuteTemplate function will be placed in on the html page. The use of {{.}} can also be used to pass in aggregate piece of data or struct of data (slice, map etc.). {{$variableName :=.}} as a variable declaration is also possible and this was used in the create order features where the item info data struct was passed into the function and declared within the .gohtml page so that the user can create customized orders based on the available item offerings.

The use of templates to write response to the client in html form is used repeatedly through the application and can be found in almost all the features that take in the argument of (res http.ResponseWriter, req *http.Request).

3. Login/Logout, cookies and sessions.

Cookies are historical data stored on user's web browser while the website is being browsed. The user's login information and session ID are stored on the client's browser when the client signs into the application and are maintained in their current session through the use of cookies.

For this application, the application checks through the http request if the user is already signed into the application. It does this check by obtaining the current session cookie from the user. If the cookie is not found, the server assigns a cookie and session ID to the user browser which will be stored on the user's browser end. Thus, if the user's browser is not already logged in, nothing will be shown on the index page except the links to sign up or log in. Conversely, if the user is already logged in, the session ID is then sent onto the server which will be compared by the server to extract user information. The user info is presented on the index page and with a valid username the user may choose to browse the menu or log out from their current session.

Description of error handling and concurrency concepts incorporated in this assignment.

1. Concurrency issues

One concurrency challenge faced is that the server may need to handle multiple submissions of orders or edit orders which will be accessing the applications data base. As such, mutex lock has been applied in the application to ensure that only one booking can be added/edited at any one time so as to prevent potential race condition allowing multiple users to order beyond the weekly capacity.

Goroutines are also used at the start of the program to load multiple user orders into the program. This is done using preLoadOrders() function at the start of program to trigger multiple Goroutines to insert various orders into the order linked list. Since order number (global variable declaration orderNumber) is required for each individual order and each go routine will race to read and write with orderNumber, the atomic function is used to increment this global variable so as to ensure only 1 Goroutine can access this variable at any single time.

2. Error handling of invalid form informations

There are quite a number of instances within the application whereby the client may be entering an invalid submission, such as trying to make a booking on a day when bookings have been maxed out. As an example, the application has been pre-loaded with 5 Saturday bookings and cannot take new Saturday bookings. Should the user try to make a booking on Saturday, the server will catch that the bookings for Saturday are full and trigger an http error response to mention that the there are no more available delivery slots for the selected booking day.

There are other examples of this type of error handling in place (mostly in the form of http.Error) that will reject invalid submissions from the client side so that the server need not process the data submitted by the clients.

3. Error handling for the application

Lastly, error handling has been incorporated on the application side in the form of panic handling that will would cover any unexpected errors that may cause panic when running the server side codes.

Short Write-up on main features.

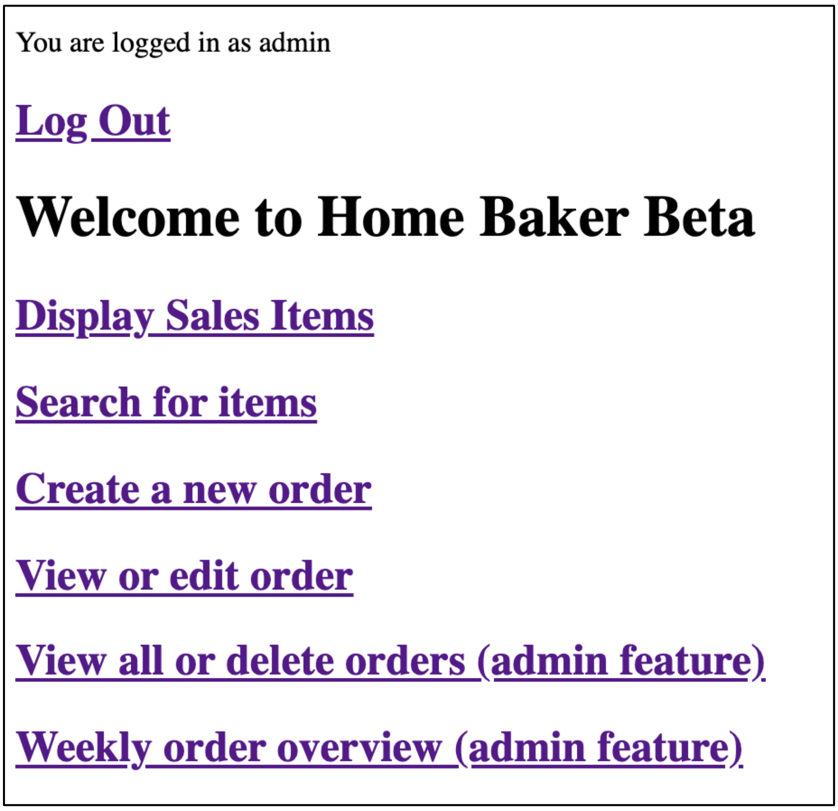
Existing Login Details:

The following details can be used to access the application. Else, a new user can also be created at index page.

User: Admin Password: password	Admin access, will be able to access view all/delete function and overview feature.
User: yanbo Password: password	Regular user without special access. Has 1 order available for editing.
User: ironman Password: password	Regular user without special access. Has 1 order available for editing.

Main Menu Feature

The main menu feature will welcome the user upon successful login and allows the user to make the following selections. Upon making a selection, the user will be redirected to the respective URLs that will be handled by the server and thus shown a client response based on the respective handler.



You are logged in as admin

Log Out

Welcome to Home Baker Beta

Display Sales Items

Search for items

Create a new order

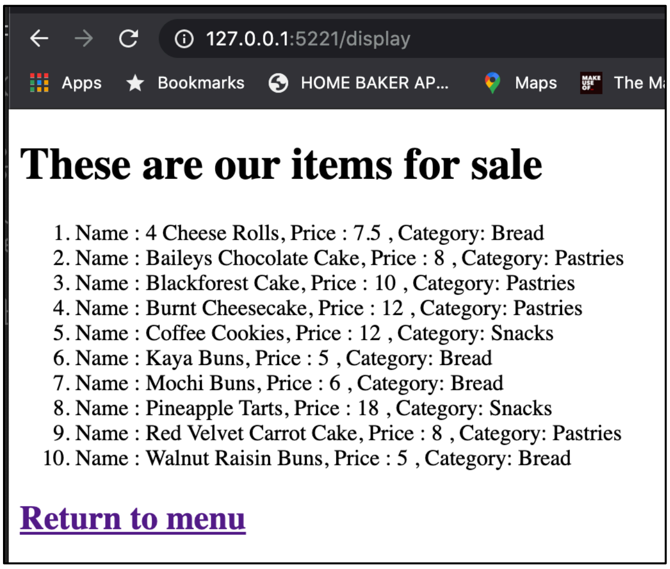
View or edit order

View all or delete orders (admin feature)

Weekly order overview (admin feature)

1. Display Sales Items

This feature will display all the items that are being sold by the home baker. The sales items are being stored in an **array based list** and are displayed based on alphabetical order due to the underlying function first sorting the array's items' names alphabetically before printing the item details based on array index.



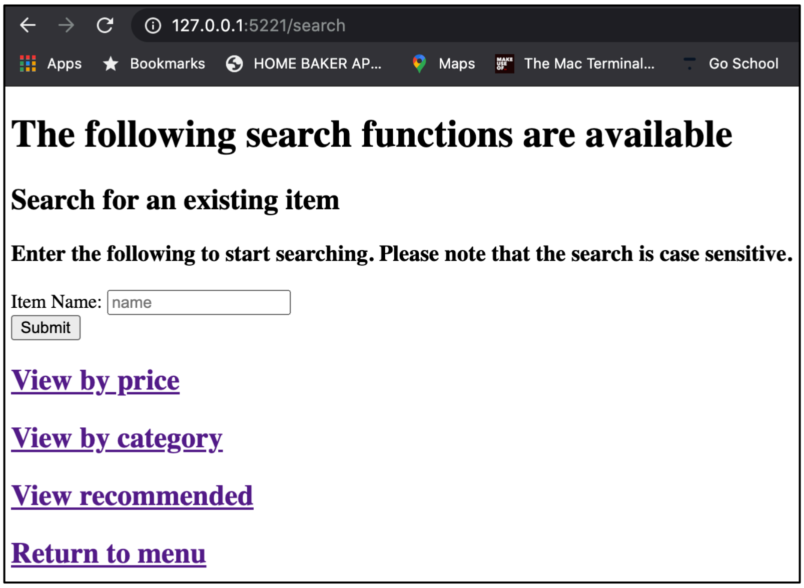
These are our items for sale

1. Name : 4 Cheese Rolls, Price : 7.5 , Category: Bread
2. Name : Baileys Chocolate Cake, Price : 8 , Category: Pastries
3. Name : Blackforest Cake, Price : 10 , Category: Pastries
4. Name : Burnt Cheesecake, Price : 12 , Category: Pastries
5. Name : Coffee Cookies, Price : 12 , Category: Snacks
6. Name : Kaya Buns, Price : 5 , Category: Bread
7. Name : Mochi Buns, Price : 6 , Category: Bread
8. Name : Pineapple Tarts, Price : 18 , Category: Snacks
9. Name : Red Velvet Carrot Cake, Price : 8 , Category: Pastries
10. Name : Walnut Raisin Buns, Price : 5 , Category: Bread

Return to menu

2. Search

The ‘Search’ feature allows the user to narrow their search option via the following options.



The following search functions are available

Search for an existing item

Enter the following to start searching. Please note that the search is case sensitive.

Item Name: name

Submit

View by price

View by category

View recommended

Return to menu

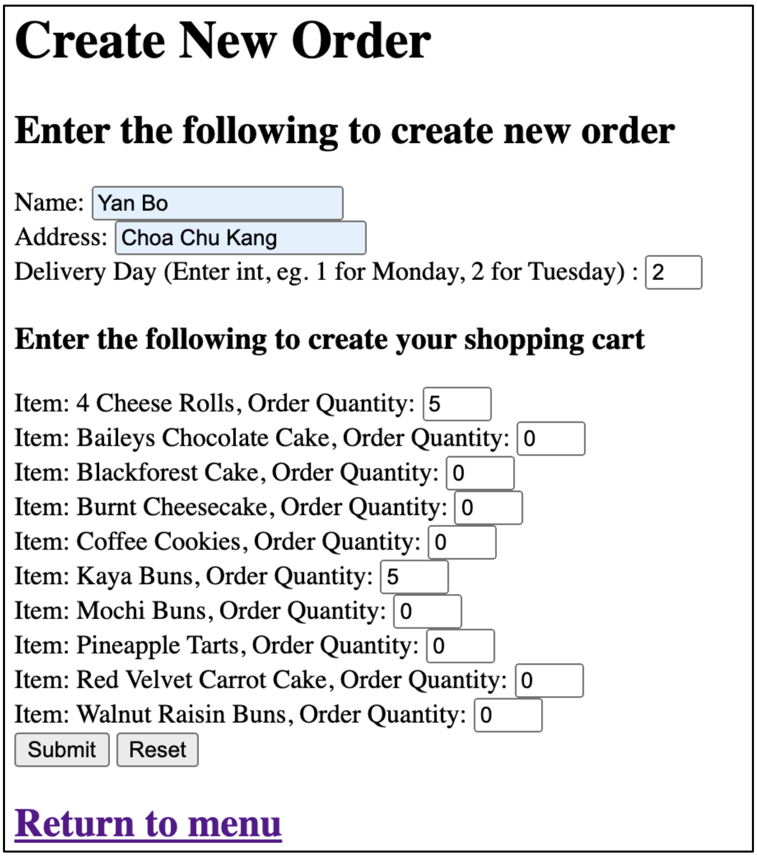
1. ‘Search by name’ returns the sales item when matched with the exact sales item name (case sensitive).
2. ‘View by price’ sorts the array using selection sort algorithm and returns the full list from lowest to highest price.
3. ‘View by category’ sorts the array using selection sort algorithm and returns the full list in category.
4. ‘View recommended’ returns the recommended sales item.

3. Create order

The ‘Create order’ feature allows the user to create a new order which will be added to the server’s order data list (linked list).

The URL displays a form that will prompt the user to insert details required to create an order. Upon submitting the form details, the server processes these details and creates a new order if processing is successful. If any of the details given are erroneous the server should detect them and return with a http bad server request error message. Once a successful order is created, the user is redirected back to the main page.

The available delivery days are processed and return to the URL via the server’s backend code and also shown to the client for ease of choosing an available delivery day.



Create New Order

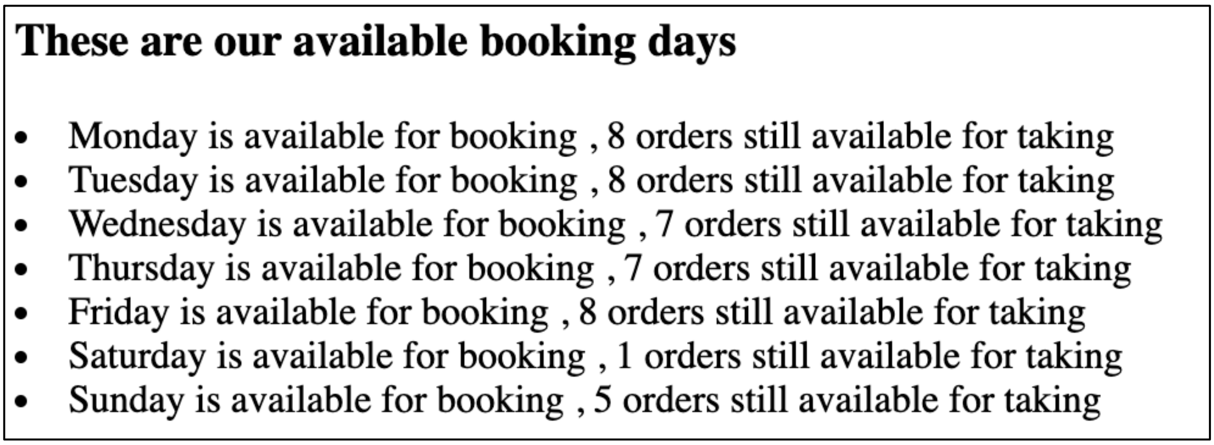
Enter the following to create new order

Name: Yan Bo
Address: Choa Chu Kang
Delivery Day (Enter int, eg. 1 for Monday, 2 for Tuesday) : 2

Enter the following to create your shopping cart

Item: 4 Cheese Rolls, Order Quantity: 5
Item: Baileys Chocolate Cake, Order Quantity: 0
Item: Blackforest Cake, Order Quantity: 0
Item: Burnt Cheesecake, Order Quantity: 0
Item: Coffee Cookies, Order Quantity: 0
Item: Kaya Buns, Order Quantity: 5
Item: Mochi Buns, Order Quantity: 0
Item: Pineapple Tarts, Order Quantity: 0
Item: Red Velvet Carrot Cake, Order Quantity: 0
Item: Walnut Raisin Buns, Order Quantity: 0

Return to menu



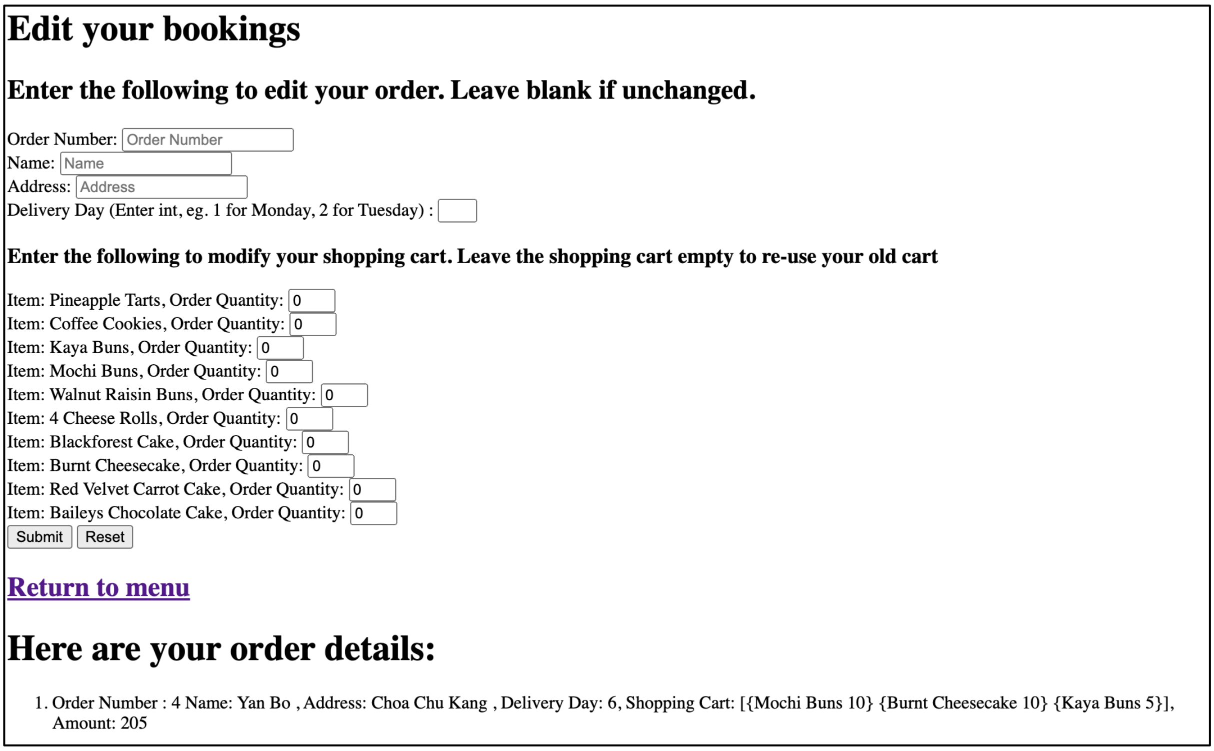
These are our available booking days

- Monday is available for booking , 8 orders still available for taking
- Tuesday is available for booking , 8 orders still available for taking
- Wednesday is available for booking , 7 orders still available for taking
- Thursday is available for booking , 7 orders still available for taking
- Friday is available for booking , 8 orders still available for taking
- Saturday is available for booking , 1 orders still available for taking
- Sunday is available for booking , 5 orders still available for taking

4. View/Edit Order feature

The 'View/Edit Order' feature would allow the user to view or edit orders based on their existing orders. The existing order is obtained via the use of client's cookies while logged into the current session. As the orders created had included the client's username as one of the required information in the order struct, this page will show all orders the user has made if any by matching the username with the orders' username field.

If the user has not made any orders, an error will appear to inform user that they do not have any current booking to view/edit.



Edit your bookings

Enter the following to edit your order. Leave blank if unchanged.

Order Number: Order Number
Name: Name
Address: Address
Delivery Day (Enter int, eg. 1 for Monday, 2 for Tuesday) :

Enter the following to modify your shopping cart. Leave the shopping cart empty to re-use your old cart

Item: Pineapple Tarts, Order Quantity: 0
Item: Coffee Cookies, Order Quantity: 0
Item: Kaya Buns, Order Quantity: 0
Item: Mochi Buns, Order Quantity: 0
Item: Walnut Raisin Buns, Order Quantity: 0
Item: 4 Cheese Rolls, Order Quantity: 0
Item: Blackforest Cake, Order Quantity: 0
Item: Burnt Cheesecake, Order Quantity: 0
Item: Red Velvet Carrot Cake, Order Quantity: 0
Item: Baileys Chocolate Cake, Order Quantity: 0

Return to menu

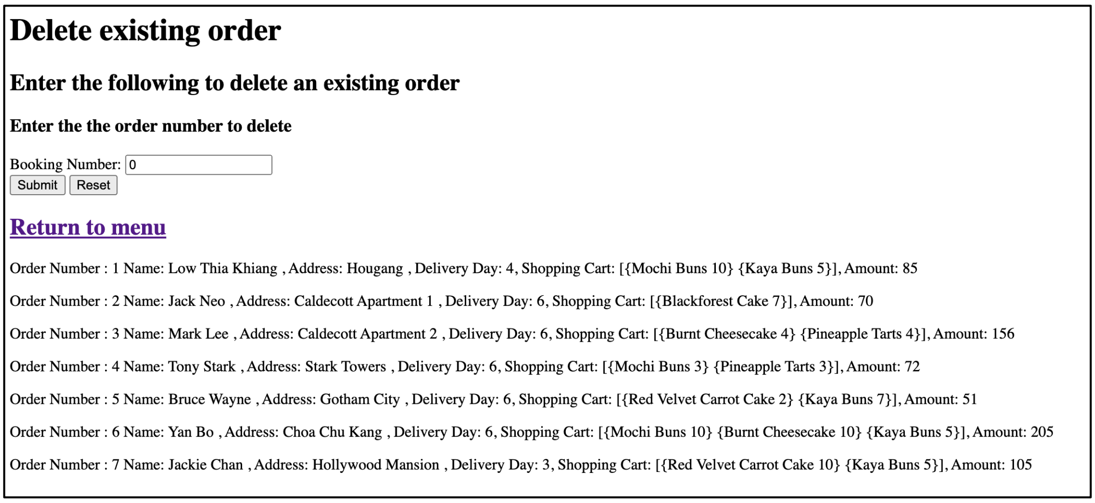
Here are your order details:

1. Order Number : 4 Name: Yan Bo , Address: Choa Chu Kang , Delivery Day: 6, Shopping Cart: [{Mochi Buns 10} {Burnt Cheesecake 10} {Kaya Buns 5}], Amount: 205

Similar to the create order feature, upon submitting the form details, the server processes these details and creates a new order if processing is successful. If any of the details given are erroneous the server should detect them and return with a http bad server request error message.

5. View all / Delete Order (Admin Feature)

This feature allows the admin to view all orders and also delete any booking number of choice. The server allows the user to access this page only if the client current session's cookie username matches that of the admin. Otherwise, access is denied for non-admin users.



Delete existing order

Enter the following to delete an existing order

Enter the the order number to delete

Booking Number: 0

Return to menu

Order Number : 1 Name: Low Thia Khiang , Address: Hougang , Delivery Day: 4, Shopping Cart: [{Mochi Buns 10} {Kaya Buns 5}], Amount: 85

Order Number : 2 Name: Jack Neo , Address: Caldecott Apartment 1 , Delivery Day: 6, Shopping Cart: [{Blackforest Cake 7}], Amount: 70

Order Number : 3 Name: Mark Lee , Address: Caldecott Apartment 2 , Delivery Day: 6, Shopping Cart: [{Burnt Cheesecake 4} {Pineapple Tarts 4}], Amount: 156

Order Number : 4 Name: Tony Stark , Address: Stark Towers , Delivery Day: 6, Shopping Cart: [{Mochi Buns 3} {Pineapple Tarts 3}], Amount: 72

Order Number : 5 Name: Bruce Wayne , Address: Gotham City , Delivery Day: 6, Shopping Cart: [{Red Velvet Carrot Cake 2} {Kaya Buns 7}], Amount: 51

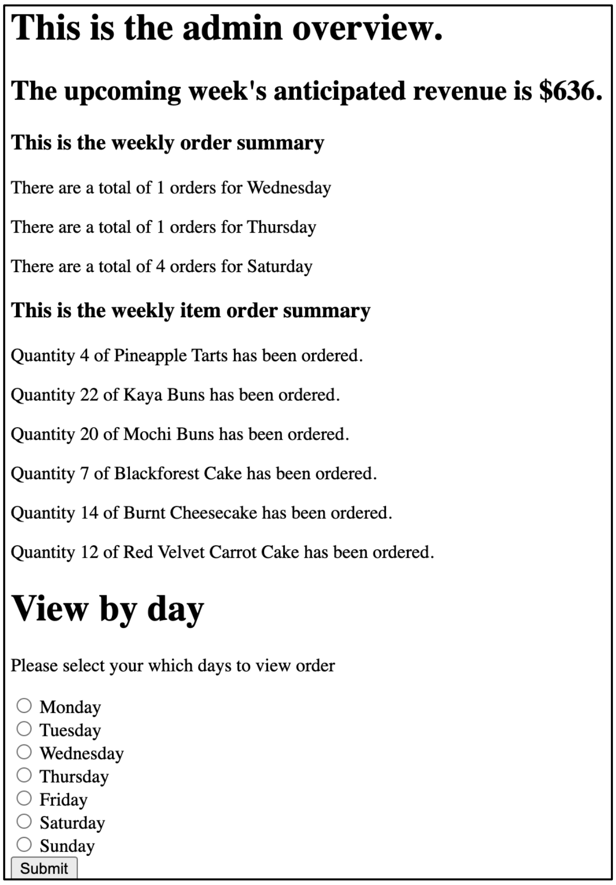
Order Number : 6 Name: Yan Bo , Address: Choa Chu Kang , Delivery Day: 6, Shopping Cart: [{Mochi Buns 10} {Burnt Cheesecake 10} {Kaya Buns 5}], Amount: 205

Order Number : 7 Name: Jackie Chan , Address: Hollywood Mansion , Delivery Day: 3, Shopping Cart: [{Red Velvet Carrot Cake 10} {Kaya Buns 5}], Amount: 105

6. Weekly Overview (Admin Feature)

The weekly overview feature is an admin access feature which allows the admin user to have a summarized view for the week's upcoming orders.

It shows the breakdown of the number of orders by day and the number of orders by item, and calculates the expected week's order revenue amount.



This is the admin overview.

The upcoming week's anticipated revenue is $636.

This is the weekly order summary

There are a total of 1 orders for Wednesday

There are a total of 1 orders for Thursday

There are a total of 4 orders for Saturday

This is the weekly item order summary

Quantity 4 of Pineapple Tarts has been ordered.

Quantity 22 of Kaya Buns has been ordered.

Quantity 20 of Mochi Buns has been ordered.

Quantity 7 of Blackforest Cake has been ordered.

Quantity 14 of Burnt Cheesecake has been ordered.

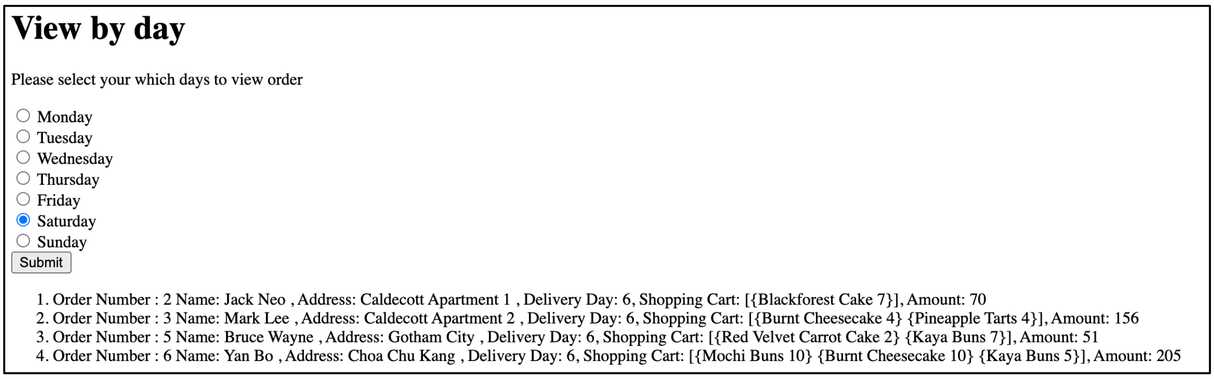
Quantity 12 of Red Velvet Carrot Cake has been ordered.

View by day

Please select your which days to view order

○ Monday
○ Tuesday
○ Wednesday
○ Thursday
○ Friday
○ Saturday
○ Sunday

It also gives the user the option to view the orders of any specific day submitting the day of choice through the radio button.



View by day

Please select your which days to view order

○ Monday
○ Tuesday
○ Wednesday
○ Thursday
○ Friday
● Saturday
○ Sunday

1. Order Number : 2 Name: Jack Neo , Address: Caldecott Apartment 1 , Delivery Day: 6, Shopping Cart: [{Blackforest Cake 7}], Amount: 70
2. Order Number : 3 Name: Mark Lee , Address: Caldecott Apartment 2 , Delivery Day: 6, Shopping Cart: [{Burnt Cheesecake 4} {Pineapple Tarts 4}], Amount: 156
3. Order Number : 5 Name: Bruce Wayne , Address: Gotham City , Delivery Day: 6, Shopping Cart: [{Red Velvet Carrot Cake 2} {Kaya Buns 7}], Amount: 51
4. Order Number : 6 Name: Yan Bo , Address: Choa Chu Kang , Delivery Day: 6, Shopping Cart: [{Mochi Buns 10} {Burnt Cheesecake 10} {Kaya Buns 5}], Amount: 205