

React App to create a web application where users can view a restaurant's menu and place an order.

Platform contains:

- **Server(Node.js)**
- **Client(React.js)**
- **Database(MongoDB)**

Server:

- It contains configuration codes like (MongoDB, serving port, etc.) and APIs.
- The node.js server is listening locally at port:8080
- For the server, the URL is: <http://localhost:8080/> or <http://127.0.0.1:8080/>
- APIs enable the server to interact with the client and the server

Platform APIs

1. **List Products**- returns the list of products to render the menu of the restaurant
URL to access this API- <http://localhost:8080/products>
method: GET
2. **Add to Cart**- add a product to the cart
URL to access this API- http://localhost:8080/add_to_cart
Method: POST
3. **List Cart Items**- returns cart items to render in the cart component
URL to access this API- http://localhost:8080/list_cart_item
Method: GET
4. **Remove the Cart Item** - This API is to remove the items from the cart or to decrease the quantity of the item in the cart
URL to access this API- http://localhost:8080/remove_cart_item
Method: DELETE
5. **Remove all items from Cart**- This API is used to remove all items from the cart collection of the database. This is used when the payment is done and the order has been placed.
URL to access this API- http://localhost:8080/remove_all_cart_item
Method: DELETE
6. **Add Order Summary**- This API is used to add detailed bills to the database for future references and to render it in the summary component
URL to access this API- http://localhost:8080/add_summary
Method: POST
7. **List Order Summary**- This API is used to list the detailed bills to render in the summary Component.
URL to access this API- http://localhost:8080/get_summary
Method: GET
8. **Payment**- This API enables the client to make payments(this will not accept actual payment) via Razorpay Gateway (test phase).
URL to access this API- <http://localhost:8080/capture/:paymentID/:total>
Method: POST

9. **Payment-** This API enables the client to create an order

URL to access this API- <http://localhost:8080/order/:total>

Method: GET

Client:

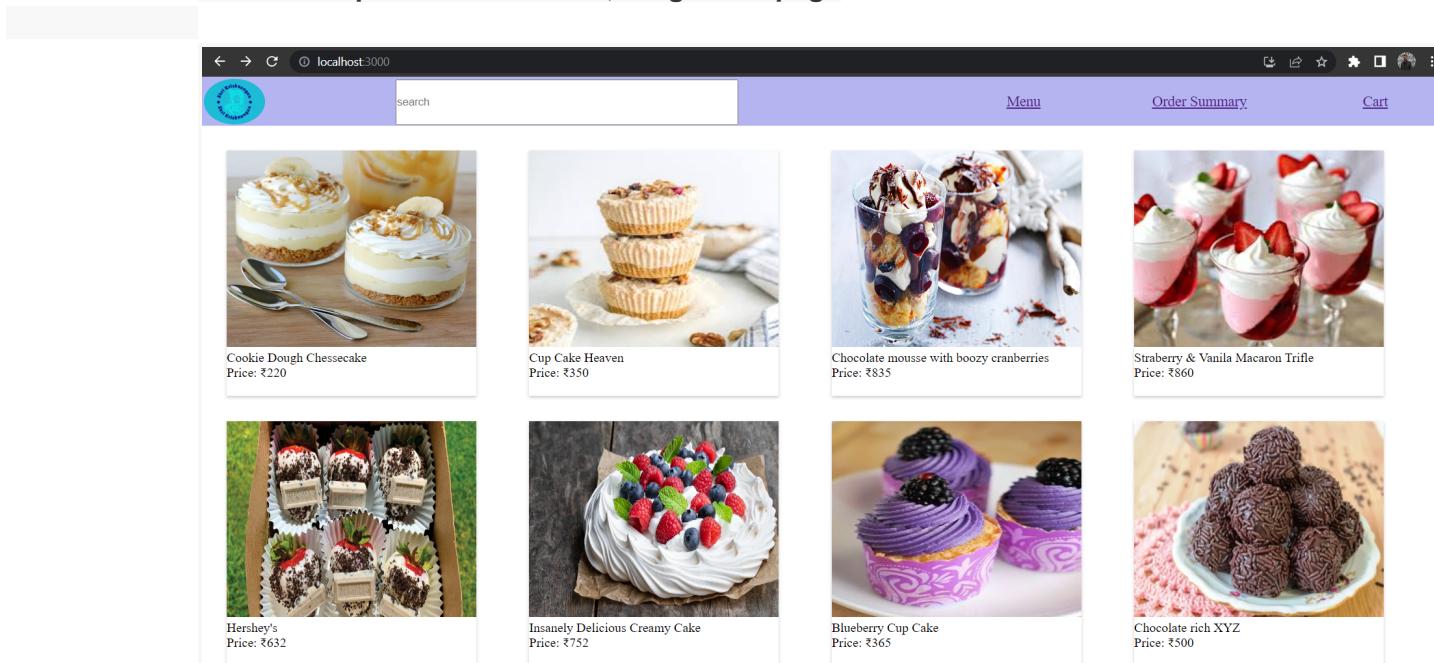
The client is a React.js application that has several sections-

URL for client - <http://localhost:3000/>

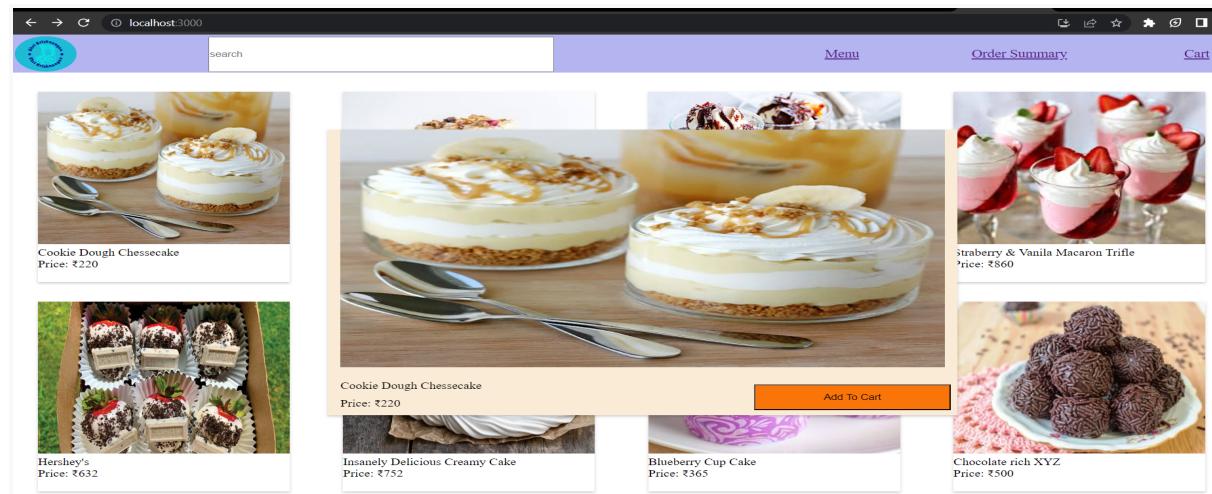
- Menu
- Cart
- Order Summary

Menu- This section renders the menu of the restaurant. Uses the API “List Product”, (1) in the API list to fetch the items from the MongoDB collection.

On URL- <http://localhost:3000/>, we get this page



If we click on any item we will get a popup like-



If we click on the “Add to Cart” button, the item will be added to the cart.

Cart- This section contains the items that were added to the cart and uses APIs- “Add to Cart”(2), “List Cart Items”(3), “Remove the Cart Item”(4), “Remove all Cart Items from Cart”(5), and “Payment”(6)&(7) in the API list.

URL Cart- <http://localhost:3000/cart>

The cart section will look-

Cookie Dough Chessecake
Price: ₹220
Quantity: 1
Payable: ₹220

Blueberry Cup Cake
Price: ₹365
Quantity: 2
Payable: ₹730

Total Payable Amount: ₹950

Pay

From this section, we can select the quantity and can pay.

When “Pay” button is clicked, we get a payment gateway as-

Fleksa

Pay With UPI QR

Scan the QR using any UPI app on your phone.

QR Code is valid for 11:40 minutes

Cards, UPI & More

Card

UPI / QR

Netbanking All Indian banks

Wallet

₹ 950 View Details

Pay Now

When payment is completed we can see our order details in the “order summary” section

Order Summary- This section renders the ordered details or the summary of the order. It makes use of API- “List Order Summary”, (7) in the API list.

Summary URL- <http://localhost:3000/summary>

localhost:3000/summary

Order ID: 64a6361c5a93f835783e65e7

Description:

Cookie Dough Chessecake
Price: ₹220
Quantity: 1
Payable: ₹220

Blueberry Cup Cake
Price: ₹365
Quantity: 2
Payable: ₹730

Total: ₹950

Menu Order Summary Cart



Database: MongoDB Atlas is being used for storage purposes.

To Run the application:

/Fleksa_Project/server/

Command: \$ npm start

/Fleksa_Project/client/

Command: \$npm start