

## UberEatsLab2: MONGO – REDUX SCREENSHOTS:

GIT URL- <https://github.com/vaishnavi333/UberEatsLab-2.git>

Activating virtual environment and mongodb (using mongosh)

The screenshot shows a VS Code interface with the following details:

- Explorer View:** Shows the project structure under "UBERESTSLAB2". The "uber\_eats\_backend" folder contains "customers", "media", "restaurants", and "venv". The "uber-eats-frontend" folder contains "vscode", "node\_modules", "public", "src", ".gitignore", "package-lock.json", "package.json", "README.md", "venv", ".gitignore", "package-lock.json", and "README.md".
- Terminal View:** The terminal window has tabs for "zsh" and "node". It displays the output of running "mongosh" in a Java runtime environment. It shows the MongoDB version (6.0.18), the use of Mongosh (2.3.3), and a warning about access control being disabled. It also lists database collections: admin, config, local, uber\_eats, test, and users.

Activating virtual environment and moving to backend folder. Downloading the requirements.txt.

```

UberEatsLab2: ~
vaishnavinanduri@Vaishnavis-MacBook-Air UberEatsLab2 % python3.10 -m venv venv
vaishnavinanduri@Vaishnavis-MacBook-Air UberEatsLab2 % source venv/bin/activate
(venv) vaishnavinanduri@Vaishnavis-MacBook-Air UberEatsLab2 % cd uber_eats_backend
(venv) vaishnavinanduri@Vaishnavis-MacBook-Air UberEatsLab2 % pip install -r requirements.txt
Collecting aspif==3.2.10 (from -r requirements.txt (line 1))
  Using cached aspif==3.2.10-py3-none-any.whl.metadata (8.5 kB)
Collecting django==3.1.12 (from -r requirements.txt (line 2))
  Using cached Django-3.1.12-py3-none-any.whl.metadata (3.6 kB)
Collecting django-cors-headers==3.5.0 (from -r requirements.txt (line 3))
  Using cached django_cors_headers==3.5.0-py3-none-any.whl.metadata (14 kB)
Collecting django-restframework==3.12.4 (from -r requirements.txt (line 4))
  Using cached django-restframework==3.12.4-py3-none-any.whl.metadata (9.9 kB)
Collecting drf-yasg==1.21.7 (from -r requirements.txt (line 5))
  Using cached drf_yasg==1.21.7-py3-none-any.whl.metadata (16 kB)
Collecting inflection==0.5.1 (from -r requirements.txt (line 6))
  Using cached inflection==0.5.1-py2.py3-none-any.whl.metadata (1.7 kB)
Collecting packaging==21.3 (from -r requirements.txt (line 7))
  Using cached packaging==21.3-py3-none-any.whl.metadata (7.5 kB)
Collecting pillow==8.3.2-cp310-cp310-macosx_11_0_arm64.whl (7.5 kB)
Collecting pytz==2021.1 (from -r requirements.txt (line 9))
  Using cached pytz==2021.1-py2.py3-none-any.whl.metadata (21 kB)
Collecting PyYAML==5.3.1 (from -r requirements.txt (line 10))
  Using cached PyYAML==5.3.1-cp310-cp310-macosx_11_0_arm64.whl (5.0 kB)
Collecting setuptools==59.3.2 (from -r requirements.txt (line 11))
  Using cached setuptools==59.3.2-py3-none-any.whl.metadata (5.0 kB)

```

## Migrating the changes and running the server.

```

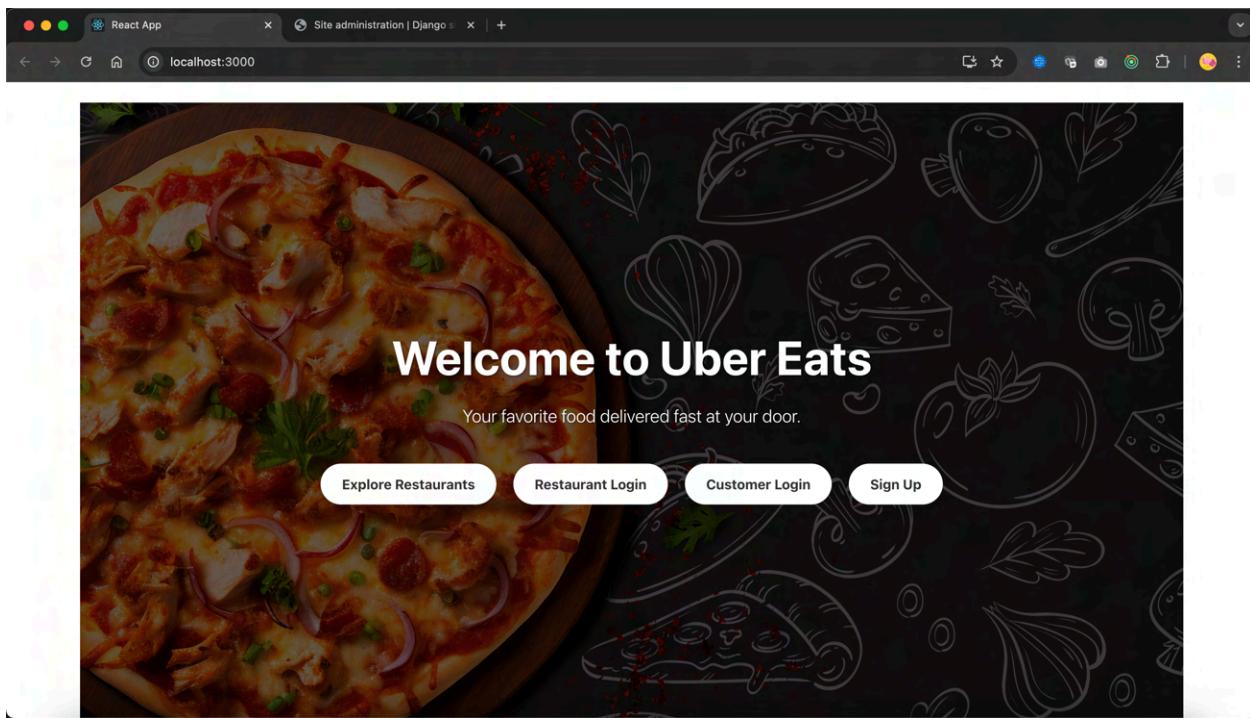
UberEatsLab2: ~
vaishnavinanduri@Vaishnavis-MacBook-Air UberEatsLab2 % python manage.py makemigrations
No changes detected
(venv) vaishnavinanduri@Vaishnavis-MacBook-Air UberEatsLab2 % python manage.py migrate
Operations to perform:
  Applying migrations: admin, auth, auth_token, contenttypes, customers, restaurants, sessions
Running migrations:
  No migrations to apply.
(venv) vaishnavinanduri@Vaishnavis-MacBook-Air UberEatsLab2 % python manage.py runserver
Watching for file changes with StatReloader
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
November 25, 2024 - 08:18:39
Django version 3.1.12, using settings 'uber_eats_backend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

```

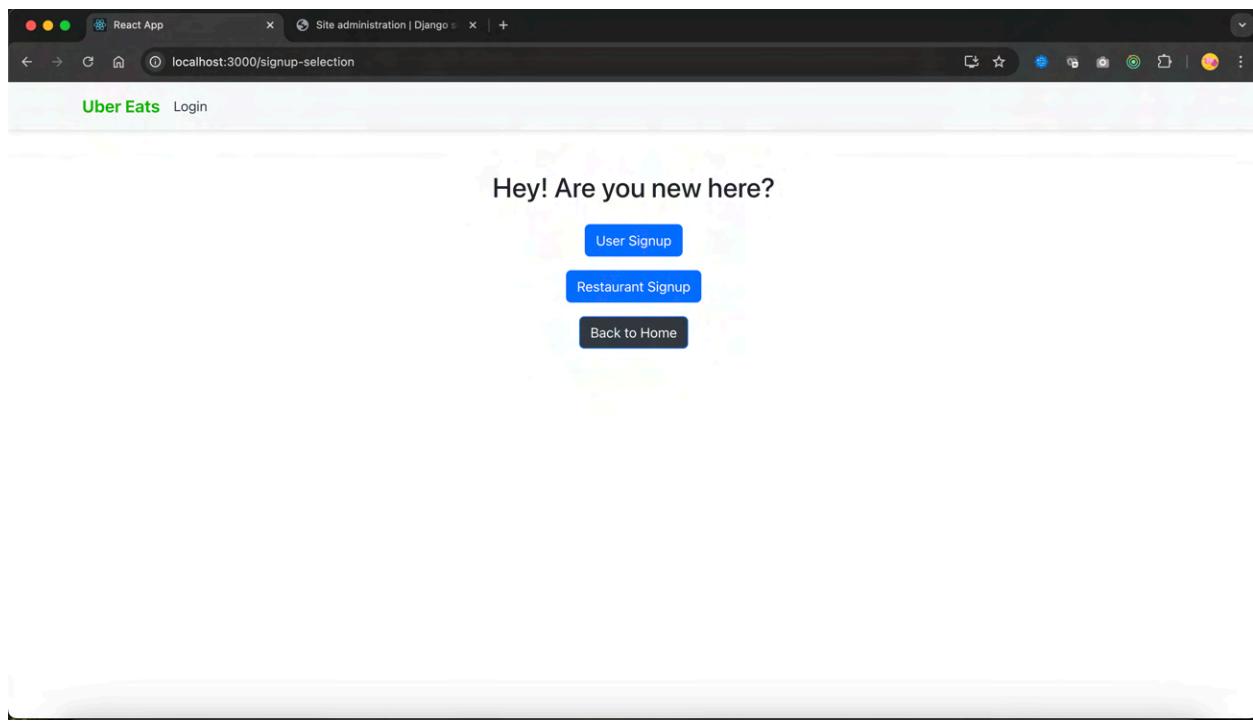
## Activating virtual environment and installing npm. Running the frontend server.

```
Compiled successfully!
You can now view uber-eats-frontend in the browser.
Local: http://localhost:3000
On Your Network: http://10.0.0.104:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
```

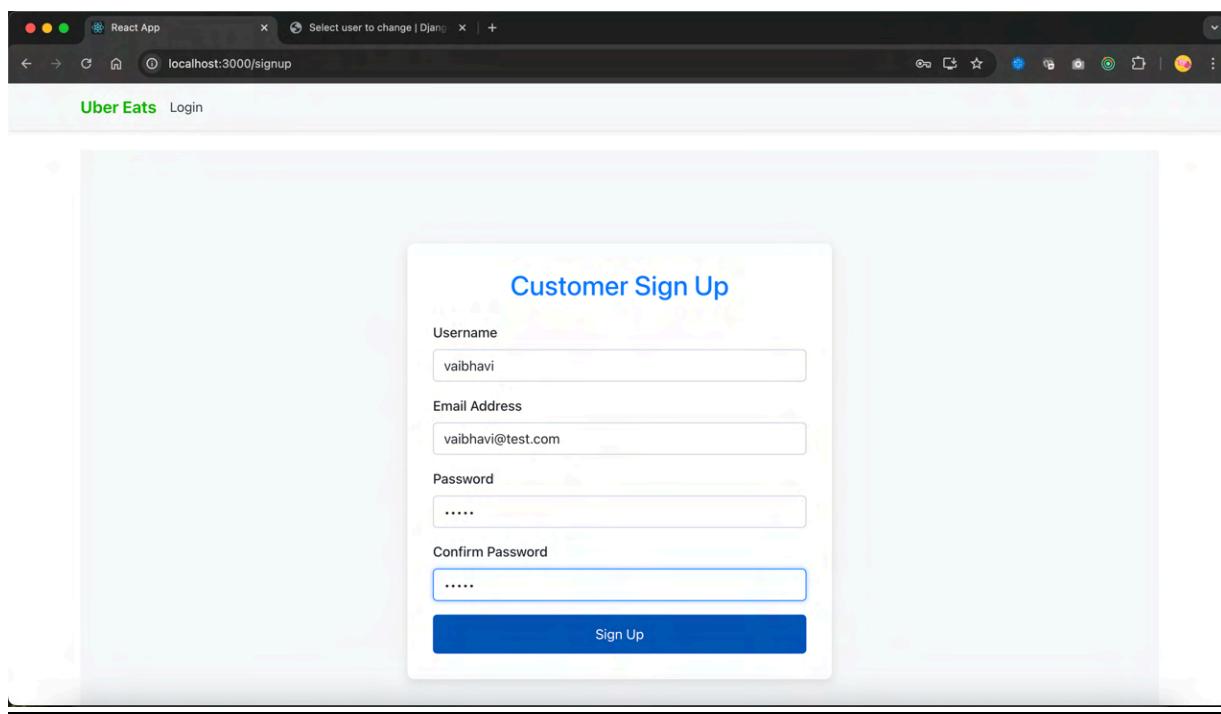
## Homepage / Landing page:



## Signup Selection:



### Cutomer signup:



### Customer Profile Update:

localhost:3000/userprofile

Uber Eats Welcome Restaurants Favorites Order History Logout

Profile updated successfully!

Full Name: vaibhavi

Nickname: vaibhu

Date of Birth: mm/dd/yyyy

Email address: vaibhavi@test.com

Phone Number: 7656987467

City: San Jose

State: CA

Country: United States

Update Profile

## Restaurant list for customer:

localhost:3000/restaurants

Uber Eats Welcome Restaurants Favorites Order History Logout

Simple and Natural Breakfast Cereals to Start Your Day

Special Offers Check out our exclusive offers! >

Available Restaurants

**South Indian Dishes**  
Only place with South Indian dishes  
[VIEW MENU](#)

**Mr. Biryani**  
Best biryanis available  
[VIEW MENU](#)

**Desserts Place**  
As sweet as sugar and you! Come visit!  
[VIEW MENU](#)

## Dishes list for a customer:

localhost:3000/restaurants/6

Uber Eats Welcome Restaurants Favorites Order History Logout

### Mr.Biryani

**Address:** Sunnyvale  
**Description:** Best biryanis available  
**Phone:** 7568376456  
**Opening Time:** 12:00 PM  
**Closing Time:** 12:00 AM

Items in Cart: 0

Menu

Image	Name	Type	Description	Price	Action
	Vegetable Biryani	Main Course	Basmati Rice, Mixed Vegetables, Yogurt, Spices (Turmeric, Cardamom, Cloves, Bay Leaves), Mint, Coriander A fragrant and flavorful rice dish cooked with fresh vegetables, aromatic spices, and herbs.	\$20.00	Add to Cart
	Paneer Biryani	Main Course	Basmati Rice, Paneer (Indian Cottage Cheese), Yogurt, Spices, Cloves, Bay Leaves, Mint, Coriander A rich and flavorful biryani with soft paneer cubes and layers of spiced rice.	\$25.00	Add to Cart
	Chicken Biryani	Main Course	Basmati Rice, Chicken, Yogurt, Spices (Turmeric, Cardamom, Cloves, Bay Leaves), Mint, Coriander A royal biryani with tender chicken pieces cooked with aromatic spices and long-grain rice.	\$30.00	Add to Cart

## Customer Favorite Restaurants:

localhost:3000/user/home

Uber Eats Welcome Restaurants Favorites Order History Logout

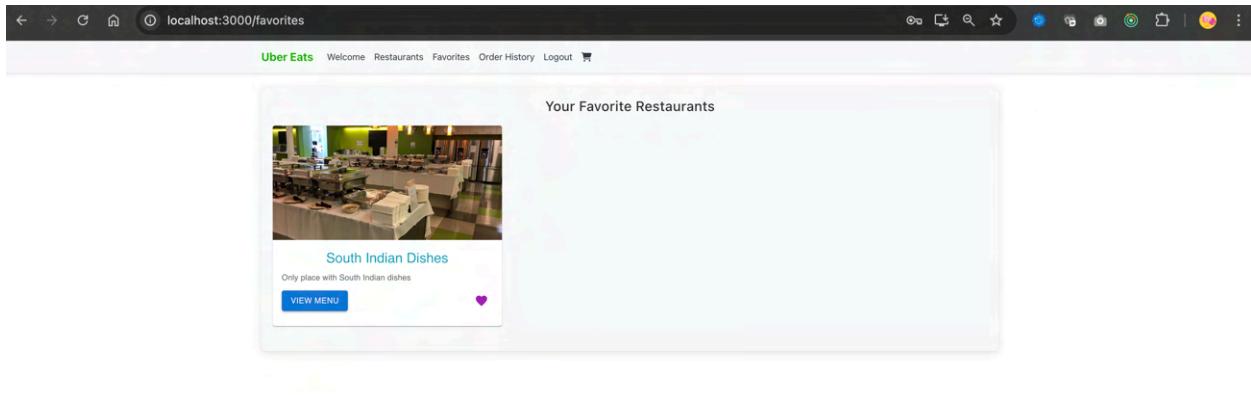
### Simple and Natural Breakfast Cereals to Start Your Day

Special Offers  
Check out our exclusive offers!

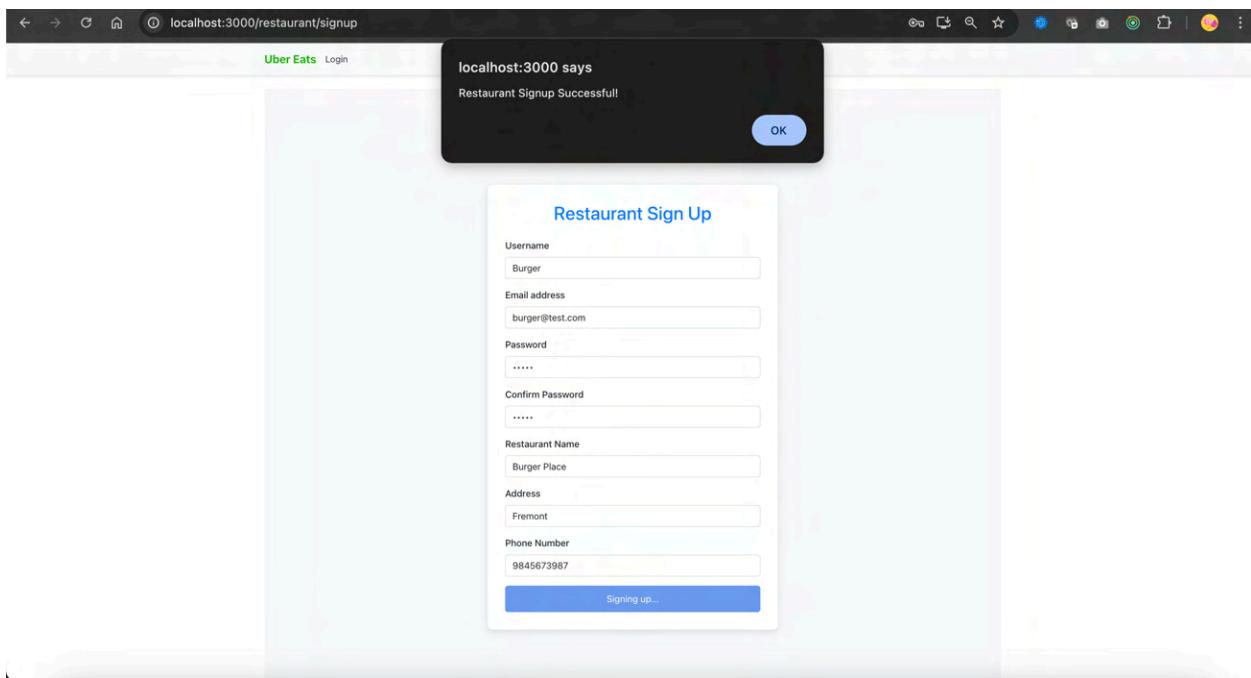


Available Restaurants

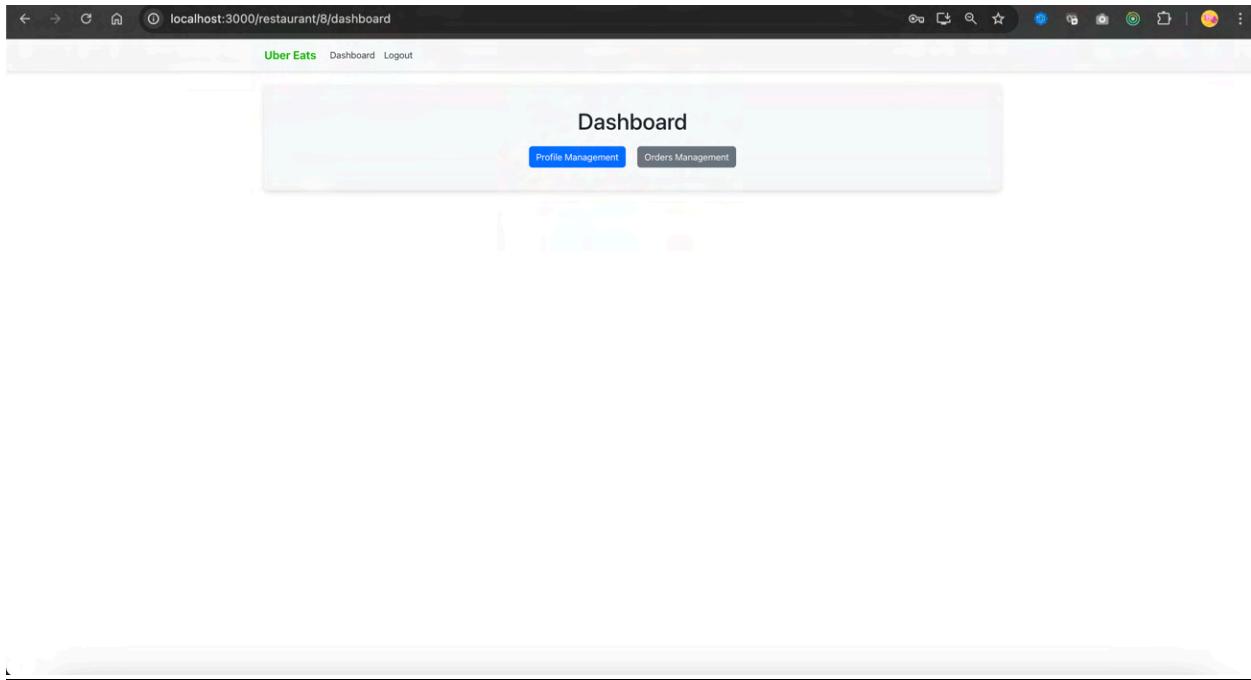
Image	Name	Description	Action
	South Indian Dishes	Only place with South Indian dishes	<a href="#">VIEW MENU</a>
	Mr.Biryani	Best biryanis available	<a href="#">VIEW MENU</a>
	Desserts Place	As sweet as sugar and you! Come visit	<a href="#">VIEW MENU</a>



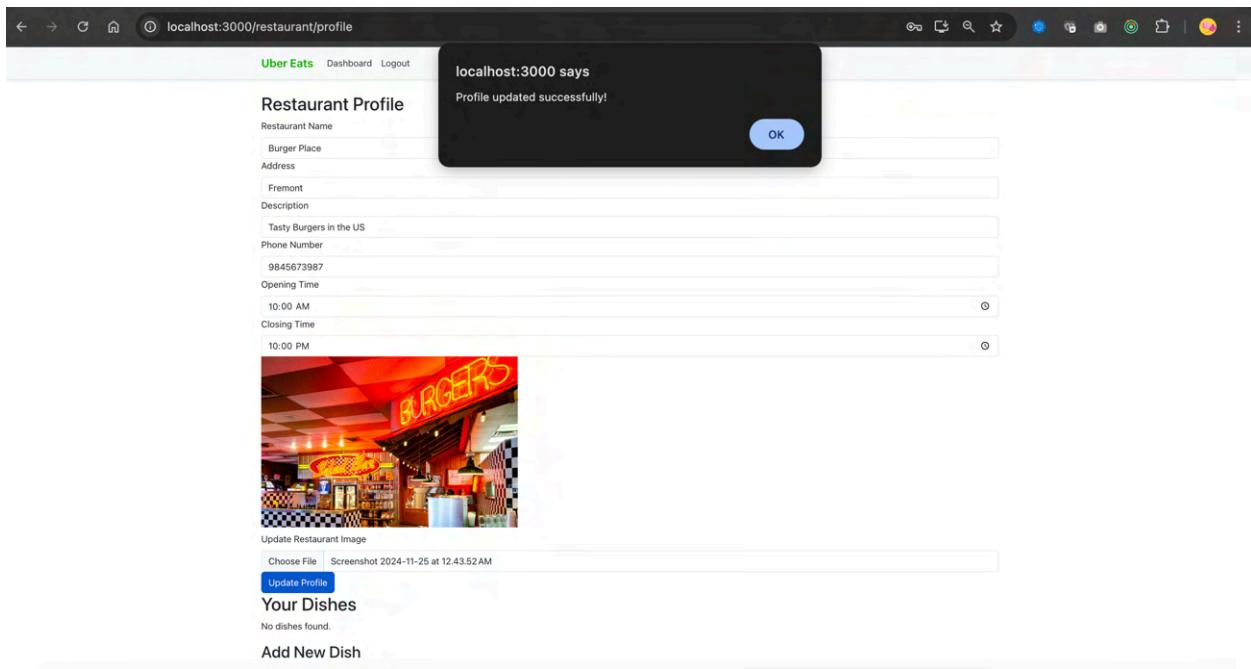
## Restaurant Signup:



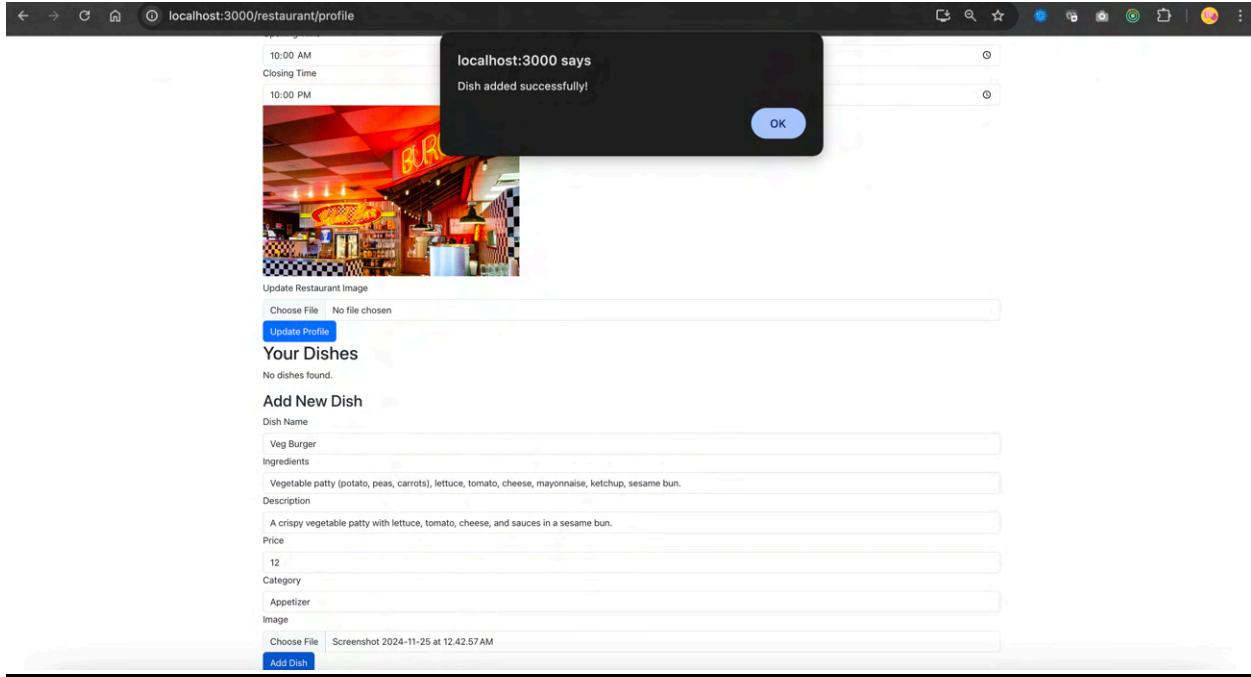
## Restaurant Dashboard:



## Restaurant Profile Update:

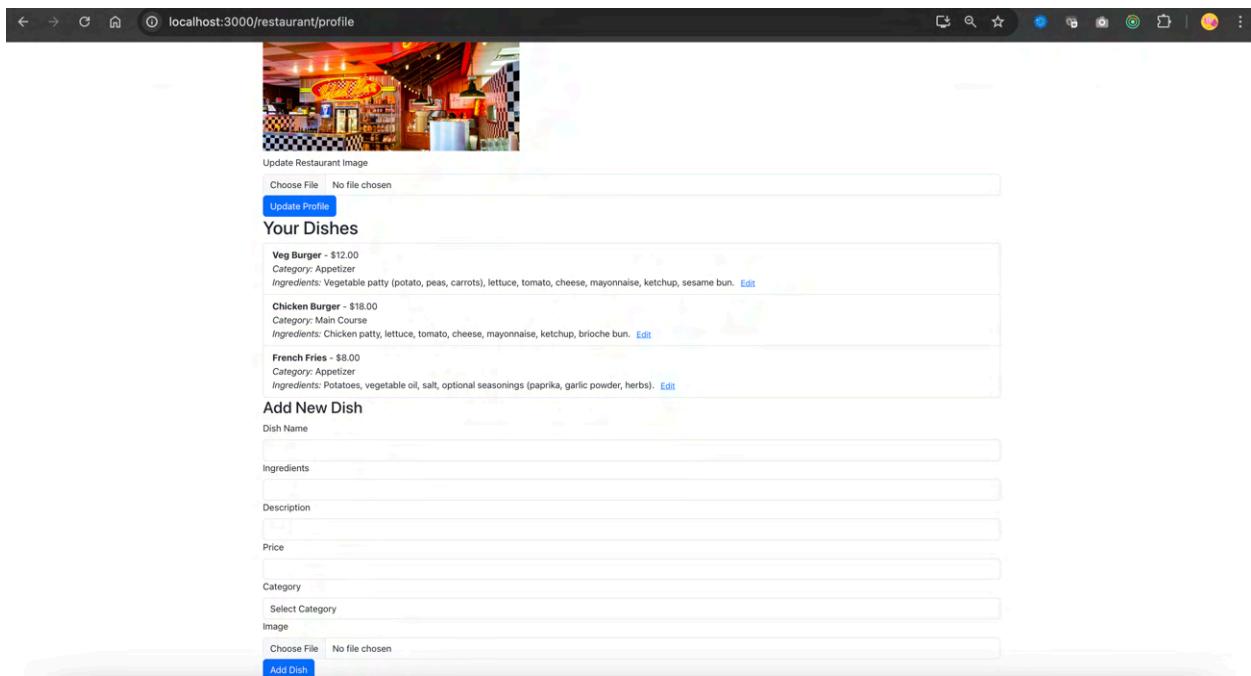


## Adding Dishes to the Restaurant:



A screenshot of a web browser window titled "localhost:3000/restaurant/profile". A modal dialog box in the center says "localhost:3000 says Dish added successfully!" with an "OK" button. The background shows a restaurant interior with a neon sign that says "BURGERS". Below the modal, there's a form for adding a new dish. The "Dish Name" field contains "Veg Burger". The "Ingredients" field contains "Vegetable patty (potato, peas, carrots), lettuce, tomato, cheese, mayonnaise, ketchup, sesame bun.". The "Description" field contains "A crispy vegetable patty with lettuce, tomato, cheese, and sauces in a sesame bun.". The "Price" field contains "12". The "Category" dropdown is set to "Appetizer". The "Image" field has a placeholder "Choose File" and "No file chosen". At the bottom right of the form is a blue "Add Dish" button.

## Dishes list:



A screenshot of a web browser window titled "localhost:3000/restaurant/profile". The main content area displays a list of dishes. The first dish is "Veg Burger - \$12.00", categorized as an "Appetizer" with the note "Ingredients: Vegetable patty (potato, peas, carrots), lettuce, tomato, cheese, mayonnaise, ketchup, sesame bun." followed by an "Edit" link. The second dish is "Chicken Burger - \$18.00", categorized as a "Main Course" with the note "Ingredients: Chicken patty, lettuce, tomato, cheese, mayonnaise, ketchup, brioche bun." followed by an "Edit" link. The third dish is "French Fries - \$8.00", categorized as an "Appetizer" with the note "Ingredients: Potatoes, vegetable oil, salt, optional seasonings (paprika, garlic powder, herbs)." followed by an "Edit" link. Below the list is a form for adding a new dish. It includes fields for "Dish Name" (placeholder "Veg Burger"), "Ingredients" (placeholder "Vegetable patty (potato, peas, carrots), lettuce, tomato, cheese, mayonnaise, ketchup, sesame bun."), "Description" (placeholder "A crispy vegetable patty with lettuce, tomato, cheese, and sauces in a sesame bun."), "Price" (placeholder "12"), "Category" (dropdown menu), "Image" (placeholder "Choose File" and "No file chosen"), and a blue "Add Dish" button at the bottom right.

## Updating a dish (amount is updated in this case) :

localhost:3000/dish/edit/12

Uber Eats Dashboard Logout

Dish Details

Name: Veg Burger

Category: Appetizer

Update Dish Image

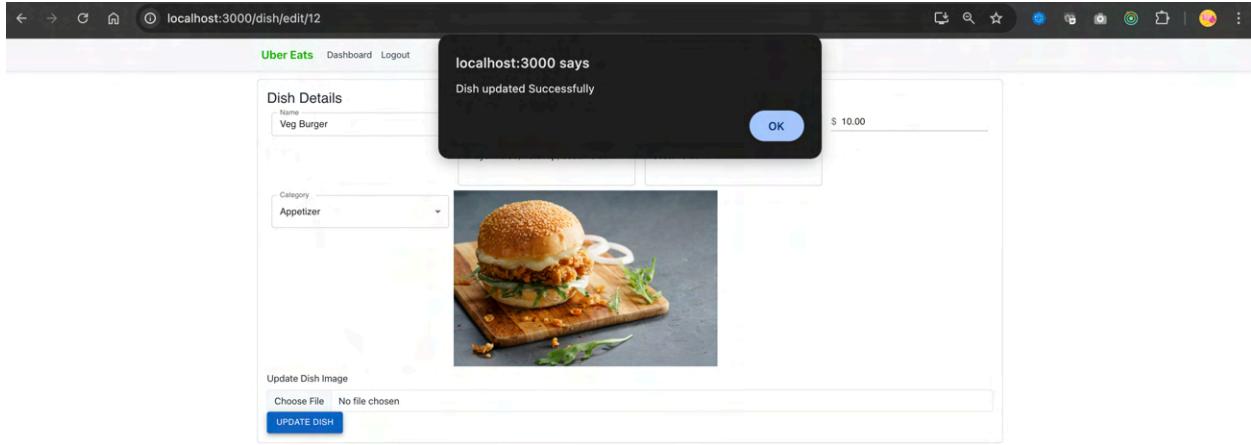
Choose File No file chosen

UPDATE DISH

localhost:3000 says  
Dish updated Successfully

OK

\$ 10.00



localhost:3000/restaurant/profile

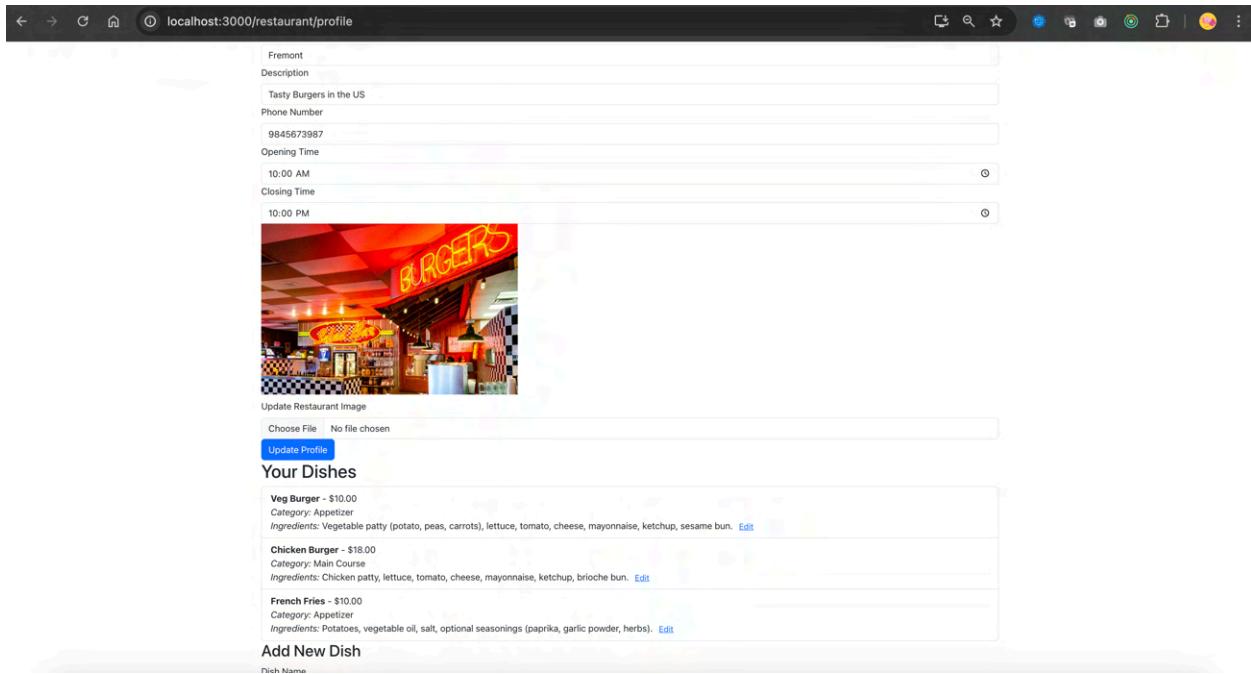
Fremont

Description: Tasty Burgers in the US

Phone Number: 9845673987

Opening Time: 10:00 AM

Closing Time: 10:00 PM



Update Restaurant Image

Choose File No file chosen

Update Profile

Your Dishes

Veg Burger - \$10.00  
Category: Appetizer  
Ingredients: Vegetable patty (potato, peas, carrots), lettuce, tomato, cheese, mayonnaise, ketchup, sesame bun. [Edit](#)

Chicken Burger - \$18.00  
Category: Main Course  
Ingredients: Chicken patty, lettuce, tomato, cheese, mayonnaise, ketchup, brioche bun. [Edit](#)

French Fries - \$10.00  
Category: Appetizer  
Ingredients: Potatoes, vegetable oil, salt, optional seasonings (paprika, garlic powder, herbs). [Edit](#)

Add New Dish

Pick Name:

## User view with the newly added restaurant (Burger Place):

The screenshot shows the Uber Eats homepage with a banner for 24/7 delivery and up to 50% savings. Below the banner, there are sections for 'Available Restaurants' and a featured restaurant, 'Burger Place'. The 'Burger Place' section includes a thumbnail image of the restaurant's interior, its name, a brief description ('Tasty Burgers in the US'), and a 'VIEW MENU' button.

## User view of dishes (restaurant – burger place):

The screenshot shows the detailed menu page for 'Burger Place'. It displays the restaurant's address (Fremont), description (Tasty Burgers in the US), phone number (9845673987), opening and closing times (10:00 AM - 10:00 PM), and the number of items in the cart (0). The menu section is titled 'Menu' and lists three items: 'Veg Burger' (Appetizer), 'Chicken Burger' (Main Course), and 'French Fries' (Appetizer). Each item has a thumbnail image, a brief description, the price (\$10.00 or \$18.00), and an 'Add to Cart' button.

## Adding dish:

localhost:3000/restaurants/8

Uber Eats Welcome Restaurants Favorites Order Delivery Support Help

localhost:3000 says  
Dish added to cart successfully!

OK

Address: Fremont  
Description: Tasty Burgers in the US  
Phone: 9845673987  
Opening Time: 10:00 AM  
Closing Time: 10:00 PM  
Items in Cart: 1

Menu

  
**Veg Burger**  
Appetizer  
Vegetable patty (potato, peas, carrots), lettuce, tomato, cheese, mayonnaise, ketchup, sesame bun.  
A crispy vegetable patty with lettuce, tomato, cheese, and sauces in a sesame bun.  
**Price:** \$10.00  
**Add to Cart**

  
**Chicken Burger**  
Main Course  
Chicken patty, lettuce, tomato, cheese, mayonnaise, ketchup, brioche bun.  
A juicy chicken patty with lettuce, cheese, sauces, and optional pickles in a brioche bun.  
**Price:** \$18.00  
**Add to Cart**

  
**French Fries**  
Appetizer  
Potatoes, vegetable oil, salt, optional seasonings (paprika, garlic powder, herbs).  
Crispy golden potato sticks, lightly salted and fried to perfection, with optional seasonings for extra flavor.  
**Price:** \$10.00  
**Add to Cart**

## Alert when customer tries to order from different restaurant:

localhost:3000/restaurants/7

Uber Eats Welcome Restaurants Favorites Order Delivery Support Help

Create New Order?

Your order contains items from Burger Place. Create a new order to add items from Desserts Place.

Cancel New Order

Address: Sunol  
Description: As sweet as sugar and you! Come visit  
Phone: 6479800987  
Opening Time: 11:00 AM  
Closing Time: 08:00 PM  
Items in Cart: 1

Menu

  
**Gulab Jamun**  
Dessert  
Milk Powder, All-purpose Flour, Ghee, Milk, Sugar Syrup flavored with Cardamom and Rose Water.  
Soft and spongy milk dumplings soaked in a sweet and fragrant sugar syrup.  
**Price:** \$5.00  
**Add to Cart**

  
**Rasmalai**  
Dessert  
Paneer (Indian Cottage Cheese), Milk, Saffron, Sugar, Cardamom.  
A rich dessert of soft paneer dumplings soaked in flavored, thickened milk.  
**Price:** \$17.00  
**Add to Cart**

  
**Carrot Halwa**  
Dessert  
Grated Carrots, Milk, Sugar, Ghee, Cardamom, Nuts (Cashews, Almonds)  
A traditional Indian dessert made by slow-cooking carrots in milk and sugar, topped with nuts.  
**Price:** \$10.00  
**Add to Cart**

localhost:3000/restaurants/7

Uber Eats Welcome Restaurants Favorites Order History Logout

localhost:300 says  
Dish added to cart successfully!

OK

Address: Sunol  
Description: As sweet as sugar and you! Come visit!  
Phone: 6479800987  
Opening Time: 11:00 AM  
Closing Time: 08:00 PM  
Items in Cart: 2

Menu

  
**Gulab Jamun**  
Dessert  
Milk Powder, All-purpose Flour, Ghee, Milk, Sugar Syrup flavored with Cardamom and Rose Water  
Soft and spongy milk dumplings soaked in a sweet and fragrant sugar syrup.  
**Price:** \$5.00  
**Add to Cart**

  
**Rasmalai**  
Dessert  
Paneer (Indian Cottage Cheese), Milk, Saffron, Sugar, Cardamom  
A rich dessert of soft paneer dumplings soaked in flavored, thickened milk.  
**Price:** \$17.00  
**Add to Cart**

  
**Carrot Halwa**  
Dessert  
Grated Carrots, Milk, Sugar, Ghee, Cardamom, Nuts (Cashews, Almonds)  
A traditional Indian dessert made by slow-cooking carrots in milk and sugar, topped with nuts.  
**Price:** \$10.00  
**Add to Cart**

## User cart view:

localhost:3000/cart

Uber Eats Welcome Restaurants Favorites Order History Logout

Your Cart

**Burger Place**

**Veg Burger**  
Price: \$10

**Total: \$10.00**

**Proceed to Checkout**

**Desserts Place**

**Rasmalai**  
Price: \$17

**Total: \$17.00**

**Proceed to Checkout**

**Clear Cart**

## Order Placement view:

The screenshot shows the 'Order Placement' view on a web browser. At the top, there's a navigation bar with links for 'Uber Eats', 'Welcome', 'Restaurants', 'Favorites', 'Order History', 'Logout', and a shopping cart icon. Below the navigation is a title 'Order Placement'. The main content area is divided into two sections: 'Cart Items' and 'Select Delivery Address'.  
**Cart Items:** This section displays a single item from 'Burger Place': 'Veg Burger - Quantity: 1'. It also shows the 'Total Price: \$10.00'.  
**Select Delivery Address:** This section shows a dropdown menu with the following address details:

- Milpitas
- San Jose
- CA
- 95035
- United States

A checked checkbox indicates 'Set as default address'. A blue button labeled 'Add Address' is visible at the bottom of the dropdown.  
At the bottom of the page is a green 'Place Order' button.

This screenshot shows the same 'Order Placement' view as above, but with a prominent black modal dialog box in the center. The dialog contains the text 'localhost:3000 says' and 'Order placed successfully!' followed by an 'OK' button. The background of the page is dimmed.  
The 'Select Delivery Address' section now shows the selected address: 'Milpitas, San Jose, CA, 95035, United States'. The 'Add New Address' link is still present.  
The green 'Place Order' button remains at the bottom of the page.

The screenshot shows a web browser window for 'Uber Eats' at the URL 'localhost:3000/cart'. The page title is 'Your Cart'. A single item, 'Rasmalai' from 'Desserts Place', is listed with a price of \$17.00. The quantity is set to 1. Below the item, the total is \$17.00 and the default delivery address is listed as Milpitas, San Jose, CA. There are buttons for 'Proceed to Checkout' and 'Clear Cart'.

## User's order history page:

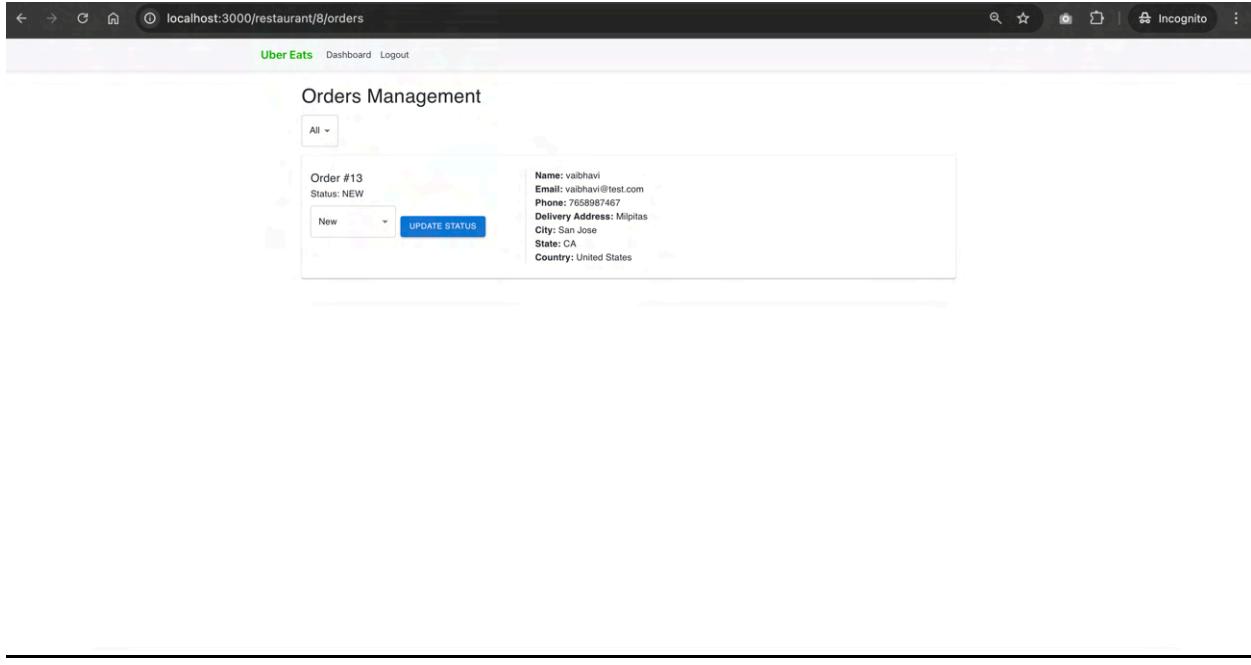
The screenshot shows a web browser window for 'Uber Eats' at the URL 'localhost:3000/order-history'. The page title is 'Order History'. Two recent orders are listed: Order #13 (Status: new, Total: \$10.00, Restaurant: Burger Place) and Order #14 (Status: new, Total: \$17.00, Restaurant: Desserts Place). Each order entry includes a summary of the order details and a list of items ordered.

Order #	Status	Total	Restaurant
Order #13	new	\$10.00	Burger Place
Order #14	new	\$17.00	Desserts Place

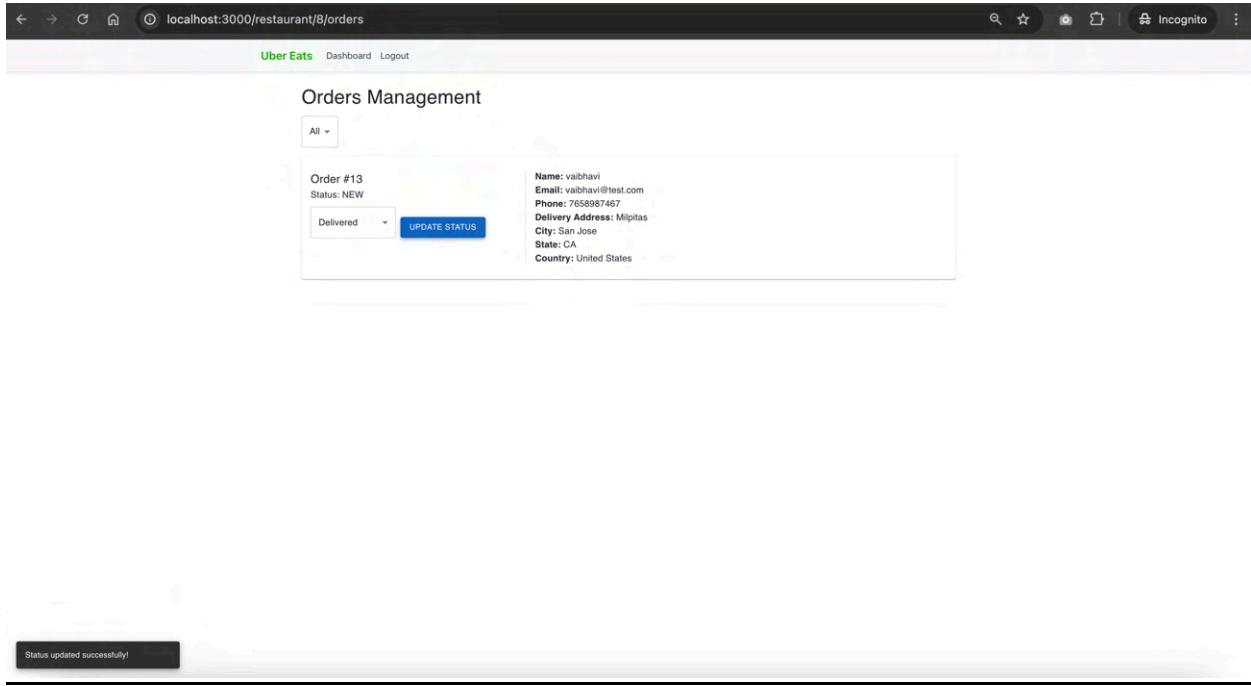
**Items:**

- Veg Burger - Quantity: 1
- Rasmalai - Quantity: 1

## Restaurant order management view:



## Status update at Restaurant end:



## Status change at User end:

The screenshot shows the 'Order History' section of the Uber Eats website. It displays two orders:

- Order #13**: Status: delivered, Total: \$10.00, Restaurant: Burger Place. Items: Veg Burger - Quantity: 1.
- Order #14**: Status: new, Total: \$17.00, Restaurant: Desserts Place. Items: Rasmalai - Quantity: 1.

## Status update at restaurant end (second dish):

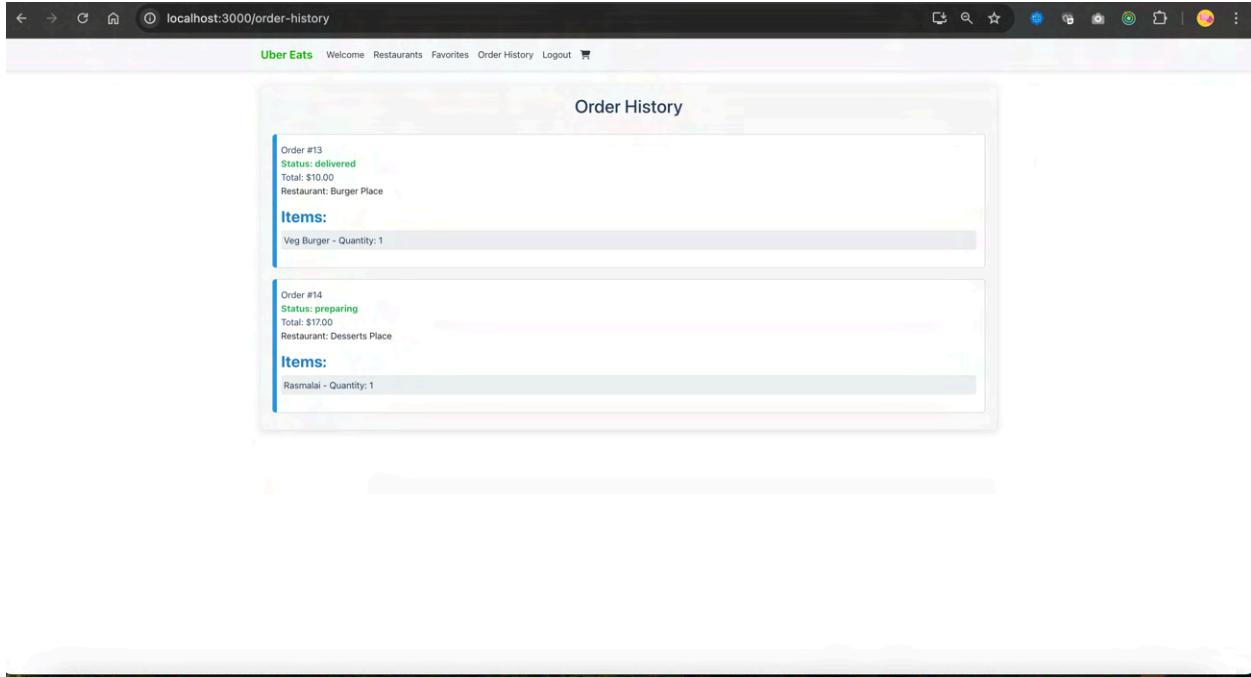
The screenshot shows the 'Orders Management' section of the Uber Eats website. It displays a single order:

**Order #14**  
Status: NEW  
Preparing

**Customer Details:**  
Name: vaibhavi  
Email: vaibhavi@test.com  
Phone: 7658987467  
Delivery Address: Milpitas  
City: San Jose  
State: CA  
Country: United States

A dark blue banner at the bottom left of the page says "Status updated successfully!"

## Status update at user end:



## Sample MongoDB related snips:

A screenshot of the Visual Studio Code (VS Code) interface. The left sidebar shows a file tree with projects like 'UBEREATSLAB2', 'uber\_eats\_backend', 'uber\_eats\_frontend', and 'src'. The 'src' folder contains files like '.gitignore', 'package-lock.json', 'package.json', 'README.md', 'venv', and 'public'. The main area shows a terminal window with the following MongoDB query and its results:

```
problems output debug console terminal ports
restaurants_restaurant
uber_eats> db.restaurants.restaurant.find().pretty()
[
  {
    "_id": ObjectId("6743d319b1ef9238e4369a62"),
    "id": 1,
    "user_id": 20,
    "name": "South Indian Dishes",
    "description": "Only place with South Indian dishes",
    "address": "Milpitas",
    "phone_number": Decimal128("76536789876"),
    "image": "restaurant_images/1840382_1055547747880075_5254293131499478601_n_yFK50ae.jpg",
    "opening_time": ISODate("1900-01-01T09:00:00.000Z"),
    "closing_time": ISODate("1900-01-01T21:00:00.000Z")
  },
  {
    "_id": ObjectId("6743d431b1ef9238e4369a69"),
    "id": 6,
    "user_id": 21,
    "name": "Mr.Biryani's",
    "description": "Best biryanis available",
    "address": "Sunnyvale",
    "phone_number": Decimal128("7568376456"),
    "image": "restaurant_images/2024-10-17.jpg",
    "opening_time": ISODate("1900-01-01T12:00:00.000Z"),
    "closing_time": ISODate("1900-01-01T08:00:00.000Z")
  },
  {
    "_id": ObjectId("6743d574b1ef9238e4369a70"),
    "id": 7,
    "user_id": 22,
    "name": "Desserts Place",
    "description": "As sweet as sugar and you! Come visit ",
    "address": "Sunny",
    "phone_number": Decimal128("6479800987"),
    "image": "restaurant_images/tempting-indian-sweets-arranged-display-sale-city-sweet-shop-glass-sh_d80wUh.webp",
    "opening_time": ISODate("1900-01-01T11:00:00.000Z"),
    "closing_time": ISODate("1900-01-01T20:00:00.000Z")
  },
  {
    "_id": ObjectId("674437b294b6140a862b03e0"),
    "id": 8,
    "user_id": 31,
    "name": "Burger Place"
  }
]
```

The right sidebar shows a 'Backend' section with 'Mongo' and 'Frontend' sections. The bottom status bar indicates 'main\*' is the active tab.

A screenshot of a dark-themed code editor, likely Visual Studio Code, displaying a MongoDB query result. The left sidebar shows a file tree with files like requirements.txt, .vscode, node\_modules, public, src, .gitignore, package-lock.json, README.md, venv, .gitignore, package-lock.json, and README.md. The main pane shows a list of delivery addresses with their details:

```
uber_eats> db.customers_deliveryaddress.find().pretty()
[ {
    "_id": ObjectId('6743da5cb1ef9238e4369a79'),
    "id": 6,
    "customer_id": 3,
    "address_line1": 'Krishnari Plaza',
    "city": 'Hyderabad',
    "state": 'Telangana',
    "postal_code": '500055',
    "country": 'India',
    "is_default": true
},
{
    "_id": ObjectId('6743ea45f7add3f8bdc09b25'),
    "id": 7,
    "customer_id": 27,
    "address_line1": 'Milpitas',
    "city": 'San Jose',
    "state": 'CA',
    "postal_code": '95035',
    "country": 'United States',
    "is_default": true
},
{
    "_id": ObjectId('67443ddc94b6140a862b03e8'),
    "id": 8,
    "customer_id": 30,
    "address_line1": 'Milpitas',
    "city": 'San Jose',
    "state": 'CA',
    "postal_code": '95035',
    "country": 'United States',
    "is_default": true
}]
```

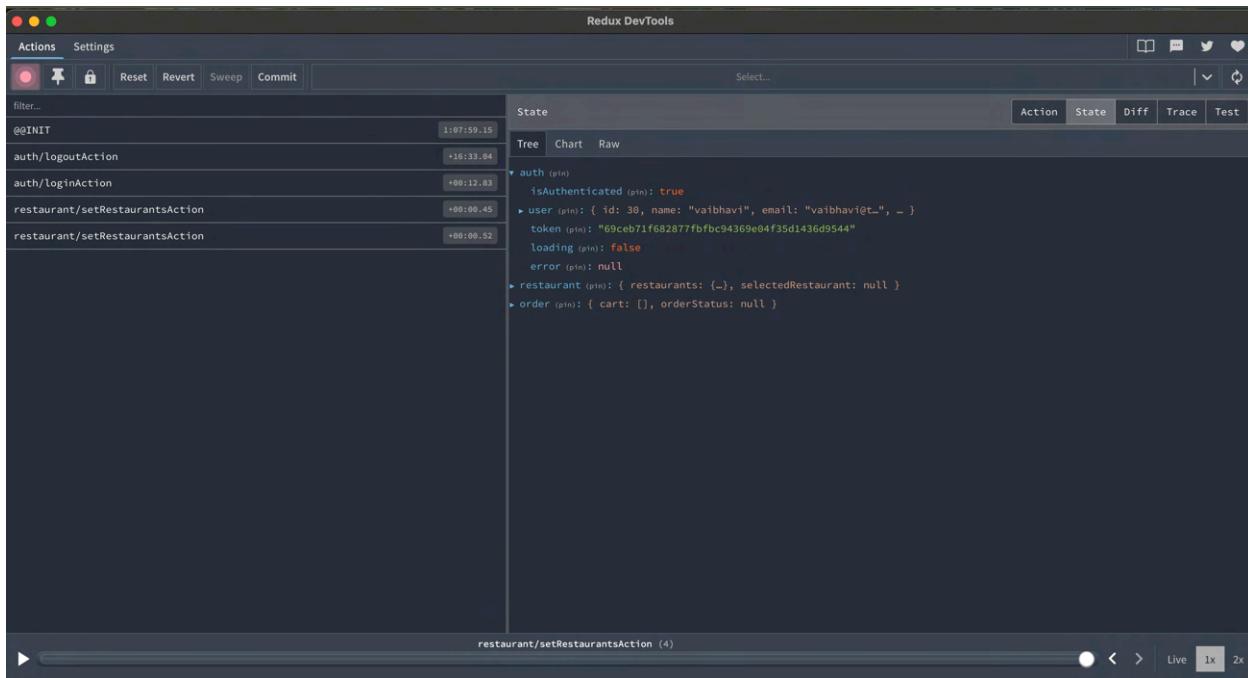
A screenshot of a dark-themed code editor showing a terminal tab with a MongoDB query result. The left sidebar is identical to the first screenshot. The terminal tab displays the same list of delivery addresses. A right sidebar shows project navigation with sections for Backend, Mongo, and Frontend uber-eats-frontend.

```
uber_eats> db.customers_deliveryaddress.find().pretty()
[ {
    "_id": ObjectId('6743da5cb1ef9238e4369a79'),
    "id": 6,
    "customer_id": 3,
    "address_line1": 'Krishnari Plaza',
    "city": 'Hyderabad',
    "state": 'Telangana',
    "postal_code": '500055',
    "country": 'India',
    "is_default": true
},
{
    "_id": ObjectId('6743ea45f7add3f8bdc09b25'),
    "id": 7,
    "customer_id": 27,
    "address_line1": 'Milpitas',
    "city": 'San Jose',
    "state": 'CA',
    "postal_code": '95035',
    "country": 'United States',
    "is_default": true
},
{
    "_id": ObjectId('67443ddc94b6140a862b03e8'),
    "id": 8,
    "customer_id": 30,
    "address_line1": 'Milpitas',
    "city": 'San Jose',
    "state": 'CA',
    "postal_code": '95035',
    "country": 'United States',
    "is_default": true
}]
```

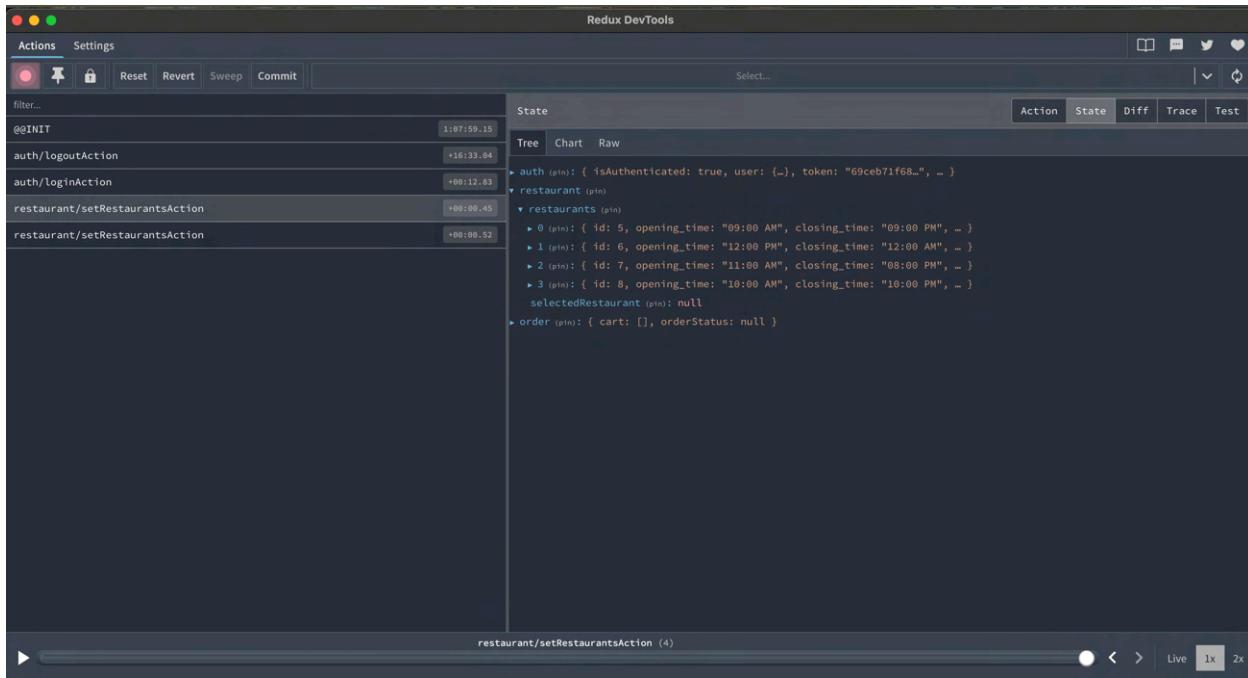
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[{"_id": ObjectId('6743da5db1ef9238e4369a7a'), "id": 8, "customer_id": 3, "restaurant_id": 5, "status": "picked_up", "total_price": Decimal128('10.00'), "created_at": ISODate('2024-11-25T02:01:01.294Z'), "delivery_address_id": 6}, {"_id": ObjectId('6743dd14557c687a179c7110'), "id": 9, "customer_id": 3, "restaurant_id": 5, "status": "new", "total_price": Decimal128('12.00'), "created_at": ISODate('2024-11-25T02:12:36.352Z'), "delivery_address_id": 6}, {"_id": ObjectId('6743ea47f7add3f8bcd09b26'), "id": 10, "customer_id": 27, "restaurant_id": 5, "status": "new", "total_price": Decimal128('10.00'), "created_at": ISODate('2024-11-25T03:08:55.362Z'), "delivery_address_id": 7}, {"_id": ObjectId('6743ee4ad50e059a40196420'), "id": 11, "customer_id": 3, "restaurant_id": 5, "status": "new", "total_price": Decimal128('10.00'), "created_at": ISODate('2024-11-25T03:26:02.547Z'), "delivery_address_id": 6}],
```

Outline Timeline Go Live

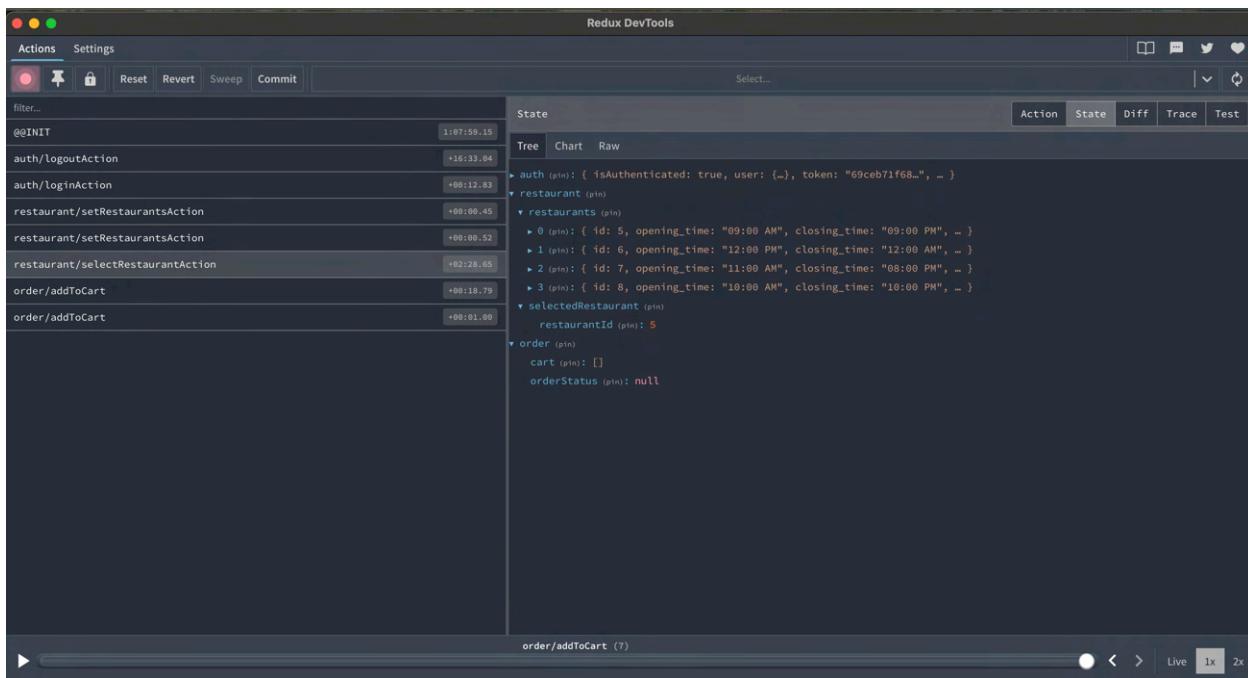
## Sample Redux state change images:



## Restaurant list:



## Selected Restaurant:



## Add to cart:

The screenshot shows the Redux DevTools interface with the "Actions" tab selected on the left. A list of actions is shown with their timestamps and types. On the right, the "State" tab is selected, displaying the current application state as a tree structure.

**Actions (7)**

- @@INIT 1:07:59.15
- auth/logoutAction +16:33.04
- auth/loginAction +00:12.83
- restaurant/setRestaurantsAction +00:00.45
- restaurant/setRestaurantsAction +00:00.52
- restaurant/selectRestaurantAction +02:28.65
- order/addToCart +00:18.79
- order/addToCart +00:01.00

**State**

Tree | Chart | Raw

```
auth (p1): { isAuthenticated: true, user: {}, token: "69ceb71f68...", ... }
restaurant (p1)
  restaurants (p1)
    0 (p1): { id: 5, opening_time: "09:00 AM", closing_time: "09:00 PM", ... }
    1 (p1): { id: 6, opening_time: "12:00 PM", closing_time: "12:00 AM", ... }
    2 (p1): { id: 7, opening_time: "11:00 AM", closing_time: "08:00 PM", ... }
    3 (p1): { id: 8, opening_time: "10:00 AM", closing_time: "10:00 PM", ... }
  selectedRestaurant (p1)
    restaurantId (p1): 5
order (p1)
  cart (p1)
    0 (p1): { id: 22, name: "Dosa", price: 10, ... }
  orderStatus (p1): null
```

order/addToCart (7)

Live 1x 2x