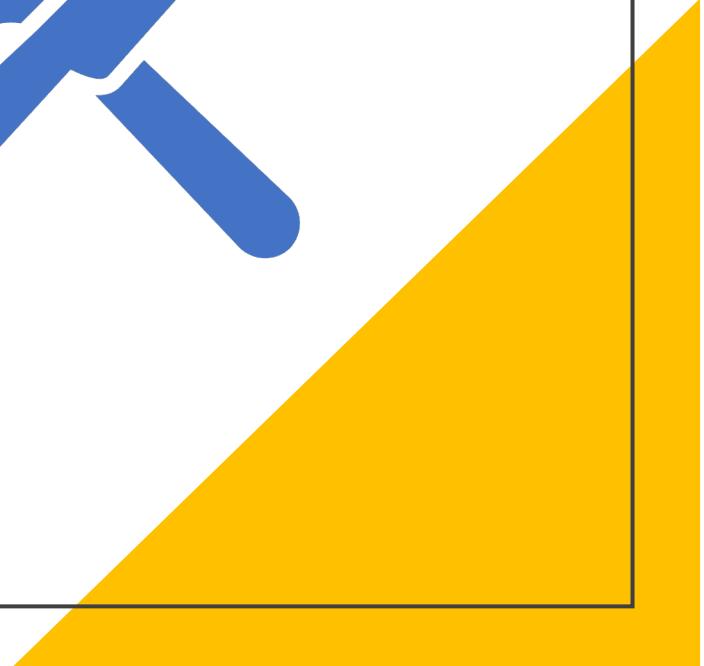


Grab and Eat

A Food Pickup App

Created by Mohammad Ali

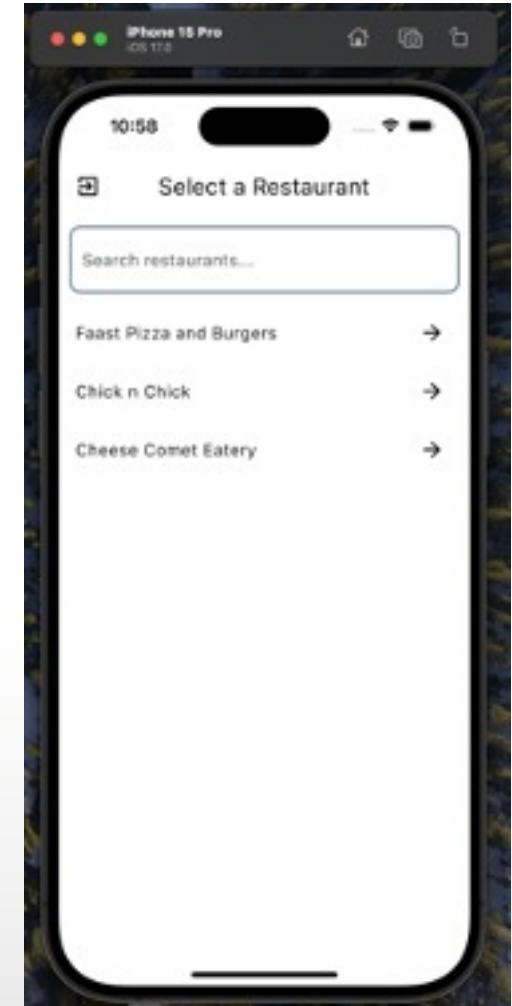
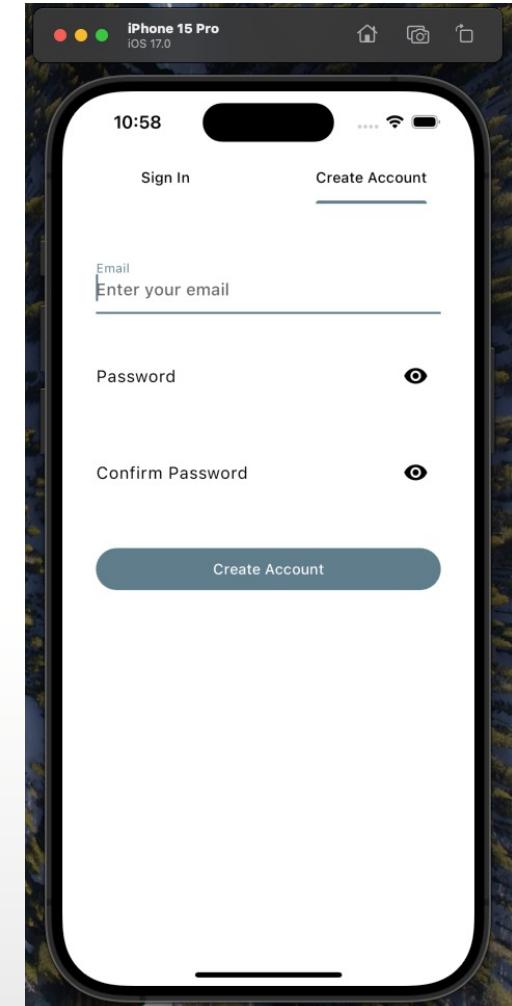
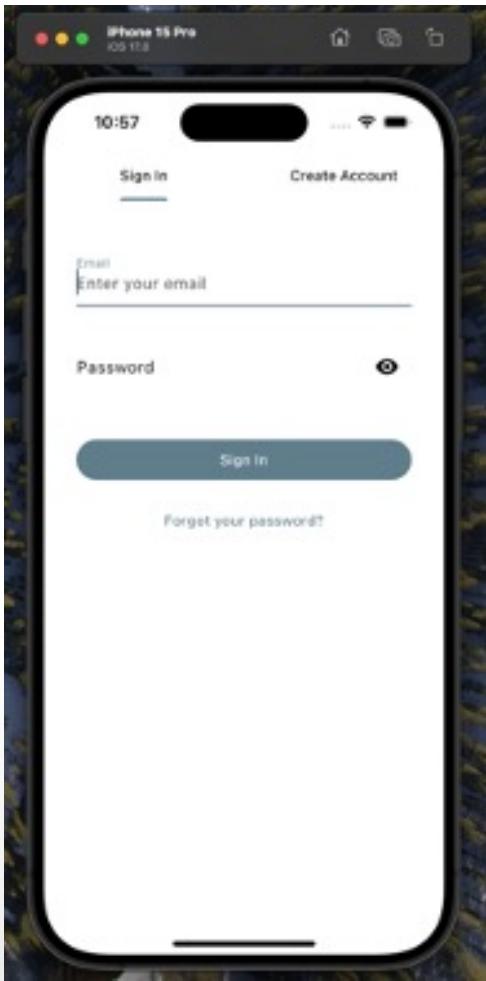


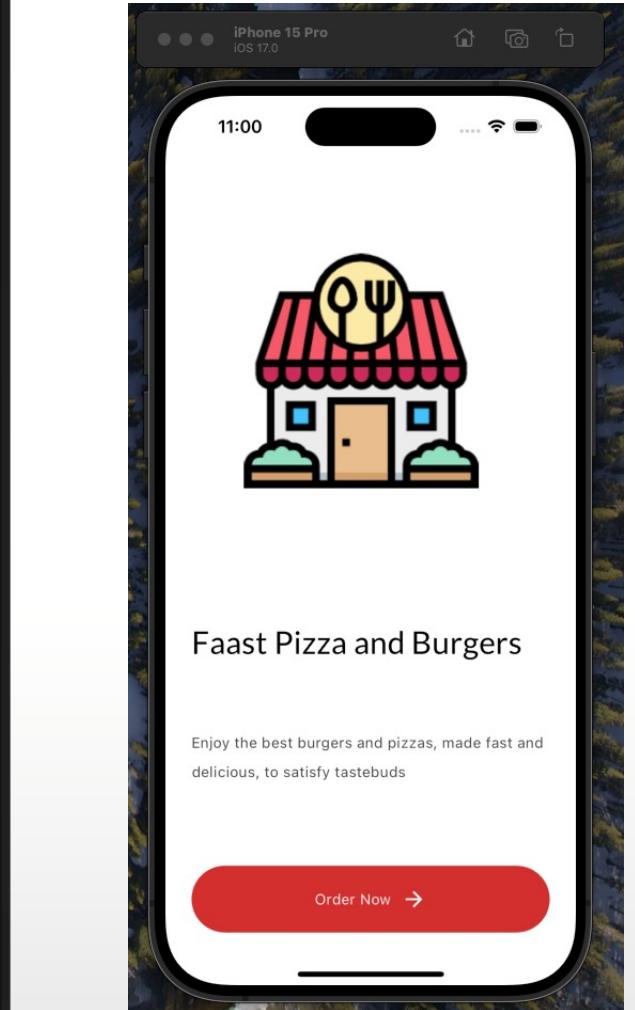
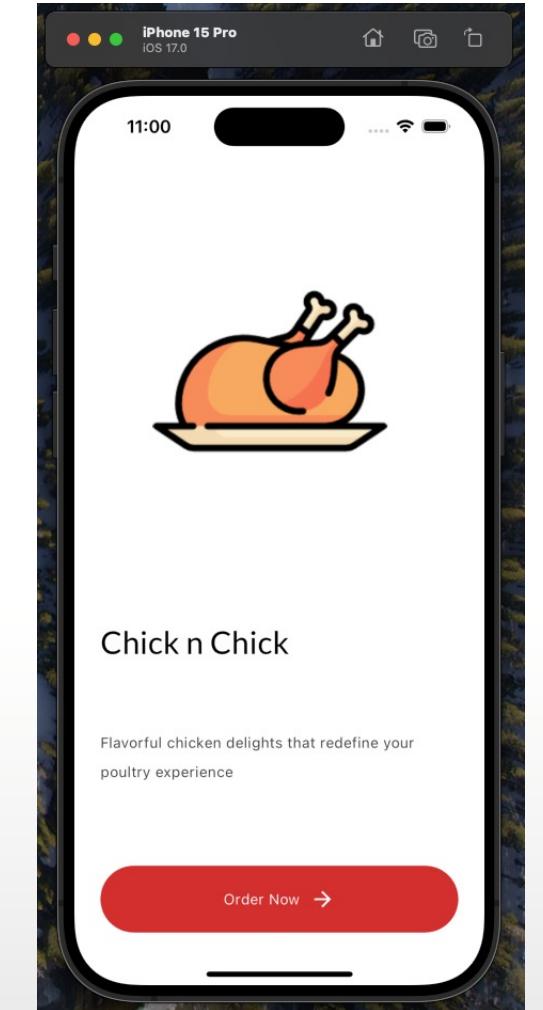
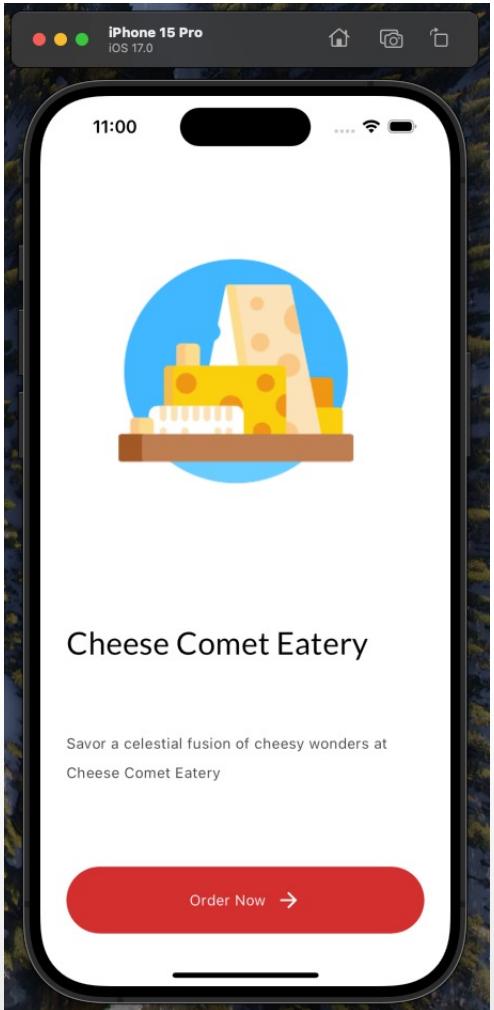
My Flutter Application

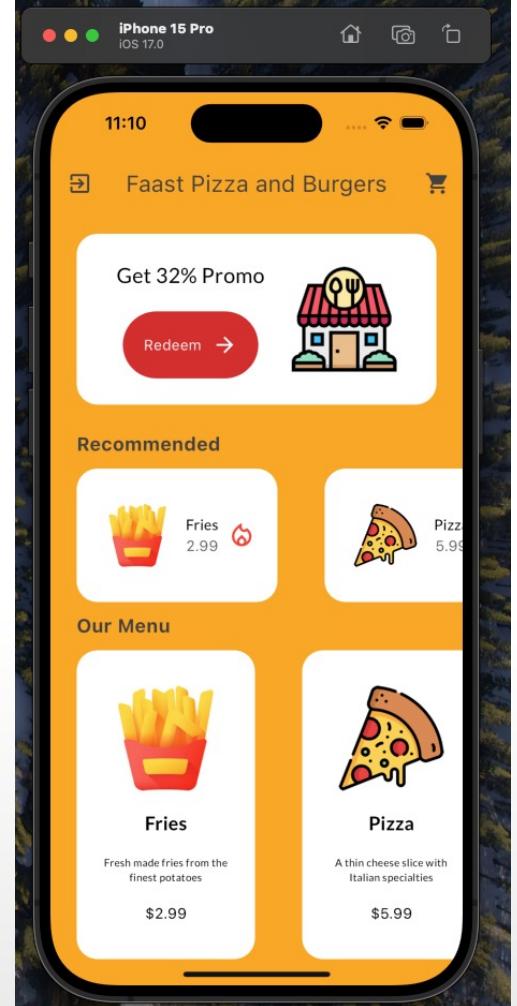
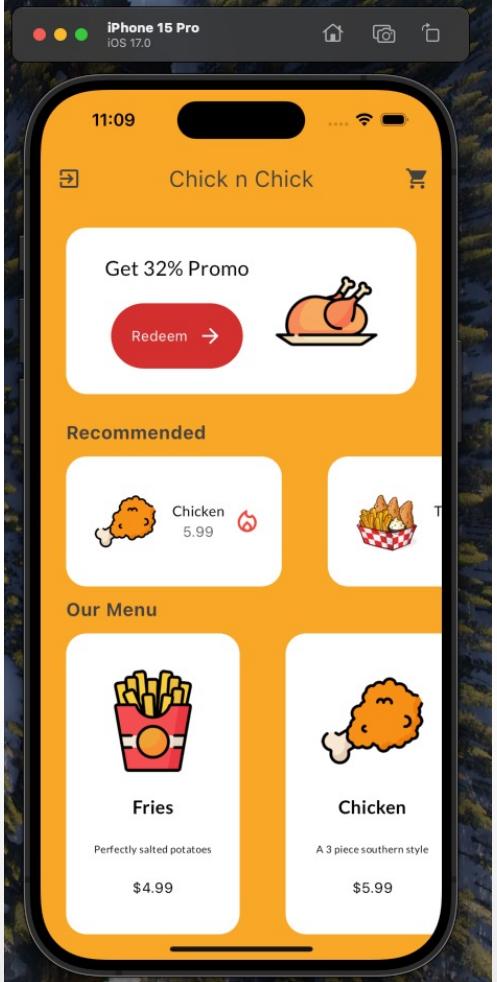
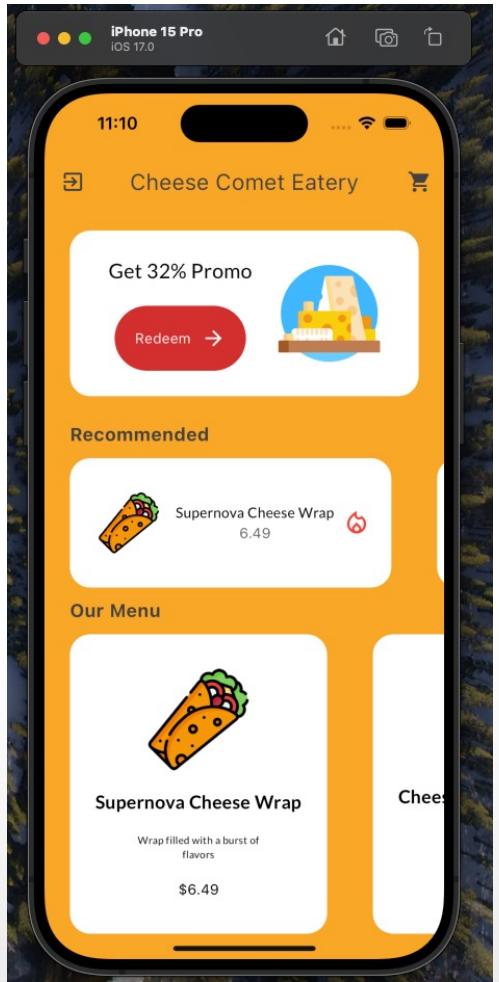
- Dart
- AWS Amplify Auth and GraphQL API for DynamoDB
- Purpose of App: Making an integrated simple/easy to use order pickup app that multiple restaurants can add to and users can use the same app for multiple restaurants.

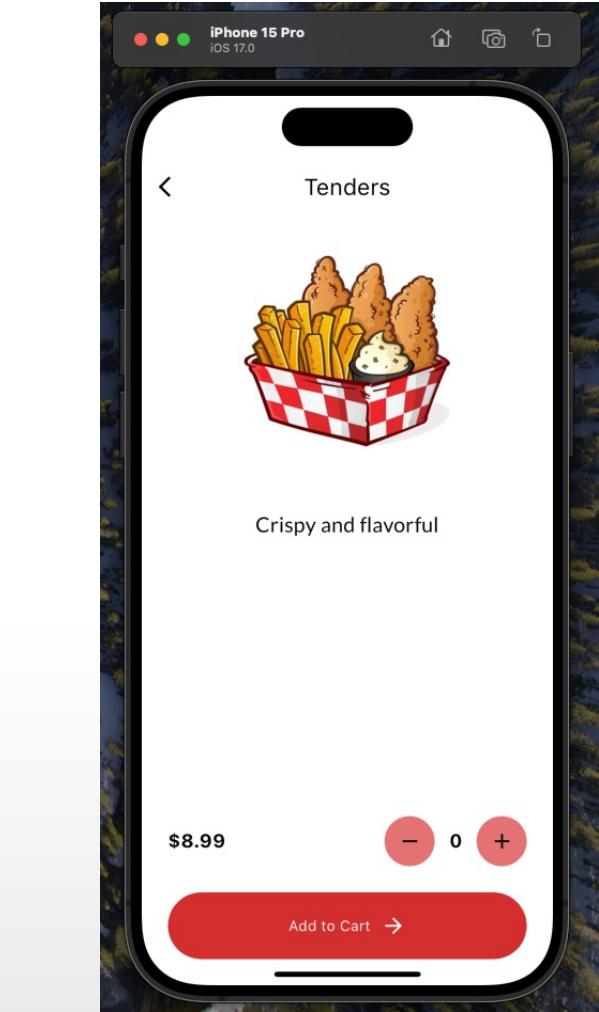
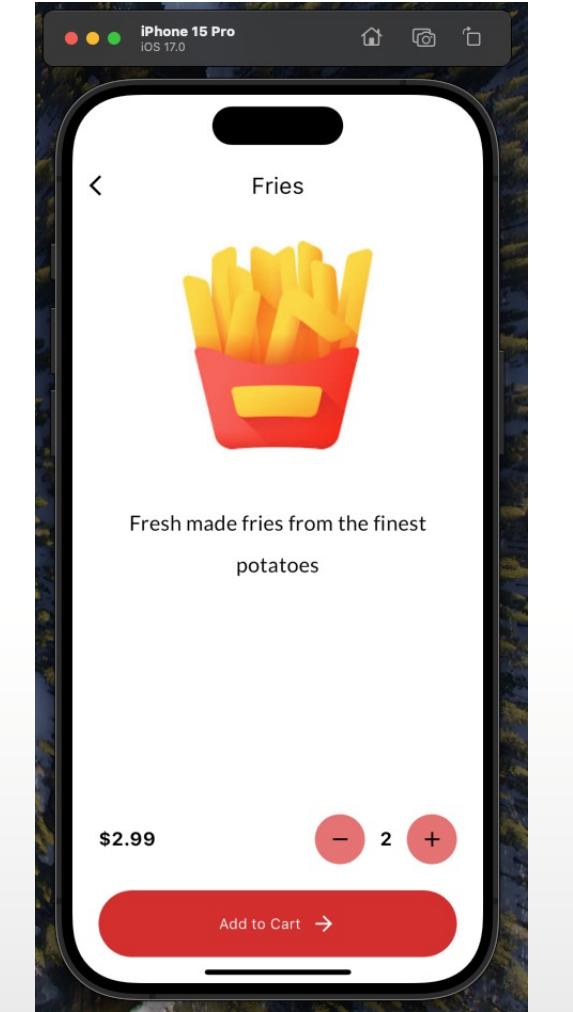
User Interface

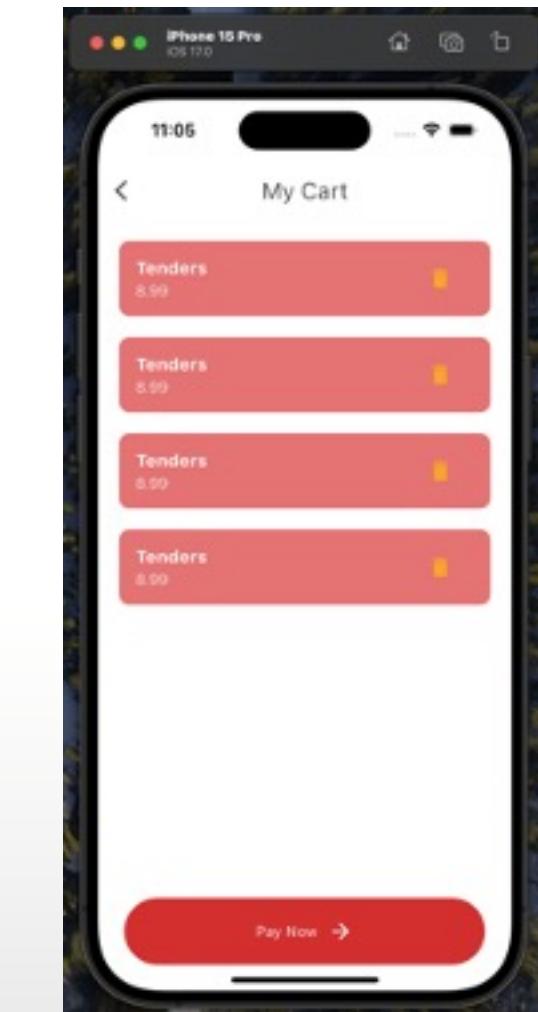
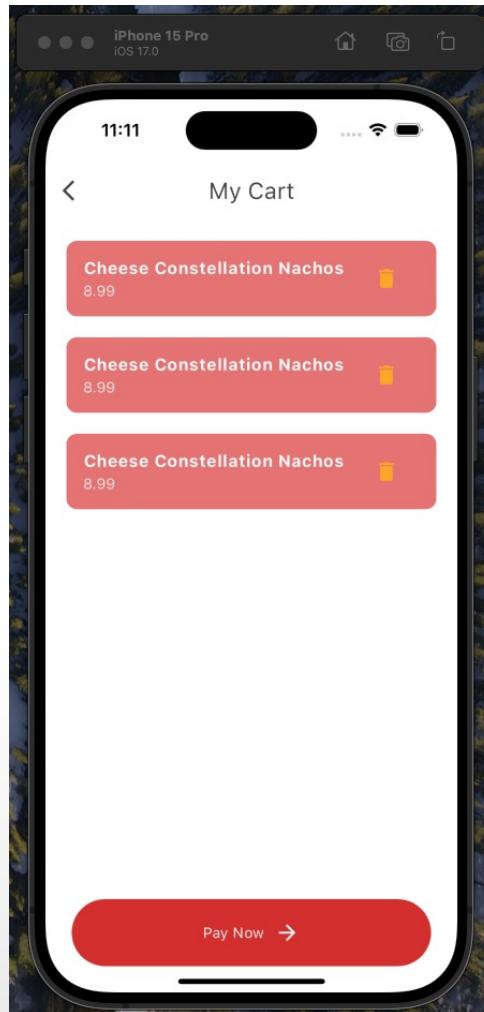
- Login Screen/Register Screen – Made using Amplify Auth; provides an interface for authenticating a user
- Restaurant Selection Screen – A list that has the current restaurant names fetched from the database, made using Amplify API/libraries. Allows for searching through the restaurants
- Main Screen: There is a promotion banner on top, a recommended list, and a menu list. Recommended list is a sub-list of the menu with a certain Boolean condition. Menu list fetches all the items associated to the selected restaurant and builds the list.
- Details/Cart Page: Details page allows for a more “in-depth” look into the item and in the future, things like screenshots, allergy notes, etc. could be shown here. The cart page simplifies has all the added items, allows for deletion, and places an order.











Data Manager

Visually manage your app content or run queries and mutations directly against the underlying GraphQL API.

[Manage content](#) [Test GraphQL API](#)

Select table [Restaurant](#)

Restaurant (3)

< 1 > |

[Actions](#) [Create Restaurant](#)

| <input type="checkbox"/> | id | name | address | shortDescription | mainImage | menu | preferences | createdAt | updatedAt |
|--------------------------|---------------------|------------------------|-------------------------|----------------------------------|-----------------------------------------------------------|---------------------------------|-----------------------------|---------------------------|---------------------------|
| <input type="checkbox"/> | 454a28e5-c4e7-4c... | Faast Pizza and Bur... | 123 Main Street, N... | Enjoy the best bur... | https://encrypted-... | [{"name": "Fries", "pr...} ...] | - | 2023-12-07T22:48... | 2023-12-07T23:50... |
| <input type="checkbox"/> | 969fb70a-6bcf-4a... | Chick n Chick | 456 Elm Avenue, B... | Flavorful chicken d... | https://encrypted-... | [{"name": "Fries", "pr...} ...] | - | 2023-12-07T22:51... | 2023-12-07T23:52... |
| <input type="checkbox"/> | 3fee5f04-0ae6-4d... | Cheese Comet Eatery | 789 Oak Lane, Que... | Savor a celestial fu... | https://encrypted-... | [{"name": "Superno...} ...] | - | 2023-12-07T22:55... | 2023-12-07T23:54... |

[Documentation](#)

[Support](#)

[Feedback](#)



 Home

Manage

 Content User management File browser

Design

 UI Library

Set up

 Data Authentication Storage Functions GraphQL API REST API Analytics Predictions Interactions Notifications

User management

Create, view, and manage users and groups for your application.

 Users Groups

Users

Actions ▾

Create user

 Find users by Email Email

Created Date

Status

 mal25@pride.hofstra.edu

December 7, 2023 6:05 PM

CONFIRMED

 wonewoh664@gearstag.com

December 7, 2023 7:18 PM

CONFIRMED

Analytics

 Documentation  Support  Feedback 

- Home
- Manage
- Content
- User management
- File browser
- Design
- UI Library
- Set up
 - Data
 - Authentication
 - Storage
 - Functions
 - GraphQL API
 - REST API
 - Analytics
 - Predictions
 - Interactions
 - Notifications
- Documentation 
- Support 
- Feedback 

Data modeling Deployed

[GraphQL API settings](#)[+ Add model](#)

Preference

| Field name | Type |
|----------------|--------|
| primaryColor | String |
| secondaryColor | String |
| bgColor | String |
| headingColor | String |
| textColor | String |
| subtitleColor | String |

[+ Add a field](#)

Item

| Field name | Type |
|------------------|----------|
| name | String! |
| price | String! |
| imagePath | String! |
| shortDescription | String! |
| isRecommended | Boolean! |

[+ Add a field](#)

Restaurant

| Field name | Type |
|------------------|------------|
| id | ID! |
| name | String! |
| address | String! |
| shortDescription | String! |
| mainImage | String! |
| menu | [Item]! |
| preferences | Preference |

[+ Add a field](#)[+ Add a relationship](#)

Inspector panel

Select a model, field or relationship to configure their properties and authorization rules.

[Learn more about data modeling](#)

File Structure and Schema

The image shows a code editor interface with several tabs at the top: 'schema.graphql' (active), 'main.dart', 'colors.dart', and 'details_page'. On the left, there's a file tree for a project named 'GRABANDEAT'.

File Tree:

- .idea
- .vscode
- amplify
- backend
 - api/grabandeat
 - build
 - resolvers
 - stacks
 - {} cli-inputs.json
 - {} parameters.json
 - schema.graphql
 - auth
 - types
 - {} backend-config.json
 - {} tags.json
 - {} cli.json
 - {} team-provider-info.json
- android
- assets
- build
- ios
- lib
 - component
 - button.dart
 - itemtile.dart
 - recommendedtile.dart
 - controller
 - restaurant_data.dart
 - model
 - current_restaurant.dart
 - shop.dart
 - models
 - Item.dart
 - ModelProvider.dart
 - Preference.dart
 - Restaurant.dart
 - theme
 - colors.dart
 - views
 - cart_page.dart
 - details_page.dart
 - intro_page.dart
 - main_page.dart
 - restaurant_selection_page.dart
 - search_page.dart
- amplifyconfiguration.dart
- main.dart
- .eslintignore

amplify > backend > api > grabandeat > schema.graphql

```
1 type Preference {  
2   primaryColor: String  
3   secondaryColor: String  
4   bgColor: String  
5   headingColor: String  
6   textColor: String  
7   subtitleColor: String  
8 }  
9  
10 type Item {  
11   name: String!  
12   price: String!  
13   imagePath: String!  
14   shortDescription: String!  
15   isRecommended: Boolean!  
16 }  
17  
18 type Restaurant @model @auth(rules: [{allow: public}]) {  
19   id: ID!  
20   name: String!  
21   address: String!  
22   shortDescription: String!  
23   mainImage: String!  
24   menu: [Item]!  
25   preferences: Preference  
26 }  
27
```

My Shop/Cart and Restaurant Model

```
lib > model > shop.dart > ...
1 import 'package:amplify_flutter/amplify_flutter.dart';
2 import 'package:flutter/material.dart';
3 import 'package:grabandeat/models/Item.dart';
4 import 'package:grabandeat/models/Restaurant.dart';
5
6 class Shop extends ChangeNotifier {
7   final List<Item> menu = [];
8
9   //fetch the Items
10 void updateMenu(List<Item> restaurantMenu) {
11   menu.clear();
12   menu.addAll(restaurantMenu);
13   notifyListeners();
14 }
15
16 Future<void> fetchItems(String restaurantName) async {
17   final restaurant = await Amplify.DataStore.query(
18     Restaurant.classType,
19     where: Restaurant.NAME.eq(restaurantName),
20   );
21   if (restaurant.isNotEmpty) {
22     final menu = restaurant.first.menu;
23     updateMenu(menu); // Update the menu in your Shop object
24     safePrint('Fetched menu for restaurant $restaurantName: $menu');
25     notifyListeners();
26   } else {
27     safePrint('No restaurant found with name: $restaurantName');
28   }
29 }
30
31 //customer cart
32 final List<Item> _cart = [];
33 //getter methods
34 List<Item> get cart => _cart;
35
36 //add to cart
37 void addToCart(Item food, int quantity) {
38   for (int i = 0; i < quantity; i++) {
39     _cart.add(food);
40   }
41   notifyListeners();
42 }
43
44 //remove from cart
45 void removeFromCart(Item food) {
46   _cart.remove(food);
47   notifyListeners();
48 }
49
50 void emptyCart() {
51   cart.clear();
52 }
53 }
```

```
shop.dart current_restaurant.dart X main_page.dart
lib > model > current_restaurant.dart > ...
1 import 'package:flutter/foundation.dart';
2
3 class RestaurantSelectionState extends ChangeNotifier {
4   String? _selectedRestaurant;
5
6   String? get selectedRestaurant => _selectedRestaurant;
7
8   void setSelectedRestaurant(String? restaurant) {
9     _selectedRestaurant = restaurant;
10    notifyListeners();
11  }
12 }
13 |
```

My Restaurant Controller

cart_page.dart

restaurant_data.dart ×

intro_page.dart

shop.dart

```
lib > controller > restaurant_data.dart > ...
1   import 'package:amplify_api/amplify_api.dart';
2   import 'package:amplify_flutter/amplify_flutter.dart';
3   import 'package:grabandeat/models/Restaurant.dart';
4
5   Future<List<Restaurant?>> queryListItems() async {
6       try {
7           final request = ModelQueries.list(Restaurant.classType);
8           final response = await Amplify.API.query(request: request).response;
9
10          if (response.data == null || response.data?.items == null) {
11              safePrint('errors: ${response.errors}');
12              return [];
13          }
14
15          final restaurantList = response.data!.items;
16          return restaurantList;
17      } on ApiException catch (e) {
18          safePrint('Query failed: $e');
19          return [];
20      }
21  }
22 }
```

The Components (MenuTile,
RecommendedTile, and a Button)

```
6 class RecommendedTile extends StatelessWidget {
7   const RecommendedTile({super.key, required this.item, required this.onTap});
8   final void Function()? onTap;
9   final Item item;
10
11   @override
12   Widget build(BuildContext context) {
13     return GestureDetector(
14       onTap: onTap,
15       child: Container(
16         decoration: BoxDecoration(
17           color: primaryColor,
18           borderRadius: BorderRadius.circular(20),
19         ), // BoxDecoration
20         margin: const EdgeInsets.only(
21           top: 10,
22           left: 25,
23           right: 20,
24           bottom: 10,
25         ), // EdgeInsets.only
26         padding: const EdgeInsets.all(20),
27         child: Row(
28           mainAxisAlignment: MainAxisAlignment.spaceBetween,
29           children: [
30             Padding(
31               padding: const EdgeInsets.all(8.0),
32               child: Column(
33                 mainAxisAlignment: MainAxisAlignment.center,
34                 children: [
35                   //image
36                   Image.network(
37                     item.imagePath,
38                     height: 60,
39                   ) // Image.network
40                 ],
41               ),
42             ), // Column
43             Padding(
44               padding: const EdgeInsets.all(8.0),
45               child: Column(
46                 mainAxisAlignment: MainAxisAlignment.center,
47                 children: [
48                   Row(
49                     children: [
50                     Text(
51                       item.name,
52                         overflow: TextOverflow.ellipsis,
53                         style: GoogleFonts.lato(fontSize: 14),
54                     ), // Text
55                   ],
56                 ),
57                 Row(
58                   children: [
59                     Text(
60                       item.price,
61                         style: TextStyle(color: Colors.grey[700]),
62                     ), // Text
63                   ],
64                 ),
65               ],
66             ), // Column
67             const Column(
68               mainAxisAlignment: MainAxisAlignment.center,
69               children: [
70               Icon(
71                 Icons.local_fire_department_outlined,
72                 color: Colors.red,
73                 size: 28,
74               ) // Icon
75             ],
76           ),
77         ],
78       ),
79     );
80   }
81 }
```

```
6 class ItemTile extends StatelessWidget {
7   const ItemTile({super.key, required this.item, required this.onTap});
8   final void Function()? onTap;
9   final Item item;
10
11   @override
12   Widget build(BuildContext context) {
13     return GestureDetector(
14       onTap: onTap,
15       child: Container(
16         decoration: BoxDecoration(
17           color: primaryColor,
18           borderRadius: BorderRadius.circular(20),
19         ), // BoxDecoration
20         margin: const EdgeInsets.only(
21           left: 25,
22           right: 20,
23         ), // EdgeInsets.only
24         padding: const EdgeInsets.all(25),
25         child: Column(
26           mainAxisAlignment: MainAxisAlignment.spaceAround,
27           crossAxisAlignment: CrossAxisAlignment.center,
28           children: [
29             Row(
30               children: [
31               //image
32               Image.network(
33                 item.imagePath,
34                 height: 100,
35               ) // Image.network
36             ],
37             Row(
38               children: [
39                 //name
40                 Text(
41                   item.name,
42                     style: GoogleFonts.lato(
43                         fontWeight: FontWeight.bold),
44                   ), // Text
45                 ],
46               ),
47               Row(
48                 children: [
49                   //description
50                   SizedBox(
51                     width: 120,
52                     child: Text(
53                       item.shortDescription,
54                         style: GoogleFonts.lato(
55                             fontWeight: FontWeight.normal),
56                         textAlign: TextAlign.center,
57                     ), // Text
58                   ), // SizedBox
59                 ],
60               ),
61               Row(
62                 children: [
63                   //price
64                   Text('\$' + item.price)
65                 ],
66               ),
67             ],
68           ],
69         ),
70       );
71   }
72 }
```

```
3 class MyButton extends StatelessWidget {
4   const MyButton({
5     super.key,
6     required this.text,
7     required this.onTap,
8   });
9   final String text;
10  final void Function()? onTap;
11
12  @override
13  Widget build(BuildContext context) {
14    return GestureDetector(
15      onTap: onTap,
16      child: Container(
17        decoration: BoxDecoration(
18          color: Colors.red[700],
19          borderRadius: BorderRadius.circular(40),
20        ), // BoxDecoration
21        padding: EdgeInsets.all(20),
22        child: Row(
23          mainAxisAlignment: MainAxisAlignment.center,
24          children: [
25            //Text for button
26            Text(
27              text,
28                style: TextStyle(color: Colors.white),
29            ), // Text
30            const SizedBox(
31              width: 10,
32            ), // SizedBox
33            //Icon for button
34            Icon(
35              Icons.arrow_forward,
36              color: Colors.white,
37            ), // Icon
38          ],
39        ),
40      );
41    }
42 }
```

The Login and Signup Page, also my Main.dart

Made using Amplify Auth UI; allowed me to add a few code snippets to get a fully functional authentication page that can be customized to have different input fields and MFA and other typical login/registration tools/features.

```
main.dart X colors.dart details_page.dart cart_page.dart intro.

lib > main.dart > ...
15 import 'amplifyconfiguration.dart';
16
17 Future<void> main() async {
18 WidgetsFlutterBinding.ensureInitialized();
19 runApp(
20 ChangeNotifierProvider(
21 create: (context) => Shop(),
22 child: const MyApp(),
23 ), // ChangeNotifierProvider
24 );
25 }
26
27 class MyApp extends StatefulWidget {
28 const MyApp({super.key});
29
30 @override
31 State<MyApp> createState() => _MyAppState();
32 }
33
34 class _MyAppState extends State<MyApp> {
35 @override
36 void initState() {
37 super.initState();
38 _configureAmplify();
39 }
40
41 Future<void> _configureAmplify() async {
42 await Amplify.addPlugin(
43 AmplifyDataStore(modelProvider: ModelProvider.instance));
44
45 final api = AmplifyAPI(modelProvider: ModelProvider.instance);
46 await Amplify.addPlugin(api);
47
48 try {
49 await Amplify.addPlugin(AmplifyAuthCognito());
50 await Amplify.configure(amplifyconfig);
51 } on Exception catch (e) {
52 safePrint('An error occurred configuring Amplify: $e');
53 }
54 }
55

56
57 @override
58 Widget build(BuildContext context) {
59 return Authenticator(
60 child: ChangeNotifierProvider(
61 create: (context) => Shop(),
62 child: MaterialApp(
63 debugShowCheckedModeBanner: false,
64 // set the default theme
65 theme: ThemeData.from(
66 useMaterial3: true,
67 colorScheme: ColorScheme.fromSwatch(
68 primarySwatch: Colors.blueGrey,
69 backgroundColor: Colors.white,
70 ), // ColorScheme.fromSwatch
71 .copyWith( // ThemeData.from
72 indicatorColor: Colors.red,
73 ),
74 // set the dark theme (optional)
75 darkTheme: ThemeData.from(
76 useMaterial3: true,
77 colorScheme: ColorScheme.fromSwatch(
78 primarySwatch: Colors.red,
79 backgroundColor: Colors.black,
80 brightness: Brightness.dark,
81 ), // ColorScheme.fromSwatch
82 ), // ThemeData.from
83 // set the theme mode to respond to the user's system preferences (optional)
84 themeMode: ThemeMode.system,
85 builder: Authenticator.builder(),
86 home: const RestaurantSelectionPage(),
87 routes: {
88 '/intropage': (context) => const IntroPage(),
89 '/mainpage': (context) => const MainPage(),
90 '/cartpage': (context) => const CartPage(),
91 '/myapp': (context) => const MyApp(),
92 '/restaurantselection': (context) =>
93 | const RestaurantSelectionPage(),
94 },
95 ), // MaterialApp
96 ), // ChangeNotifierProvider
97 ); // Authenticator
98 }
99 }
```

Intro Page (The Restaurant Get Started Page)

```
55
56     @override
57     Widget build(BuildContext context) {
58         return Authenticator(
59             child: ChangeNotifierProvider(
60                 create: (context) => Shop(),
61                 child: MaterialApp(
62                     debugShowCheckedModeBanner: false,
63                     // set the default theme
64                     theme: ThemeData.from(
65                         useMaterial3: true,
66                         colorScheme: ColorScheme.fromSwatch(
67                             primarySwatch: Colors.blueGrey,
68                             backgroundColor: Colors.white,
69                         ), // ColorScheme.fromSwatch
70                         .copyWith( // ThemeData.from
71                             indicatorColor: Colors.red,
72                         ),
73                     // set the dark theme (optional)
74                     darkTheme: ThemeData.from(
75                         useMaterial3: true,
76                         colorScheme: ColorScheme.fromSwatch(
77                             primarySwatch: Colors.red,
78                             backgroundColor: Colors.black,
79                             brightness: Brightness.dark,
80                         ), // ColorScheme.fromSwatch
81                     ), // ThemeData.from
82                     // set the theme mode to respond to the user's system preferences (optional)
83                     themeMode: ThemeMode.system,
84                     builder: Authenticator.builder(),
85                     home: const RestaurantSelectionPage(),
86                     routes: {
87                         '/intropage': (context) => const IntroPage(),
88                         '/mainpage': (context) => const MainPage(),
89                         '/cartpage': (context) => const CartPage(),
90                         '/myapp': (context) => const MyApp(),
91                         '/restaurantselection': (context) =>
92                             const RestaurantSelectionPage(),
93                     },
94                 ), // MaterialApp
95             ), // ChangeNotifierProvider
96         ); // Authenticator
97     }
98 }
99 }
```

Main Page (The Menu Page Basically)

```
shop.dart current_restaurant.dart main_page.dart itemtile.dart

lib > views > main_page.dart > _MainPageState > build
  import 'package:provider/provider.dart';
  ...
  class MainPage extends StatefulWidget {
    final String? selectedRestaurant;
    ...
    const MainPage({super.key, this.selectedRestaurant});
    ...
    @override
    State<MainPage> createState() => _MainPageState();
  }
  ...
  class _MainPageState extends State<MainPage> {
    void goToMainApp() {
      //give a message of confirmation
      showDialog(
        context: context,
        barrierDismissible: true,
        builder: (context) => AlertDialog(
          backgroundColor: primaryColor,
          content: Text(
            "Confirm to go back to restaurant selection",
            style: TextStyle(color: textColor),
            textAlign: TextAlign.center,
          ), // Text
          actions: [
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                MyButton(
                  text: 'Confirm',
                  onTap: () =>
                      Navigator.pushNamed(context, '/restaurantselection'),
                ), // MyButton
              ],
            ) // Row
          ],
        ), // AlertDialog
      );
    }

    //Navigate to item's page
    void navigateToDetail(int index) {
      //Get shop and it's menu
      final shop = context.read<Shop>();
      final menu = shop.menu;
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => DetailsPage(
            food: menu[index],
          ), // DetailsPage
        ), // MaterialPageRoute
      );
    }

    @override
    Widget build(BuildContext context) {
      //get the menu
      final args = ModalRoute.of(context)!.settings.arguments as Map?;
      final menu = args?['menu'];
      final mainImage = args?['mainImage'];
      //rest of the app
      return Scaffold(
        backgroundColor: bgColor,
        appBar: AppBar(
          ...
        ),
        ...
      );
    }
  }
}
```

```
shop.dart current_restaurant.dart main_page.dart itemtile.dart

lib > views > main_page.dart > _MainPageState
  appBar: AppBar(
    ...
    title: Text(widget.selectedRestaurant ?? 'Selected Restaurant'),
    actions: [
      // cart button
      IconButton(
        onPressed: () =>
            Navigator.pushNamed(context, '/cartpage'),
        icon: const Icon(Icons.shopping_cart),
      ), // IconButton
    ],
  ), // AppBar
  body: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      const SizedBox(
        height: 20,
      ), // SizedBox
      //A promotional/messsage banner
      Container(
        decoration: BoxDecoration(
          color: primaryColor, borderRadius: BorderRadius.circular(20),
        ),
        margin: const EdgeInsets.symmetric(horizontal: 25),
        padding: const EdgeInsets.all(25),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
            Column(
              children: [
                Text(
                  'Get 32% Promo',
                  style: GoogleFonts.lato(fontSize: 20, color: textColor),
                ), // Text
                const SizedBox(
                  height: 20,
                ), // SizedBox
                MyButton(
                  text: 'Redeem',
                  onTap: () {},
                ), // MyButton
              ],
            ), // Column
            //Insert an image/icon for promotional/message banner
            Image.network(
              mainImage ?? '', height: 100, // Handle null case
            ), // Image.network
          ],
        ), // Row
      ), // Container
      const SizedBox(
        height: 25,
      ), // SizedBox
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 25.0),
        ...
      ),
    ],
  ),
}
```

```
shop.dart current_restaurant.dart main_page.dart itemtile.dart

lib > views > main_page.dart > _MainPageState > build
  const SizedBox(
    height: 25,
  ), // SizedBox
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 25.0),
    child: Text(
      'Recommended',
      style: TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 18,
        color: headingColor),
    ), // Text
  ), // Padding
  //recommended food items
  Expanded(
    flex: 1,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: menu.length,
      itemBuilder: (context, index) {
        final food = menu[index];
        if (food.isRecommended == true) {
          return RecommendedTile(
            item: food,
            onTap: () => navigateToDetail(index),
          ); // RecommendedTile
        } else {
          return Container();
        }
      },
    ), // ListView.builder
  ), // Expanded
  //menu of items
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 25.0),
    child: Text(
      'Our Menu',
      style: TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 18,
        color: headingColor),
    ), // Text
  ), // Padding
  const SizedBox(
    height: 10,
  ), // SizedBox
  //List View to display the menu
  Expanded(
    flex: 2,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: menu.length,
      itemBuilder: (context, index) {
        final food = menu[index];
        return ItemTile(
          item: food,
          onTap: () => navigateToDetail(index),
        ); // ItemTile
      },
    ), // ListView.builder
  ), // Expanded
}
```

Details Page (The Specific Item Page Basically)

```
details_page.dart cart_page.dart intro_page.dart Preference.dart

lib > views > details_page.dart > _DetailsPageState > addToCart
  import 'package:provider/provider.dart';
  ...
  class DetailsPage extends StatefulWidget {
    final Item food;
    const DetailsPage({super.key, required this.food});
  }
  ...
  @override
  State<DetailsPage> createState() => _DetailsPageState();
}

class _DetailsPageState extends State<DetailsPage> {
  //quantity
  int quantityCount = 0;

  void decrementQuantity() {
    if (quantityCount >= 1) {
      setState(() {
        quantityCount--;
      });
    }
  }

  void incrementQuantity() {
    if (quantityCount <= 14) {
      setState(() {
        quantityCount++;
      });
    }
  }

  //add to cart
  void addToCart() {
    //only add to cart if there is something in the cart
    if (quantityCount > 0) {
      //get access to shop
      final shop = context.read<Shop>();

      //add the actual item to the cart
      shop.addToCart(widget.food, quantityCount);

      //give a message of confirmation
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (context) => AlertDialog(
          backgroundColor: primaryColor,
          content: Text(
            "Successfully added to cart",
            style: TextStyle(color: textColor),
            textAlign: TextAlign.center,
          ),
          actions: [
            //confirm button
            IconButton(
              onPressed: () {
                Navigator.pop(context);
                Navigator.pop(context);
              },
              icon: Icon(
                Icons.done,
                color: textColor,
              ),
            ),
          ],
        ),
      );
    }
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.transparent,
      title: Text(widget.food.name),
      elevation: 0,
    ), // AppBar
    body: Column(
      children: [
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 25.0),
            child: ListView(
              children: [
                const SizedBox(
                  height: 25,
                ), // SizedBox
                //image
                Image.network(
                  widget.food.imagePath,
                  height: 200,
                ), // Image.network

                const SizedBox(
                  height: 25,
                ), // SizedBox
                const SizedBox(
                  height: 25,
                ), // SizedBox

                Text(
                  widget.food.shortDescription,
                  style: GoogleFonts.lato(fontSize: 20, height: 2),
                  textAlign: TextAlign.center,
                ), // Text
              ],
            ), // ListView
          ), // Padding
        ), // Expanded
        Container(
          color: primaryColor,
          padding: const EdgeInsets.all(25),
          child: Column(children: [
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                //price
                Text(
                  '\$${widget.food.price}',
                  style: TextStyle(
                    color: textColor,
                    fontWeight: FontWeight.bold,
                    fontSize: 18), // TextStyle
                ), // Text
              ],
            ),
            Row(
              children: [
                //minus button
                Container(
                  decoration: BoxDecoration(
                    color: secondaryColor, shape: BoxShape.circle), // Box
                  child: IconButton(
                    icon: Icon(
                      Icons.remove,
                      color: textColor,
                    ), // Icon
                    onPressed: decrementQuantity,
                  ), // IconButton
                ), // Container
                //quantity counter
                SizedBox(
                  width: 40,
                  child: Center(
                    child: Text(
                      quantityCount.toString(),
                      style: TextStyle(
                        color: textColor,
                        fontWeight: FontWeight.bold,
                        fontSize: 18), // TextStyle
                    ), // Text
                  ), // Center
                ), // SizedBox
                //plus button
                Container(
                  decoration: BoxDecoration(
                    color: secondaryColor, shape: BoxShape.circle), // Box
                  child: IconButton(
                    icon: Icon(
                      Icons.add,
                      color: textColor,
                    ), // Icon
                    onPressed: incrementQuantity,
                  ), // IconButton
                ), // Container
              ],
            ), // Row
          ],
        ), // Row
        const SizedBox(
          height: 25,
        ), // SizedBox
        //add to cart button
        MyButton(text: "Add to Cart", onTap: addToCart),
      ],
    ), // Column
  ), // Container
), // Column
); // Scaffold
}

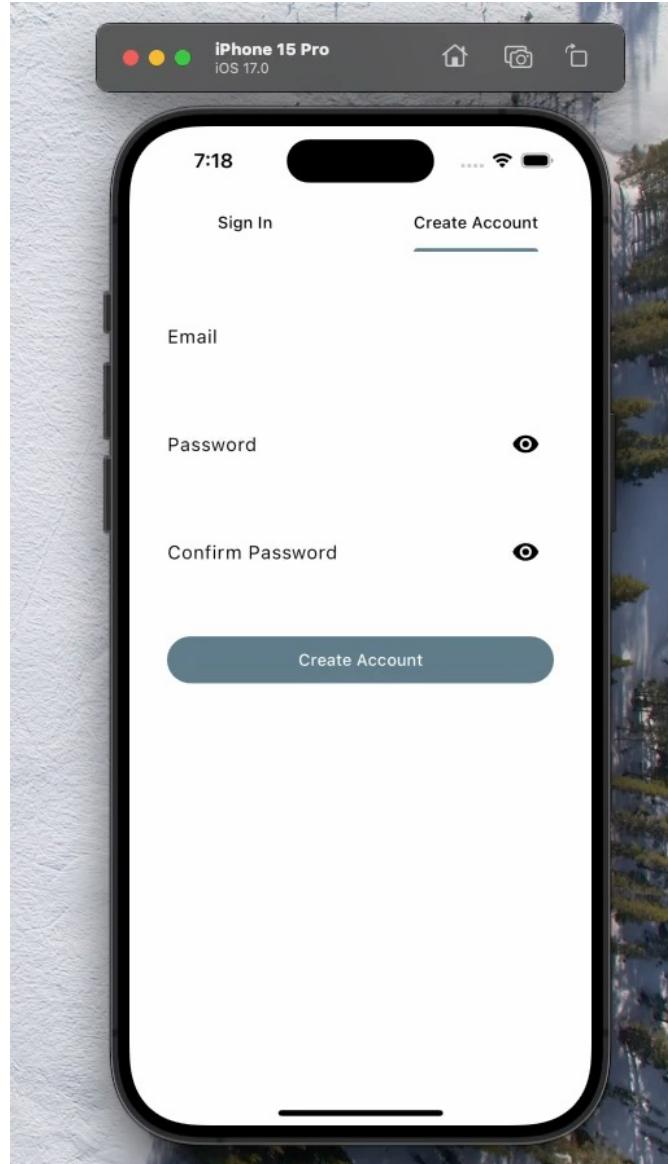
```

Cart Page

```
lib > views > cart_page.dart > CartPage > build
/
8 class CartPage extends StatelessWidget {
9   const CartPage({super.key});
10
11   void removeFromCart(Item food, BuildContext context) {
12     //getting access to shop
13     final shop = context.read<Shop>();
14     //actually removing the cart
15     shop.removeFromCart(food);
16   }
17
18   @override
19   Widget build(BuildContext context) {
20     return Consumer<Shop>(
21       builder: (context, value, child) => Scaffold(
22         backgroundColor: primaryColor,
23         appBar: AppBar(
24           title: const Text('My Cart'),
25           elevation: 0,
26           backgroundColor: primaryColor,
27           foregroundColor: headingColor,
28         ), // AppBar
29         body: Column(
30           children: [
31             //customer cart section
32             Expanded(
33               child: ListView.builder(
34                 itemCount: value.cart.length,
35                 itemBuilder: (context, index) {
36                   //get food from cart
37                   final Item food = value.cart[index];
38                   //get food name
39                   final String foodName = food.name;
40                   //get food price
41                   final String foodPrice = food.price;
42                   //return list tile
43                   return Container(
44                     decoration: BoxDecoration(
45                       color: secondaryColor,
46                       borderRadius: BorderRadius.circular(10), // BoxDecoration
47                     margin: const EdgeInsets.only(left: 20, top: 20, right: 20),
48                     child: ListTile(
49                       title: Text(
50                         foodName,
51                         style: TextStyle(
52                           color: primaryColor, fontWeight: FontWeight.bold), // Text
53                         ), // Text
54                         subtitle: Text(
55                           foodPrice,
56                           style: TextStyle(
57                             color: subtitleColor,
58                           ), // TextStyle
59                         ), // Text
60                         trailing: IconButton(
61                           icon: Icon(
62                             Icons.delete,
63                             color: bgColor,
64                           ), // Icon
65                           onPressed: () => removeFromCart(food, context),
66                         ), // IconButton
67                         ), // Container
68                         ), // ListView.builder
69                         ), // Expanded
70                         );
71   
```

```
lib > views > cart_page.dart > ...
/
66   ), // IconButton
67   ), // ListTile
68   ); // Container
69   ),
70   ), // ListView.builder
71   ), // Expanded
72   ),
73   ),
74   ),
75   ),
76   ),
77   ),
78   ),
79   ),
80   ),
81   ),
82   ),
83   ),
84   ),
85   ),
86   ),
87   ),
88   ),
89   ),
90   ),
91   ),
92   ),
93   ),
94   ),
95   ),
96   ),
97   ),
98   ),
99   ),
100  ),
101  ),
102  ),
103  ),
104  ),
105  ),
106  ),
107  ),
108  ),
109  ),
110  ),
111  ),
112  ),
113  ),
114  ),
115  ),
116  ),
117  ),
118  ),
119  ),
120  ),
121  ),
122  );
123  }
124  
```

Video Demo



Incase video doesn't work:

https://drive.google.com/file/d/1ix3whOe_WG3qB_6bKH6DGonQuuZcUZhw/view?usp=sharing

How can it improve?

- In the restaurant selection page, add a button to go to a new page that has a form where you can add a new restaurant and its related properties. Only 1 restaurant per authenticated user would be allowed. The restaurant would have edit button for each form as well. Would use text controllers and a function to feed the information, once submitted, over to the database.
- Images, allergies, etc. could be used for the details page for each item to display more information about it. Would need to add import mechanism for photos gallery to database.
- Alongside the form to add/modify a restaurant, the restaurant could change preferences which would update the colors.dart page to make each restaurant have their own theme/style.