Restaurant App UI Using Flutter & Dart

Complete code of the project :-

Main.dart :-

```
//                ignore_for_file:                use_key_in_widget_constructors,
avoid_function_literals_in_foreach_calls,             prefer_const_constructors,
prefer_const_literals_to_create_immutables,
prefer_interpolation_to_compose_strings,               sized_box_for_whitespace,
avoid_unnecessary_containers

import 'package:flutter/material.dart';
import 'package:restaurantappui/login_page.dart';
import 'package:restaurantappui/home_page.dart';


void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      // home: LoginPage(),
      routes: {
        "/": (context) => LoginPage(),
        "/home": (context) => HomePage(),
        "/login": (context) => LoginPage(),

      },
    );
  }
}
```

loign_page.dart :-

```
//                ignore_for_file:                use_key_in_widget_constructors,
avoid_function_literals_in_foreach_calls,             prefer_const_constructors,
prefer_const_literals_to_create_immutables,
prefer_interpolation_to_compose_strings,               sized_box_for_whitespace,
avoid_unnecessary_containers

import 'package:flutter/material.dart';
import 'package:restaurantappui/login_page.dart';
import 'package:restaurantappui/home_page.dart';


void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
```

```
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      // home: LoginPage(),
      routes: {
        "/": (context) => LoginPage(),
        "/home": (context) => HomePage(),
        "/login": (context) => LoginPage(),

      },
    );
  }
}
```

home_page.dart :-

```
//                  ignore_for_file:                  prefer_const_constructors,
prefer_const_literals_to_create_immutables,         avoid_unnecessary_containers,
use_key_in_widget_constructors,                       sized_box_for_whitespace,
prefer_interpolation_to_compose_strings

import 'dart:convert';
import 'package:flutter/material.dart';
import
'package:material_design_icons_flutter/material_design_icons_flutter.dart';
import 'package:restaurantappui/drawer.dart';

var bannerItems = ["Pizza", "Burger", "Cheese Chilly", "Noodles", "Geleto"];
var bannerOffers = [
  "Flate 20% Off",
  "More than 40% Off",
  "Flate 35% Off",
  "Free Coke Can",
  "More than 15% Off"
];
var bannerImage = [
  "images/pizza.jpg",
  "images/burger.jpg",
  "images/cheesechilly.jpg",
  "images/noodles.jpg",
  "images/Sassy_Spoon.jpg"
];

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var screenHeight = MediaQuery.of(context).size.height;
    var screenWidth = MediaQuery.of(context).size.width;

    Future<List<Widget>> createList() async {
      List<Widget> items = [];
      String dataString =
```

```
    await DefaultAssetBundle.of(context).loadString("data.json");
List<dynamic> dataJSON = jsonDecode(dataString);

for (var object in dataJSON) {
  String finalString = "";
  List<dynamic> dataList = object["placeItems"];
  for (var item in dataList) {
    // finalString = finalString + item + " | ";
    finalString = finalString + item + " | ";
  }

  items.add(Padding(
    padding: EdgeInsets.all(2.0),
    child: Container(
      decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.all(Radius.circular(10.0)),
          boxShadow: [
            BoxShadow(
                color: Colors.black12,
                spreadRadius: 2.0,
                blurRadius: 5.0),
          ]),
      margin: EdgeInsets.all(5.0),
      child: Row(
        mainAxisSize: MainAxisSize.max,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          ClipRRect(
            borderRadius: BorderRadius.only(
                topLeft: Radius.circular(10.0),
                bottomLeft: Radius.circular(10.0)),
            child: Image.asset(
              object["placeImage"],
              width: 80,
              height: 80,
              fit: BoxFit.cover,
            ),
          ),
          SizedBox(
            width: 250,
            child: Padding(
              padding: const EdgeInsets.all(8.0),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                  Text(object["placeName"]),
                  Padding(
                    padding: const EdgeInsets.only(top: 2.0, bottom: 2.0),
                    child: Text(
                      finalString,
                      overflow: TextOverflow.ellipsis,
```

```
                      style: TextStyle(
                        fontSize: 12.0,
                        color: Colors.black54,
                      ),
                      maxLines: 1,
                    ),
                  ),
                  Text(
                    "Min. Order: ${object["minOrder"]}",
                    style:
                        TextStyle(fontSize: 12.0, color: Colors.black54),
                  )
                ],
              ),
            ),
          )
        ],
      ),
    ),
  ));
  }

  return items;
}

return Scaffold(
  appBar: AppBar(
    title: Text(
      "Master Chef's Lounge",
      style: TextStyle(fontSize: 50, fontFamily: "Samantha"),
    ),
    titleTextStyle: TextStyle(fontSize: 20, color: Colors.white),
    backgroundColor: Colors.red.shade400,
  ),
  body: Container(
    height: screenHeight,
    width: screenWidth,
    child: SafeArea(
        child: SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          Padding(
            padding: const EdgeInsets.fromLTRB(10, 5, 10, 5),
          ),
          BannerWidgetArea(),
          Container(
              child: FutureBuilder<List<Widget>>(
            future: createList(),
            builder: (context, snapshot) {
              if (snapshot.connectionState == ConnectionState.waiting) {
                return CircularProgressIndicator();
```

```
                } else if (snapshot.hasError) {
                  return Text('Error loading data: ${snapshot.error}');
                } else if (snapshot.hasData) {
                  // Perform a null check before using snapshot.data
                  final List<Widget> data = snapshot.data!;
                  return Padding(
                    padding: EdgeInsets.all(8.0),
                    child: ListView(
                      primary: false,
                      shrinkWrap: true,
                      children: data,
                    ),
                  );
                } else {
                  return Text('No data available');
                }
              },
            ))
          ],
        ),
      )),
    ),
    floatingActionButton: FloatingActionButton(
        onPressed: () {},
        backgroundColor: Colors.red.shade400,
        child: Icon(
          MdiIcons.food,
          color: Colors.white,
        )),
    drawer: MyDrawer(),
  );
 }
}

class BannerWidgetArea extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var screenWidth = MediaQuery.of(context).size.width;

    PageController controller =
        PageController(viewportFraction: 0.8, initialPage: 2);

    List<Widget> banners = [];

    for (int i = 0; i < bannerItems.length; i++) {
      var bannerView = Padding(
        padding: EdgeInsets.all(10.0),
        child: Container(
          child: Stack(
            fit: StackFit.expand,
            children: <Widget>[
              Container(
```

```
            decoration: BoxDecoration(
                borderRadius: BorderRadius.all(Radius.circular(20.0)),
                boxShadow: [
                  BoxShadow(
                      color: Colors.black38,
                      offset: Offset(2.0, 2.0),
                      blurRadius: 5.0,
                      spreadRadius: 1.0)
                ]),
          ),
          ClipRRect(
            borderRadius: BorderRadius.all(Radius.circular(20.0)),
            child: Image.asset(
              bannerImage[i],
              fit: BoxFit.cover,
            ),
          ),
          Container(
            decoration: BoxDecoration(
                borderRadius: BorderRadius.all(Radius.circular(20.0)),
                gradient: LinearGradient(
                    begin: Alignment.topCenter,
                    end: Alignment.bottomCenter,
                    colors: [Colors.transparent, Colors.black87])),
          ),
          Padding(
            padding: EdgeInsets.all(10.0),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.end,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: <Widget>[
                Text(
                  bannerItems[i],
                  style: TextStyle(fontSize: 25.0, color: Colors.white),
                ),
                Text(
                  bannerOffers[i],
                  // "More than 40% Off",
                  style: TextStyle(fontSize: 12.0, color: Colors.white),
                )
              ],
            ),
          )
        ],
      ),
    ),
  );
  banners.add(bannerView);
}

return Container(
  width: screenWidth,
```

```
      height: screenWidth * 9 / 16,
      child: PageView(
        controller: controller,
        scrollDirection: Axis.horizontal,
        children: banners,
      ),
    );
  }
}
```

drawer.dart :-

```
//                  ignore_for_file:                  prefer_const_constructors,
prefer_const_literals_to_create_immutables, unnecessary_import

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:restaurantappui/login_page.dart';

class MyDrawer extends StatelessWidget {
  const MyDrawer({super.key});

  @override
  Widget build(BuildContext context) {
    return Drawer(
      child: Container(
        height: double.infinity,
        decoration: BoxDecoration(
            gradient: LinearGradient(
          colors: [Colors.white, Colors.white, Colors.white],
          begin: Alignment.topCenter,
          end: Alignment.bottomCenter,
        )),
        child: Padding(
          padding: const EdgeInsets.all(8.0),
          child: ListView(
            children: [
              DrawerHeader(
                padding: EdgeInsets.zero,
                child: UserAccountsDrawerHeader(
                  accountName: Text(
                    name,
                    style: TextStyle(
                        color: Colors.white,
                        // fontWeight: FontWeight.bold,
                        fontSize: 16),
                  ),
                  accountEmail: Text(
                    "sangramsupalkar123@gmail.com",
                    // "",
                    style: TextStyle(color: Colors.white, fontSize: 14),
                  ),
```

```
    decoration: BoxDecoration(
      color: Colors.red.shade400,
      borderRadius: BorderRadius.only(
          bottomRight: Radius.circular(10),
          topLeft: Radius.circular(10),
          topRight: Radius.circular(10),
          bottomLeft: Radius.circular(10)),
    ),
    currentAccountPicture: CircleAvatar(
      backgroundColor: Colors.white,
      backgroundImage:
          // NetworkImage("https://i.ibb.co/vzrShBq/Sangram.jpg"),
          NetworkImage("https://i.ibb.co/vzrShBq/Sangram.jpg"),
    ),
  ),
),
ListTile(
  leading: Icon(
    Icons.account_circle_outlined,
    color: Colors.black,
  ),
  title: Text(
    "Account",
    textScaleFactor: 1.2,
    style: TextStyle(
        color: Colors.black,
        // fontWeight: FontWeight.bold,
        fontSize: 13),
  ),
),
ListTile(
  leading: Icon(
    CupertinoIcons.bell,
    color: Colors.black,
  ),
  title: Text(
    "Notifications",
    textScaleFactor: 1.2,
    style: TextStyle(
        color: Colors.black,
        // fontWeight: FontWeight.bold,
        fontSize: 13),
  ),
),
ListTile(
  leading: Icon(
    CupertinoIcons.star,
    color: Colors.black,
  ),
  title: Text(
    "Favourite",
    textScaleFactor: 1.2,
```

```
    style: TextStyle(
        color: Colors.black,
        // fontWeight: FontWeight.bold,
        fontSize: 13),
  ),
),
ListTile(
  leading: Icon(
    CupertinoIcons.creditcard,
    color: Colors.black,
  ),
  title: Text(
    "Payment",
    textScaleFactor: 1.2,
    style: TextStyle(
        color: Colors.black,
        // fontWeight: FontWeight.bold,
        fontSize: 13),
  ),
),
ListTile(
  leading: Icon(
    CupertinoIcons.clock,
    color: Colors.black,
  ),
  title: Text(
    "History",
    textScaleFactor: 1.2,
    style: TextStyle(
        color: Colors.black,
        // fontWeight: FontWeight.bold,
        fontSize: 13),
  ),
),
ListTile(
  leading: Icon(
    CupertinoIcons.settings_solid,
    color: Colors.black,
  ),
  title: Text(
    "Settings",
    textScaleFactor: 1.2,
    style: TextStyle(
        color: Colors.black,
        // fontWeight: FontWeight.bold,
        fontSize: 13),
  ),
),
ListTile(
  leading: Icon(
    Icons.help_outline_rounded,
    color: Colors.black,
```

```
                ),
                title: Text(
                  "Help",
                  textScaleFactor: 1.2,
                  style: TextStyle(
                      color: Colors.black,
                      // fontWeight: FontWeight.bold,
                      fontSize: 13),
                ),
              ),
              ListTile(
                leading: Icon(
                  CupertinoIcons.phone,
                  color: Colors.black,
                ),
                title: Text(
                  "Contact Us",
                  textScaleFactor: 1.2,
                  style: TextStyle(
                      color: Colors.black,
                      // fontWeight: FontWeight.bold,
                      fontSize: 13),
                ),
              ),
              ListTile(
                leading: Icon(
                  CupertinoIcons.power,
                  color: Colors.black,
                ),
                title: Text(
                  "Log Out",
                  textScaleFactor: 1.2,
                  style: TextStyle(
                      color: Colors.black,
                      // fontWeight: FontWeight.bold,
                      fontSize: 13),
                ),
                onTap: () {
                  Navigator.pushNamed(context, "/login");
                  name = '';
                },
              ),
            ],
          ),
        ),
      ),
    );
  }
}

Padding buildContactAvatar(String name, String filename, BuildContext context) {
  return Padding(
```

```
    padding: const EdgeInsets.only(right: 25),
    child: GestureDetector(
      onTap: () {
        _showPopUp(context, name);
      },
      child: Column(
        children: [
          // UserAvatar(filename: filename),
          SizedBox(
            height: 5,
          ),
          Text(
            name,
            style: const TextStyle(
                color: Colors.black, fontSize: 15, fontWeight: FontWeight.bold),
          ),
        ],
      ),
    ),
  );
}

void _showPopUp(BuildContext context, String name) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text("Contact"),
        content: Text("You tapped on $name"),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: Text("Close", style: TextStyle(color: Colors.blueAccent)),
          ),
        ],
      );
    },
  );
}
```

data.json :-

```
[
  {
    "placeImage": "images/GOOD_FLIPPIN_BURGERS.jpg",
    "placeName": "GOOD FLIPPIN' BURGERS",
    "placeItems": ["Burgers","Chinese","Fast Food","Italian","Juice"],
    "minOrder": "30"
  },
  {
```

```
  "placeImage": "images/Mainland_China.jpg",
  "placeName": "Mainland China",
  "placeItems": ["Burgers","Chinese","Fast Food","Italian","Juice"],
  "minOrder": "50"
},
{
  "placeImage": "images/Shamiana.jpg",
  "placeName": "Shamiana",
  "placeItems": ["Fish Stew","Indian","Mughlia","Desert","Juice"],
  "minOrder": "50"
},
{
  "placeImage": "images/14_41_Pizzeria.jpg",
  "placeName": "14° 41° Pizzeria",
  "placeItems": ["Pizza","Chinese","Fast Food","Italian","Juice"],
  "minOrder": "30"
},
{
  "placeImage": "images/Mystique_Delight.jpg",
  "placeName": "Mystique Delight",
  "placeItems": ["Pestries","Italian","Western","Desert","Juice"],
  "minOrder": "30"
},
{
  "placeImage": "images/bluehill.jpg",
  "placeName": "Blue Hill",
  "placeItems": ["Stakes","Spainish","BBQ","Lamb","Lebanese","Soda"],
  "minOrder": "60"
},
{
  "placeImage": "images/The_Foo.jpg",
  "placeName": "The Foo",
  "placeItems": ["Sushi","Japanese","Mughlia","Desert","Juice"],
  "minOrder": "50"
},
{
  "placeImage": "images/Sassy_Spoon.jpg",
  "placeName": "Sassy Spoon",
  "placeItems": ["Geleto","Italian","Smoothie","Desert","Juice"],
  "minOrder": "30"
},
{
  "placeImage": "images/Sequel.jpg",
  "placeName": "Sequel",
  "placeItems": ["Salad","Mongolian","Organic","Meat","Mojito"],
  "minOrder": "40"
},
{
  "placeImage": "images/Behrouz.jpg",
  "placeName": "Behrouz",
  "placeItems": ["Biryani","Mughlia","Kebab","Sweets","Sarbat"],
  "minOrder": "60"
```

```
  }
]
```

<end_of_code>

Explanation :-

1. ***What is the project?***
2. ***How was this project done?***
3. ***What are the features?***
4. ***What are the technologies used?***
5. ***What concepts were used ?***

Ans)

The project is a restaurant app UI, inspired by Zomato, with both a login page and a home page. However, the app is not fully functional, as it doesn't interact with a backend or provide actual data.

1. **Project Description**:
   The project is a restaurant app UI, designed to resemble Zomato, which allows users to view a list of restaurants and their details. The app consists of two main screens: the login page and the home page. The login page allows users to log in or navigate to the home page directly. The home page displays a list of restaurants with their names, menu items, and other details.

2. **Project Implementation**:
   The project is implemented using the Flutter framework, which allows for cross-platform mobile app development. The UI design is done using various Flutter widgets and layout components to create a visually appealing and responsive user interface.

Complete explanation of code

1. **main.dart**:
   - The `main.dart` file serves as the entry point of the application.
   - The `MyApp` class is the root widget of the application. It extends `StatelessWidget`.
   - The `build` method of `MyApp` returns a `MaterialApp` widget, which sets up the main structure of the app and handles routing between different screens.
   - The `debugShowCheckedModeBanner` property is set to `false` to hide the debug banner on the top-right corner of the app.
   - The `routes` property defines named routes for navigation within the app. It maps route names to the corresponding widget classes.
   - In this case, three routes are defined:
     - `/`: Maps to the `LoginPage` widget. When the app launches, it will display the login page.
     - `/home`: Maps to the `HomePage` widget, which is the main screen of the app, displaying the list of restaurants.
     - `/login`: Also maps to the `LoginPage` widget, providing users with the option to navigate back to the login page if they are already logged in.

2. **login_page.dart**:
   - The `LoginPage` class is defined, extending `StatelessWidget`.
   - The `build` method of `LoginPage` returns a `Scaffold` widget, which provides the basic structure for the login page.

- The login page contains a centered logo (not present in the code provided) and a button that allows users to log in.
- The "Log In" button is linked to the `/home` route using `Navigator.pushNamed(context, "/home")`. When the button is pressed, it navigates to the home page.

3. **home_page.dart**:
- The `HomePage` class is defined, extending `StatelessWidget`.
- The `build` method of `HomePage` returns a `Scaffold` widget, which provides the basic structure for the home page.
- The `AppBar` widget is displayed at the top of the page, with the title "Master Chef's Lounge."
- The `BannerWidgetArea` class is a custom widget used to display a horizontal scrolling banner with various restaurant items and offers. It contains a `PageView` widget to allow users to scroll through different banner items.
- The `FutureBuilder` widget is used to asynchronously load data from the `data.json` file and display the list of restaurants on the home page.
- The `createList()` function is an asynchronous function that reads the JSON data from the file and converts it into a list of widgets representing individual restaurants.
- Inside `createList()`, the JSON data is fetched and decoded using `jsonDecode`.
- The function iterates through each object in the JSON array, extracts the relevant information (place image, name, items, and minimum order), and creates a `Container` widget for each restaurant.
- The `Container` displays the restaurant image, name, menu items, and minimum order amount in a visually appealing manner.
- The function returns a list of these restaurant `Container` widgets.
- The `FutureBuilder` utilizes the `createList()` function to build the list of restaurants. While the data is being fetched and processed, a circular progress indicator is shown. Once the data is ready, the list of restaurants is displayed on the home page.

4. **drawer.dart**:
- The `MyDrawer` class is a `StatelessWidget` that returns a `Drawer` widget.
- The `Drawer` is a sliding panel that appears from the left side of the screen, providing navigation options for the user.
- The `Container` with a `LinearGradient` decoration is used to create a gradient background for the drawer.
- Inside the drawer, a `ListView` widget is used to display multiple `ListTile` widgets representing various menu options for the user.
- The user's profile information is displayed at the top of the drawer using the `UserAccountsDrawerHeader` widget. It includes the user's name, email, and profile picture.
- The user's name is fetched from a variable called `name`. However, in the current code, it's not clear where `name` is defined or initialized.
- The `ListTile` widgets represent various menu options, such as "Account," "Notifications," "Favourite," "Payment," "History," "Settings," "Help," "Contact Us," and "Log Out."
- When the user taps on the "Log Out" option, it navigates to the login page (`/login`) and sets the user's name to an empty string.
- The `buildContactAvatar` function appears to be a helper function but is not used in the current code provided. It seems to be related to displaying contact avatars with associated pop-up dialogs, but it's not called or utilized in the provided code.

5. **data.json**:
- The `data.json` file contains an array of restaurant objects, each representing a restaurant's details.
- Each restaurant object has properties such as `placeImage` (path to the restaurant image), `placeName` (restaurant name), `placeItems` (list of menu items), and `minOrder` (minimum order amount).

Overall, the app provides a basic restaurant app UI with a login page, a home page displaying a list of restaurants, and a custom drawer for navigation options. The restaurant data is loaded asynchronously from the `data.json` file and displayed on the home page using the `FutureBuilder`. The drawer displays user-related options, including a "Log Out" option that navigates the user to the login page. The app is intended to showcase the UI design and loading of static restaurant data. For a fully functional restaurant app, further development and backend integration would be required.

<end_of_code_explanation>

3. **Features**:
   - Login Page: Users can log in to the app using their credentials or navigate directly to the home page.
   - Home Page: Displays a list of restaurants along with their images, names, menu items, and minimum order details.

4. **Technologies Used**:
   - Flutter: A popular open-source UI software development kit (SDK) by Google for building natively compiled applications for mobile, web, and desktop.
   - Dart: The programming language used by Flutter for building the app logic and UI components.

5. **Concepts Used**:
   - UI Design: The code demonstrates various UI design concepts, including using containers, columns, rows, padding, images, and text elements to create a visually appealing layout.
   - Asynchronous Programming: The `FutureBuilder` is used to asynchronously load data from a JSON file and display it in the list of restaurants.
   - Navigation: The app utilizes the `MaterialApp`'s `routes` property to define named routes for navigation between different screens (login and home page).
   - Statelessness: The app is mainly stateless, as indicated by the extensive use of stateless widgets. However, the `FutureBuilder` handles stateful operations for asynchronous data loading.

It's important to note that while the UI design looks promising, it's not functional without the actual backend integration. For a fully operational restaurant app like Zomato, you would need to implement features such as user authentication, restaurant data retrieval from a server, search functionality, user reviews, and ordering capabilities.

In summary, this project showcases a well-designed UI for a restaurant app, but it requires further development to become a fully functional and feature-rich application. If you plan to continue working on it, you'll need to implement the backend functionalities to turn it into a functional restaurant app.