Ankara Üniversitesi

BLM4537-A IOS İle Mobil Uygulama Geliştirme I

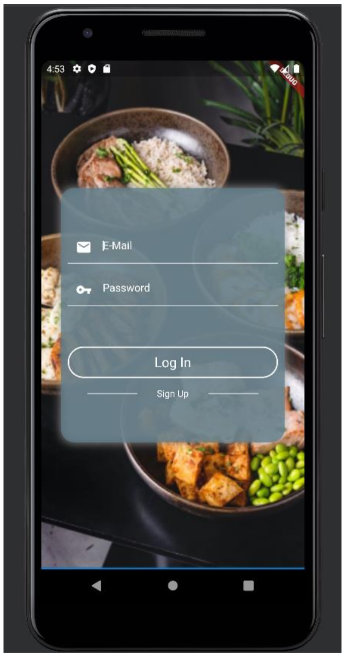
MAMOUN J.S. ZAID

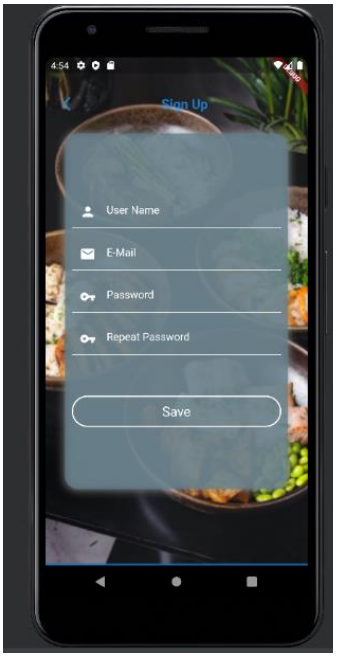
20290961

Yemek Tarifi Uygulaması

The aim of this project is to develop a small mobile application using the Flutter framework that allows users to browse and search for food recipes.

The design of the app was based on a simple and user-friendly interface. The main screen of the app displays a list of recipes, which can be filtered by category. Each recipe has a picture, a title, and a brief description. When a recipe is selected, the user is taken to a detailed view that includes the recipe's ingredients and instructions.





Log in screen:

This is the login page of the app, it is a stateful widget that contains two text editing controllers, one for the email and the other for the password.

It is using the AuthService class which is probably responsible for handling the authentication process with Firebase.

The page contains two text fields, one for the email and the other for the password.

The page also has a button that triggers the sign in process and navigate to the homepage if the login is successful.

It also uses MediaQuery to get the screen size and adjust the design accordingly.

Register screen:

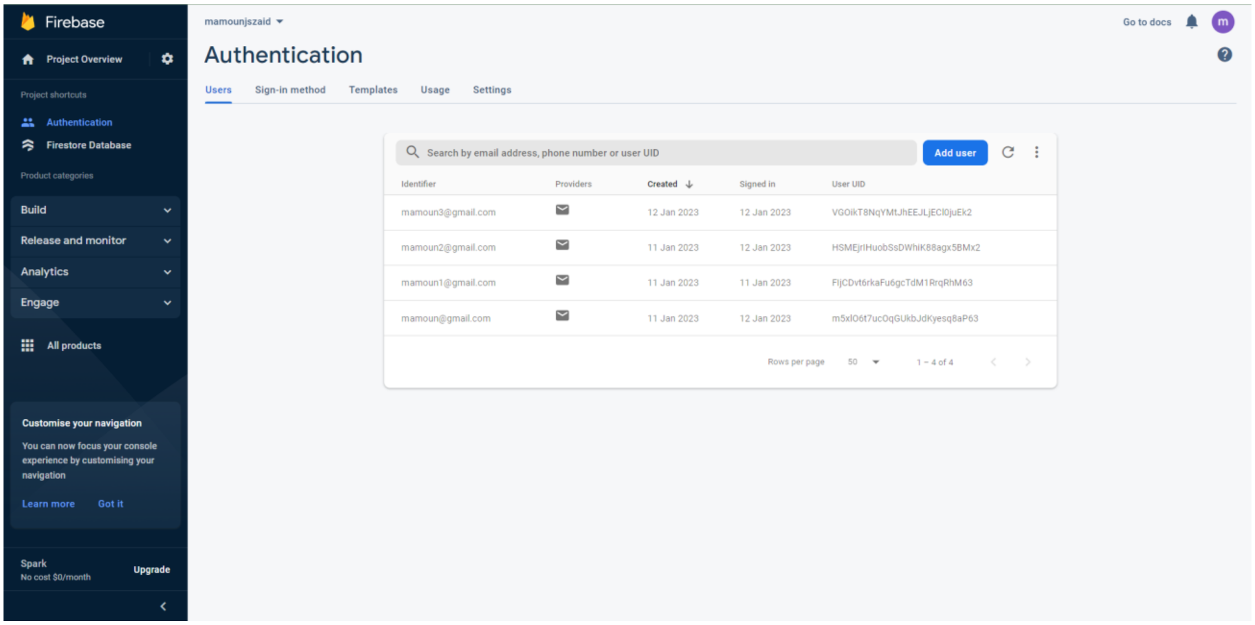
This is the register page of the app, it is also a stateful widget that contains four text editing controllers, one for the name, one for the email, one for the password, and the other for confirming the password.

It's using the AuthService class which is responsible for handling the registration process with Firebase.

The page contains four text fields, one for the name, one for the email, one for the password, and one for the password confirmation.

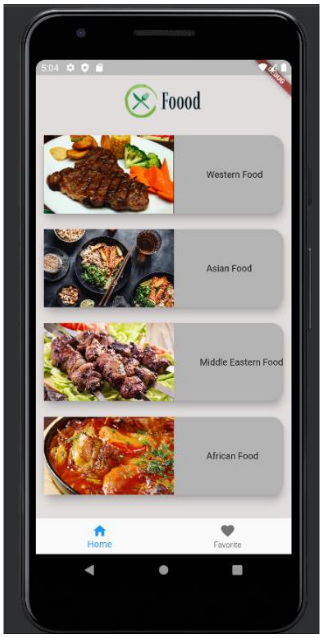
The page also has a button that triggers the registration process, and navigate to the login page if the registration is successful.

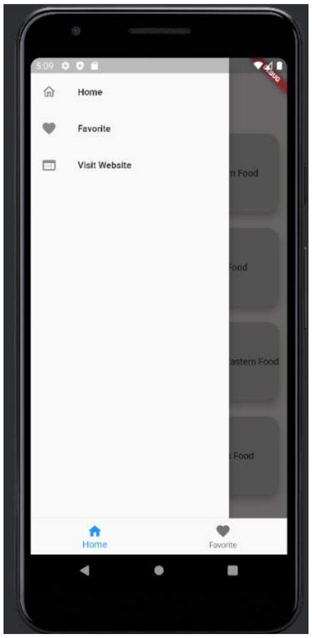
The design of this page is similar to the login page with a Scaffold widget that gives it the basic layout structure of a Material Design screen in flutter and it uses MediaQuery to get the screen size and adjust the design accordingly.



Identifier	Providers	Created ↓	Signed in	User UID
mamoun3@gmail.com	✉	12 Jan 2023	12 Jan 2023	VGOikT8NqYMtJhEEJLjECl0juEk2
mamoun2@gmail.com	✉	11 Jan 2023	12 Jan 2023	HSMEjrIHuobSsDWhiK88agx5BMx2
mamoun1@gmail.com	✉	11 Jan 2023	11 Jan 2023	FIjCDvt6rkaFu6gcTdM1RrqRhM63
mamoun@gmail.com	✉	11 Jan 2023	12 Jan 2023	m5xlO6t7ucOqGUkbJdKyesq8aP63

Here is the firebase showing the Authentication page with a list of all the signed in users.





BottomNavigationBar:

This is the bottom navigation bar function of the app, it is stateful widget that contains two screens :

HomePage

Favorite

This class is responsible for displaying the correct screen based on the selected tab in the bottom navigation bar, it has a currentIndex variable that keeps track of the currently selected tab and a screens list that contains the widgets that correspond to each tab.

The BottomNavigationBar widget is used to display the tabs at the bottom of the screen, it has an onTap callback that is called when a tab is selected, this callback updates the currentIndex variable and rebuilds the widget to display the corresponding screen.

Also, it has items property that contains a list of BottomNavigationBarItem, which are used to display the icons and labels of each tab.

Home screen:

This is the home page of the app, it's a stateful widget that contains a Scaffold widget that gives it the basic layout structure of a Material Design screen in flutter.

The Scaffold widget contains a drawer property that is set to a constant NavigationDrawer widget, this widget contains the navigation menu of the app.

The body of the Scaffold widget is a ListView that contains several child widgets including:

Image.asset : this widget is used to display an image from the project's assets folder.

Material : this widget is used to create a Material Design card, it has elevation, color, and borderRadius properties.

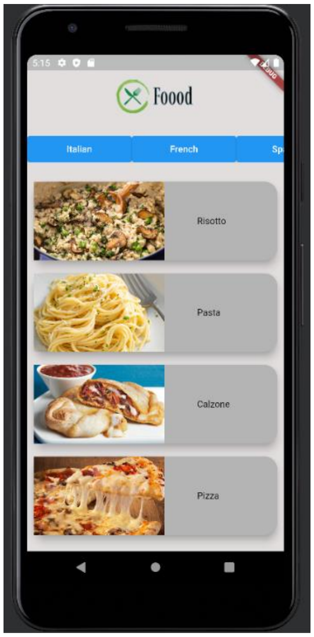
InkWell : this widget is used to make the card interactive, it has an onTap callback that navigates to different pages of the app.

Text : this widget is used to display text.

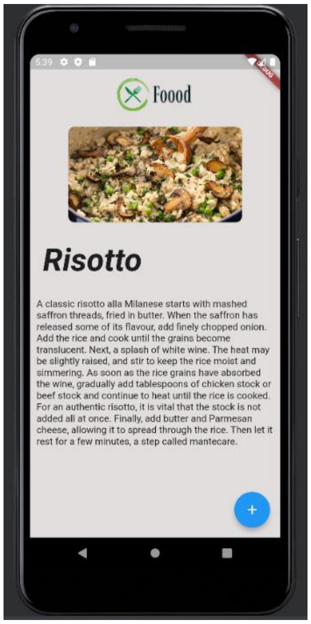
Ink.image : this widget is used to create an image with ink splash effect on tap.

SizedBox : this widget is used to add space between the widgets.

this page contains several sections that each section is displaying a different type of food, and each section contains an image and text, and when the user taps on the image it navigates to the corresponding food page.



"westren" screen which displays different types of Western cuisine and allowing the user to navigate to different screens for each type (Italian, French, Spanish, Greek). The screen includes a horizontal list of buttons for each type of cuisine, and an image and a Material button that leads to the "Risotto" recipe with the name and recipe stored in the Rglobals file. The screen also includes a drawer for navigation.



Risotto

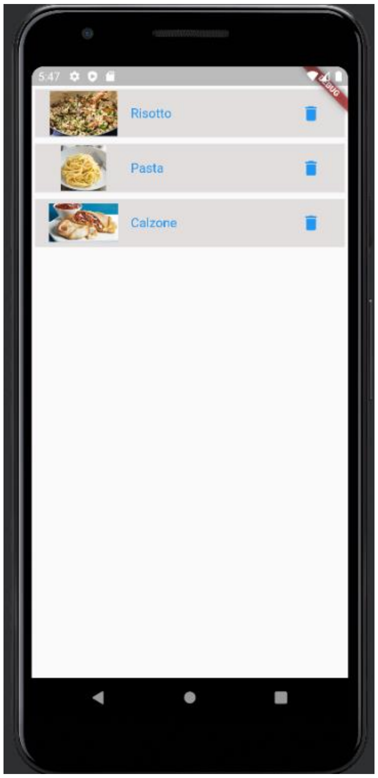
A classic risotto alla Milanese starts with mashed saffron threads, fried in butter. When the saffron has released some of its flavour, add finely chopped onion. Add the rice and cook until the grains become translucent. Next, a splash of white wine. The heat may be slightly raised, and stir to keep the rice moist and simmering. As soon as the rice grains have absorbed the wine, gradually add tablespoons of chicken stock or beef stock and continue to heat until the rice is cooked. For an authentic risotto, it is vital that the stock is not added all at once. Finally, add butter and Parmesan cheese, allowing it to spread through the rice. Then let it rest for a few minutes, a step called mantecare.

Recipes page:

It uses the Scaffold widget to create the basic layout of the page, and includes a FloatingActionButton that, when pressed, add the recipe to the favorite list.

The body of the Scaffold is a ListView that contains a Container with a nested ListView of other Container widgets. These containers hold various widgets such as text, images, and buttons. One of the containers is using the Expanded widget with a SingleChildScrollView to allow for scrolling of the text within it.

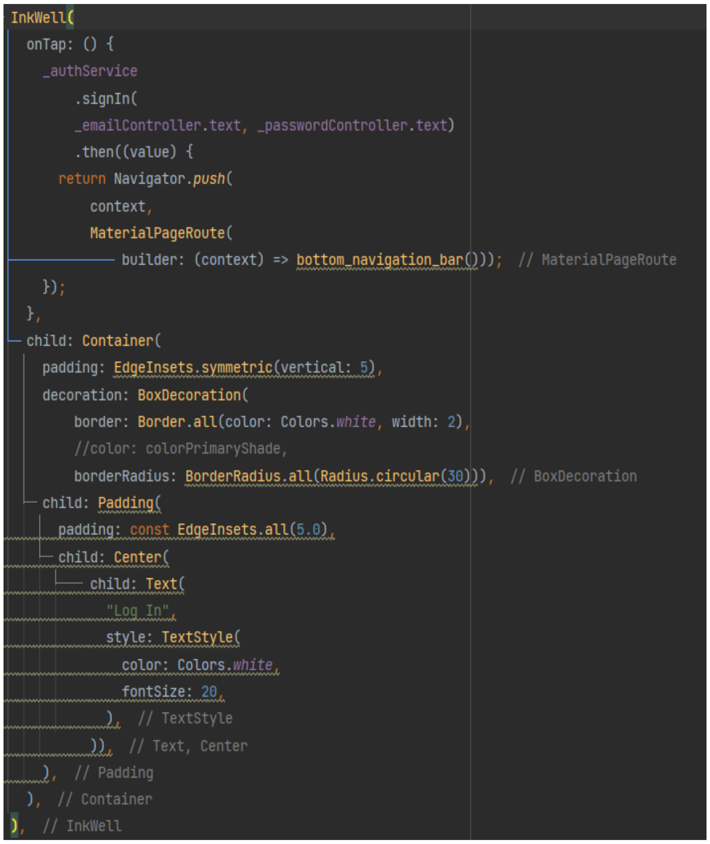
It also has a NavigationDrawer that is called on this page.



Favorite page:

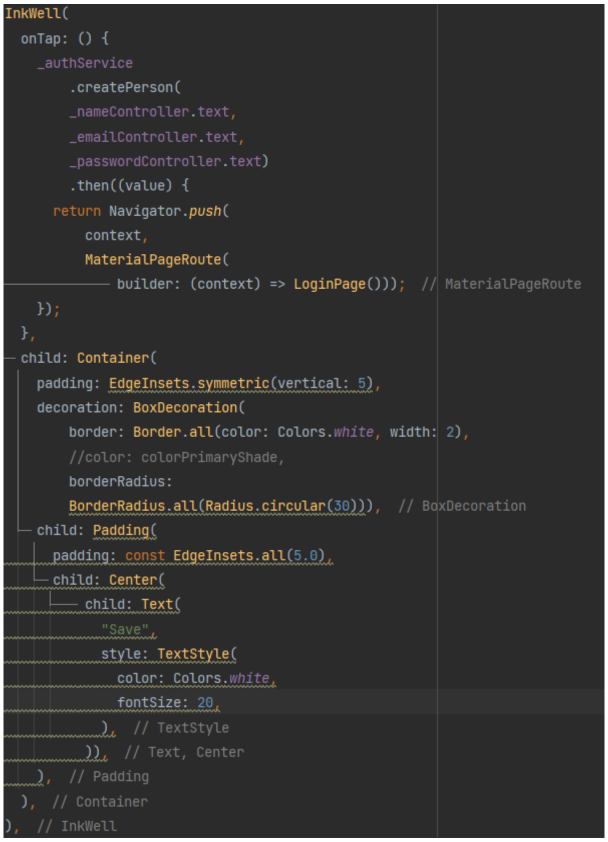
The favorite class is a stateful widget that displays a list of items that have been marked as favorites by the user. The FavoriteList class, which is a stateful widget, is the main widget that is displayed in the Scaffold. The UI is made up of a ListView that displays a list of items that have been marked as favorites. Each item in the list is represented by a ListTile, which is a widget that contains a leading image, a title, and a trailing icon button. The leading image is an asset, the title is the name of the recipe and the trailing icon button is the delete button. If the user taps on the delete button, the recipe will be removed from the list of favorites. If the user taps on the recipe, it will be directed to the recipe page and the recipe will be displayed.

Backend:



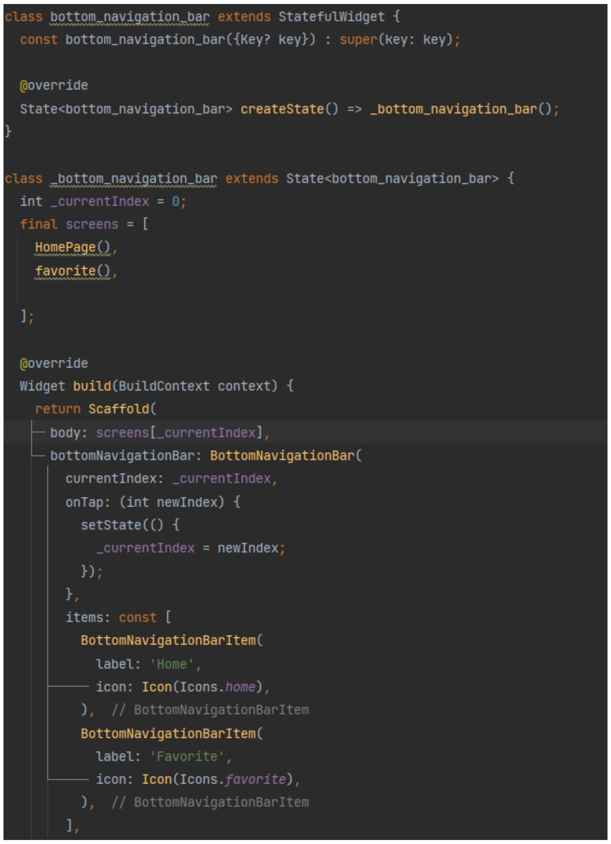
```dart
InkWell(
  onTap: () {
    _authService
        .signIn(
        _emailController.text, _passwordController.text)
        .then((value) {
      return Navigator.push(
          context,
          MaterialPageRoute(
              builder: (context) => bottom_navigation_bar()));  // MaterialPageRoute
    });
  },
  child: Container(
    padding: EdgeInsets.symmetric(vertical: 5),
    decoration: BoxDecoration(
        border: Border.all(color: Colors.white, width: 2),
        //color: colorPrimaryShade,
        borderRadius: BorderRadius.all(Radius.circular(30))),  // BoxDecoration
    child: Padding(
      padding: const EdgeInsets.all(5.0),
      child: Center(
          child: Text(
            "Log In",
            style: TextStyle(
              color: Colors.white,
              fontSize: 20,
            ),  // TextStyle
          )),  // Text, Center
    ),  // Padding
  ),  // Container
),  // InkWell
```

This code is a part of the login page of a Flutter app. It creates an "InkWell" widget. When the InkWell is tapped, it calls the "signIn" function of an instance of _authService, passing in the text values of two TextFields (_emailController and _passwordController), which are expected to be the email and password respectively entered by the user. If the sign in is successful, it then navigates to the bottom_navigation_bar() page.



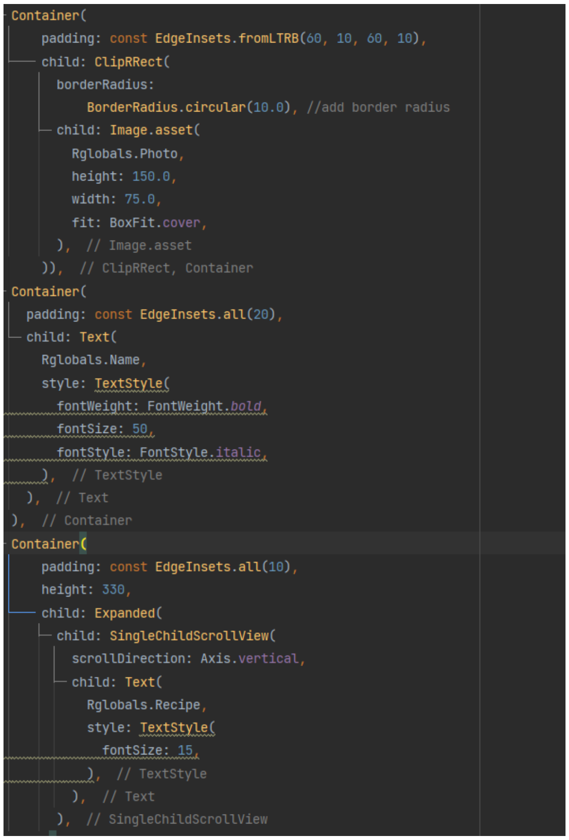
```dart
InkWell(
  onTap: () {
    _authService
        .createPerson(
        _nameController.text,
        _emailController.text,
        _passwordController.text)
        .then((value) {
      return Navigator.push(
          context,
          MaterialPageRoute(
              builder: (context) => LoginPage()));  // MaterialPageRoute
    });
  },
  child: Container(
    padding: EdgeInsets.symmetric(vertical: 5),
    decoration: BoxDecoration(
        border: Border.all(color: Colors.white, width: 2),
        //color: colorPrimaryShade,
        borderRadius:
        BorderRadius.all(Radius.circular(30))),  // BoxDecoration
    child: Padding(
      padding: const EdgeInsets.all(5.0),
      child: Center(
          child: Text(
            "Save",
            style: TextStyle(
              color: Colors.white,
              fontSize: 20,
            ),  // TextStyle
          )),  // Text, Center
    ),  // Padding
  ),  // Container
),  // InkWell
```

This code is a widget that is used for creating a new account in the app, when the user taps on the "Save" button, it will call the createPerson method from _authService object which is an authentication service. The method takes three arguments, _nameController.text, _emailController.text, and _passwordController.text which are the user's name, email and password that the user inputs in the form.



```
class bottom_navigation_bar extends StatefulWidget {
  const bottom_navigation_bar({Key? key}) : super(key: key);

  @override
  State<bottom_navigation_bar> createState() => _bottom_navigation_bar();
}

class _bottom_navigation_bar extends State<bottom_navigation_bar> {
  int _currentIndex = 0;
  final screens = [
    HomePage(),
    favorite(),

  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: screens[_currentIndex],
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _currentIndex,
        onTap: (int newIndex) {
          setState(() {
            _currentIndex = newIndex;
          });
        },
        items: const [
          BottomNavigationBarItem(
            label: 'Home',
            icon: Icon(Icons.home),
          ),  // BottomNavigationBarItem
          BottomNavigationBarItem(
            label: 'Favorite',
            icon: Icon(Icons.favorite),
          ),  // BottomNavigationBarItem
        ],
```

This code defines a class called bottom_navigation_bar which is a stateful widget that creates a bottom navigation bar. The bottom navigation bar displays two items: "Home" and "Favorite". When the user taps on one of the items, it sets the current index to the tapped index and updates the displayed screen accordingly. The screens that are displayed are defined in the screens list which includes HomePage() and favorite(). The HomePage() is the first screen that is displayed when the app is opened.



```
Container(
    padding: const EdgeInsets.fromLTRB(60, 10, 60, 10),
    child: ClipRRect(
      borderRadius:
          BorderRadius.circular(10.0), //add border radius
      child: Image.asset(
        Rglobals.Photo,
        height: 150.0,
        width: 75.0,
        fit: BoxFit.cover,
      ),  // Image.asset
    )),  // ClipRRect, Container
Container(
  padding: const EdgeInsets.all(20),
  child: Text(
    Rglobals.Name,
    style: TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 50,
      fontStyle: FontStyle.italic,
    ),  // TextStyle
  ),  // Text
),  // Container
Container(
    padding: const EdgeInsets.all(10),
    height: 330,
    child: Expanded(
      child: SingleChildScrollView(
        scrollDirection: Axis.vertical,
        child: Text(
          Rglobals.Recipe,
          style: TextStyle(
            fontSize: 15,
          ),  // TextStyle
        ),  // Text
      ),  // SingleChildScrollView
```

This code creates three containers in a row to form the recipes page. The first container display an image wich is loaded from an asset file specified by the Rglobals.Photo variable (wich is the recipe photo). The second container displays text wich is the value of the Rglobals.Name variable (wich is the recipe name). The third container creates an Expanded widget that contains a SingleChildScrollView widget which allows the recipe text to be scrolled vertically. The recipe text is taken from Rglobals.Recipe variable.