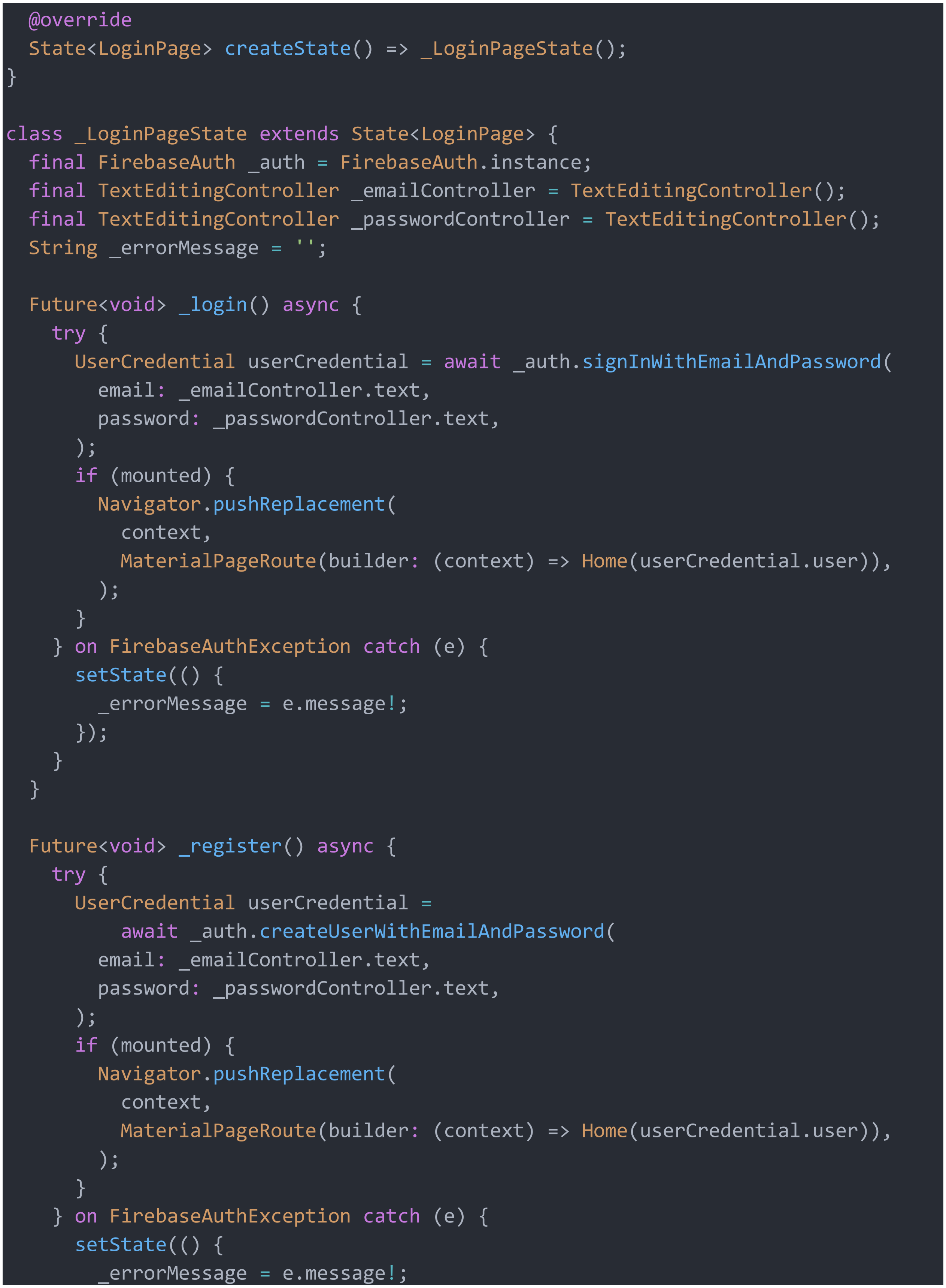
Main.dart

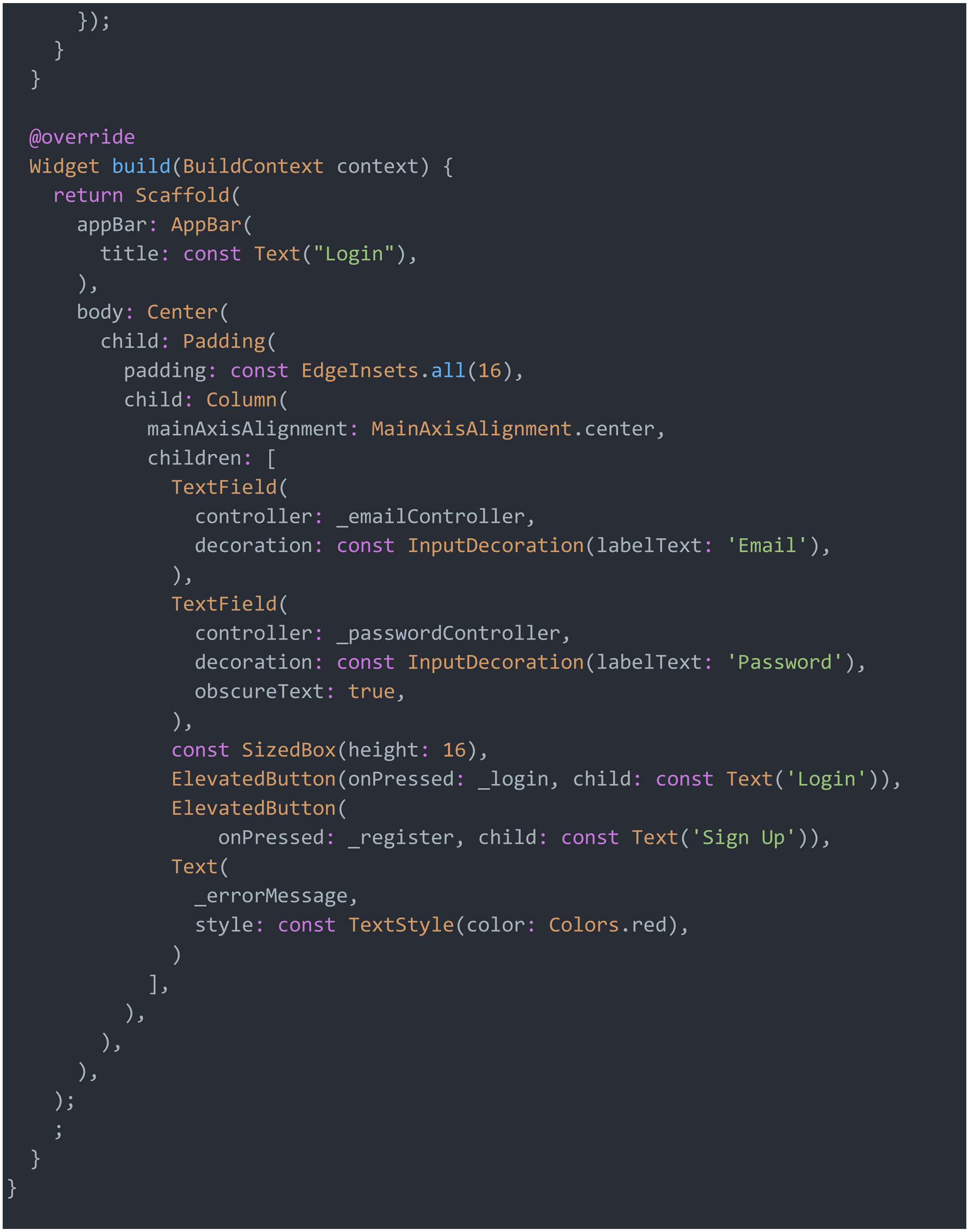
```
import 'package:ctse_lab03_inclass_01/screens/Home.dart';
import 'package:ctse_lab03_inclass_01/screens/login_page.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Recipe App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const LoginPage(),
    );
  }
}
```

login_page.dart

```
import 'package:ctse_lab03_inclass_01/screens/home.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';

class LoginPage extends StatefulWidget {
  const LoginPage({super.key});
```

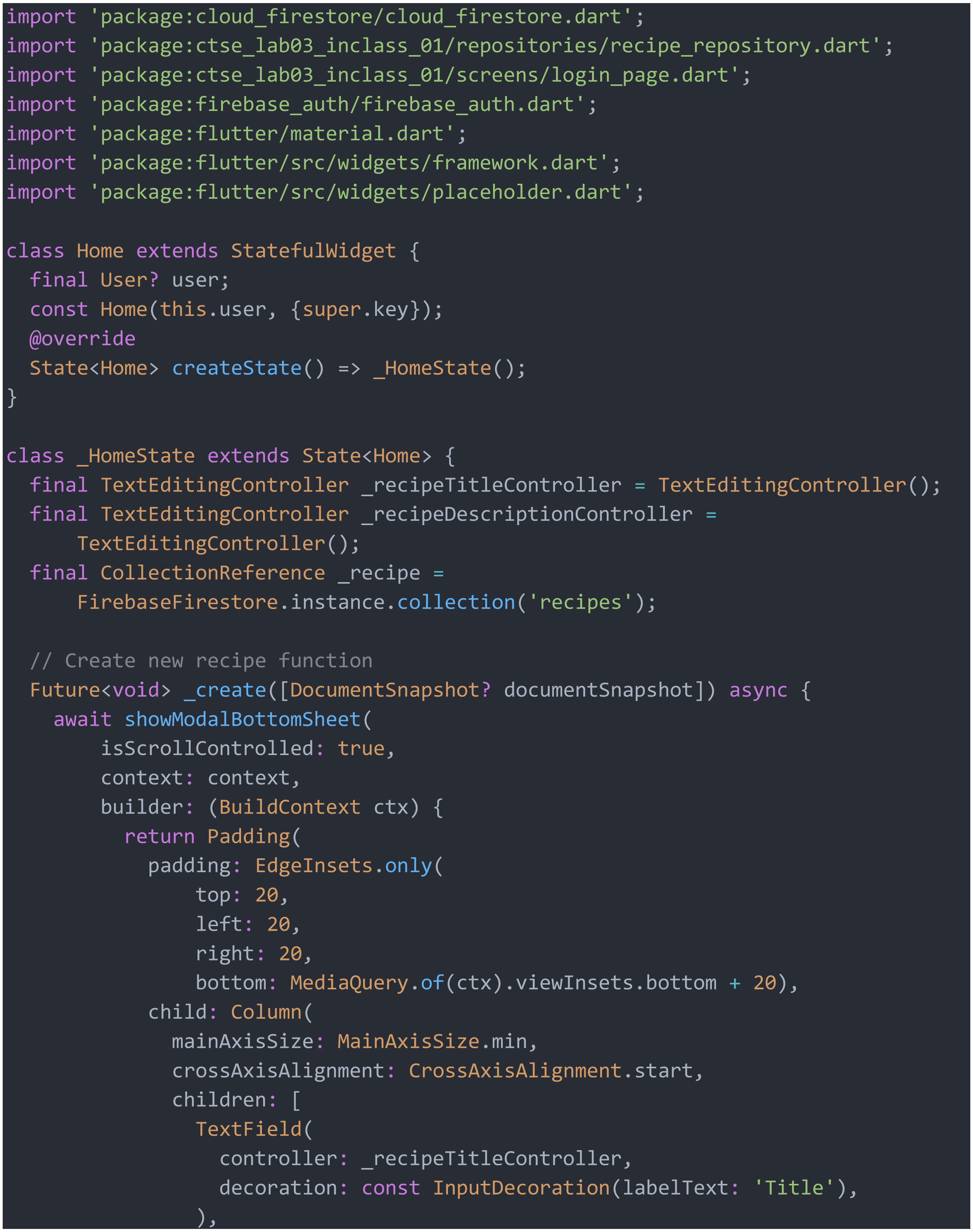


```
  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  String _errorMessage = '';

  Future<void> _login() async {
    try {
      UserCredential userCredential = await _auth.signInWithEmailAndPassword(
        email: _emailController.text,
        password: _passwordController.text,
      );
      if (mounted) {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => Home(userCredential.user)),
        );
      }
    } on FirebaseAuthException catch (e) {
      setState(() {
        _errorMessage = e.message!;
      });
    }
  }

  Future<void> _register() async {
    try {
      UserCredential userCredential =
          await _auth.createUserWithEmailAndPassword(
        email: _emailController.text,
        password: _passwordController.text,
      );
      if (mounted) {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => Home(userCredential.user)),
        );
      }
    } on FirebaseAuthException catch (e) {
      setState(() {
        _errorMessage = e.message!;
```

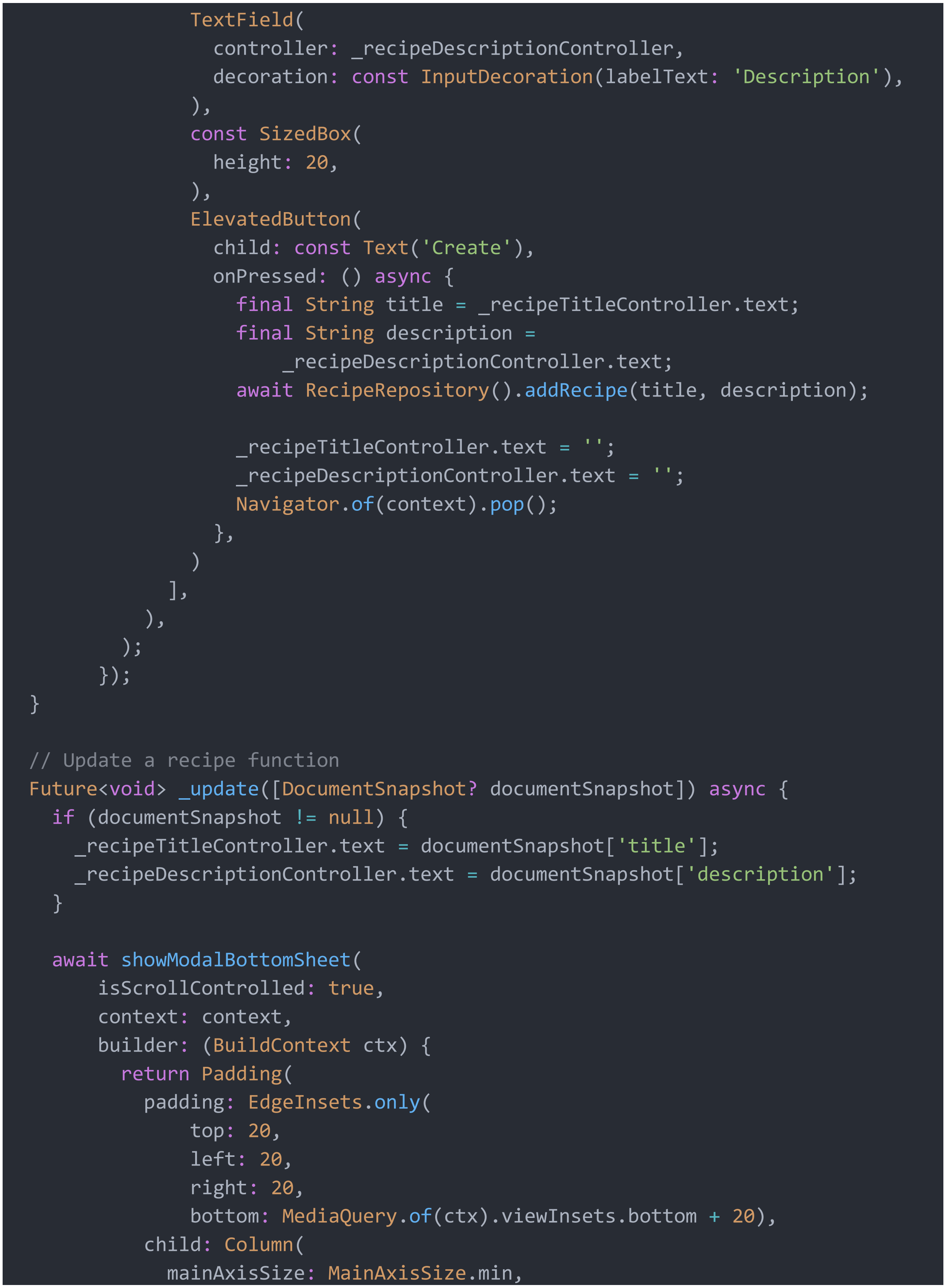


```dart
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Login"),
      ),
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(16),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              TextField(
                controller: _emailController,
                decoration: const InputDecoration(labelText: 'Email'),
              ),
              TextField(
                controller: _passwordController,
                decoration: const InputDecoration(labelText: 'Password'),
                obscureText: true,
              ),
              const SizedBox(height: 16),
              ElevatedButton(onPressed: _login, child: const Text('Login')),
              ElevatedButton(
                  onPressed: _register, child: const Text('Sign Up')),
              Text(
                _errorMessage,
                style: const TextStyle(color: Colors.red),
              )
            ],
          ),
        ),
      ),
    );
    ;
  }
}
```

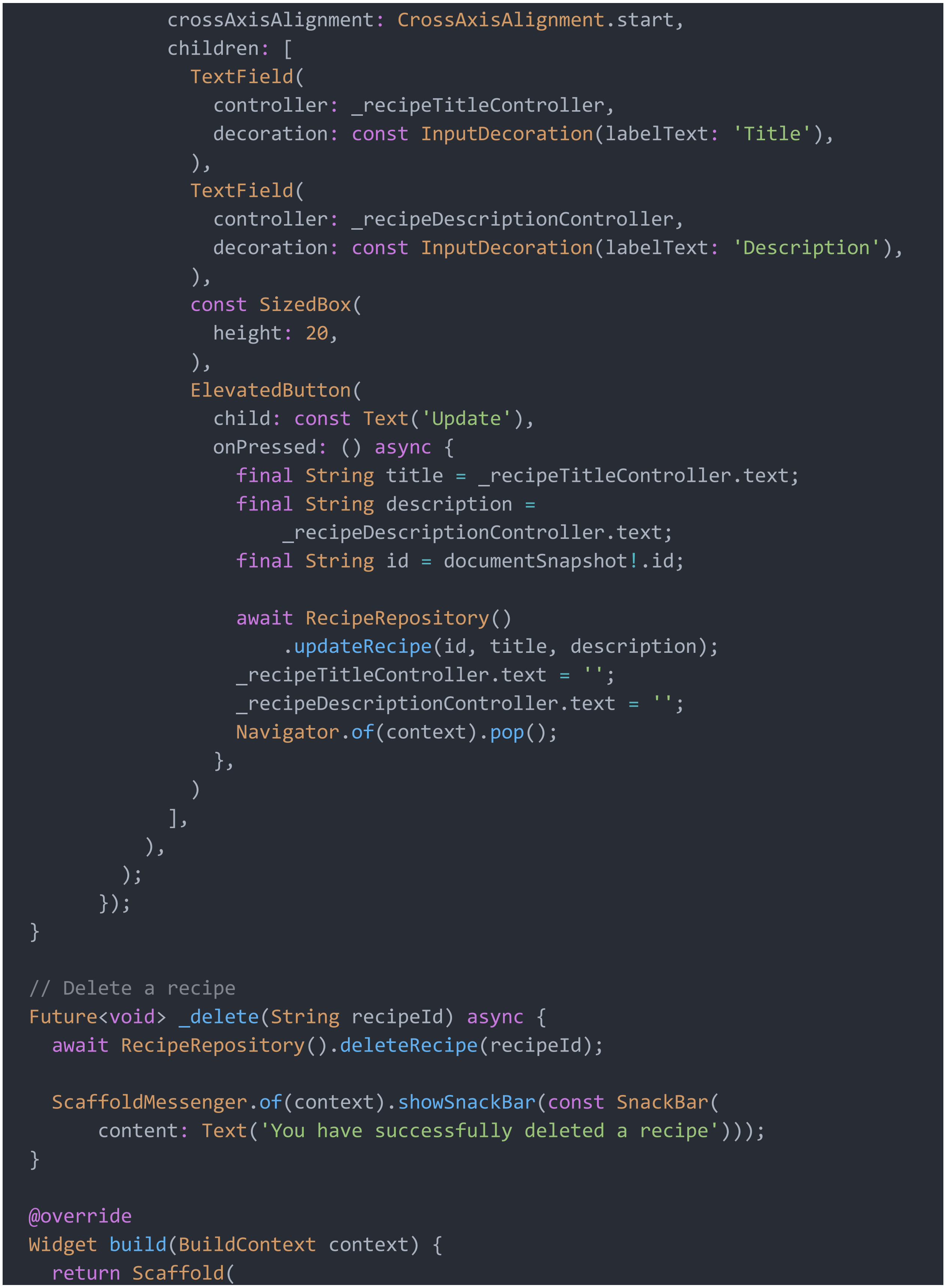
Home.dart



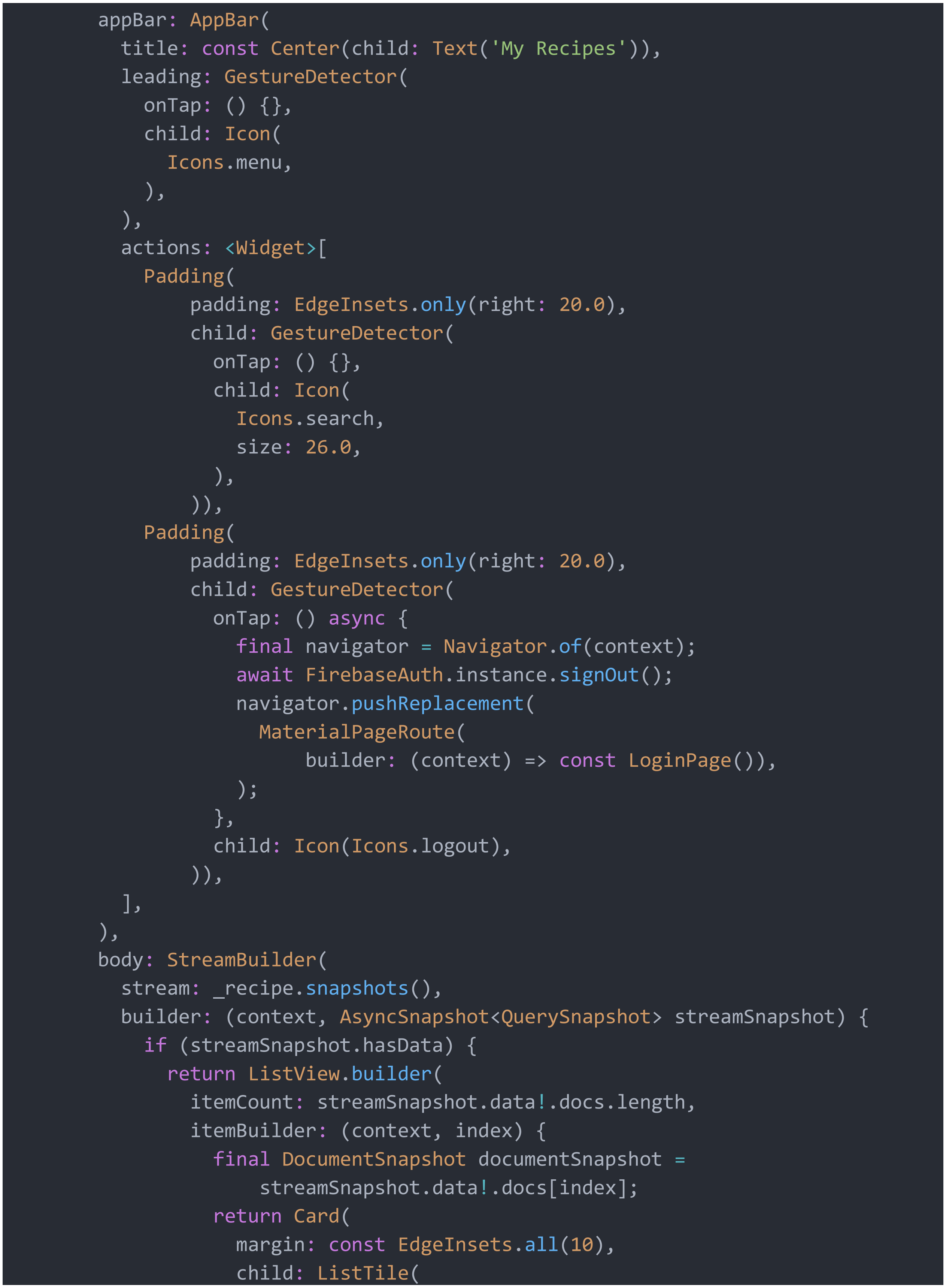
```dart
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:ctse_lab03_inclass_01/repositories/recipe_repository.dart';
import 'package:ctse_lab03_inclass_01/screens/login_page.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';

class Home extends StatefulWidget {
  final User? user;
  const Home(this.user, {super.key});
  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  final TextEditingController _recipeTitleController = TextEditingController();
  final TextEditingController _recipeDescriptionController =
      TextEditingController();
  final CollectionReference _recipe =
      FirebaseFirestore.instance.collection('recipes');

  // Create new recipe function
  Future<void> _create([DocumentSnapshot? documentSnapshot]) async {
    await showModalBottomSheet(
        isScrollControlled: true,
        context: context,
        builder: (BuildContext ctx) {
          return Padding(
            padding: EdgeInsets.only(
                top: 20,
                left: 20,
                right: 20,
                bottom: MediaQuery.of(ctx).viewInsets.bottom + 20),
            child: Column(
              mainAxisSize: MainAxisSize.min,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                TextField(
                  controller: _recipeTitleController,
                  decoration: const InputDecoration(labelText: 'Title'),
                ),
```



```
              TextField(
                controller: _recipeDescriptionController,
                decoration: const InputDecoration(labelText: 'Description'),
              ),
              const SizedBox(
                height: 20,
              ),
              ElevatedButton(
                child: const Text('Create'),
                onPressed: () async {
                  final String title = _recipeTitleController.text;
                  final String description =
                      _recipeDescriptionController.text;
                  await RecipeRepository().addRecipe(title, description);

                  _recipeTitleController.text = '';
                  _recipeDescriptionController.text = '';
                  Navigator.of(context).pop();
                },
              )
            ],
          ),
        );
      });
}

// Update a recipe function
Future<void> _update([DocumentSnapshot? documentSnapshot]) async {
  if (documentSnapshot != null) {
    _recipeTitleController.text = documentSnapshot['title'];
    _recipeDescriptionController.text = documentSnapshot['description'];
  }

  await showModalBottomSheet(
      isScrollControlled: true,
      context: context,
      builder: (BuildContext ctx) {
        return Padding(
          padding: EdgeInsets.only(
              top: 20,
              left: 20,
              right: 20,
              bottom: MediaQuery.of(ctx).viewInsets.bottom + 20),
          child: Column(
            mainAxisSize: MainAxisSize.min,
```



```
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              TextField(
                controller: _recipeTitleController,
                decoration: const InputDecoration(labelText: 'Title'),
              ),
              TextField(
                controller: _recipeDescriptionController,
                decoration: const InputDecoration(labelText: 'Description'),
              ),
              const SizedBox(
                height: 20,
              ),
              ElevatedButton(
                child: const Text('Update'),
                onPressed: () async {
                  final String title = _recipeTitleController.text;
                  final String description =
                      _recipeDescriptionController.text;
                  final String id = documentSnapshot!.id;

                  await RecipeRepository()
                      .updateRecipe(id, title, description);
                  _recipeTitleController.text = '';
                  _recipeDescriptionController.text = '';
                  Navigator.of(context).pop();
                },
              )
            ],
          ),
        );
      });
}

// Delete a recipe
Future<void> _delete(String recipeId) async {
  await RecipeRepository().deleteRecipe(recipeId);

  ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
      content: Text('You have successfully deleted a recipe')));
}

@override
Widget build(BuildContext context) {
  return Scaffold(
```



```
        appBar: AppBar(
          title: const Center(child: Text('My Recipes')),
          leading: GestureDetector(
            onTap: () {},
            child: Icon(
              Icons.menu,
            ),
          ),
          actions: <Widget>[
            Padding(
                padding: EdgeInsets.only(right: 20.0),
                child: GestureDetector(
                  onTap: () {},
                  child: Icon(
                    Icons.search,
                    size: 26.0,
                  ),
                )),
            Padding(
                padding: EdgeInsets.only(right: 20.0),
                child: GestureDetector(
                  onTap: () async {
                    final navigator = Navigator.of(context);
                    await FirebaseAuth.instance.signOut();
                    navigator.pushReplacement(
                      MaterialPageRoute(
                          builder: (context) => const LoginPage()),
                    );
                  },
                  child: Icon(Icons.logout),
                )),
          ],
        ),
        body: StreamBuilder(
          stream: _recipe.snapshots(),
          builder: (context, AsyncSnapshot<QuerySnapshot> streamSnapshot) {
            if (streamSnapshot.hasData) {
              return ListView.builder(
                itemCount: streamSnapshot.data!.docs.length,
                itemBuilder: (context, index) {
                  final DocumentSnapshot documentSnapshot =
                      streamSnapshot.data!.docs[index];
                  return Card(
                    margin: const EdgeInsets.all(10),
                    child: ListTile(
                      title: Text(documentSnapshot['title']),
                      subtitle: Text(documentSnapshot['description']),
                      trailing: SizedBox(
                        width: 100,
                        child: Row(
                          children: [
                            IconButton(
                                icon: const Icon(Icons.edit),
                                onPressed: () => _update(documentSnapshot)),
                            IconButton(
                                icon: const Icon(Icons.delete),
                                onPressed: () => _delete(documentSnapshot.id)),
                          ],
                        ),
                      ),
                    ),
                  );
                },
              );
            }

            return const Center(
              child: CircularProgressIndicator(),
            );
          },
        ),
        floatingActionButton: FloatingActionButton(
          onPressed: () => _create(),
          child: const Icon(Icons.add),
        ),
        floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat);
  }
}
```

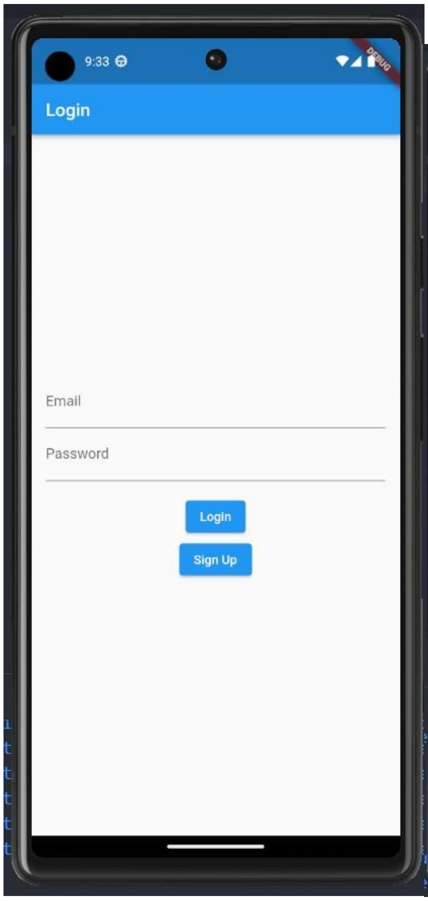
Recipe_repository.dart

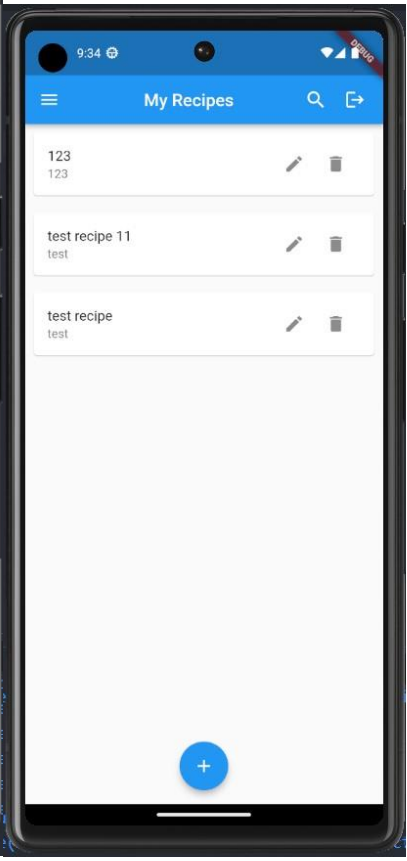
```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/cupertino.dart';

class RecipeRepository {
  final CollectionReference _recipes =
      FirebaseFirestore.instance.collection('recipes');

  // Get all recipes
  Future getAllRecipes() async {
    return _recipes;
  }

  // Add new recipe
  Future addRecipe(String title, String description) async {
    return await _recipes.add({
      "title": title,
      "description": description,
      "ingredients": ["sample01", "sample02"],
    });
  }

  // Update recipe
  Future<void> updateRecipe(String id, String title, String description) async {
    return await _recipes.doc(id).update({
      "title": title,
      "description": description,
      "ingredients": ["sample01Updated", "sample02Updated"],
    });
  }

  // Delete recipe
  Future deleteRecipe(id) async {
    await _recipes.doc(id).delete();
  }
}
```

Recipe.dart

```
import 'dart:ffi';

class Recipe {
  String title;
  String description;
  Array ingredients;

  Recipe({
    required this.title,
    required this.description,
    required this.ingredients,
  });
}
```

SCREEN SHOTS:

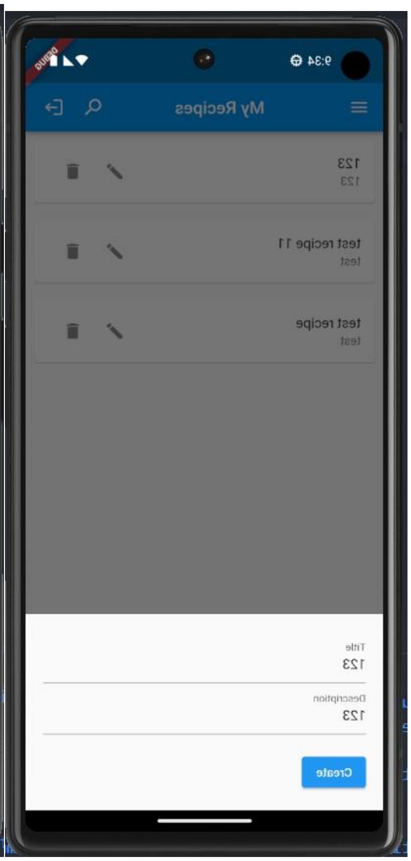




123
123

test recipe 11
test

test recipe
test



123
123

test recipe 11
test

test recipe
test

Title
123

Description
123

Repo Link : https://github.com/ManulThisuraka/recipe-app-in-class-lab