IT20238612

H.A. Hasith Deminda

main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:lab_assignment/screen/loginPage.dart';
import 'firebase_options.dart';

//Async main function
void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );

  runApp(const MyApp());
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Lab Assignment | Recipe App',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      home: const LoginPage(),
    );
```

```
 }
}
```

loginPage.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lab_assignment/screen/recipeScreens/displayRecipe.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) => const RecipePage()));
   }
  } on FirebaseAuthException catch (e) {
```

```
    setState(() {
      _errorMessage = e.message!;
    });
  }
}

Future<void> _register() async {
  try {
    UserCredential userCredential =
        await _auth.createUserWithEmailAndPassword(
      email: _emailController.text,
      password: _passwordController.text,
    );

    setState(() {
      _emailController.clear();
      _passwordController.clear();
    });

    _showDialog('Message', 'User Registered Successfully!');
  } on FirebaseAuthException catch (e) {
    setState(() {
      _errorMessage = e.message!;
    });
  }
}

//show dialog box
Future<void> _showDialog(String type, String message) async {
  return showDialog<void>(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text(type),
```

```
      content: Text(message),
      actions: [
        TextButton(
          onPressed: () => Navigator.of(context).pop(),
          child: const Text('OK'),
        ),
      ],
    );
  },
 );
}

@override
Widget build(BuildContext context) {
 return Scaffold(
   appBar: AppBar(
    title: const Text('Login | Register '),
    titleTextStyle: const TextStyle(color: Colors.white, letterSpacing: 2),
   ),
   body: Padding(
    padding: const EdgeInsets.all(8.0),
    child: Column(
     children: [
       TextField(
         controller: _emailController,
         decoration: const InputDecoration(
           suffixIcon: Icon(Icons.email),
           border: OutlineInputBorder(),
           labelText: 'Email',
         ),
       ),
       const SizedBox(
         height: 10,
       ),
```

```
TextField(
  controller: _passwordController,
  decoration: const InputDecoration(
    suffixIcon: Icon(Icons.lock),
    border: OutlineInputBorder(),
    labelText: 'Password',
  ),
  obscureText: true,
),
const SizedBox(
  height: 10,
),
Text(
  _errorMessage,
  style: const TextStyle(color: Colors.red),
),
const SizedBox(
  height: 10,
),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    ElevatedButton(
      onPressed: _login,
      child: const Text('Login'),
    ),
    const SizedBox(
      width: 10,
    ),
    ElevatedButton(
      onPressed: _register,
      child: const Text('Register'),
    ),
  ],
```

```
          )
        ],
      ),
    ),
  );
 }
}
```

addRecipe.dart

```
import 'package:flutter/material.dart';
import 'package:lab_assignment/model/recipe.dart';
import 'package:lab_assignment/repository/recipeRepository.dart';
import 'package:lab_assignment/screen/recipeScreens/displayRecipe.dart';

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

 @override
 State<AddRecipePage> createState() => _AddRecipePageState();
}

class _AddRecipePageState extends State<AddRecipePage> {
 List<String> ingredients = [];
 final TextEditingController _titleController = TextEditingController();
 final TextEditingController _descriptionController = TextEditingController();
 final TextEditingController _ingredientsController = TextEditingController();

 final RecipeRepository _recipeRepository = RecipeRepository();
 void _addRecipe(Recipe recipe) {
  setState(() {
   _recipeRepository.addRecipe(recipe);
  });

  Navigator.push(context, MaterialPageRoute(builder: (context) {
```

```
   return const RecipePage();
  }));
}

@override
Widget build(BuildContext context) {
  return Scaffold(
      appBar: AppBar(
        title: const Text('Add a Recipe'),
        titleTextStyle:
            const TextStyle(color: Colors.white, letterSpacing: 2),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8),
        child: Column(
          children: [
            TextField(
              controller: _titleController,
              decoration: const InputDecoration(
                  border: OutlineInputBorder(),
                  labelText: 'Title',
                  hintText: "Enter title"),
            ),
            const SizedBox(
              height: 10,
            ),
            TextField(
              controller: _descriptionController,
              decoration: const InputDecoration(
                  border: OutlineInputBorder(),
                  labelText: 'Description',
                  hintText: "Enter description"),
            ),
            const SizedBox(
```

```
  height: 10,
),
TextField(
  controller: _ingredientsController,
  decoration: const InputDecoration(
     border: OutlineInputBorder(),
     labelText: 'Ingredients',
     hintText: "Enter ingredients"),
),
const SizedBox(
  height: 10,
),
ElevatedButton(
  onPressed: () {
    setState(() {
      ingredients.add(_ingredientsController.text);
      _ingredientsController.clear();
    });
  },
  child: const Text('Add Ingredient'),
),
const SizedBox(
  height: 10,
),
Text("Enterd Ingredient :  ${ingredients.toString()}"),
const SizedBox(
  height: 10,
),
ElevatedButton(
  onPressed: () {
    _addRecipe(Recipe(
       title: _titleController.text,
       description: _descriptionController.text,
       ingredients: ingredients));
```

```
            },
            child: const Text('Save Recipe'),
          ),
        ],
      ),
    ));
  }
}
```

displayRecipe.dart

```
import 'package:flutter/material.dart';
import 'package:lab_assignment/model/recipe.dart';
import 'package:lab_assignment/repository/recipeRepository.dart';
import 'package:lab_assignment/screen/recipeScreens/addRecipe.dart';
import 'package:lab_assignment/screen/recipeScreens/oneRecipe.dart';
import 'package:lab_assignment/screen/recipeScreens/updateRecipe.dart';

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

  @override
  State<RecipePage> createState() => _RecipePageState();
}

class _RecipePageState extends State<RecipePage> {
  @override
  Widget build(BuildContext context) {
    final RecipeRepository recipeRepository = RecipeRepository();

    //delete function
    void _DeleteRecipe(String id) {
      setState(() {
        recipeRepository.deleteRecipe(id);
      });
```

```
}

return Scaffold(
  appBar: AppBar(
    title: const Text('All Recipes List'),
    titleTextStyle: const TextStyle(color: Colors.white, letterSpacing: 2),
  ),
  body: Padding(
    padding: const EdgeInsets.all(8.0),
    child: Column(
      children: [
        Expanded(
          child: FutureBuilder<List<Recipe>>(
            future: recipeRepository.getAllRecipes(),
            builder: (context, snapshot) {
              if (snapshot.hasData) {
                final List<Recipe>? recipes = snapshot.data;
                return ListView.builder(
                  itemCount: recipes!.length,
                  itemBuilder: (context, index) {
                    final recipe = recipes[index];
                    return Card(
                      elevation: 5,
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(10.0),
                      ),
                      child: ListTile(
                        title: Text(recipe.title.toString()),
                        subtitle: Text(recipe.description.toString()),
                        trailing: Row(
                          mainAxisSize: MainAxisSize.min,
                          children: [
                            IconButton(
                              onPressed: () {
```

```
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) =>
                            UpdateRecipe(recipe),
                      ),
                    );
                  },
                  icon: const Icon(Icons.edit,
                      color: Colors.blue),
                ),
                IconButton(
                  onPressed: () {
                    _DeleteRecipe(recipe.id.toString());
                  },
                  icon: const Icon(Icons.delete,
                      color: Colors.red),
                ),
              ],
            ),
          ),
        );
      },
    );
  } else if (snapshot.hasError) {
    return Text("${snapshot.error}");
  }
  return const CircularProgressIndicator();
},
),
),
],
),
),
```

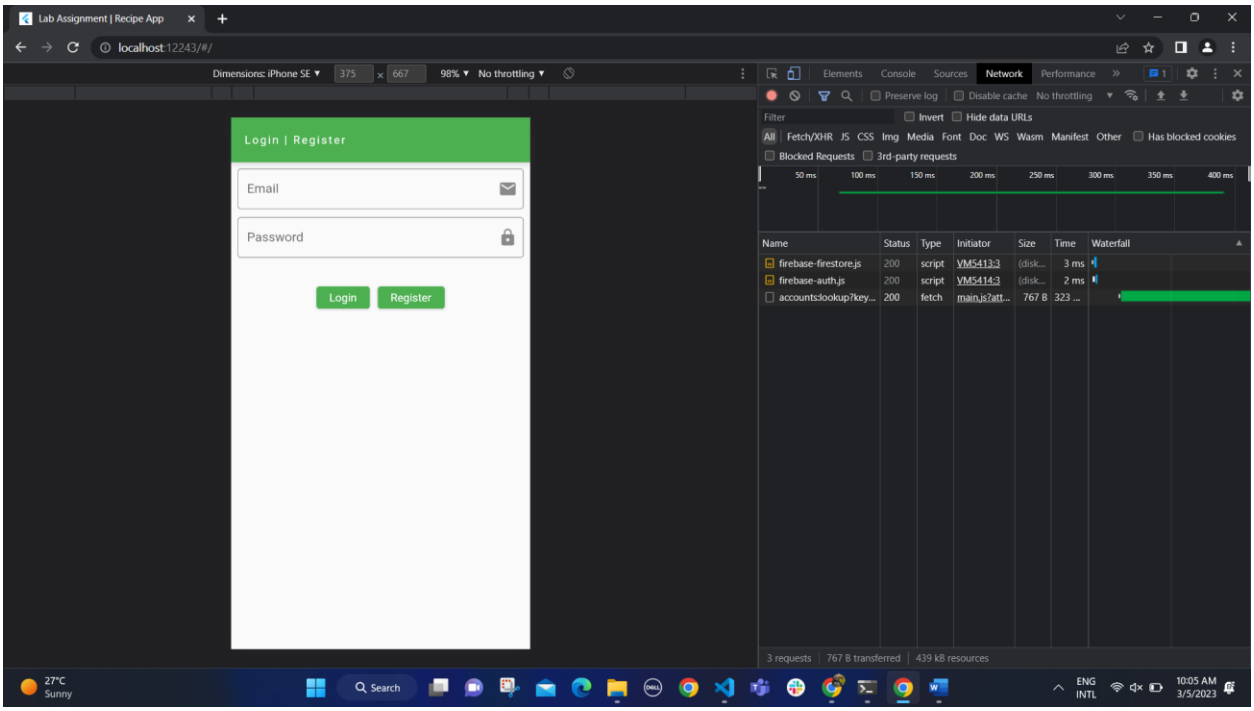
```
    //Add recipe button
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => const AddRecipePage(),
          ),
        );
      },
      child: const Icon(Icons.add, color: Colors.white),
    ),
   );
 }
}
```

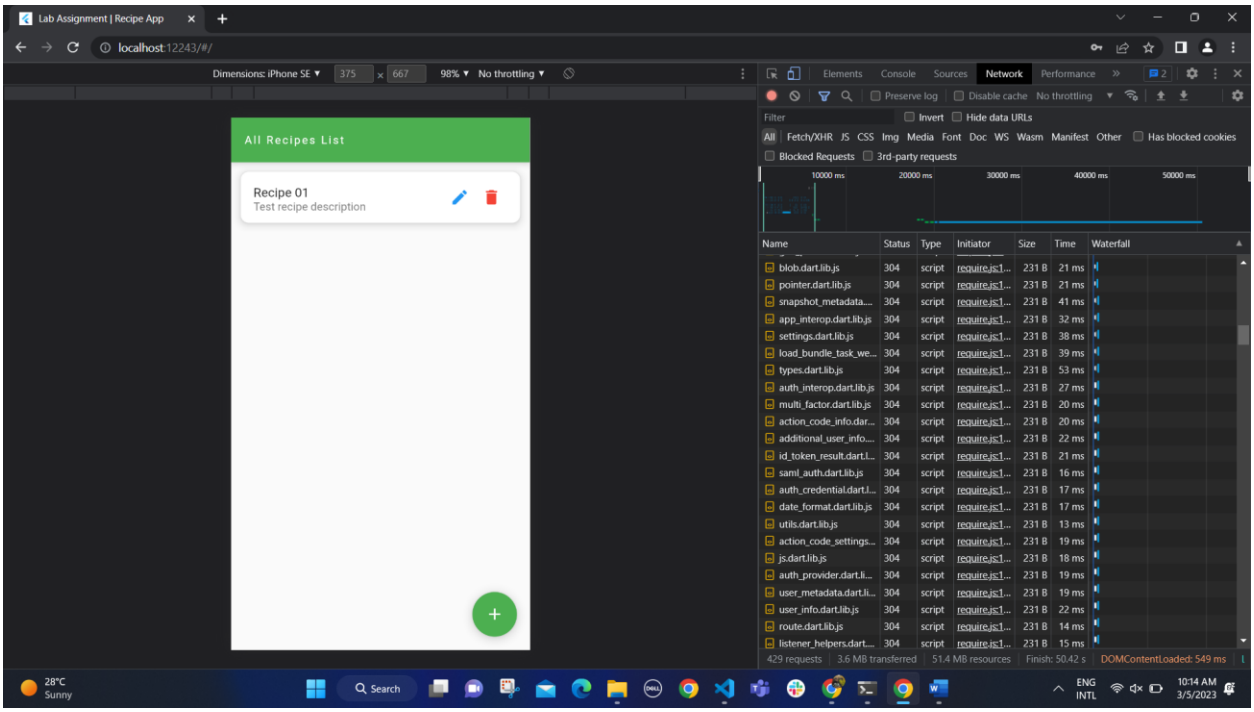
updateRecipe.dart

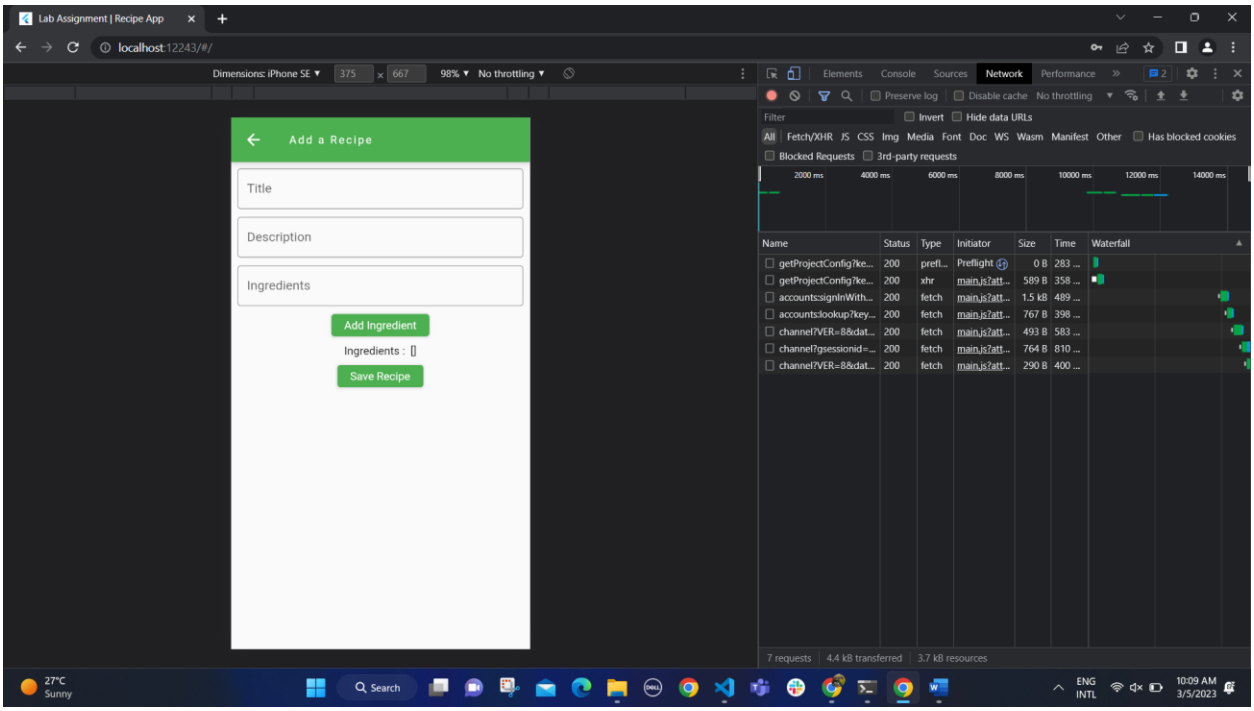
```
import 'package:flutter/material.dart';
import 'package:lab_assignment/model/recipe.dart';
import 'package:lab_assignment/repository/recipeRepository.dart';
import 'package:lab_assignment/screen/recipeScreens/displayRecipe.dart';

class UpdateRecipe extends StatefulWidget {
  final Recipe? recipe;
  const UpdateRecipe(this.recipe, {super.key});

  @override
  State<UpdateRecipe> createState() => _UpdateRecipeState();
}

class _UpdateRecipeState extends State<UpdateRecipe> {
  final TextEditingController _titleController = TextEditingController();
  final TextEditingController _descriptionController = TextEditingController();
  final RecipeRepository recipeRepository = RecipeRepository();
```

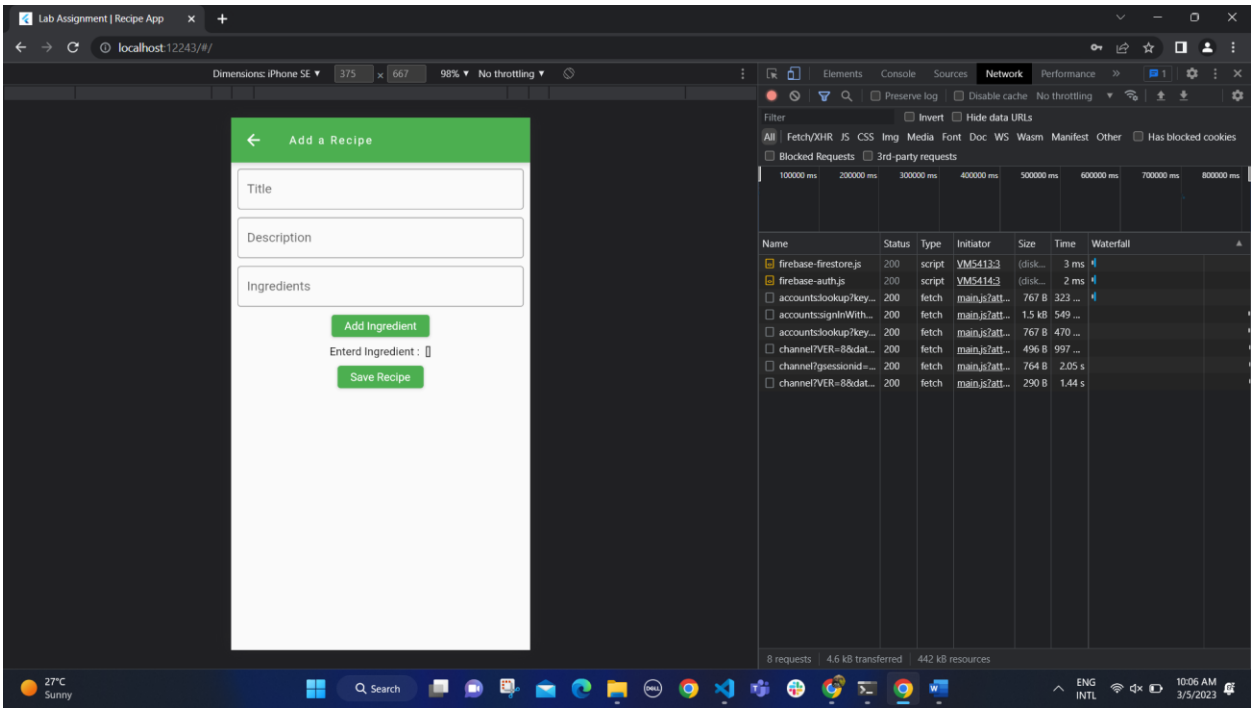
```
@override
Widget build(BuildContext context) {
 _titleController.text = widget.recipe!.title.toString();
 _descriptionController.text = widget.recipe!.description.toString();


 void _updateRecipe(Recipe recipe) {
  setState(() {
   recipeRepository.updateRecipe(recipe);
  });


  Navigator.push(context, MaterialPageRoute(builder: (context) {
   return const RecipePage();
  }));
 }


 return Scaffold(
  appBar: AppBar(
   title: const Text('Update Recipe'),
  ),
  body: Padding(
   padding: const EdgeInsets.all(8),
   child: Column(
    children: [
     TextField(
      controller: _titleController,
      decoration: const InputDecoration(
          border: OutlineInputBorder(),
          labelText: 'Title',
          hintText: "Enter title"),
     ),
     const SizedBox(
      height: 10,
     ),
```

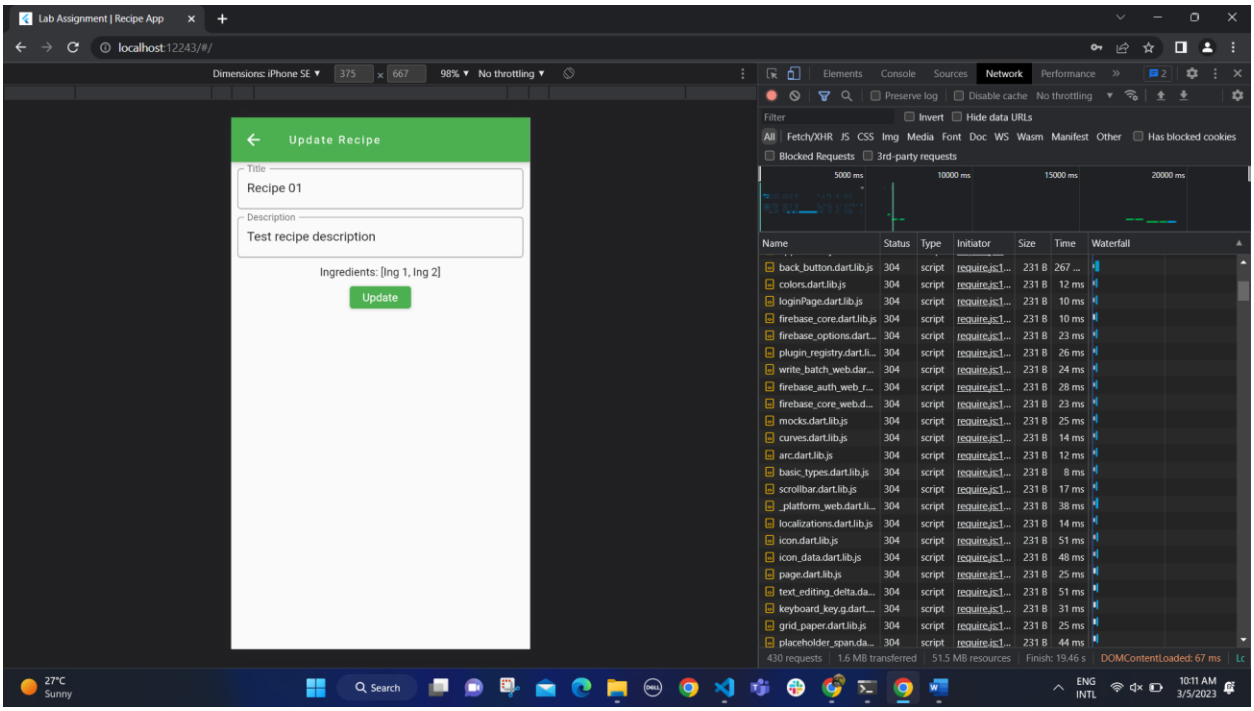
```
          TextField(
            controller: _descriptionController,
            decoration: const InputDecoration(
                border: OutlineInputBorder(),
                labelText: 'Description',
                hintText: "Enter description"),
          ),
          const SizedBox(
            height: 10,
          ),
          Text("Ingredients: ${widget.recipe!.ingredients}"),
          const SizedBox(
            height: 10,
          ),
          ElevatedButton(
            onPressed: () {
              _updateRecipe(Recipe(
                  id: widget.recipe!.id,
                  title: _titleController.text,
                  description: _descriptionController.text,
                  ingredients: widget.recipe!.ingredients));
            },
            child: const Text('Update Recipe'),
          ),
        ],
      ),
    ),
  );
 }
}
```


Lab Assignment | Recipe App
localhost:12243/#/
Login | Register
Email
Password
Login
Register
Elements
Console
Sources
Network
Performance
Name
Status
Type
Initiator
Size
Time
Waterfall
firebase-firestore.js
firebase-auth.js
accounts:lookup?key...
3 requests
767 B transferred
439 kB resources


Lab Assignment | Recipe App
localhost:12243/#/
All Recipes List
Recipe 01
Test recipe description
Elements
Console
Sources
Network
Performance
Name
Status
Type
Initiator
Size
Time
Waterfall
429 requests
3.6 MB transferred
51.4 MB resources
Finish: 50.42 s
DOMContentLoaded: 549 ms


Lab Assignment | Recipe App
localhost:12243/#/
Add a Recipe
Title
Description
Ingredients
Add Ingredient
Ingredients : []
Save Recipe
Elements Console Sources Network Performance
7 requests 4.4 kB transferred 3.7 kB resources


Lab Assignment | Recipe App
localhost:12243/#/
Add a Recipe
Title
Description
Ingredients
Add Ingredient
Enterd Ingredient : []
Save Recipe
Elements Console Sources Network Performance
8 requests 4.6 kB transferred 442 kB resources


Lab Assignment | Recipe App
localhost:12243/#/
Dimensions: iPhone SE
375
667
98%
No throttling
Update Recipe
Title
Recipe 01
Description
Test recipe description
Ingredients: [Ing 1, Ing 2]
Update
Elements
Console
Sources
Network
Performance
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
3rd-party requests
5000 ms
10000 ms
15000 ms
20000 ms
Name
Status
Type
Initiator
Size
Time
Waterfall
back_button.dart.lib.js 304 script require.js:1... 231 B 267 ...
colors.dart.lib.js 304 script require.js:1... 231 B 12 ms
loginPage.dart.lib.js 304 script require.js:1... 231 B 10 ms
firebase_core.dart.lib.js 304 script require.js:1... 231 B 10 ms
firebase_options.dart... 304 script require.js:1... 231 B 23 ms
plugin_registry.dart.li... 304 script require.js:1... 231 B 26 ms
write_batch_web.dar... 304 script require.js:1... 231 B 24 ms
firebase_auth_web_r... 304 script require.js:1... 231 B 28 ms
firebase_core_web.d... 304 script require.js:1... 231 B 23 ms
mocks.dart.lib.js 304 script require.js:1... 231 B 25 ms
curves.dart.lib.js 304 script require.js:1... 231 B 14 ms
arc.dart.lib.js 304 script require.js:1... 231 B 12 ms
basic_types.dart.lib.js 304 script require.js:1... 231 B 8 ms
scrollbar.dart.lib.js 304 script require.js:1... 231 B 17 ms
_platform_web.dart.li... 304 script require.js:1... 231 B 38 ms
localizations.dart.lib.js 304 script require.js:1... 231 B 14 ms
icon.dart.lib.js 304 script require.js:1... 231 B 51 ms
icon_data.dart.lib.js 304 script require.js:1... 231 B 48 ms
page.dart.lib.js 304 script require.js:1... 231 B 25 ms
text_editing_delta.da... 304 script require.js:1... 231 B 51 ms
keyboard_key.g.dart... 304 script require.js:1... 231 B 31 ms
grid_paper.dart.lib.js 304 script require.js:1... 231 B 25 ms
placeholder_span.da... 304 script require.js:1... 231 B 44 ms
430 requests
1.6 MB transferred
51.5 MB resources
Finish: 19.46 s
DOMContentLoaded: 67 ms
27°C
Sunny
Search
ENG
INTL
10:11 AM
3/5/2023