

Radiant Retail

Mobile App Development



Supervised By

Mr. Amjad Farooq

Submitted By

Fahad Karim Khan	2021-CS-14
Muhammad Tahir	2021-CS-34
Ali Haider	2021-CS-38

Department of Computer Science
University of Engineering and Technology, Lahore
Pakistan

Contents

1	Introduction	2
1.1	Background	2
1.2	Objective	2
1.3	Scope	2
1.4	Significance	2
2	Target Audience	2
3	Features	3
3.1	Admin Portal	3
3.2	User Portal	3
4	Screen Layout with Navigation Flow	4
5	Directory Structure	5
6	Packages and Widgets Used	6
7	Play Store URL	7
8	Brief Description of Database Integration via API's	7
9	Description of FireBase Database and Connectivity	7
10	Use Of Shared Preference	8
11	Conclusion	8
11.1	Summary of Achievements	8
11.2	Challenges Faced	8
11.3	Lessons Learned	8
11.4	Future Enhancements	8

1 Introduction

Radiant Retail, powered by Flutter and Dart, stands as an advanced e-commerce mobile application. This documentation offers a thorough examination of Radiant Retail, focusing on its background, objectives, scope, and significance within the realm of next-generation shopping applications.

1.1 Background

In response to the dynamic e-commerce landscape, Radiant Retail emerges as a technologically advanced solution. By leveraging the capabilities of Flutter and Dart, this application aims to redefine the standards of mobile shopping. The background section delves into the intersection of technology and consumer demands, laying the foundation for the innovative features Radiant Retail brings to the forefront.

1.2 Objective

Radiant Retail is driven by the primary objective of delivering an unparalleled shopping experience. Utilizing the robust features of Flutter and Dart, the application aims to excel in user interface design, personalization, and transaction security. The objective section details how Radiant Retail aspires to streamline and elevate the user's shopping journey through a commitment to excellence.

1.3 Scope

Radiant Retail's scope encompasses a diverse range of product categories, meeting the varied preferences of modern shoppers. From fashion and electronics to home goods, the application caters to a broad audience. The documentation explores the cross-platform functionality, ensuring a seamless experience for users on both Android and iOS devices. Additionally, it outlines the ongoing commitment to updates that keep Radiant Retail aligned with industry trends.

1.4 Significance

The significance of Radiant Retail lies in its role as a transformative force in the e-commerce landscape. The application provides a secure and personalized environment, reshaping the conventional shopping experience. The significance section highlights Radiant Retail's commitment to being a user-centric platform, symbolizing a new era of efficient and sophisticated online shopping.

2 Target Audience

Radiant Retail is thoughtfully crafted to be an inclusive and user-friendly application, catering to a broad spectrum of end users. Whether you're a student, a teacher, a professional, or someone simply exploring the world of online shopping, this app is designed for you. The target audience of Radiant Retail is diverse and inclusive, embracing individuals from various backgrounds and interests. With its intuitive interface and versatile features, Radiant Retail ensures a delightful shopping experience for all users, making it the go-to platform for anyone seeking convenience and style in their online retail journey.

3 Features

3.1 Admin Portal

Login and Authentication

By writing the correct credentials of Admin, Admin can log in to the application and will redirect to dashboard

Dashboard

On the dashboard, the Admin will be able to view the status of how many users registered, the total product, and the total orders placed.

Home Page

Admin should be able to open a home page where all products show with opening products of the specific category.

Manage Products

Admin should also be able to manage the products doing all CRUD operations.

Manage Account

Admin Should be able to manage the account and can view its account information.

Orders History

Admin can view all order history.

3.2 User Portal

Login and Authentication

Users can log in to the application and will be redirected to the dashboard.

Sign Up

Users can sign up for the application by providing their personal information.

Products

Users can view all products and products of specific categories.

Buy Product

Users can buy the product by opening the product.

Add to Cart

Users can add products to the cart.

Buy Cart Product

Users can buy all cart products which saves in shared preference.

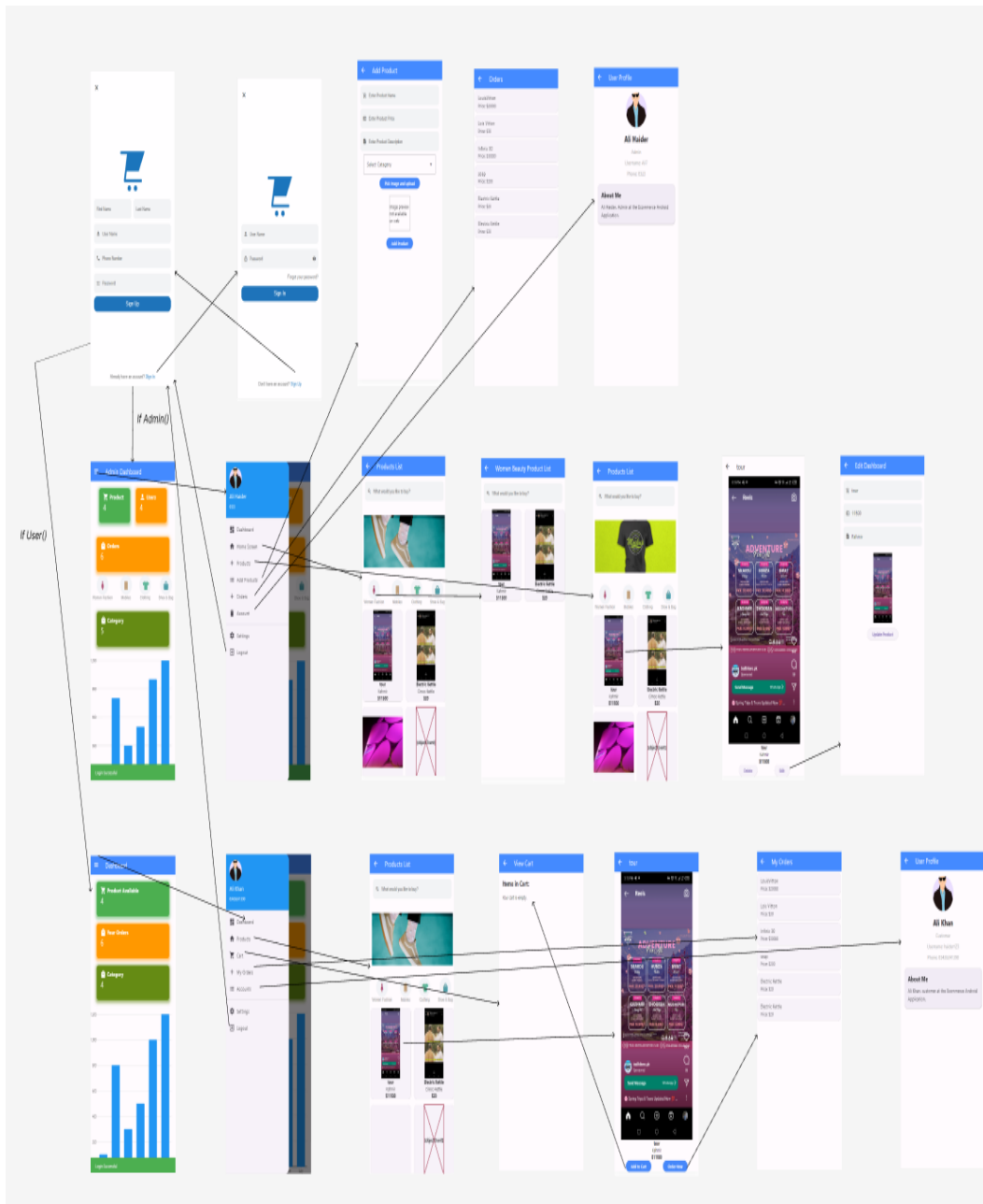
Orders History

Users can view its order history.

Manage Account

Users can manage and view their account information.

4 Screen Layout with Navigation Flow



5 Directory Structure

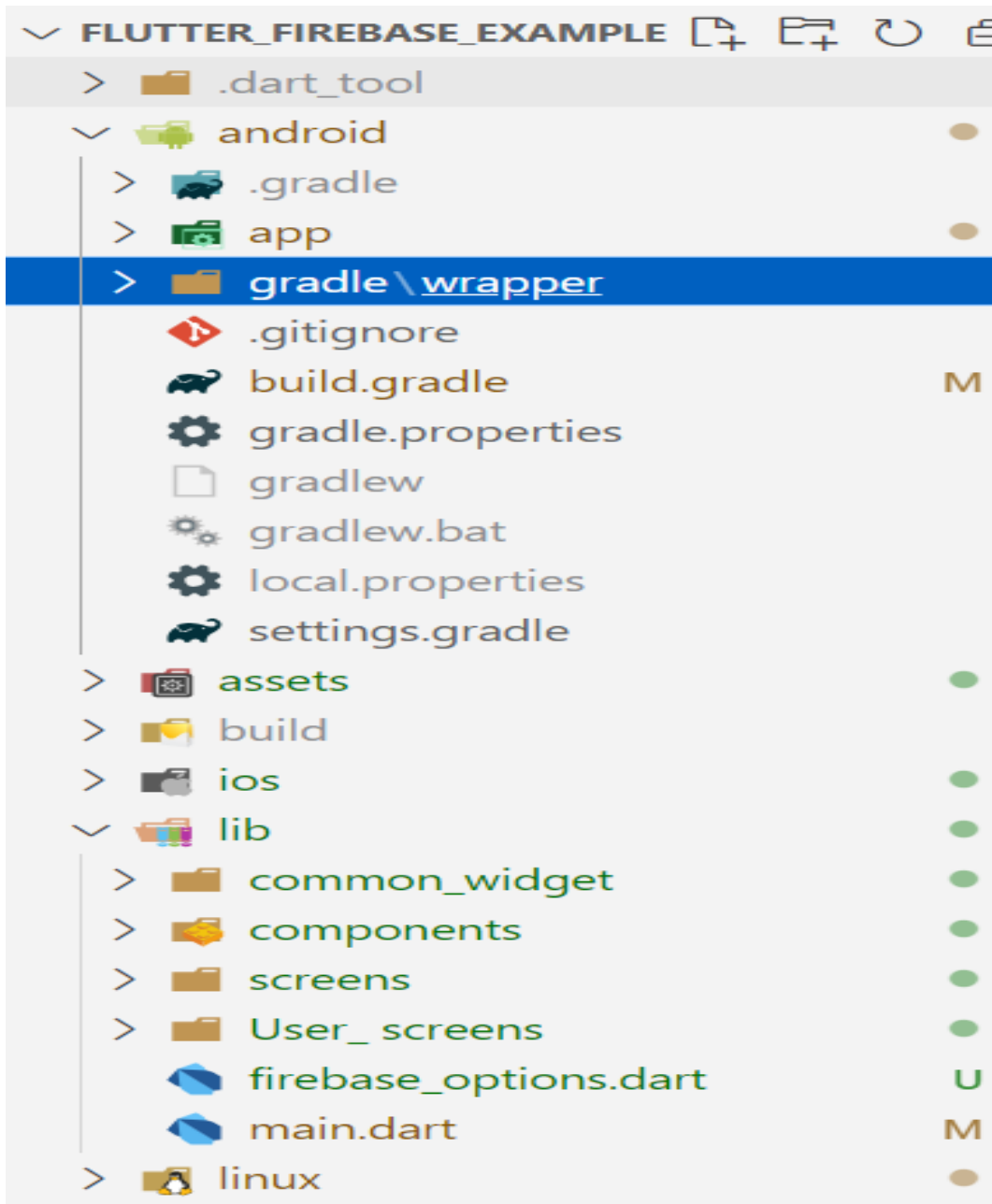


Figure 1: Directory Structure of Application

6 Packages and Widgets Used

Packages

```
name: firebase_setup
description: A new Flutter project.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev
version: 1.0.0+1
environment:
  sdk: '>=3.1.0 <4.0.0'
```

dependencies:

```
carousel_slider: ^4.0.0
flutter:
  sdk: flutter
cupertino_icons: ^1.0.2
firebase_core: ^2.22.0
firebase_auth: ^4.13.0
cloud_firestore: ^4.13.0
font_awesome_flutter: ^9.0.0
http: ^0.13.3
image_picker: ^0.8.0
firebase_storage: ^11.5.1
charts_flutter: ^0.12.0
shared_preferences: ^2.0.7
sqflite: ^2.0.0+3
```

Figure 2: Packages of App

Widgets

The widgets that are used in my app are listed below:

- Scaffold Widget
- SingleChildScrollView Widget
- Row Widget
- Column Widget
- Center Widget

- Drawer Widget
- InkWell Widget
- Positioned Widget
- Stack Widget
- Container Widget
- Padding Widget
- ListTile Widget
- SafeArea Widget
- Expanded Widget
- Stream Builder Widget
- Card Widget

7 Play Store URL

8 Brief Description of Database Integration via API's

AddUser() Method

- Purpose: Post the user data by using API. - HTTP Request: GET request to the specified API endpoint. - Headers: Includes the 'Content-Type' header with the value 'application/json'. - Response Handling: Converts the data to JSON format. - Error Handling: Throws an error if an exception occurs during the process.

Login() Method

- Purpose: Post to check user credentials. - HTTP Request: GET request to the specified API. - Headers: Includes the 'Content-Type' header with the value 'application/json'. - Error Handling: Throws an error if an exception occurs during the process.

9 Description of FireBase Database and Connectivity

Install the plugin by running the following command from the project root:

flutter pub add firebase – core

To initialize FlutterFire, call the initializeApp method on the Firebase class.

dart pub global activate flutterfire – cli

Generated options need to be provided to the initializeApp method. Since this is an asynchronous operation, the main function can be modified to ensure initialization is complete before running the application. First import the firebase – core plugin and generated firebase – options.dart file:

import 'package:firebase – core firebase – core.dart

import 'firebase – options.dart

Next, within the main function, ensure WidgetsFlutterBinding is initialized and then initialize Firebase:

void main() async{


```
WidgetsFlutterBinding.ensureInitialized();
await firebase.initializeApp(
options: DefaultFirebaseOptions.currentPlatform;
)
runApp(myApp());
}
```

10 Use Of Shared Preference

Shared Preference is used to store the data on the local device. Shared Preference is lightweight and very useful for simple data.

CartProduct() Method

When a product is added to the cart it adds it to the shared preference ID "cartProducts" with all the necessary information of the product.

11 Conclusion

11.1 Summary of Achievements

As we conclude this documentation, it's essential to reflect on the milestones and accomplishments of Radiant Retail. From its inception, Radiant Retail has successfully delivered an exceptional shopping experience to users, marked by innovative features, robust functionality, and a commitment to user satisfaction. The summary of achievements underscores the strides made in redefining the e-commerce landscape and meeting the diverse needs of our users.

11.2 Challenges Faced

In any journey, challenges are inevitable, and Radiant Retail is no exception. This section outlines the challenges faced during the development and implementation phases. These challenges, whether technical, logistical, or user-related, have served as opportunities for growth and improvement. Understanding and overcoming these hurdles have contributed to the resilience and adaptability of Radiant Retail.

11.3 Lessons Learned

The development and deployment of Radiant Retail have been rich with insights and lessons. This section encapsulates the valuable learnings garnered throughout the process. From user interactions to technical considerations, each lesson learned has been instrumental in refining the application and shaping the future direction of Radiant Retail. These lessons serve as a foundation for continuous improvement and innovation.

11.4 Future Enhancements

Looking ahead, Radiant Retail is poised for ongoing evolution and enhancement. This section outlines the roadmap for future developments, including planned features, updates, and optimizations. The commitment to staying at the forefront of industry trends and addressing user feedback is paramount. Radiant Retail's journey doesn't conclude here; it extends into an

exciting future marked by continuous innovation and a dedication to providing an increasingly refined shopping experience.