BSc (Hons) in Information Technology

Year 4

CTSE Individual Report

SE4010– Current Trends in Software

Software Engineering **Semester 1,2020**

LEARN A FRUIT

Flutter Report

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

May 2020

BSc (Hons) in Information Technology
Year 4

CTSE Individual Report

SE4010– Current Trends in Software
Software Engineering **Semester 1,2020**

LEARN A FRUIT

CTSE Final Project Report

W.G.M.V.S. Wijesundara

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

May 2020

BSc (Hons) in Information Technology

Year 4

CTSE Individual Report

SE4010– Current Trends in Software

Software Engineering Semester 1,2020

Declaration

“I declare that this is my own work and this dissertation does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print electronic or other medium. I retain the right to use this content in whole or part in future works.

BSc (Hons) in Information Technology
Year 4

CTSE Individual Report

SE4010– Current Trends in Software
Software Engineering **Semester 1,2020**

Abstract

Flutter is a new application building cross platform SDK to create attractive high-performance detailed application for android, web, desktop environment and IOS with reusable codes. Using lot of widgets. Goal to provide same appearance in all app environment with high fidelity user experience. Fruits are more healthy foods to improve our natural beauty, to get more important vitamins, and many more etc. this application is developed to provide a knowledge of lot of fruits peoples can get those with available countries and its fruit family. This app is mainly developed to work with both iOS and android app environment to scale up the usage of the application to give among more users to give a valuable knowledge about users.

In here inside the application has developed to add more fruits without getting stuck and browse their available countries and family with high fidelity user interfaces user can browse each fruit and add to the favorite list as their wishes. Users can change the application UI mode with dark and light mode with their choices.

TABLE OF CONTENTS

List of Figures

SE4010– Current Trends in Software
Software Engineering **Semester 1,2020**

1. Introduction

Learn A Fruit is a cross platform application to provide 1000+ fruits to know about its family, available country and detailed description of each fruits. Flutter enables it's with fast crud operations to provide better user experience and provide high fidelity user interfaces to surf through the application environment to use its full benefit. First before the usage user need to create or the login using their email and password. Then automatically application will be showing the fruit list and other services inside user interfaces.

In the application level users can add update delete fruits efficiently through the fruit management forms can add multiple countries to an array to save a collection of available countries in a single fruit detail. Here provides fruit name, fruit image, fruit family and available country for each fruit as the attributes of the fruits can add fruit images from both camera accessibility and galley uploading method. In the main screen user can see favorite list, profile, about us and fruit home page as the sub links. User can change the dark mode or to the light mode using the profile page or the about us side navigation as users prefer.

2. Application User Interfaces

2.1 Main Screen of the App

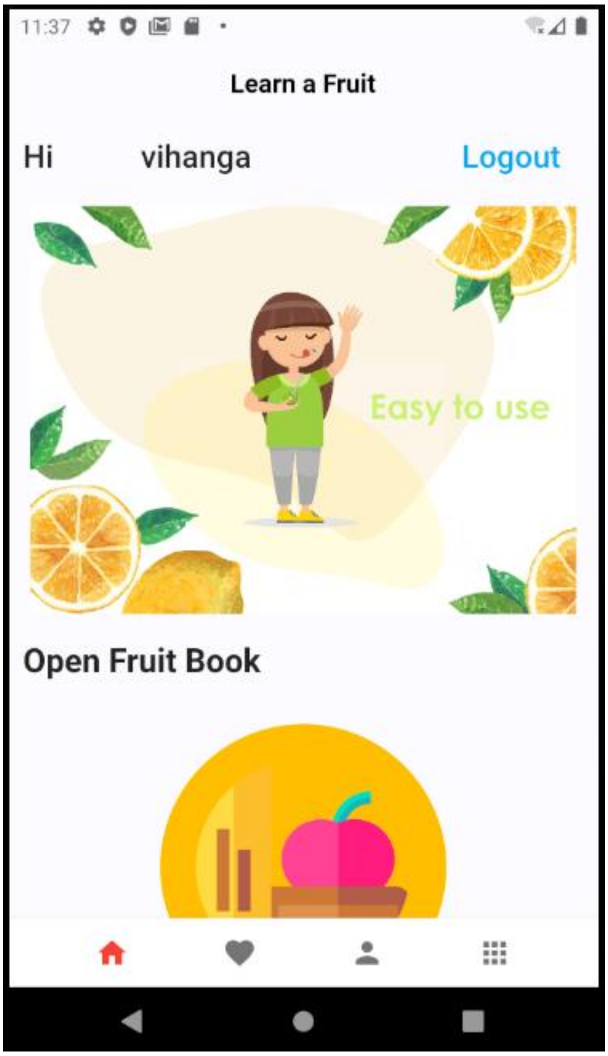
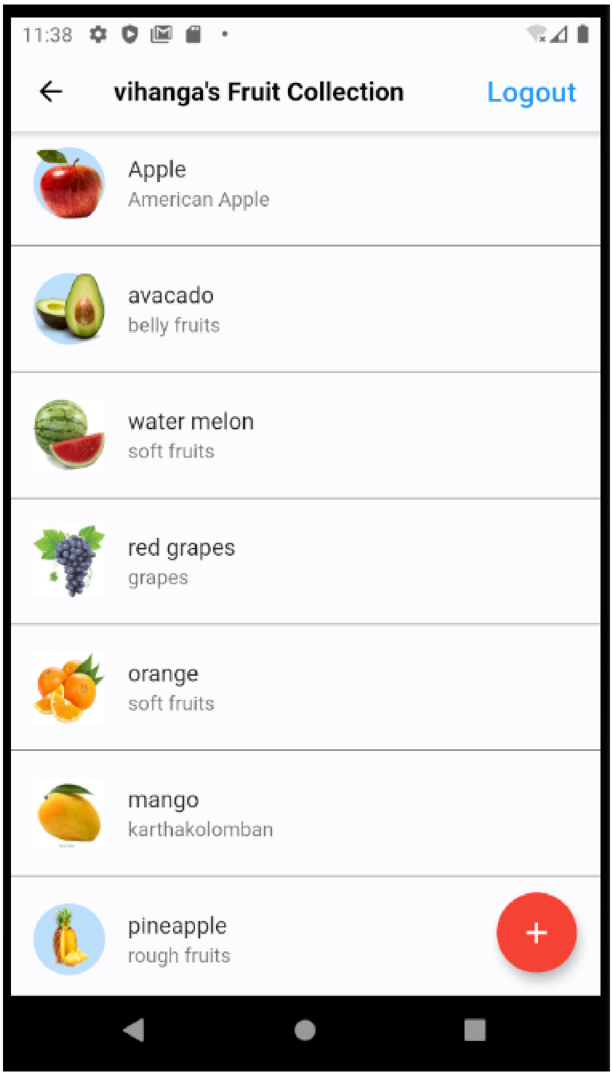


Figure 2.1.1: Main Screen UI

This is the home screen for the fruit learner this page allows to access favorite screen, profile, about us and fruit list using the below footer.

2.2 Fruit List



11:38

vihanga's Fruit Collection Logout

Apple
American Apple

avacado
belly fruits

water melon
soft fruits

red grapes
grapes

orange
soft fruits

mango
karthakolomban

pineapple
rough fruits

Figure 2.2.1: Fruit List View

This is list view page for showing the list of the fruits available in the firebase. Can add new fruits using red button below right side and select any fruit and can view the details of the relevant fruit.

2.3 Adding Fruits Form

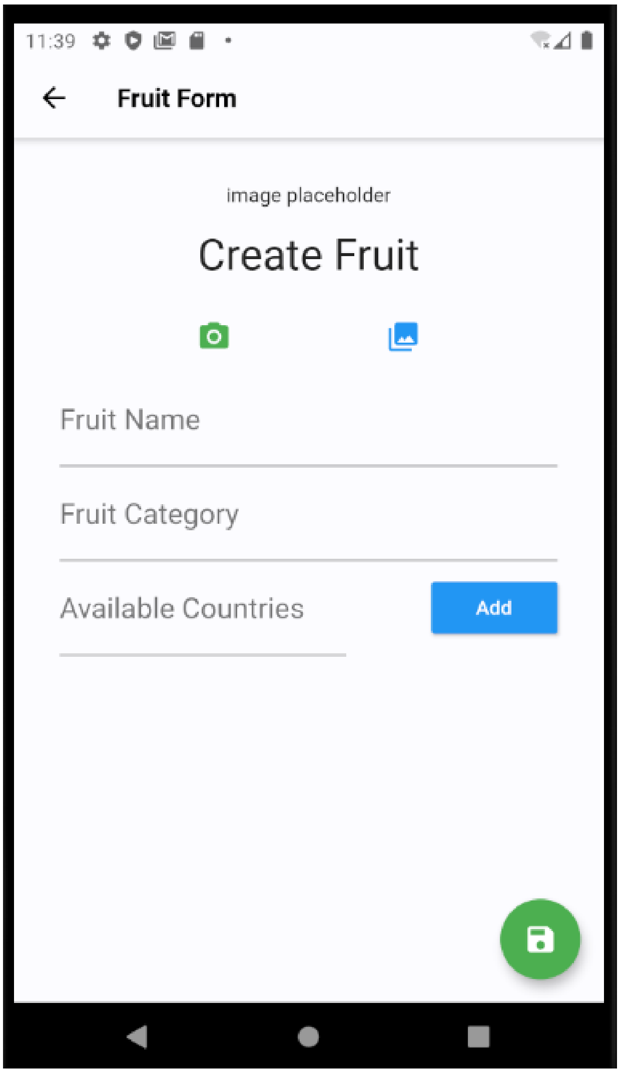
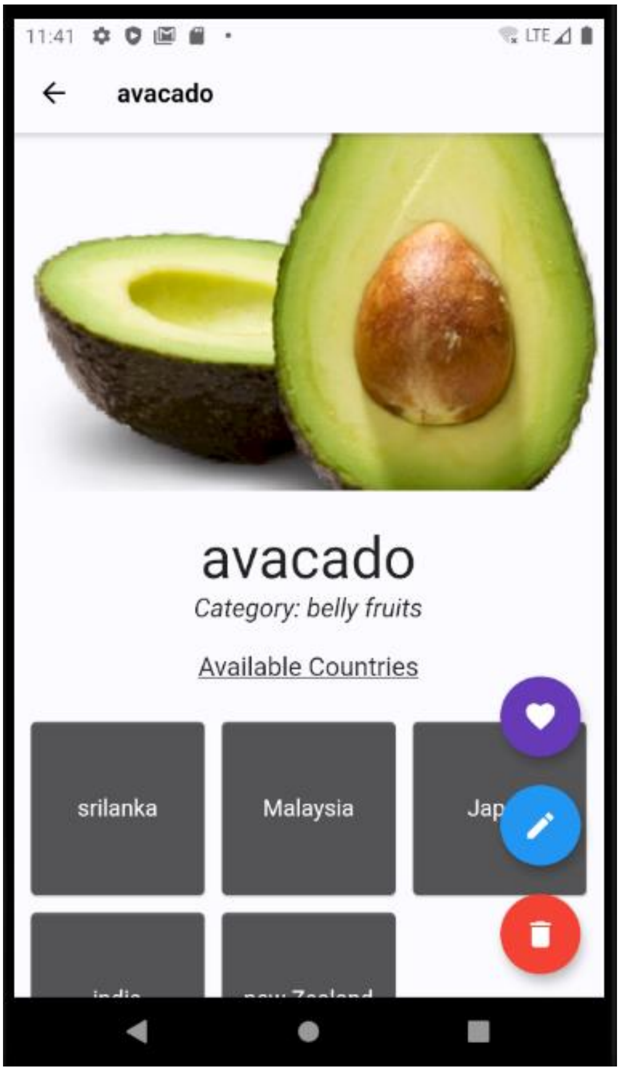


Figure 2.3.1: Fruit Add Form

In the above interface used to add fruit image, fruit name, fruit category and multiple available countries for the fruit crud. Here can add from the blue color button new image for the relevant fruit. Then after adding all green color button performs the saving details.

2.4 View Fruit details



avacado

avacado

Category: belly fruits

Available Countries

srilanka

Malaysia

Jap

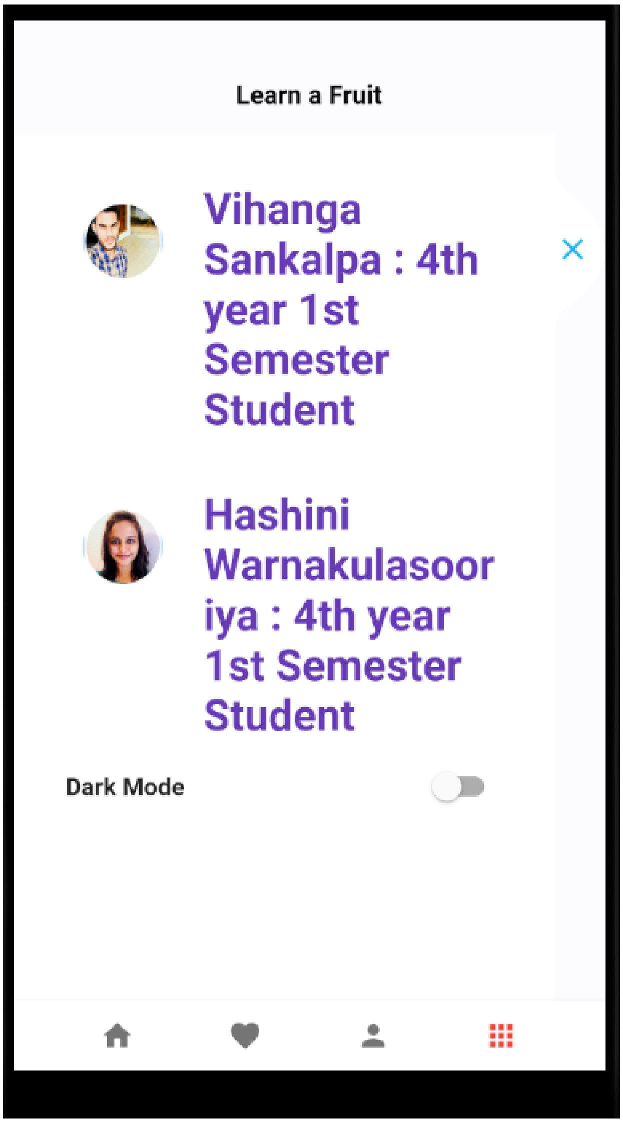
india

new Zealand

Figure 2.4.1: View Fruit Details

In this interface can view the fruit details and update the details of the fruits using the blue color button. And using the purple color heart button can add to favorite list and using red color button can delete the relevant fruit.

2.5 About Us



Learn a Fruit

Vihanga Sankalpa : 4th year 1st Semester Student

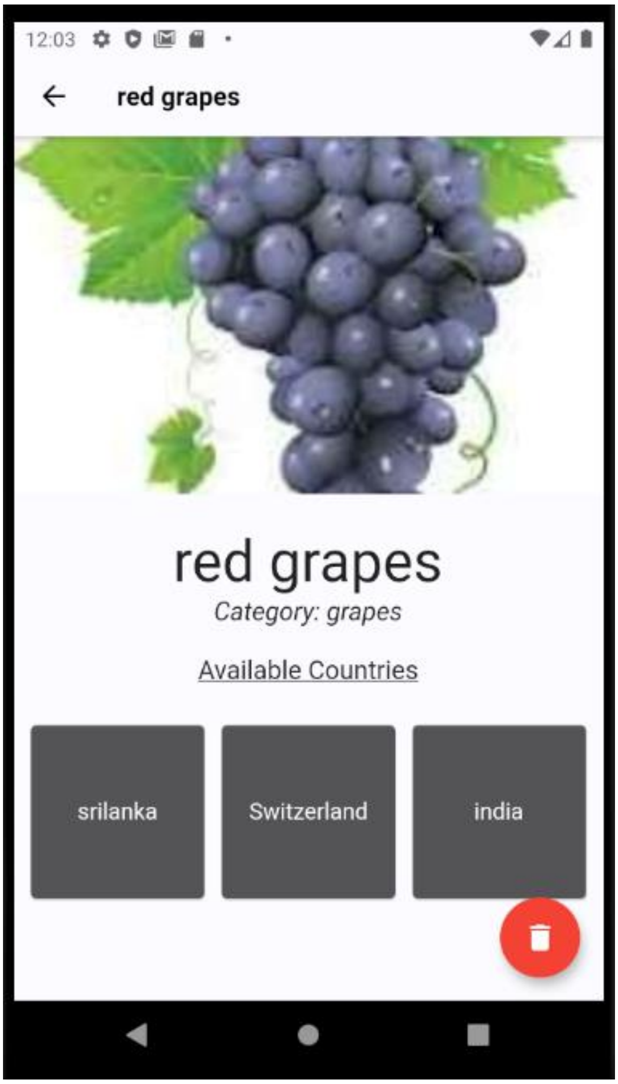
Hashini Warnakulasooriya : 4th year 1st Semester Student

Dark Mode

Figure 2.5.1: About Us View

In here can see the members details with image and switch ability to dark mode inside navigational bar

2.6 Favorite Fruit View

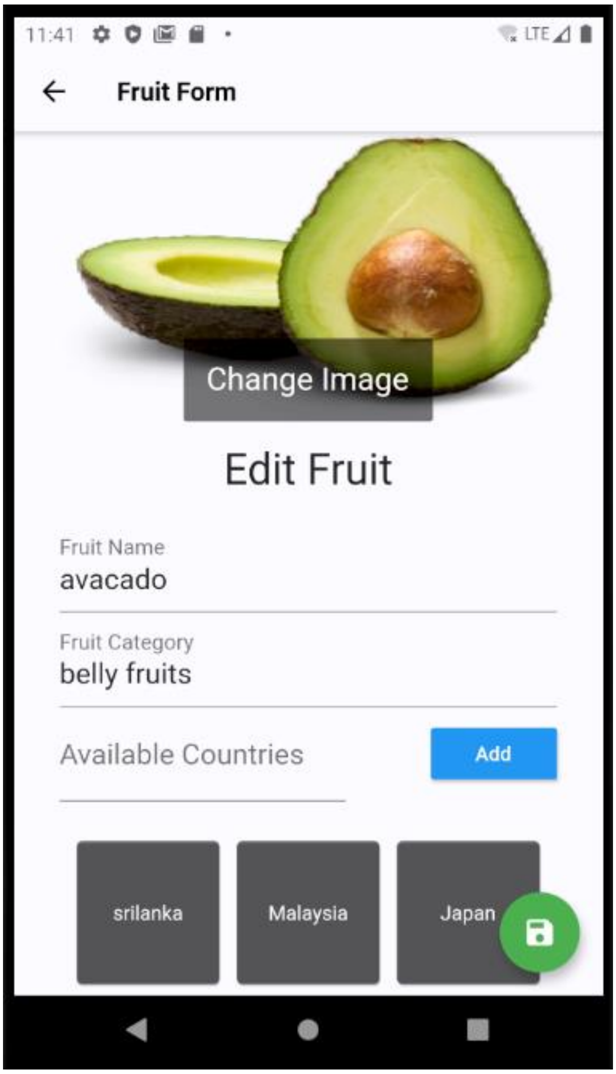

red grapes

red grapes
Category: grapes

Figure 2.6.1: Favorite Fruit View

In here is showing the favorite fruit details and can delete using the red color button.

2.7 Update Fruit Details



Edit Fruit

Fruit Name
avacado

Fruit Category
belly fruits

Available Countries

Figure 2.7.1: Update Fruit View

This interface designed to edit the fruit detail and save in the existing fruit document to perform update relevant fruit details.

2.8 Gallery Access

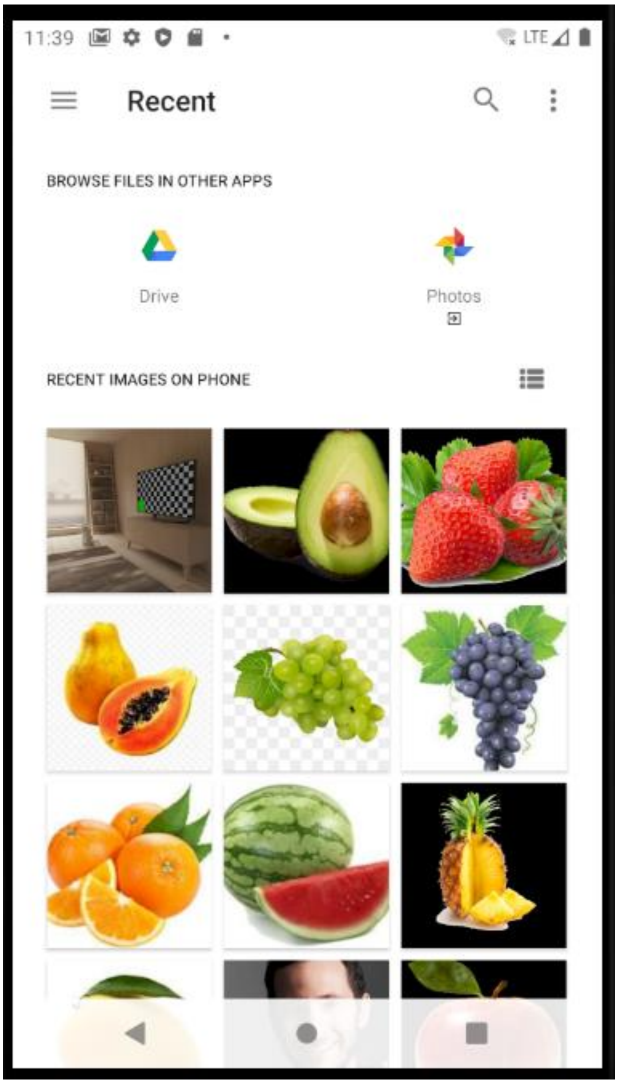
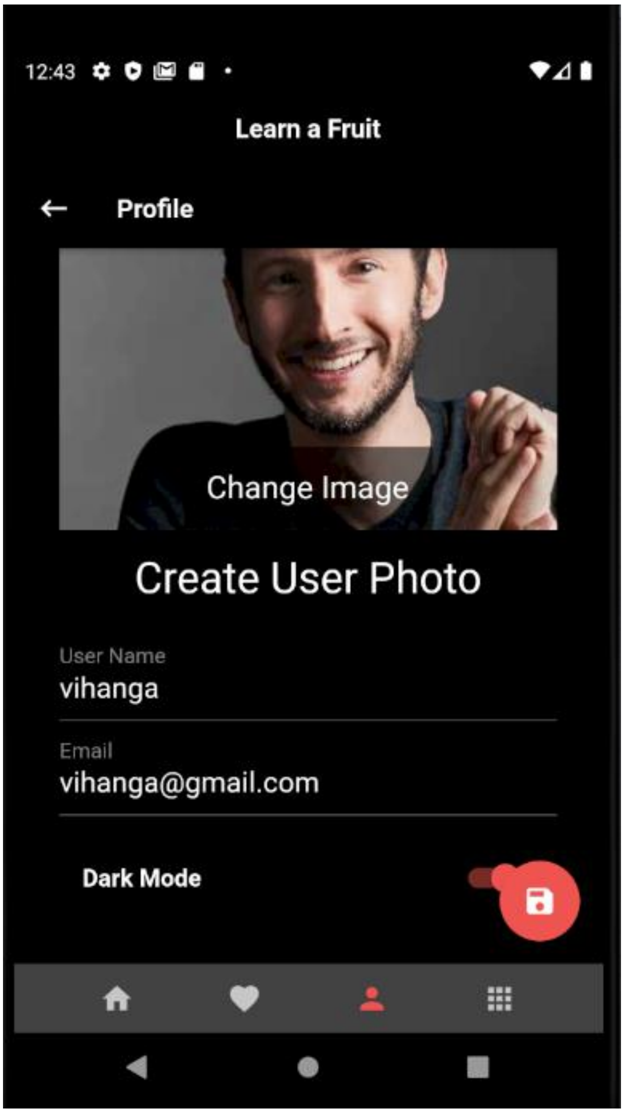


Figure 2.8.1: Gallery Access View

When Press the blue color galley button in fruit add form galley will be open and look like this in the interface.

2.9 Dark Mode



User Name
vihanga

Email
vihanga@gmail.com

Figure 2.9.1: Dark Mode View

Using Dark Mode Button can change the appearance of the color to dark mode or to white mode.

3. Source Codes

UICollectionHandler.dart

```
import 'package:finalproject/DisplayUI/ProfilePageDisplay.dart';
import 'package:finalproject/LearnAFruitUtilities/constColourAttributer.dart';
import 'package:finalproject/LearnAFruit_widgets/SideBarMain.dart';
import 'package:finalproject/LearnAFruit_widgets/sidecomponents.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'favorite_screen.dart';
import 'MainPageDisplay.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the page navigation handler
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */


//creating the ui screen to load all footer bar and page navigation handler in
main screen view and handle the state
class UICollectionHandler extends StatefulWidget {
  @override
  _UICollectionHandlerState createState() => _UICollectionHandlerState();
}
//creating the ui screen to load all footer bar and page navigation handler in
main screen view
class _UICollectionHandlerState extends State<UICollectionHandler> with PageState
{
  //creating private page controller to route for relevant page
  PageController _pageController;
  //creating private pages count
  int _page = 0;

  @override
  Widget build(BuildContext context) {
    return WillPopScope(
      onWillPop: ()=>Future.value(false),
      child: Scaffold(
        appBar: AppBar(
          automaticallyImplyLeading: false,
          centerTitle: true,
          title: Text(
            Constants.appName,
          ),
          elevation: 0.0,
          actions: <Widget>[
          ],
        ),

        body: PageView(
          physics: NeverScrollableScrollPhysics(),
          controller: _pageController,
          onPageChanged: onPageChanged,
          children: <Widget>[
            HomeScreenUI(),
            FavoriteScreen(),
            ProfileUI(isUpdating: false),
            SideBarLayout(),
          ],
        ),

        bottomNavigationBar: BottomAppBar(
          child: new Row(
            mainAxisSize: MainAxisSize.max,
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: <Widget>[
              SizedBox(width:7),
              IconButton(
                icon: Icon(
                  Icons.home,
                  size: 24.0,
                ),
                color: _page == 0
                    ? Theme.of(context).accentColor
                    : Theme
                    .of(context)
                    .textTheme.caption.color,
                onPressed: ()=>_pageController.jumpToPage(0),
              ),

              IconButton(
                icon:Icon(
                  Icons.favorite,
                  size: 24.0,
                ),
                color: _page == 1
                    ? Theme.of(context).accentColor
                    : Theme
                    .of(context)
                    .textTheme.caption.color,
                onPressed: ()=>_pageController.jumpToPage(1),
              ),



              IconButton(
                icon: Icon(
                  Icons.person,
                  size: 24.0,
                ),
                color: _page == 2
                    ? Theme.of(context).accentColor
                    : Theme
                    .of(context)
                    .textTheme.caption.color,
                onPressed: ()=>_pageController.jumpToPage(2),
              ),


              IconButton(
                icon:Icon(
                  Icons.apps,
                  size: 24.0,
                ),
                color: _page == 3
                    ? Theme.of(context).accentColor
                    : Theme
                    .of(context)
                    .textTheme.caption.color,
                onPressed: ()=>_pageController.jumpToPage(3),
              ),

              SizedBox(width:7),
            ],
          ),
        ),
       ),
      );
     }
     //handle the relevant page routing when tap on the relevant icon with int page
    number
     void navigationTapped(int page) {
      _pageController.jumpToPage(page);
     }
     //set the initial state as the page controller state to show relevant page at
    once
     @override
     void initState() {
      super.initState();
      _pageController = PageController();
     }
     //close the page after disposing
     @override
     void dispose() {
      super.dispose();
      _pageController.dispose();
     }
```

```
  //set the page value to the state when tap on the icon in below footer
  void onPageChanged(int page) {
    setState(() {
      this._page = page;
    });
  }
}
```

LearnAFruit_provider.dart

```
import 'package:finalproject/LearnAFruitUtilities/constColourAttributer.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:shared_preferences/shared_preferences.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the About Us Page
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */

//creating the class to handle the dark mode and white mode provider mode
class LearnAFruitProvider extends ChangeNotifier{
  //loading the theme in constructor
  LearnAFruitProvider(){
    checkTheme();
  }
  // reference: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
  //declare theme variable assign its default to light theme
  ThemeData theme = Constants.lightTheme;
  //declare unique key
  Key key = UniqueKey();
  //declaring the global navigator key to maintain navigation state
  GlobalKey<NavigatorState> navigatorKey = GlobalKey<NavigatorState>();

  //setting the key value
  void setKey(value) {
    key = value;
    notifyListeners();
  }
 //setting the navigation value
  void setNavigatorKey(value) {
    navigatorKey = value;
    notifyListeners();
  }
  //setting the theme to dark mode to white mode vise versa
  // reference: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
  void setTheme(value, c) {
    theme = value;
    SharedPreferences.getInstance().then((prefs){
      prefs.setString("theme", c).then((val){
        SystemChrome.setEnabledSystemUIOverlays(SystemUiOverlay.values);
        SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle(
          statusBarColor: c == "dark" ? Constants.darkPrimary :
Constants.lightPrimary,
          statusBarIconBrightness: c == "dark" ? Brightness.light:Brightness.dark,
        ));
      });
    });
    notifyListeners();
  }
  //return theme state
  ThemeData getTheme(value) {
    return theme;
  }
  //checking previous state of the theme and return
 // reference: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
  Future<ThemeData> checkTheme() async{
    SharedPreferences prefs = await SharedPreferences.getInstance();
    ThemeData t;
    String r = prefs.getString("theme") == null ? "light" : prefs.getString(
        "theme");

    if(r == "light"){
      t = Constants.lightTheme;
      setTheme(Constants.lightTheme, "light");
    }else{
      t = Constants.darkTheme;
      setTheme(Constants.darkTheme, "dark");
    }

    return t;
  }
}
```

sidewidgets.dart

```
import 'dart:async';
import 'package:finalproject/CrudControllers/authentication_Controller.dart';
import 'package:finalproject/LearnAFruitproviders/LearnAFruit_provider.dart';
import 'package:finalproject/LearnAFruitUtilities/constColourAttributer.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:rxdart/rxdart.dart';
/*
Author       : W.G.M.V.S Wijesundara  IT17035118
description : Creating the About Us Page
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */

//Creating the UI Level to Display In About Us
//reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
class SideBar extends StatefulWidget {
  @override
  _SideBarState createState() => _SideBarState();
}

class _SideBarState extends State<SideBar> with
SingleTickerProviderStateMixin<SideBar> {
  AnimationController _animationController;
  StreamController<bool> isSidebarOpenedStreamController;
  Stream<bool> isSidebarOpenedStream;
  StreamSink<bool> isSidebarOpenedSink;
  final _animationDuration = const Duration(milliseconds: 500);

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(vsync: this, duration:
_animationDuration);
    isSidebarOpenedStreamController = PublishSubject<bool>();
    isSidebarOpenedStream = isSidebarOpenedStreamController.stream;
    isSidebarOpenedSink = isSidebarOpenedStreamController.sink;
  }

  @override
  void dispose() {
    _animationController.dispose();
    isSidebarOpenedStreamController.close();
    isSidebarOpenedSink.close();
    super.dispose();
  }
  void onIconPressed() {
    final animationStatus = _animationController.status;
    final isAnimationCompleted = animationStatus == AnimationStatus.completed;

    if (isAnimationCompleted) {
      isSidebarOpenedSink.add(false);
      _animationController.reverse();
    } else {
      isSidebarOpenedSink.add(true);
      _animationController.forward();
    }
  }

  @override
  Widget build(BuildContext context) {
    final screenWidth = MediaQuery.of(context).size.width;
    AuthenticationController authNotifier =
Provider.of<AuthenticationController>(context);

    return StreamBuilder<bool>(
      initialData: false,
      stream: isSidebarOpenedStream,
      builder: (context, isSideBarOpenedAsync) {
        return AnimatedPositioned(
          duration: _animationDuration,
          top: 0,
          bottom: 0,
          left: isSideBarOpenedAsync.data ? 0 : -screenWidth,
          right: isSideBarOpenedAsync.data ? 0 : screenWidth - 45,
          child: Row(
            children: <Widget>[
              Expanded(
                child: Container(
                  padding: const EdgeInsets.symmetric(horizontal: 20),
                  color: Colors.white,
                  child: Column(
                    children: <Widget>[
                      SizedBox(
                        height: 30,
                      ),
                  ListTile(
                    title: Text(
                      "Vihanga Sankalpa : 4th year 1st Semester Student",
                      style: TextStyle(color: Colors.deepPurple, fontSize: 30,
fontWeight: FontWeight.w800),
                    ),
                      leading: CircleAvatar(
                       child   :Image.asset(
                      'assets/v.bmp',
                      width: 670.0,
                      height: 300.0,
                      ),
                        radius: 40,
                      ),
                  ),
                      Divider(
                        height: 30,
                        thickness: 0.5,
                        color: Colors.white.withOpacity(0.3),
                        indent: 32,
                        endIndent: 32,
                      ),

                      ListTile(
                        title: Text(
                          "Hashini Warnakulasooriya : 4th year 1st Semester 
Student",
                          style: TextStyle(color: Colors.deepPurple, fontSize: 30, 
fontWeight: FontWeight.w800),
                        ),
                        leading: CircleAvatar(
                          child   :Image.asset(
                            'assets/hashini.bmp',
                            width: 670.0,
                            height: 300.0,
                          ),
                          radius: 40,
                        ),
                      ),
                    //showing the dark mode white mode changer
                      ListTile(
                        title: Text(
                          "Dark Mode",
                          style: TextStyle(
                            fontSize: 17,
                            fontWeight: FontWeight.w700,
                          ),
                        ),

                        trailing: Switch(
                          value: Provider.of<LearnAFruitProvider>(context).theme 
== Constants.lightTheme
                              ? false
                              : true,
                          onChanged: (v) async{
                            if (v) {
                              Provider.of<LearnAFruitProvider>(context, listen: 
false)
                                  .setTheme(Constants.darkTheme, "dark");
                            } else {
                              Provider.of<LearnAFruitProvider>(context, listen: 
```

```
false)
                                  .setTheme(Constants.lightTheme, "light");
                            }
                          },
                          activeColor: Theme.of(context).accentColor,
                        ),
                      ),


                    ],
                  ),
                ),
              ),
              Align(
                alignment: Alignment(0, -0.9),
                child: GestureDetector(
                  onTap: () {
                    onIconPressed();
                  },
                  child: ClipPath(
                    clipper: MenuWidget(),
                    child: Container(
                      width: 35,
                      height: 110,
                      color: Colors.white,
                      alignment: Alignment.centerLeft,
                      child: AnimatedIcon(
                        progress: _animationController.view,
                        icon: AnimatedIcons.menu_close,
                        color: Color(0xFF1BB5FD),
                        size: 25,
                      ),
                    ),
                  ),
                ),
              ),
            ],
          ),
        );
      },
    );
  }
}

class MenuWidget extends CustomClipper<Path> {
  @override
  Path getClip(Size size) {
    Paint paint = Paint();
    paint.color = Colors.white;

    final width = size.width;
    final height = size.height;

    Path path = Path();
    path.moveTo(0, 0);
    path.quadraticBezierTo(0, 8, 10, 16);
    path.quadraticBezierTo(width - 1, height / 2 - 20, width, height / 2);
    path.quadraticBezierTo(width + 1, height / 2 + 20, 10, height - 16);
    path.quadraticBezierTo(0, height - 8, 0, height);
    path.close();
    return path;
  }

  @override
  bool shouldReclip(CustomClipper<Path> oldClipper) {
    return true;
  }
}
```

sidecomponents.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'SideBarMain.dart';
import 'sidewidgets.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the About Us Page
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */
//reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
//Creating the building block of About Us Side Navigation
class SideBarLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: BlocProvider<Paginization>(
        create: (context) => Paginization(),
        child: Stack(
          children: <Widget>[
            BlocBuilder<Paginization, PageState>(
              builder: (context, navigationState) {
                return navigationState as Widget;
              },
            ),
            SideBar(),
          ],
        ),
      ),
    );
  }
}
```

SideBarMain.dart

```
import 'package:bloc/bloc.dart';
import 'sidebarinvoker.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the About Us Page
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */

//Creating the Abstract class to load About Us


//reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
abstract class PageState {}

class Paginization extends Bloc<NavigationEvents, PageState> {

  @override
  PageState get initialState => AboutUs();

  @override
  Stream<PageState> mapEventToState(event) {
    // TODO: implement mapEventToState
  }
}

class NavigationEvents {
}
```

sidebarinvoker.dart

```
import 'package:flutter/material.dart';
import 'SideBarMain.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the About Us Page
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */
//reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
//Creating the About Us Page
class AboutUs extends StatelessWidget with PageState {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Text(
        "About Us",
        style: TextStyle(fontWeight: FontWeight.w900, fontSize: 28),
      ),
    );
  }
}
```

FruitBook.dart

```
import 'package:finalproject/LearnAFruit_Api/Fruit_Api_Handler.dart';
import 'package:finalproject/CrudControllers/authentication_Controller.dart';
import 'package:finalproject/CrudControllers/Fruit_Controller.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'detail.dart';
import 'fruit_form.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the main screen to direct to the fruit list
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */
//creating the class to surf as the main screen for to go the fruit list  and
```

```
manage the state of the page
class FruitBook extends StatefulWidget {
  @override
  _FruitBookState createState() => _FruitBookState();
}

//creating the class to surf as the main screen for to go the fruit list
class _FruitBookState extends State<FruitBook> {
  //maintaining the state of the fruit controller to get fruits list
  @override
  void initState() {
    FruitController fruitNotifier = Provider.of<FruitController>(context, listen:
false);
    getFruits(fruitNotifier);
    super.initState();
  }
//calling the authentication class to get the display name
//calling the fruit controller class to get the list of fruits to display
  @override
  Widget build(BuildContext context) {
    AuthenticationController authNotifier =
Provider.of<AuthenticationController>(context);
    FruitController fruitNotifier = Provider.of<FruitController>(context);
    //showing the list of fruits in list  view builder
    Future<void> _refreshList() async {
      getFruits(fruitNotifier);
    }

    print("Opening Fruit Book");
    return Scaffold(
      appBar: AppBar(
        title: Text(
          //if authentic user state isn't null then show the display name as email
display name in the appbar
          authNotifier.user != null ? authNotifier.user.displayName + "'s Fruit
Collection" : "Favourite Fruit Book",
        ),
        actions: <Widget>[
          // action button
          FlatButton(
            onPressed: () => signout(authNotifier),
            child: Text(
              "Logout",
              style: TextStyle(fontSize: 20, color: Colors.blue),
            ),
          ),
        ],
      ),
      body: new RefreshIndicator(
        child: ListView.separated(
          itemBuilder: (BuildContext context, int index) {
            return ListTile(
              leading: CircleAvatar(
                radius: 25.0,
                child: Image.network(
                fruitNotifier.fruitList[index].image != null
                    ? fruitNotifier.fruitList[index].image
                    : 'https://www.testingxperts.com/wp-
content/uploads/2019/02/placeholder-img.jpg',
                width: 150,height: 1000,
                fit: BoxFit.fill,
              ),
              ),

              title: Text(fruitNotifier.fruitList[index].name),
              subtitle: Text(fruitNotifier.fruitList[index].category),
              onTap: () {
                fruitNotifier.currentFruit = fruitNotifier.fruitList[index];
                Navigator.of(context).push(MaterialPageRoute(builder:
(BuildContext context) {
                  return FruitDetail();
                }));
              },
            );

          },

          itemCount: fruitNotifier.fruitList.length,
          separatorBuilder: (BuildContext context, int index) {


            InkWell(
              child: ListView(
                shrinkWrap: true,
                primary: false,
                children: <Widget>[
                  Stack(
                    children: <Widget>[
                      Positioned(
                        right: -10.0,
                        bottom: 3.0,
                        child: RawMaterialButton(
                          onPressed: (){},
                          fillColor: Colors.blue,
                          shape: CircleBorder(),
                          elevation: 4.0,
                          child: Padding(
                            padding: EdgeInsets.all(5),
                            child: Icon(
                                  Icons.favorite,
                              color: Colors.red,
                              size: 17,
                            ),
                          ),
                        ),
                      ),
                    ],
                  ),
                ],
              ),
            );


            return Divider(
              color: Colors.black,
            );
          },
        ),
        onRefresh: _refreshList,
      ),

      floatingActionButton: FloatingActionButton(
        onPressed: () {
          fruitNotifier.currentFruit = null;
          Navigator.of(context).push(
            MaterialPageRoute(builder: (BuildContext context) {
              return FruitForm(
                isUpdating: false,
              );
            }),
          );
        },
        //invoke to create new fruits
        child: Icon(Icons.add),
        foregroundColor: Colors.white,
      ),
    );
  }
}
```

fruit_form.dart

```
import 'dart:io';
import 'package:finalproject/LearnAFruit_Api/Fruit_Api_Handler.dart';
import 'package:finalproject/Crudmodel/FruitCrudModel.dart';
import 'package:finalproject/CrudControllers/Fruit_Controller.dart';
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:provider/provider.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the input screen for fruit crud
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */
//creating the class to input fruit details for adding the details of the fruit
class FruitForm extends StatefulWidget {
  //checking whether needs to update or not
  final bool isUpdating;
  //loading the details to the constructor
  FruitForm({@required this.isUpdating});
  //handle the fruit form state
  @override
  _FruitFormState createState() => _FruitFormState();
}
//creating the class to input fruit details for adding the details of the fruit
class _FruitFormState extends State<FruitForm> {
  //declaring the global form key to maintain form state
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  //declaring the global scaffold key to maintain scaffold state
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  //creating the list handler to store countries
  List _countries = [];
  //creating the current fruit details object
  FruitCrudModel _currentFruit;
  //creating the image url for the fruit
  String _imageUrl;
  //creating the image file to store in cloud store bucket
  File _imageFile;
  //creating the text field controller to country adder
  TextEditingController countriesController = new TextEditingController();

  //maintaining the state of the fruit controller
  @override
  void initState() {
    super.initState();
    FruitController fruitNotifier = Provider.of<FruitController>(context, listen:
false);
```

```
  //checking if current fruit object is null then load the current details
  if (fruitNotifier.currentFruit != null) {
   _currentFruit = fruitNotifier.currentFruit;
  } else {
   _currentFruit = FruitCrudModel();
  }

  _countries.addAll(_currentFruit.countries);
  _imageUrl = _currentFruit.image;
 }
 //check if the the file and image url is null then show in the placeholder as 
image placeholder otherwise print in console as showing image from local file
 _showImage() {
  if (_imageFile == null && _imageUrl == null) {
   return Text("image placeholder");
  } else if (_imageFile != null) {
   print('showing image from local file');
   //designing the ui level
   return Stack(
    alignment: AlignmentDirectional.bottomCenter,
    children: <Widget>[
     Image.file(
      _imageFile,
      fit: BoxFit.cover,
      height: 250,
     ),
     FlatButton(
      padding: EdgeInsets.all(16),
      color: Colors.black54,
      child: Text(
       'Change Image',
       style: TextStyle(color: Colors.white, fontSize: 22, fontWeight: 
FontWeight.w400),
      ),
      onPressed: () => _getLocalImage(),
     )
    ],
   );
  } else if (_imageUrl != null) {
   print('showing image from url');

   return Stack(
    alignment: AlignmentDirectional.bottomCenter,
    children: <Widget>[
     Image.network(
      _imageUrl,
      width: MediaQuery.of(context).size.width,
      fit: BoxFit.fill,
      height: 200,
     ),
     FlatButton(
      padding: EdgeInsets.all(16),
      color: Colors.black54,
      child: Text(
       'Change Image',
       style: TextStyle(color: Colors.white, fontSize: 22, fontWeight: 
FontWeight.w400),
      ),
      onPressed: () => _getLocalImage(),
     )
    ],
   );
  }
 }
 //getting the image file from image gallery to load image in the placeholder
 _getLocalImage() async {
  File imageFile =
    await ImagePicker.pickImage(source: ImageSource.gallery, imageQuality: 
100, maxWidth: 400);

  if (imageFile != null) {
   setState(() {
    _imageFile = imageFile;
   });
  }
 }
 //getting the image file from mobile camera to load image in the placeholder
 _openCamera() async {
  File imageFile =
  await ImagePicker.pickImage(source: ImageSource.camera, imageQuality: 100, 
maxWidth: 400);

  if (imageFile != null) {
   setState(() {
    _imageFile = imageFile;
   });
  }
 }

 //handle the name field with validation using text form controller
 Widget _buildNameField() {
  return TextFormField(
   decoration: InputDecoration(labelText: 'Fruit Name'),
   initialValue: _currentFruit.name,
   keyboardType: TextInputType.text,
   style: TextStyle(fontSize: 20),
   validator: (String value) {
    if (value.isEmpty) {
     return 'Name is required';
    }

    if (value.length < 3 || value.length > 20) {
          return 'Name must be more than 3 and less than 20';
        }

        return null;
      },
      onSaved: (String value) {
        _currentFruit.name = value;
      },
    );
  }
  //handle the fruit family field with validation using text form controller
  Widget _buildCategoryField() {
    return TextFormField(
      decoration: InputDecoration(labelText: 'Fruit Category'),
      initialValue: _currentFruit.category,
      keyboardType: TextInputType.text,
      style: TextStyle(fontSize: 20),
      validator: (String value) {
        if (value.isEmpty) {
          return 'Category is required';
        }

        if (value.length < 3 || value.length > 20) {
          return 'Category must be more than 3 and less than 20';
        }

        return null;
      },
      onSaved: (String value) {
        _currentFruit.category = value;
      },
    );
  }
  //handle the available country field with validation using text form controller
  _buildCountryField() {
    return SizedBox(
      width: 200,
      child: TextField(
        controller: countriesController,
        keyboardType: TextInputType.text,
        decoration: InputDecoration(labelText: 'Available Countries'),
        style: TextStyle(fontSize: 20),
      ),
    );
  }
  //sending the uploaded fruit details to add to the cloud store
  _onFruitUploaded(FruitCrudModel fruit) {
    FruitController fruitNotifier = Provider.of<FruitController>(context, listen:
false);
    fruitNotifier.addFruit(fruit);
    Navigator.pop(context);
  }
  //controlling the text form state whether empty or add the text if it is not
empty
  _addCountry(String text) {
    if (text.isNotEmpty) {
      setState(() {
        _countries.add(text);
      });
      countriesController.clear();
    }
  }
   //save the current state of the form text forms
  _saveFruit() {
    print('saveFruit Called');
    if (!_formKey.currentState.validate()) {
      return;
    }

    _formKey.currentState.save();

    print('form saved');

    _currentFruit.countries = _countries;

    uploadFruitAndImage(_currentFruit, widget.isUpdating, _imageFile,
_onFruitUploaded);

    print("name: ${_currentFruit.name}");
    print("category: ${_currentFruit.category}");
    print("Countries: ${_currentFruit.countries.toString()}");
    print("_imageFile ${_imageFile.toString()}");
    print("_imageUrl $_imageUrl");
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      key: _scaffoldKey,
      appBar: AppBar(title: Text('Fruit Form')),
      body: SingleChildScrollView(
        padding: EdgeInsets.all(32),
        child: Form(
          key: _formKey,
          autovalidate: true,
          child: Column(children: <Widget>[
            _showImage(),
            SizedBox(height: 16),
            Text(
              widget.isUpdating ? "Edit Fruit" : "Create Fruit",
              textAlign: TextAlign.center,
              style: TextStyle(fontSize: 30),
            ),
            SizedBox(height: 16),
            _imageFile == null && _imageUrl == null
                ? Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: <Widget>[
                Align(
                  alignment: Alignment.centerLeft,
                  child: Container(
                    child: Column(children: <Widget>[
                      Align(alignment: Alignment.centerLeft,
                        child: IconButton(
                          icon: Icon(
                            Icons.photo_camera,
                            color: Colors.green,
                          ),
                          onPressed: () {
                            _openCamera();
                          },),
                      ),
                    ],),
                  ),
                ),
                Align(
                  alignment: Alignment.centerRight,
                  child: Container(
                    child: IconButton(
                      icon: Icon(
                        Icons.photo_library,
                        color: Colors.blue,
                      ),
                      onPressed: () {
                        _getLocalImage();
                      },),
                  ),
                )
              ],
            )
                : SizedBox(height: 0),
            _buildNameField(),
            _buildCategoryField(),
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: <Widget>[
                _buildCountryField(),
                ButtonTheme(
                  child: RaisedButton(
                    child: Text('Add', style: TextStyle(color: Colors.white)),
                    onPressed: () => _addCountry(countriesController.text),
                  ),
                )
```

```
              ],
            ),
            SizedBox(height: 16),
            GridView.count(
              shrinkWrap: true,
              scrollDirection: Axis.vertical,
              padding: EdgeInsets.all(8),
              crossAxisCount: 3,
              crossAxisSpacing: 4,
              mainAxisSpacing: 4,
              children: _countries
                  .map(
                    (ingredient) => Card(
                  color: Colors.black54,
                  child: Center(
                    child: Text(
                      ingredient,
                      style: TextStyle(color: Colors.white, fontSize: 14),
                    ),
                  ),
                ),
              )
                  .toList(),
            )
          ]),
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () {
          FocusScope.of(context).requestFocus(new FocusNode());
          _saveFruit();
        },
        child: Icon(Icons.save),
        backgroundColor: Colors.green,
        foregroundColor: Colors.white,
      ),


    );
  }
}
```

detail.dart

```
import 'package:finalproject/LearnAFruit_Api/Fruit_Api_Handler.dart';
import 'package:finalproject/Crudmodel/FruitCrudModel.dart';
import 'package:finalproject/CrudControllers/Fruit_Controller.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'favourite_form.dart';
import 'fruit_form.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the fruit detail class show details of the each fruits
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */

//creating the class to display the fruit details from each fruits with update
delete adding favorite list actions in fruit main screen
class FruitDetail extends StatelessWidget {
  //checking whether needs to update or not
  final bool isUpdating;
  //loading the updateble details to the constructor
  FruitDetail({@required this.isUpdating});
  //declaring the global scaffold key to maintain scaffold state
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  //creating the text controller in form section to add new countries to the list
of the each fruit
  TextEditingController countriesController = new TextEditingController();

  @override
  Widget build(BuildContext context) {
    //calling the fruit controller class
    FruitController fruitNotifier = Provider.of<FruitController>(context);
    //calling the delete fruit method when invoke the onfruitdeleted method
    _onFruitDeleted(FruitCrudModel fruit) {
      Navigator.pop(context);
      fruitNotifier.deleteFruit(fruit);
    }

    //design the ui level
    return Scaffold(
      key: _scaffoldKey,
      appBar: AppBar(
        title: Text(fruitNotifier.currentFruit.name),
      ),
      body: SingleChildScrollView(
        child: Center(
          child: Container(
            child: Column(
              children: <Widget>[
                //showing the relevant fruit image for each fruit detail page or 
providing to enter new
                Image.network(
                  fruitNotifier.currentFruit.image != null
                      ? fruitNotifier.currentFruit.image
                      : 'https://www.testingxperts.com/wp-
content/uploads/2019/02/placeholder-img.jpg',
                  width: MediaQuery.of(context).size.width,
                  height: 250,
                  fit: BoxFit.fitWidth,
                ),
                SizedBox(height: 24),
                //showing the relevant fruit name for each fruit detail page or 
providing to enter new
                Text(
                  fruitNotifier.currentFruit.name,
                  style: TextStyle(
                    fontSize: 40,
                  ),
                ),
                //showing the relevant fruit family name for each fruit detail 
page or providing to enter new
                Text(
                  'Category: ${fruitNotifier.currentFruit.category}',
                  style: TextStyle(fontSize: 18, fontStyle: FontStyle.italic),
                ),
                SizedBox(height: 20),
                //showing the relevant fruit multiple countries for each fruit 
detail page  to enter new
                Text(
                  "Available Countries",
                  style: TextStyle(fontSize: 18, decoration: 
TextDecoration.underline),
                ),
                SizedBox(height: 16),
                //showing the added countries in the list before update
                GridView.count(
                  shrinkWrap: true,
                  scrollDirection: Axis.vertical,
                  padding: EdgeInsets.all(8),
                  crossAxisCount: 3,
                  crossAxisSpacing: 4,
                  mainAxisSpacing: 4,
                  children: fruitNotifier.currentFruit.countries
                      .map(
                        (ingredient) => Card(
                          color: Colors.black54,
                          child: Center(
                            child: Text(
                              ingredient,
                              style: TextStyle(color: Colors.white, fontSize: 16),
                            ),
                          ),
                        ),
                      )
                      .toList(),
                )
              ],
            ),
          ),
        ),
      ),
      floatingActionButton: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        children: <Widget>[
          //page button action to add to the favorite collection
          FloatingActionButton(
            heroTag: 'button0',
            onPressed: () {
              Navigator.of(context).push(
                MaterialPageRoute(builder: (BuildContext context) {
                  return FavouriteForm(
                    isUpdating: true,
                  );
                }),
              );
            },
            child: Icon(Icons.favorite),
            backgroundColor: Colors.deepPurple,
            foregroundColor: Colors.white,
          ),
          SizedBox(height: 20),
          //page button action  to the edit details
          FloatingActionButton(
            heroTag: 'button1',
            onPressed: () {
              Navigator.of(context).push(
                MaterialPageRoute(builder: (BuildContext context) {
                  return FruitForm(
                    isUpdating: true,
                  );
                }),
              );
            },
            child: Icon(Icons.edit),
            backgroundColor: Colors.blue,
            foregroundColor: Colors.white,
          ),
          SizedBox(height: 20),
          //page button action  to the delete details of the relevant fruit
          FloatingActionButton(
            heroTag: 'button2',
            onPressed: () => deleteFruit(fruitNotifier.currentFruit,
_onFruitDeleted),
            child: Icon(Icons.delete),
            backgroundColor: Colors.red,
            foregroundColor: Colors.white,
          ),
        ],
      ),
    );
  }
}
```

UserCrudModel.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the fruit crud model to perform crud operations
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */
//creating the class to create the attributes for the user crud model to handle
the user details
class UserCrudModel {
  //declare string name for the each user crud  to display after login in main
screen
  String displayName;
  //declare unique string email for the each user crud
  String email;
  //declare string password for the each user crud
  String password;
  //declare string image for the each user crud
  String image;
  //declare creating time to record the created time for the each user crud
  Timestamp createdAt;
  //declare updating time to record the updated time for the each user crud
  Timestamp updatedAt;

  //default constructor implementation
  UserCrudModel();


  //mapping the string based json data to document using the user crud model
  UserCrudModel.fromMap(Map<String, dynamic> data) {
    displayName = data['displayName'];
    email = data['email'];
    image = data['image'];
    createdAt = data['createdAt'];
    updatedAt = data['updatedAt'];
  }
  //returning the created data using user crud model
  Map<String, dynamic> toMap() {
    return {
      'displayName': displayName,
      'email': email,
      'image': image,
      'createdAt': createdAt,
      'updatedAt': updatedAt
    };
  }


}
```

FruitCrudModel.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the fruit crud model to perform crud operations
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */

//creating the class to create the attributes for the fruit crud model to handle
the fruit details
class FruitCrudModel {
  //declare string unique id for the each fruit crud
  String id;
  //declare string name for the each fruit crud
  String name;
  //declare string fruit category for the each fruit family crud
  String category;
  //declare string image url for the each fruit crud
  String image;
  //declare list for available countries to store multiple countries at once for
each fruit
  List countries = [];
  //declare creating time to record the created time for the each fruit crud
  Timestamp createdAt;
  //declare updating time to record the updated time for the each fruit crud
  Timestamp updatedAt;

  //default constructor implementation
  FruitCrudModel();

  //mapping the string based json data to document using the fruit crud model
  FruitCrudModel.fromMap(Map<String, dynamic> data) {
    id = data['id'];
    name = data['name'];
    category = data['category'];
    image = data['image'];
    countries = data['countries'];
    createdAt = data['createdAt'];
    updatedAt = data['updatedAt'];
  }
 //returning the created data using fruit crud model
  Map<String, dynamic> toMap() {
    return {
      'id': id,
      'name': name,
      'category': category,
      'image': image,
      'countries': countries,
      'createdAt': createdAt,
      'updatedAt': updatedAt
    };
  }


}
```

Fruit_Controller.dart

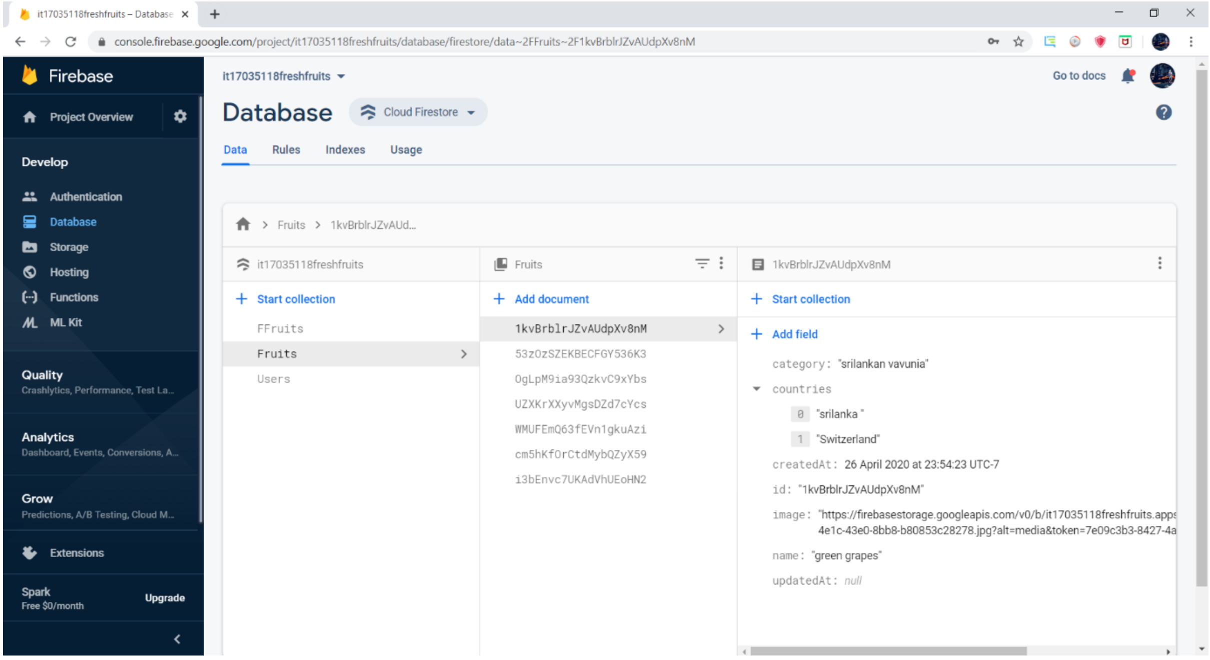
```
import 'dart:collection';
import 'package:finalproject/Crudmodel/FruitCrudModel.dart';
import 'package:finalproject/Crudmodel/UserCrudModel.dart';
import 'package:flutter/cupertino.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the firebase fruit object to set all the data to the notify
classes
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */
```

```
//creating the Fruit details list  controller class for fruit object handling
class FruitController with ChangeNotifier {
  //creating a list object to store private fruit details
  List<FruitCrudModel> _fruitList = [];
  //creating a list object to store private user details
  List<UserCrudModel> _userList= [];
  //creating a object to store private current loading fruit details
  FruitCrudModel _currentFruit;
  //creating a  object to store private after login user details in current
session
  UserCrudModel _currentUser;
  //creating a list view to get fruit details
  UnmodifiableListView<FruitCrudModel> get fruitList =>
UnmodifiableListView(_fruitList);
  //creating a list view to get user details
  UnmodifiableListView<UserCrudModel> get userList =>
UnmodifiableListView(_userList);
  //getting the current fruit details
  FruitCrudModel get currentFruit => _currentFruit;
  //getting the current user details
  UserCrudModel get currentUser => _currentUser;

  //creating the fruit details setter object to set the fruit data
  set fruitList(List<FruitCrudModel> fruitList) {
    _fruitList = fruitList;
    notifyListeners();
  }
  //creating the user details setter object to set the user data
  set userList(List<UserCrudModel> userList) {
    _userList = userList;
    notifyListeners();
  }
  //creating the current loading fruit details setter object to set its relevant
fruit data
  set currentFruit(FruitCrudModel fruit) {
    _currentFruit = fruit;
    notifyListeners();
  }

 //creating the current loading user details setter object to set its relevant
user data after authentication
  set currentUser(UserCrudModel user) {
    _currentUser = user;
    notifyListeners();
  }
  //creating the adding fruit method to add fruit details to store
  addFruit(FruitCrudModel fruit) {
    _fruitList.insert(0, fruit);
    notifyListeners();
  }
  //creating the adding user method to add user details to store
  addUser(UserCrudModel user) {
    _userList.insert(0, user);
    notifyListeners();
  }
  //creating the deleting fruit method to delete fruit details from fire cloud
store
  deleteFruit(FruitCrudModel fruit) {
    _fruitList.removeWhere((_fruit) => _fruit.id == fruit.id);
    notifyListeners();
  }
  //creating the deleting fruit method to delete favorite fruit details from fire
cloud store
  deleteFavouriteFruit(FruitCrudModel fruit) {
    _fruitList.removeWhere((_fruit) => _fruit.id == fruit.id);
    notifyListeners();
  }


}
```

authentication_Controller.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/widgets.dart';
/*
Author      : W.G.M.V.S Wijesundara  IT17035118
description : Creating the firebase user object to set all the data to the notify
classes
reference1:  https://github.com/JulianCurrie/CwC_Flutter
reference2: https://www.youtube.com/watch?v=bjMw89L61FI
reference3: https://github.com/TechieBlossom/sidebar_animation_flutter
reference4: https://apkpure.com/flutter-mobile-restaurantui-
kit/com.jideguru.restaurant_ui_kit
 */

//creating the authentication  controller class for user object handling
class AuthenticationController with ChangeNotifier {

  //Firebase private user object declaration
  FirebaseUser _user;

  //creation of getter of user details to the user object
  FirebaseUser get user => _user;

  //creating the user details setter object to set the user data
  void setUser(FirebaseUser user) {
    _user = user;
    notifyListeners();
  }
}
```

4. Screenshots of Database


it17035118freshfruits – Database
console.firebase.google.com/project/it17035118freshfruits/database/firestore/data~2FFruits~2F1kvBrblrJZvAUdpXv8nM
Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics, Performance, Test La...
Analytics
Dashboard, Events, Conversions, A...
Grow
Predictions, A/B Testing, Cloud M...
Extensions
Spark
Free $0/month
Upgrade
it17035118freshfruits
Go to docs
Database
Cloud Firestore
Data
Rules
Indexes
Usage
Fruits > 1kvBrblrJZvAUd...
Start collection
Add document
Start collection
Add field

it17035118freshfruits

- FFruits
- Fruits
- Users

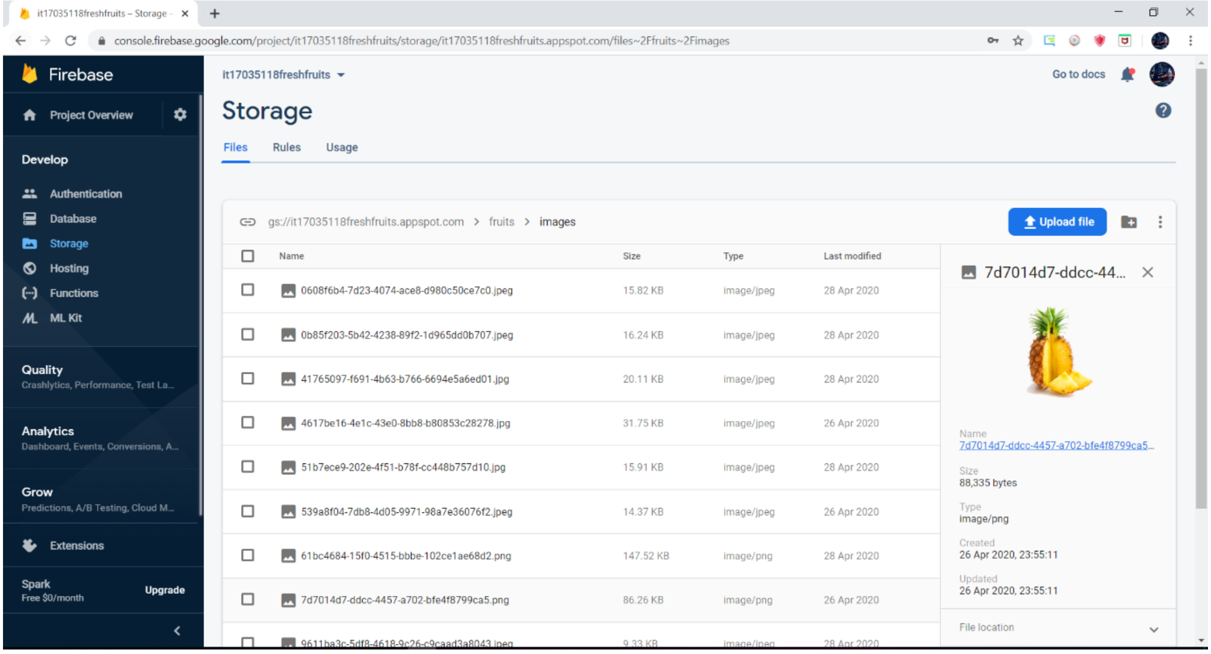
Fruits

- 1kvBrblrJZvAUdpXv8nM
- 53z0zSZEKBECFGY536K3
- OgLpM9ia93QzkvC9xYbs
- UZXKrXXyvMgsDZd7cYcs
- WMUFEmQ63fEVn1gkuAzi
- cm5hKf0rCtdMybQZyX59
- i3bEnvc7UKAdVhUEoHN2

1kvBrblrJZvAUdpXv8nM

```
category: "srilankan vavunia"
countries
    0 "srilanka "
    1 "Switzerland"
createdAt: 26 April 2020 at 23:54:23 UTC-7
id: "1kvBrblrJZvAUdpXv8nM"
image: "https://firebasestorage.googleapis.com/v0/b/it17035118freshfruits.apps
       4e1c-43e0-8bb8-b80853c28278.jpg?alt=media&token=7e09c3b3-8427-4a
name: "green grapes"
updatedAt: null
```

Figure 4.1: Collection in Firebase


it17035118freshfruits – Storage
console.firebase.google.com/project/it17035118freshfruits/storage/it17035118freshfruits.appspot.com/files~2Ffruits~2Fimages
Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics, Performance, Test La...
Analytics
Dashboard, Events, Conversions, A...
Grow
Predictions, A/B Testing, Cloud M...
Extensions
Spark
Free $0/month
Upgrade
it17035118freshfruits
Go to docs
Storage
Files
Rules
Usage
Upload file

gs://it17035118freshfruits.appspot.com > fruits > images

Name	Size	Type	Last modified
0608f6b4-7d23-4074-ace8-d980c50ce7c0.jpeg	15.82 KB	image/jpeg	28 Apr 2020
0b85f203-5b42-4238-89f2-1d965dd0b707.jpeg	16.24 KB	image/jpeg	28 Apr 2020
41765097-f691-4b63-b766-6694e5a6ed01.jpg	20.11 KB	image/jpeg	28 Apr 2020
4617be16-4e1c-43e0-8bb8-b80853c28278.jpg	31.75 KB	image/jpeg	26 Apr 2020
51b7ece9-202e-4f51-b78f-cc448b757d10.jpg	15.91 KB	image/jpeg	28 Apr 2020
539a8f04-7db8-4d05-9971-98a7e36076f2.jpeg	14.37 KB	image/jpeg	26 Apr 2020
61bc4684-15f0-4515-bbbe-102ce1ae68d2.png	147.52 KB	image/png	28 Apr 2020
7d7014d7-ddcc-4457-a702-bfe4f8799ca5.png	86.26 KB	image/png	26 Apr 2020
9611ba3c-5df8-4618-9c26-c9caad3a8043.jpeg	9.33 KB	image/jpeg	28 Apr 2020

7d7014d7-ddcc-44...

Name
7d7014d7-ddcc-4457-a702-bfe4f8799ca5...

Size
88,335 bytes

Type
image/png

Created
26 Apr 2020, 23:55:11

Updated
26 Apr 2020, 23:55:11

File location

Figure 4.2: Storage View in Firebase

5. References

[1] Julian Curry “Code With Curry ”, https://github.com/JulianCurrie/CwC_Flutter [Accessed:20-Jan-2020]

[2] Julian Curry “Code With Curry ”, https://www.youtube.com/watch?v=bjMw89L61FI [Accessed: 20-Jan-2020]

[3]TechieBlossom“SideBarNavigationalPage”https://github.com/TechieBlossom/sidebar_animation_flutter [Accessed: 02-Feb-2020]

[4] Restaurant UI Kit “Flutter Front UI Design” https://apkpure.com/flutter-mobile-restaurant-ui-kit/com.jideguru.restaurant_ui_kit [Accessed: 06-Mar-2020]