DIET AND FITNESS APP

A PROJECT REPORT SUBMITTED TO THE ENGINEERING FACULTY

OF

NEAR EAST UNIVERSITY

BY

SEDAT DAYAN

Advisor: MSc Kezban Alpan

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

IN

SOFTWARE ENGINEERING

NICOSIA, 2023

ACKNOWLEDGEMENTS

The completion of this study would not have been possible without the guidance of my supervisor Kezban Alpan, who was an amazing teacher all through the time of this project work. I would also like to thank Assoc. Prof. Dr. Boran Şekeroğlu. Lastly, I would love to acknowledge my friends; Can Ilgu, Hanifi Adiguzel,Eren Esen, who tested the application related to this thesis and gave me their critical points and for been a blessing to my life throughout this project. Above all, my heartfelt gratitude to my parents for their great confidence in me.

ABSTRACT

Today, there are many people who are dealing with the problem of obesity. These people do not have much information about how to eat properly and what kind of exercises to do, or there is no source that can be guided correctly. This project is a mobile application where people with obesity can make their nutrition more accurate and find exercise movements covering many body regions.This project is going to develop by using Dart work to implement backend. For frontend part I have used the Flutter framework. Finally, to store all the data I used the Firestore database.In recent, these technologies are very popular around mobile application development. By following these technologies, I have tried to develop a quality mobile application project as a student.

TABLE OF CONTENTS

LIST OF FIGURES

List Of Abbreviations

APP:	Application
MAD:	Mobile Application Development
MAE:	Mobile Application
IDM:	Iterative Development Method
BMI:	Body Mass Index

CHAPTER 1

1. Introduction

Nowadays, people can easily handle most of their work on mobile devices. Therefore, mobile applications have become very popular and important. There are many free applications with diet and fitness content separately, but there are not many applications with both diet and fitness content. I decided to create a mobile application to take advantage of this opportunity.

I felt the need to make an application where people can see how they can do fitness movements correctly, how they can work properly within the right program, and what people who want to diet can eat according to their BMA during the day. Thanks to this application, you can easily access the application from your mobile phone, which you always have with you and which is easy to carry, and you will be able to continue the program easily.

Figure 1: Mobile app development (MAD)

1.1.Aim Of the Project

The aim of this project is to focus specifically on people who go to sports or want to diet. People who go to sports or want to diet often do not know exactly what they should do or how

they should do it. The main purpose of this thesis is to design a mobile application called Diet and Fitness App. Diet and Fitness App is a platform where users can see their exercise movements, also calculate their BMI and see what they can eat according to the result. It will be very easy for all people to access and benefit from them. The biggest problems of people, especially people who have just started that business They do not know exactly where they should start and what they should do, and they hesitate to consult those who know. This is exactly where the Diet and Fitness App will come into play and solve this problem. Diet and Fitness App's mission is to help people live healthier and feel better with the right exercises and nutrition.

With this application, I aim to encourage people to do sports throughout their lives, to eat healthier according to their BMI, and to enable them to be more successful in life. Not only people who do sports or diet, but also people who want to calculate their BMI or want to find out how many points they eat during the day can use this application.

1.2. Limitations Of the Project

This project, like every project, has certain limits.This project appeals to almost all age groups except little then 18 years old.Anyone who does not have any health problems or special conditions can use this application easily.However, my main target audience in this application will be people with obesity disease. I will definitely be making this reminder in the application. In the application, there will be only a certain number of products that are known and consumed by everyone as a start. These products will also have calorie scoring. In the future, the number of products for the purpose of developing this project may be increased.

Initially, only English language support will be available within the application. In the future, new languages can be added. In the future, an area can be added where users can contact the application owner directly so that they can easily express the deficiencies or errors they see in the application.

If the application attracts attention, we can make an agreement with experts in the field of sports and diet in order to further develop the project, and enable users to communicate with them from within the application. In this case, we will have the chance to offer user-specific training programs and diet programs.

1.3. Requirements Analysis

Obesity is increasingly becoming an important public health concern among all age groups in most of the developed world.A World Health Organization (WHO) Consultation described obesity as a chronic disease that is so prevalent in both developed and developing countries that it is replacing the more traditional public health concerns, such as undernutrition and infectious diseases, as a significant contributor to ill health/ In the United States, about one-third of Americans are currently classified as obese. Being overweight or obese is associated with elevated risks for an array of chronic disease and disabilities including diabetes, heart disease, high blood pressure, stroke, cancers, gallbladder disease, obstructive sleep apnea syndrome, and metabolic syndrome, more problems with activities of daily living and lower scores on self reported quality of life data. When I did research on the subjects I mentioned, I could not see a completely free mobile application that people could use on this subject. I decided to create such a mobile application to take advantage of this opportunity.

In this application, I used the weight watchers diet for the diet part. Weight Watchers is the most widely used commercial diet in the world. The Weight Watchers program involves some optional but commonly used tools such as a diet plan, weekly meetings for support, and tips for exercise and mindful eating. Weight Watchers diet programs have always been based on using a moderate calorie restriction for weight loss. Weight Watchers tends to be a reasonably balanced plan compared to healthy eating guidelines, especially its higher carbohydrate plan as it allows for more fiber intake. Weight Watchers seems to have greater adherence than other more restrictive diets since it allows greater flexibility in food choices and participants are able to buy traditional grocery foods.

In recent years, much work has been done in that field. I have done some research below and discussed.

Figure 2: Mobile app example (MAE)

MyFitnessPal provides personalized diet program and calorie planning services. After logging into the application, you need to enter height, weight and age.

This platform is similar to my project, but if you want to use this platform, you have to pay monthly or annually (Evans, 2017)

In 2005, MyFitnessPal was developed and created by Albert Lee and Mike Lee.
On February 4, 2015, MyFitnessPal was acquired by athletic apparel maker, Under Armour, in a deal worth $475 million. MyFitnessPal had 80 million users at the time.
On October 30, 2020, Under Armour announced that MyFitnessPal would be sold to the private equity firm Francisco Partners for $345 million.

1.4. Software Development Methodology

For Diet and Fitness App I will be using and following part of Agile methodology which is Iterative and incremental development. Iterative and incremental development is any combination of both iterative design or iterative method and incremental build model for development. (Larman, Craig, June 2003).

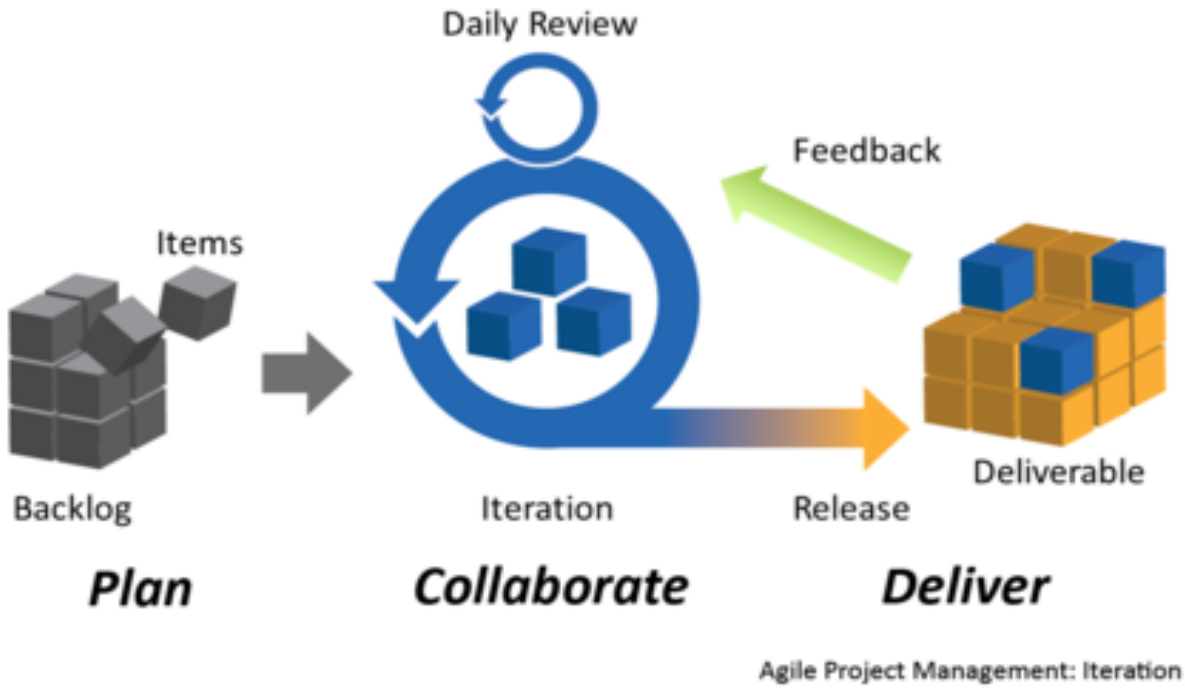

Figure 3: Iterative development model (Planbox, 22 May 2012)

The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take

advantage of what was learned during development of earlier parts or versions of the system. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added

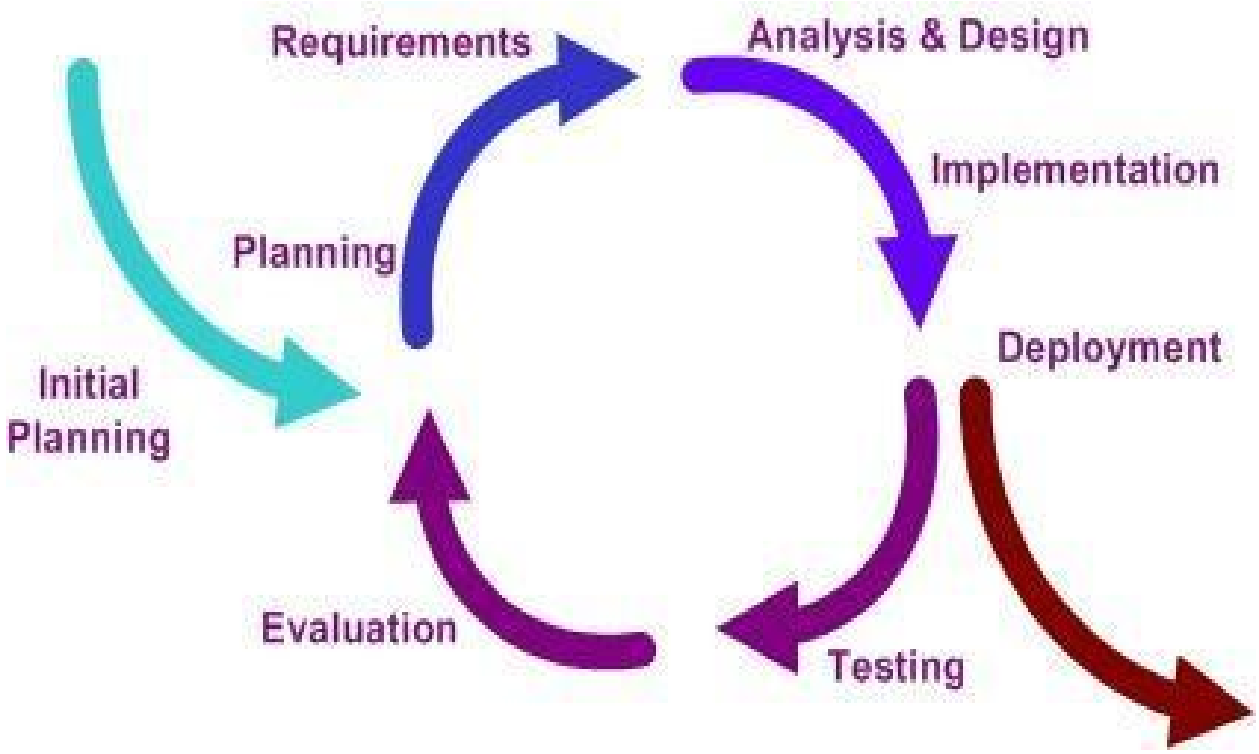

Figure 3: Iterative development model (IDM, 22 Jul 2014)

I have chosen this model because:

- Potential problems and defects are spotted early.
- Development of functional prototypes takes place early in the project life cycle.
- Progress is easy to measure.
- Less time spent on documenting, more on designing.
- Changes to the project are easier to implement and less costly.

- Most risks can be identified during iteration and dealing with bigger risks can be prioritized.
- Operating time is significantly reduced.

So, iterative development is a software development life cycle model which works through small iterations and phases. This model is apt for large software that needs changes based on feedback and review from time to time rather than at the very end. Following every step of the iterative development model properly with the right tech talent on board, will lead to a good quality product that is aligned with the desired functionality in the end. It fits the Diet and Fitness App project.

2. Conceptual Framework

In this project, I will be using the technologies listed below.

- Flutter
- Dart
- Firebase, Firestore
- Python

Flutter is a cross-platform framework that aims at developing high-performance mobile applications. Flutter is publicly released at 2016 by Google. Not only can Flutter applications run on Android and iOS, but also Fuschia, Google's next generation operating system, chooses Flutter as its application-level framework (Wu, 2018).

In Flutter, all applications are written with Dart. Dart is a programming language that is developed and maintained by Google. It is widely used inside of Google and it has been proved to have the capability to develop massive web applications, such as AdWords (Wu, 2018)

Dart was originally developed as a replacement and successor of JavaScript. Thus, it implements most of the important characteristics of JavaScript's next standard (ES7), such as keywords "async" and "await". However, in order to attract developers that are not familiar with JavaScript, Dart has a Java-like syntax (Wu, 2018)

Akin to other systems that utilize reactive views, Flutter application refreshes the view tree on every new frame. This behavior leads to a drawback that many objects, which may live for only one frame, will be created. Dart, as a modern programming language, is optimized to handle this scenario in memory level with the help of "Generational Garbage Collection".

Firebase is a mobile application development platform. It provides services for building, developing, and scaling applications. Firebase offers the possibility to manage necessary but tedious services, such as authentication, databases, or analytics, and allows developers to focus on the app experience itself (Faust, 2020)

Cloud Firestore offers a real-time, cloud-based, scalable database service (Stevenson2018). It ensures data synchronization between client applications and web and mobile applications' responsiveness if the Internet connection is not available. Furthermore, Cloud Firestore is a non-relational, document-oriented database for storing and syncing data (Faust, 2020)

Python is an interpreted, interactive, object-oriented programming language. It provides high-level data structures such as list and associative arrays (called dictionaries), dynamic typing and dynamic binding, modules, classes, exceptions, automatic memory management, etc.. It has a remarkably simple and elegant syntax and yet is a powerful and general purpose programming language. It was designed in 1990 by Guido van Rossum. Like many other scripting languages it is free, even for commercial purposes, and it can be run on practically any modern computer. A python program is compiled automatically by the interpreter into platform independent bytecode which is then interpreted

3. EXPLANATION OF THE PROJECT

Below, you can see screenshots of the project. Also i will add some explanation to get you know more about Diet and Fitness App. In addition this project is a concept work that introduces how Diet and FitnessApp does look like.

3.1 Launch Page

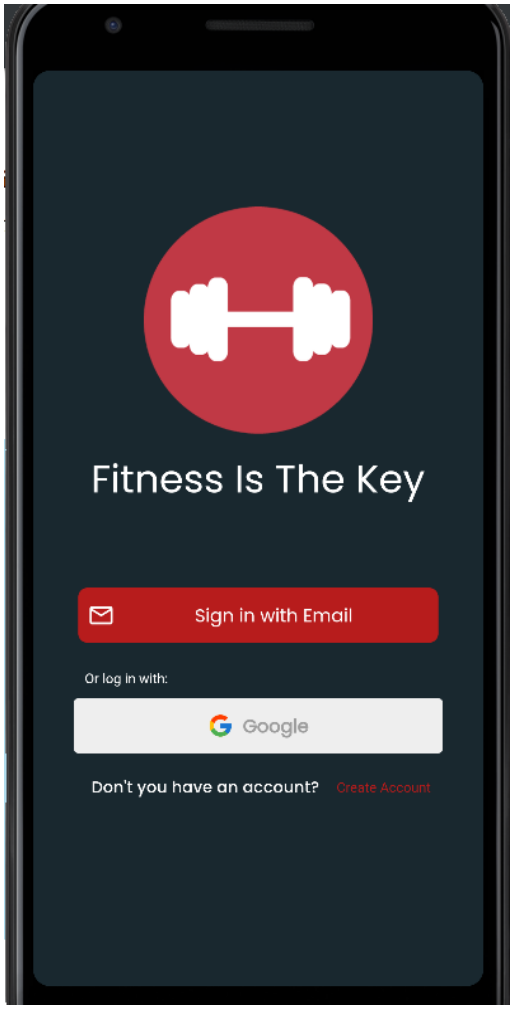

Figure 4. Launch Page

As seen in Figure 4, the user will see this page when they open the application for the first time. User Create an account by clicking Create Account who does not have any account on the platform The page will greet the user. The user will be directed to the login page by clicking Login with Mail. In addition, the user can also use the Login with google option if he/she wants.

3.2 Log In Page

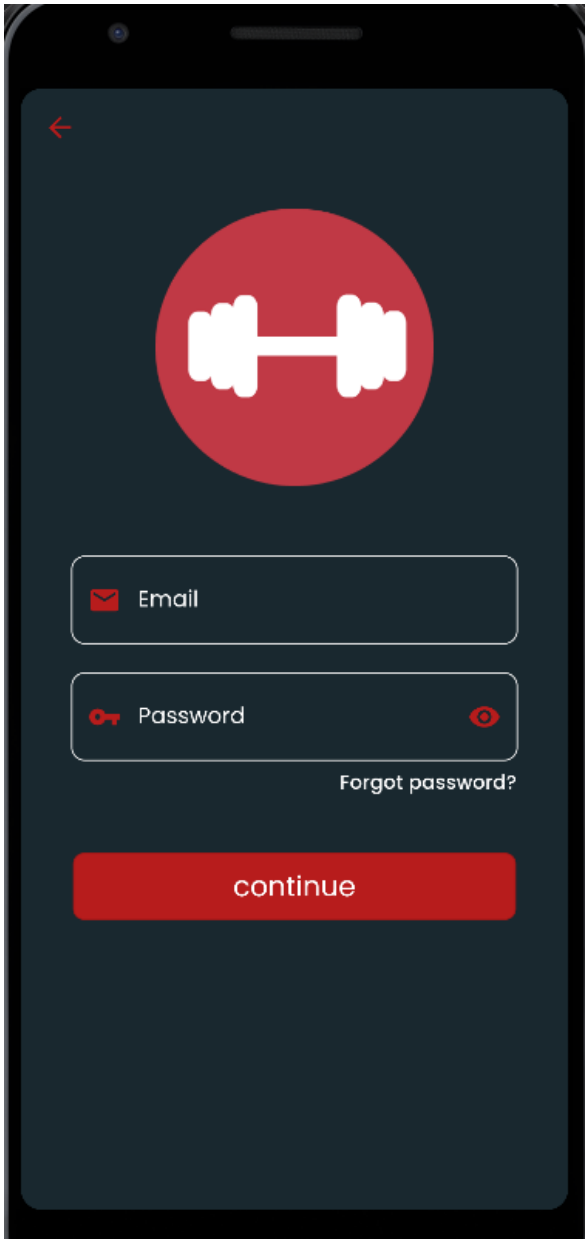

Figure 5. Login Page

As shown in Figure 5, the user will use this page to log in to their account and continue. If user has forgotten his password on the platform, he will be directed to the forgot password page by clicking Forgot Password.

3.3 Forgot Password Page

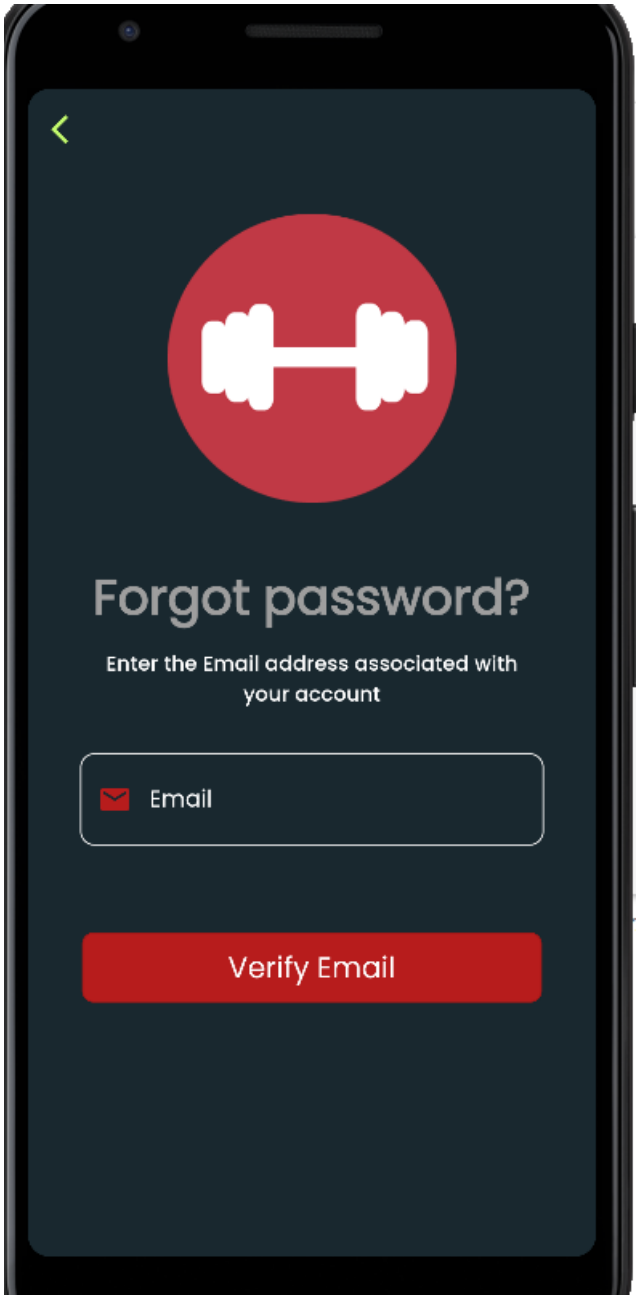

Figure 6. Forgot Password Page

As shown in Figure 6, the user will be able to renew his password via the email sent to him by entering the registered email address to renew his forgotten password.

3.4 Register Page

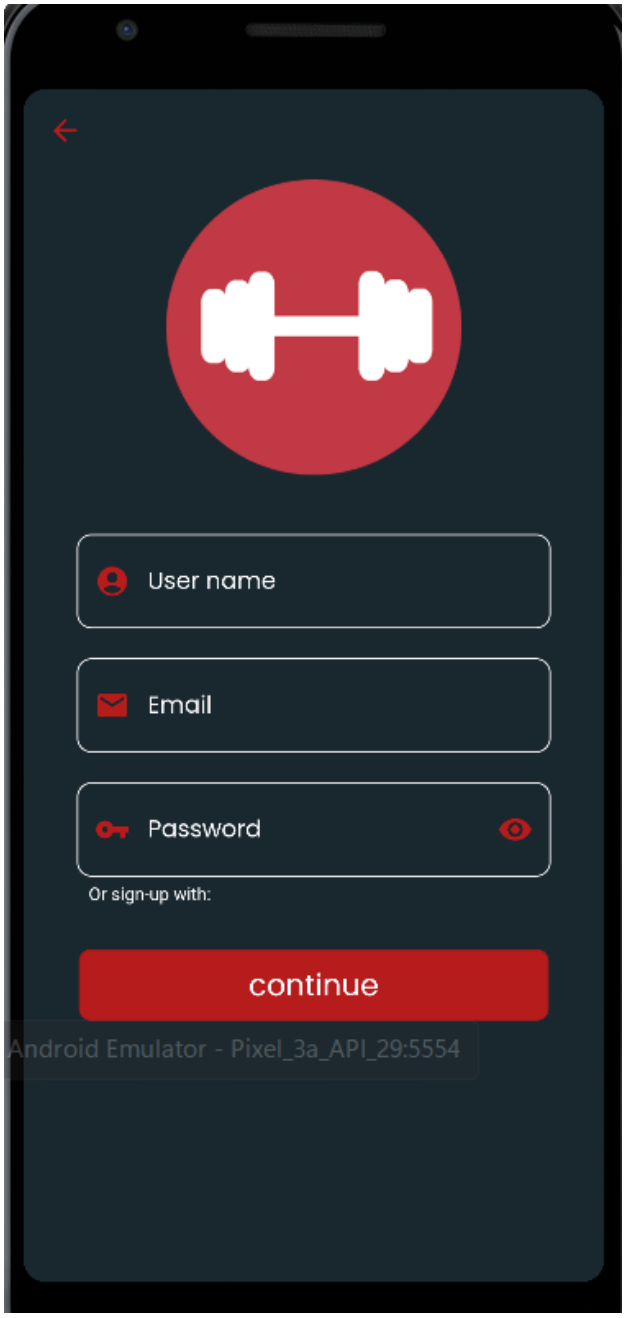

Figure 7. Register Page

As shown in Figure 7, the user will first use this page to create an account. By clicking the Continue button, he will be greeted with the other required pages for registration.

3.5 Name Page

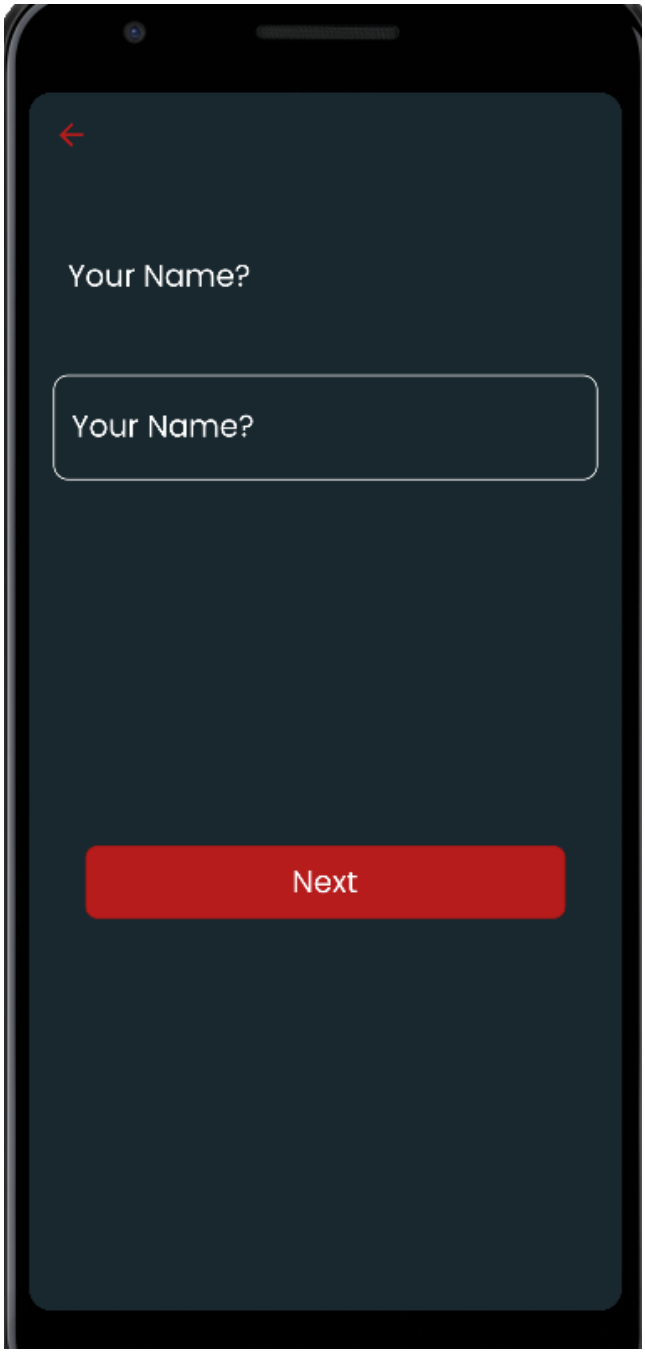

Figure 8. Name Page

As shown in Figure 8, the user will enter their own name on this page.

3.6 Gender Page

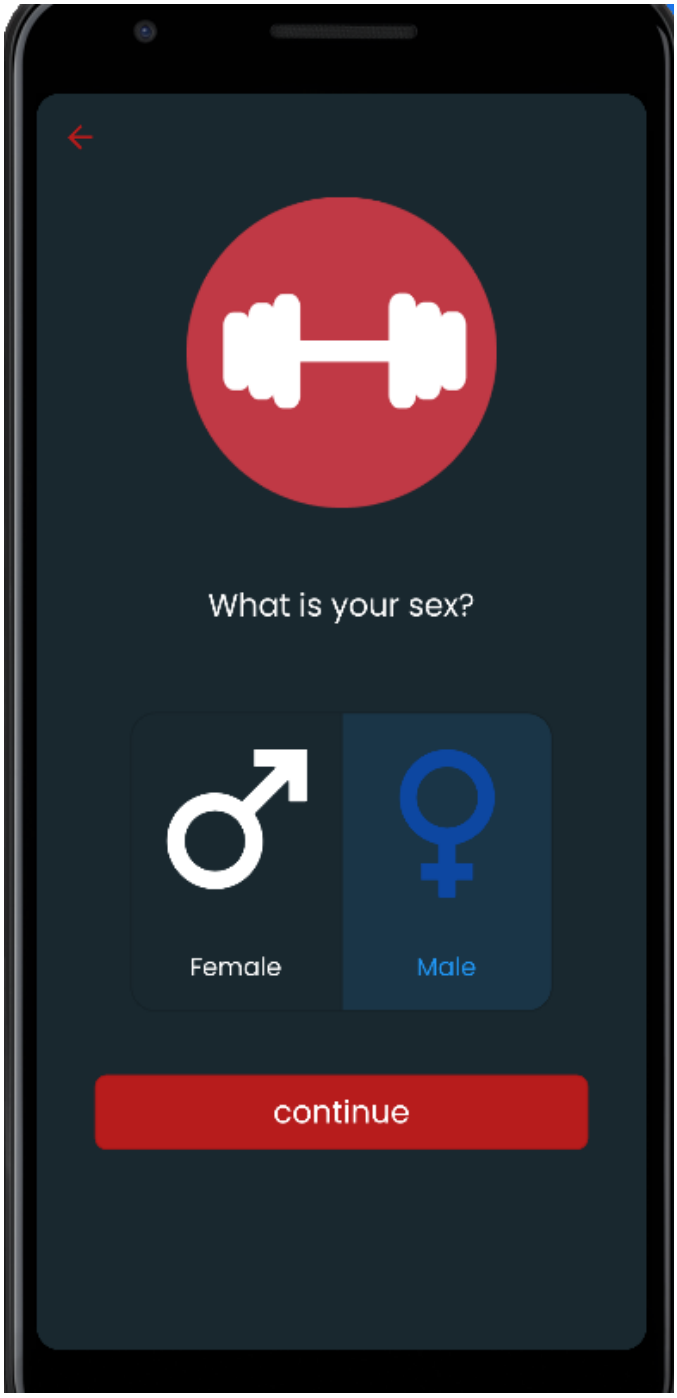

Figure 9. Gender Page

As shown in Figure 9, the user will select their gender on this page.

3.7 Birth of Date Page

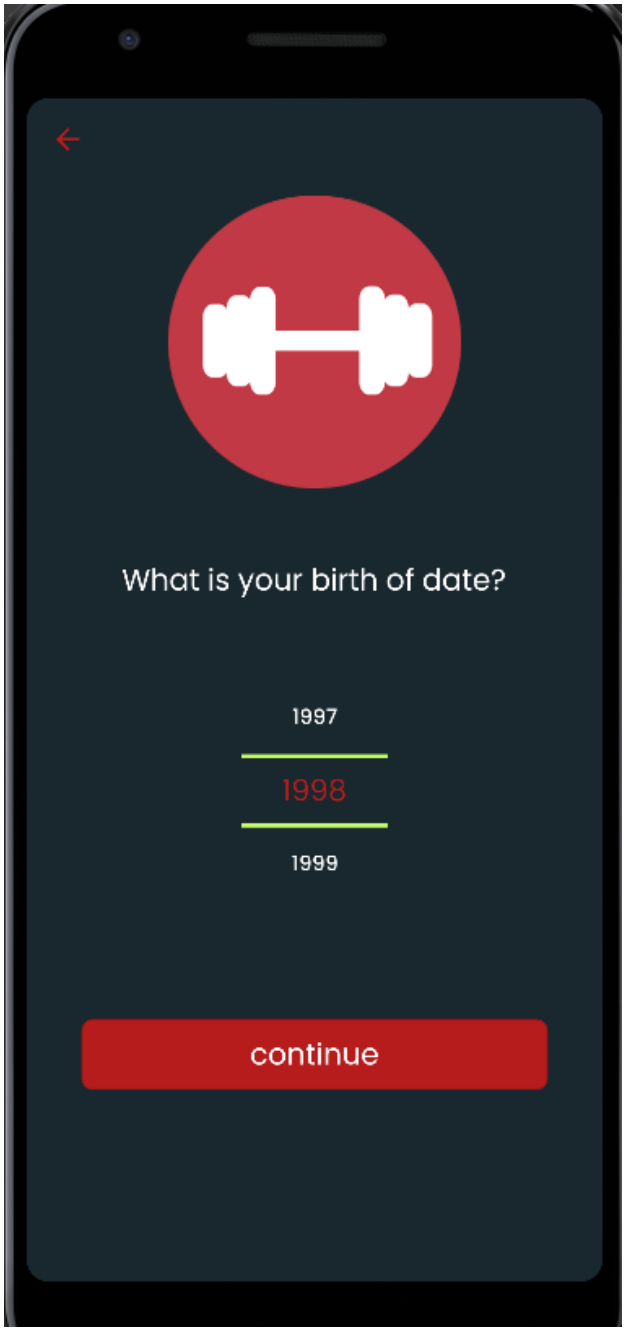

Figure 10. Birth of Date Page

As shown in Figure 10, the user will select the year of birth on this page.

3.8 Mobility Page

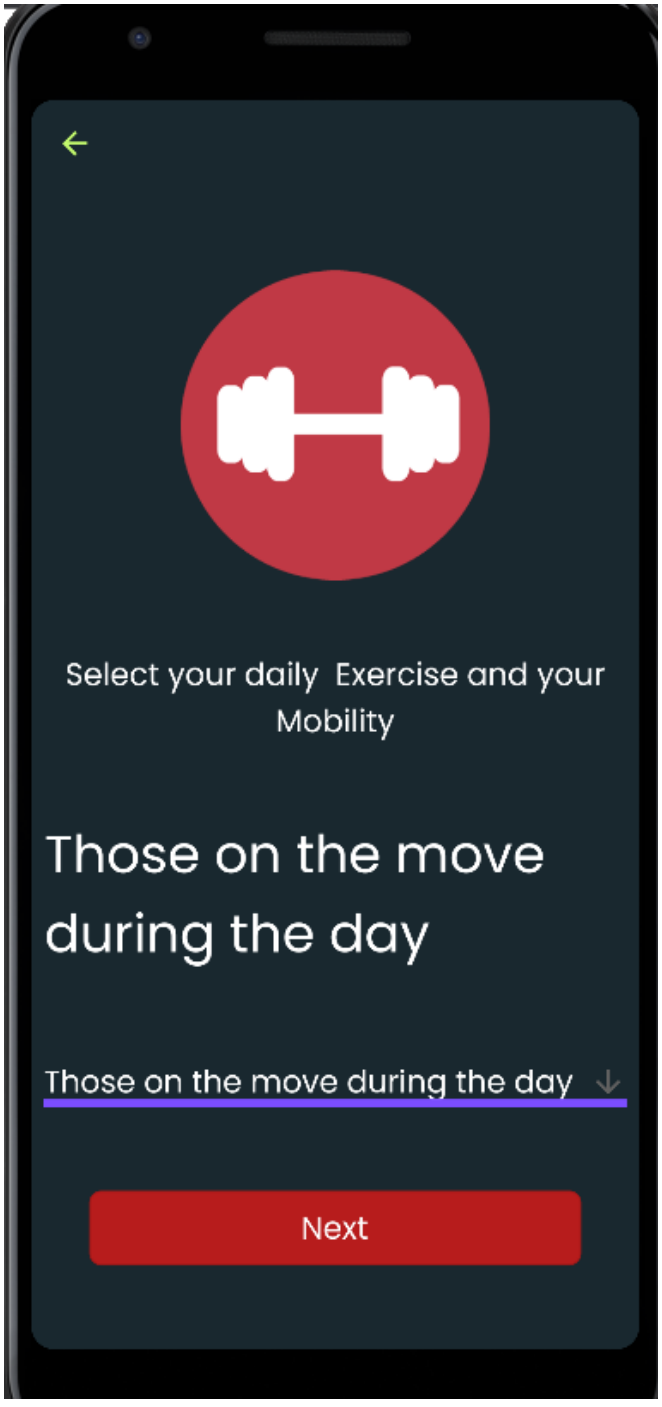

Figure 11.Mobility Page

As shown in Figure 11, the user will select the daily movement tempo on this page.

3.9 Height Page

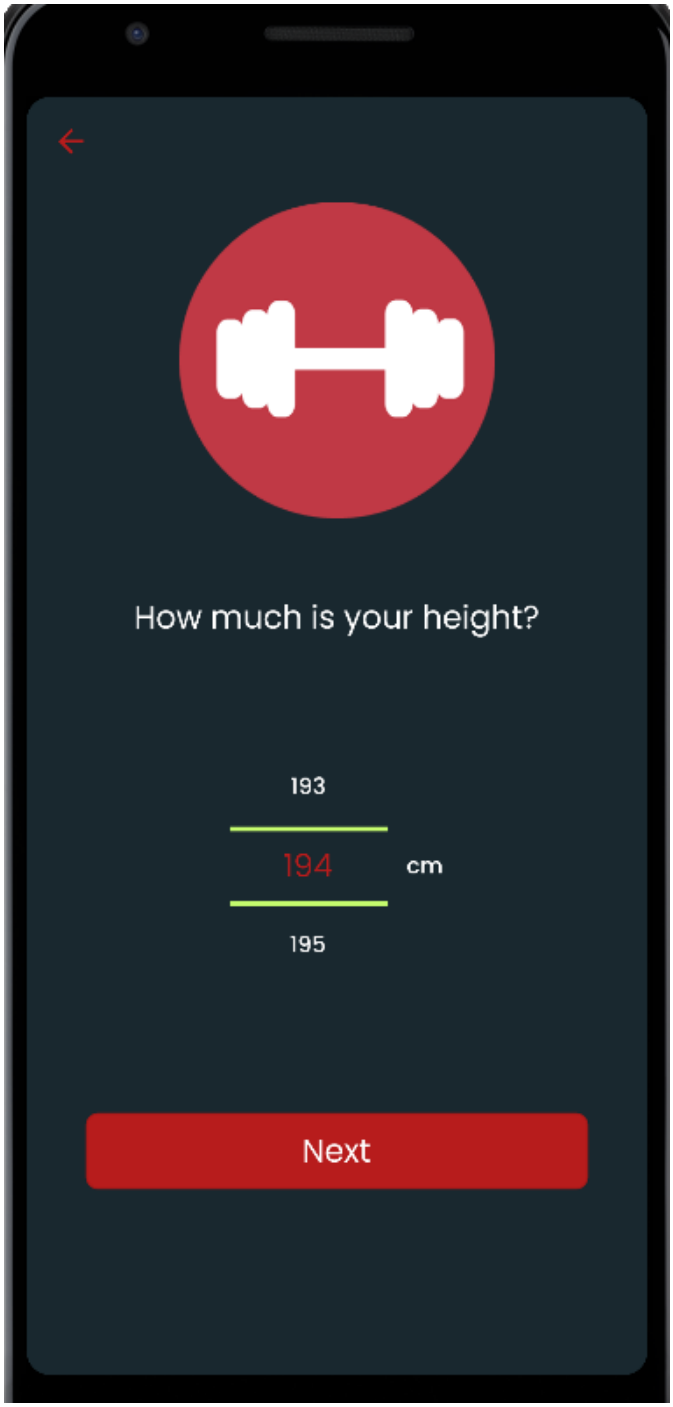

Figure 12. Height Page

As shown in Figure 12, the user will choose the height on this page.

3.10 Weight Page

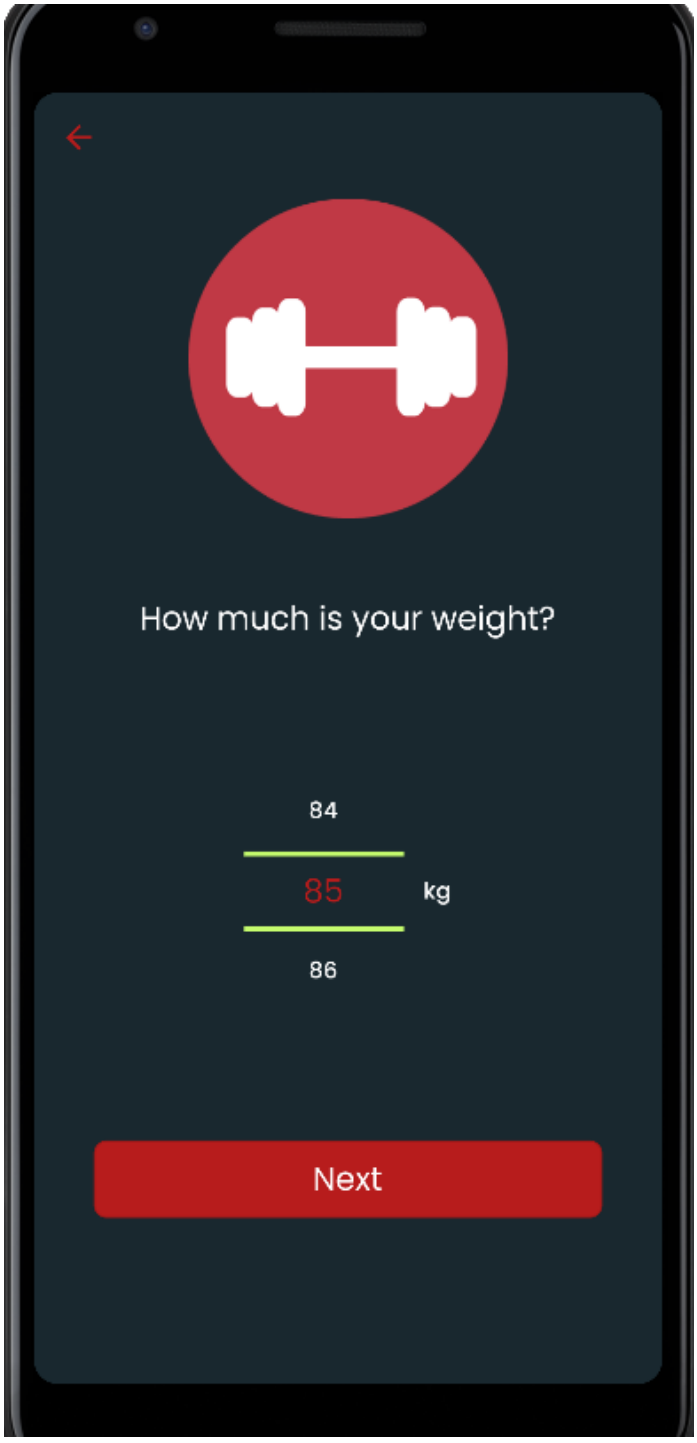

Figure 13. Weight Page

As shown in Figure 13, the user will select their weight on this page.And after this page the user has to verify his/her email addresses.After that user can login to Main Page.

3.11 Main Page

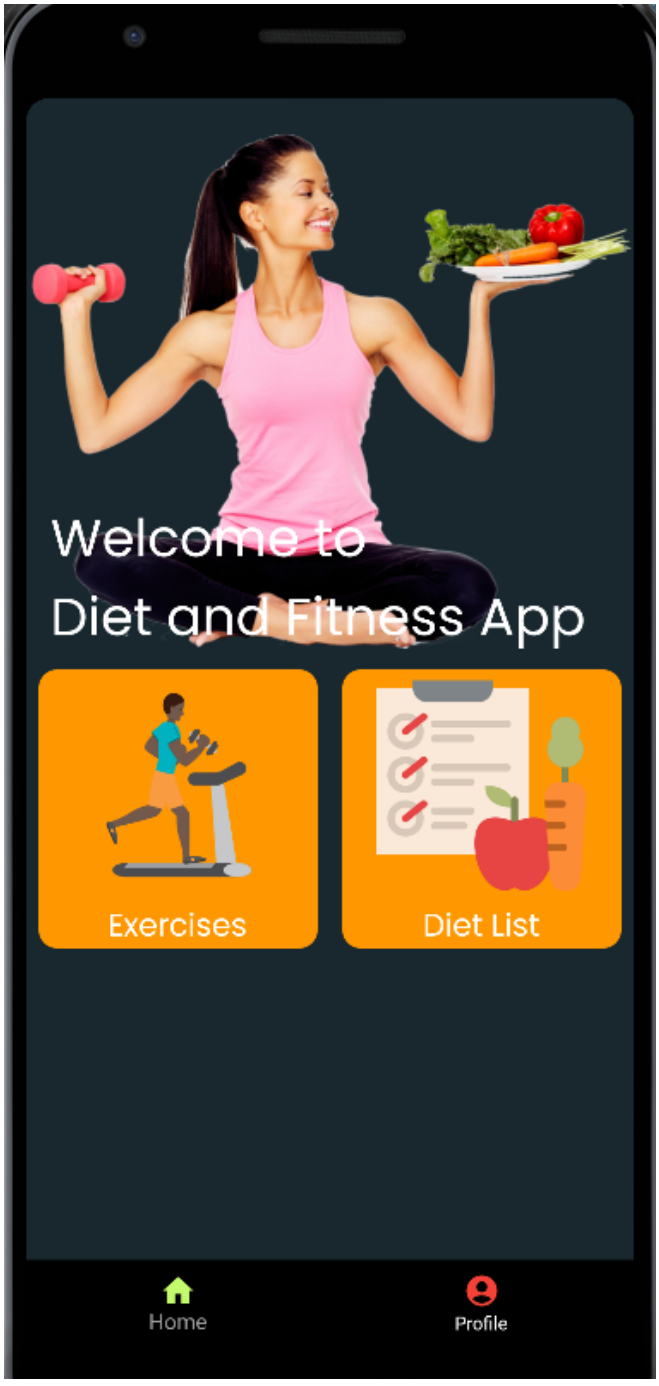

Figure 14. Main Page

This screen welcomes you when you log in.As you can see in Figure 14, which exercise or diet list page the user can go to. Also, they can go to their profile page.

3.12 Exercises Page

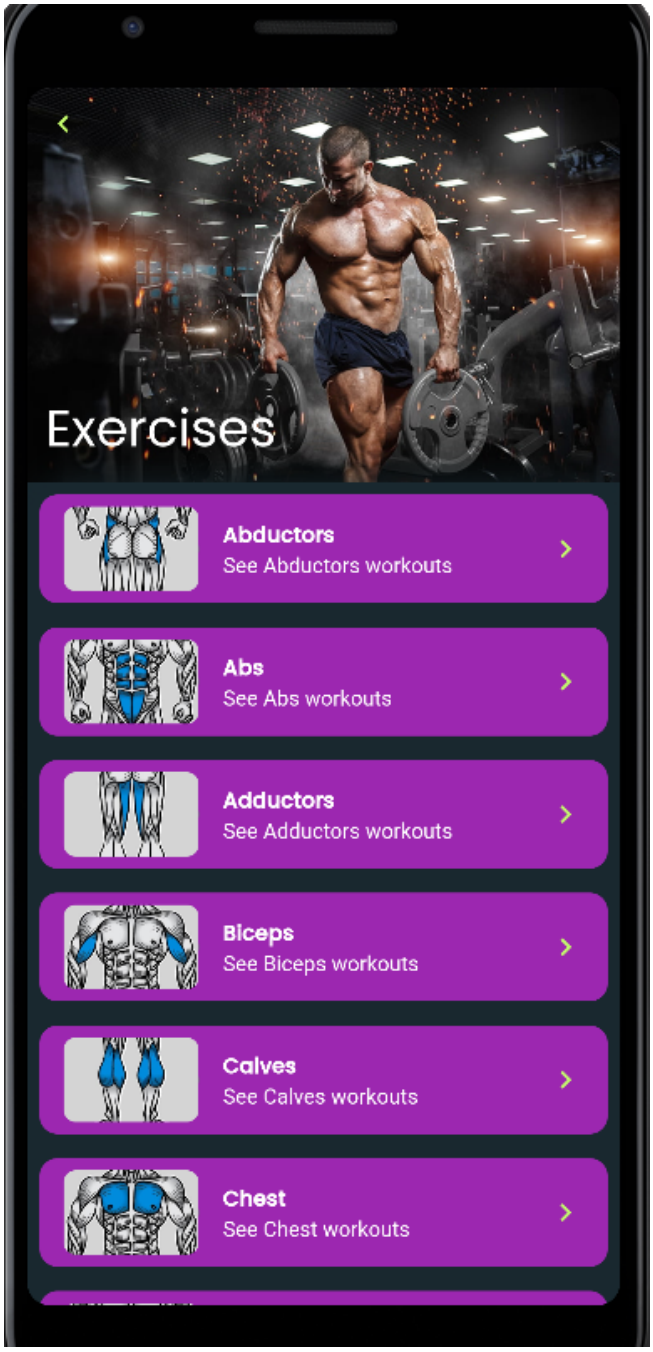

Figure 15. Exercise Page

As shown in Figure 15, the user can select the desired region according to the body regions on this page.

3.13 Specific Exercise Page

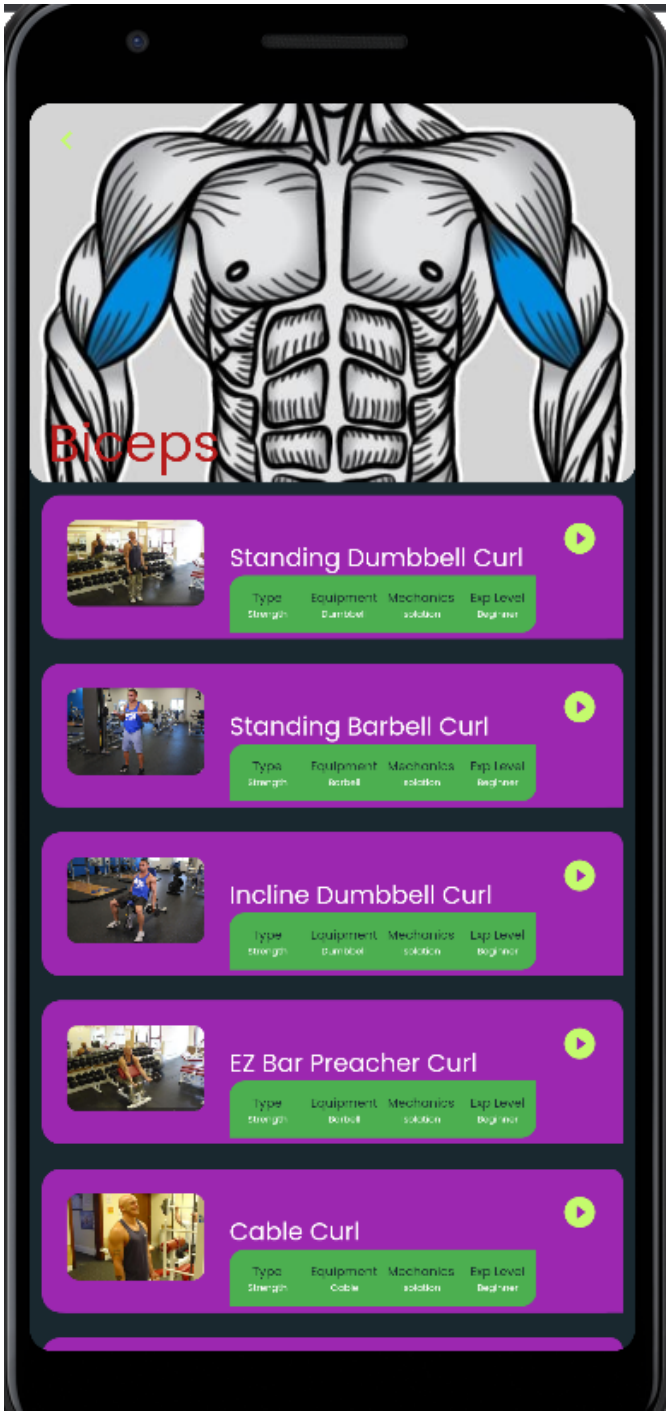

Figure 16.Specific Exercise Page

As shown in Figure 16, on this page, the user can see multiple movement types of the body which region they selected earlier.

3.14 Detail Exercise Page

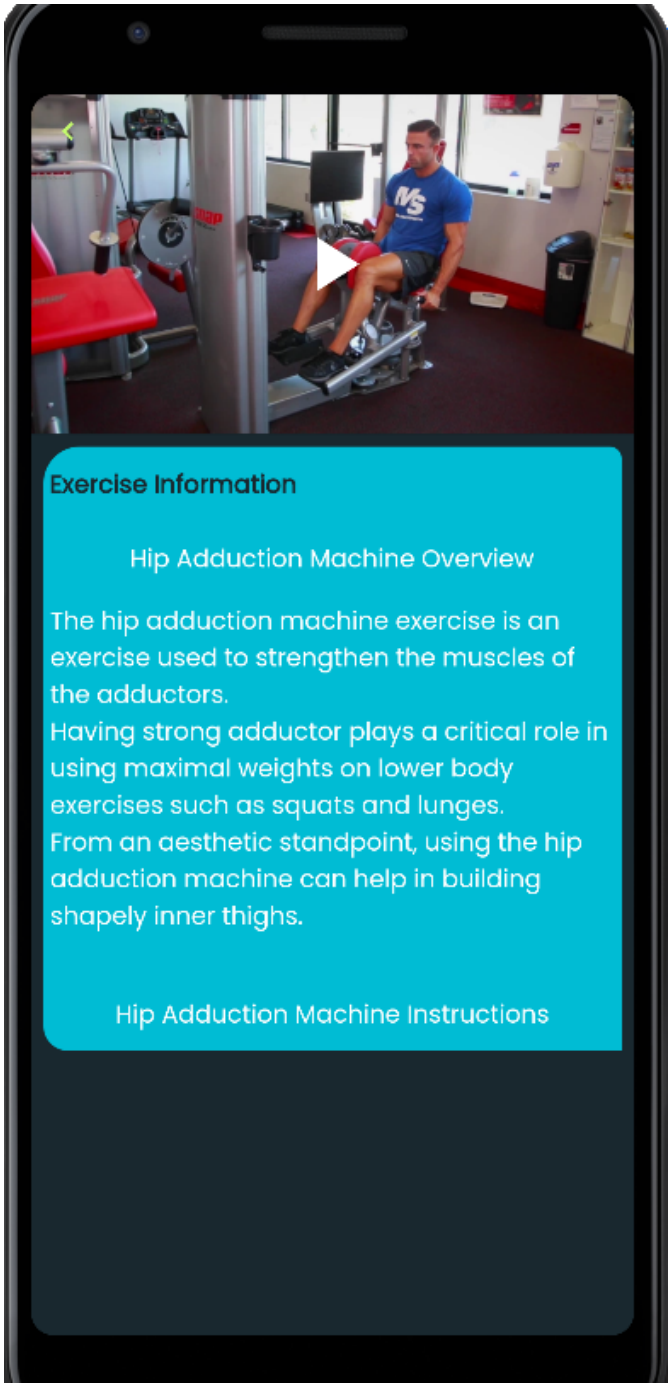

Figure 17. Detail Exercise Page

As shown in Figure 17, the user will see the video of the exercise movement they want to do and information about how to do the movement.

3.15 Diet List Page

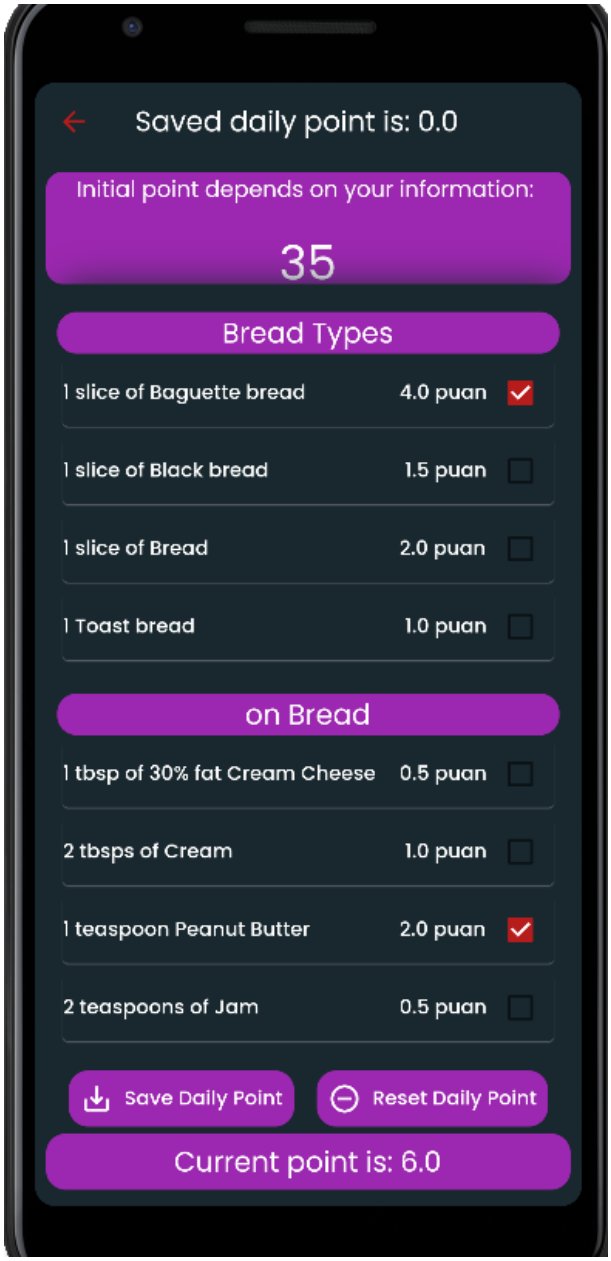

Figure 18. Diet List Page

As shown in Figure 18, the user will create the daily score by selecting the food and beverages consumed during the day on this page. At the top of the page, there is also the right of points calculated according to the information entered by the user while registering. In addition, by clicking the Save Daily Point button after selecting the products, he will be able to keep his daily score without errors until another day during the day.

3.16 Profile Page

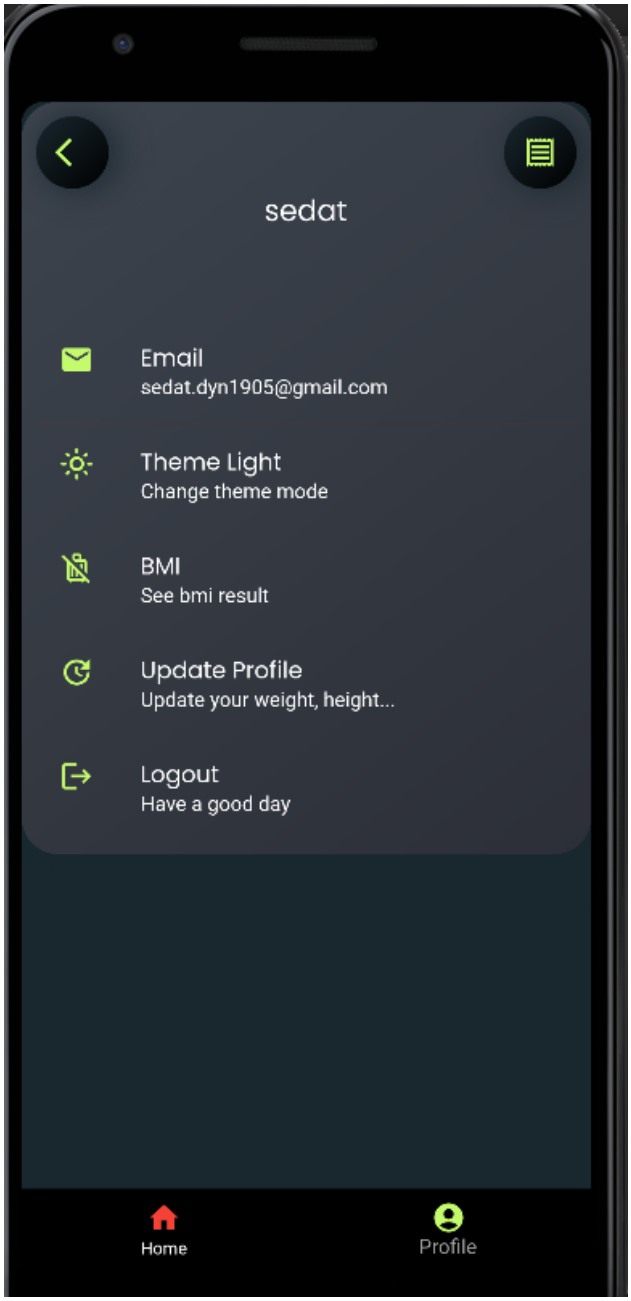

Figure 19. Profile Page

As shown in Figure 19, the user will be able to see the registered username and email account on the profile page. You can also change the theme. By clicking on BMI, he will be directed to the BMI page where he can see the bmi result.The user can also log out by clicking the Log Out button and update their current weight and height by clicking the Update my information button.

3.17 BMI Page

Figure 20. BMI Page

As shown in Figure 20, on this page, the user can see the result of the BMI calculated according to the weight and height entered before.

3.18 Update Page

Figure 21. Update Page

As shown in Figure 21, the user will be able to update his height and weight on this page and by clicking the update button he will be directed to the bmi page to see the new bmi result.

Conclusion

In this project, I created a completely free mobile application that anyone can use, but targeting people with the main obesity disease. Diet and Fitness Application is an application that will show which exercise movements work which region, how to perform exercise movements, and which foods they can consume according to the BMI calculation results. It will be developed with Dart and Flutter Framework. Firebase will be used for users to sign in with google. On the other hand, I used Firestore as a database to store the user's entered information. With this application, people will be able to access both from a single application without the need to use separate applications for diet and sports.

References:

Barnett, M. (2018). The Weight Watchers Diet. Clinical Guide to Popular Diets, 113-126.

Evans, D. (2017). MyFitnessPal. BMJ Journals, 1101-1102.

Faust, S. (2020). *Using Google´s Flutter Framework for the Development of a Large-Scale Reference Application.* Germany: Fakultät 10 .

Hubbard, V. S. (2000). Defining overweight and obesity: what are the issues?. The American journal of clinical nutrition, 72(5), 1067-1068.

Wang, H. Y., Liao, C., & Yang, L. H. (2013). What affects mobile application use? The roles of consumption values. *International Journal of Marketing Studies*, *5*(2), 11.

Wu, W. (2018). *React Native vs Flutter, cross-platform mobile application frameworks.* Finland: Metropolia University of Applied Sciences Bachelor of Engineering.

Salihu, H. M., Bonnema, S. M., & Alio, A. P. (2009). Obesity: what is an elderly population growing into?. Maturitas, 63(1), 7-12.

Sanner, M. F. (1999). Python: a programming language for software integration and development. J Mol Graph Model, 17(1), 57-61.

Planbox (22 May 2012) from (https://commons.wikimedia.org/wiki/File:Agile_Project_Management_by_Planbox.png)

MAD from (https://www.pngfind.com/mpng/iRxbxwh_mobile-application-development-mobile-app-development-hd-png/)

MAE from (https://www.pngfind.com/mpng/iiRiJTm_wherever-you-go-train-with

mayweather-boxing-fitness/)

IDM from

(https://upload.wikimedia.org/wikipedia/commons/a/ac/Iterative_development_model_V2.jpg)

APPENDIX

Here are the source codes of this mobile application that I have been coding.

main.dart

```
import 'package:fistness_app_firebase/product/global/theme_control.dart';

import 'package:fistness_app_firebase/views/home/bottomNavigateBar/navigare_bar.dart';

import 'package:fistness_app_firebase/views/service/foods_exercises_service.dart';

import 'package:fistness_app_firebase/views/service/project_network.dart';

import 'package:fistness_app_firebase/views/views_shelf.dart';

import 'package:flutter_native_splash/flutter_native_splash.dart';

import 'package:provider/provider.dart';

import 'package:shared_preferences/shared_preferences.dart';


import 'firebase_options.dart';


Future<void> main() async {

  WidgetsBinding widgetsBinding = WidgetsFlutterBinding.ensureInitialized();

  FlutterNativeSplash.preserve(widgetsBinding: widgetsBinding);


  await SystemChrome.setEnabledSystemUIMode(SystemUiMode.immersive);

  await Firebase.initializeApp(

    options: DefaultFirebaseOptions.currentPlatform,

  );
```

```
  runApp(MultiProvider(
    providers: [
      ChangeNotifierProvider<ThemeNotifier>(
        create: (context) => ThemeNotifier(),
      )
    ],
    builder: (context, child) => const MyApp(),
  ));
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const MyHomePage();
  }
}

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

  @override
```

```
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  Future<bool?> initialization() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();

    bool? isSuccess =
        await GeneralService(ProjectNetworkManager.instance.service, "token")
            .checkToken(prefs.getString("token"));
    if (isSuccess!) {
      FlutterNativeSplash.remove();
      return true;
    } else {
      FlutterNativeSplash.remove();
      return false;
    }
  }

  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
        future: initialization(),
```

```
      builder: (context, AsyncSnapshot<dynamic> data) {
        if (data.connectionState != ConnectionState.waiting) {
          if (data.data) {
            return MaterialApp(
              debugShowCheckedModeBanner: false,
              theme: context.watch<ThemeNotifier>().currentTheme,
              home: const MainPage(),
            );
          }
          return MaterialApp(
            debugShowCheckedModeBanner: false,
            theme: context.watch<ThemeNotifier>().currentTheme,
            home: const LaunchPage(),
          );
        } else {
          return const Center(child: CircularProgressIndicator());
        }
      });
  }
}
```

constAppBar.dart

```
import 'package:fistness_app_firebase/core/extensions/theme_extension.dart';
import 'package:flutter/material.dart';


class CommonAppBar extends StatelessWidget with PreferredSizeWidget {
  const CommonAppBar({
    Key? key,
  }) : super(key: key);


  @override
  Widget build(BuildContext context) {
    return AppBar(
      backgroundColor: context.scfBackColor,
      leading: IconButton(
        icon: Icon(
          Icons.arrow_back_outlined,
          color: context.mainColor,
        ),
        onPressed: () => Navigator.of(context).pop(),
      ),
    );
  }
```

```
  @override
  Size get preferredSize => const Size.fromHeight(kToolbarHeight);
}
```

constButton.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:flutter/material.dart';

class CommonButton extends StatelessWidget {
  final String text;
  final dynamic onPressed;
  const CommonButton({
    Key? key,
    required this.text,
    required this.onPressed,
  }) : super(key: key);

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

```
    child: SizedBox(
      height: context.height * 0.06,
      width: context.width * 0.81,
      child: OutlinedButton(
        style: ButtonStyle(
          shape: MaterialStateProperty.all(RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(8.0))),
          foregroundColor:
              MaterialStateProperty.all<Color>(context.textColor),
          backgroundColor: MaterialStateProperty.all(context.mainColor),
        ),
        onPressed: onPressed,
        child: Text(text, style: context.headline6(context)),
      ),
    ),
  );
 }
}
```

constContainer.dart

```
import 'package:flutter/material.dart';
import 'const_shelf.dart';

Widget topBox(
        BuildContext context,
        double width,
        double height,
        Widget child,
        Color color1,
        Color color2,
        Color color3,
        EdgeInsets padding,
        EdgeInsets margin) =>
    Container(
      width: width,
      height: height,
      decoration: commonBoxDec(color1, color2, color3),
      padding: padding,
      margin: margin,
      child: child,
    );
```

constDecoration.dart

```
import 'package:fistness_app_firebase/views/views_shelf.dart';

BoxDecoration commonBoxDec(Color? color1, Color? color2, Color? color3) =>
    BoxDecoration(
      borderRadius: BorderRadius.circular(25.0),
      boxShadow: [
        BoxShadow(
          offset: const Offset(5, 1),
          blurRadius: 20,
          color: color1!,
        )
      ],
      gradient: LinearGradient(
          colors: [
            color2!,
            color3!,
          ],
          begin: const FractionalOffset(0.0, 0.0),
          end: const FractionalOffset(1.0, 1.0),
          stops: const [0.0, 1.0],
```

```
        tileMode: TileMode.clamp),
    );
```

constLoading.dart

```
import 'package:fistness_app_firebase/core/extensions/theme_extension.dart';
import 'package:flutter/material.dart';

class LoadingPage extends StatelessWidget {
  const LoadingPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Center(
      child: CircularProgressIndicator(
        color: context.greenColor,
      ),
    );
  }
}
```

constLogoBody.dart

```
import 'package:flutter/material.dart';


class LogoBody extends StatelessWidget {
  const LogoBody({Key? key}) : super(key: key);


  @override
  Widget build(BuildContext context) {
    double height = 200.0;
    return Container(
      height: height,
      alignment: Alignment.center,
      child: LogoPath.logo.toWidget(),
    );
  }
}


enum LogoPath { logo }


extension LogoPathExtension on LogoPath {
  String path() {
    return "assets/${LogoPath.logo.name}.png";
```

```
  }

  Widget toWidget() {
    return Image.asset(path());
  }
}
```

constText.dart

```
import 'package:fistness_app_firebase/core/extensions/theme_extension.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';

class ConstText extends StatelessWidget {
  final String text;
  const ConstText({
    Key? key,
    required this.text,
  }) : super(key: key);

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

```
    return Text(
      text,
      style: context.headline6(context),
    );
  }
}
```

constWarningToast.dart

```
import 'package:fistness_app_firebase/core/extensions/theme_extension.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';

Future<bool?> warningToast(BuildContext context, String text,
    {Color? color = Colors.red}) {
  return Fluttertoast.showToast(
      msg: text,
      timeInSecForIosWeb: 2,
      toastLength: Toast.LENGTH_SHORT,
      gravity: ToastGravity.CENTER,
      backgroundColor: color,
      textColor: context.textColor,
```

```
    fontSize: 14);
}
```

constEdgeInsests.dart

```
import 'package:flutter/material.dart';

extension AllPaddings on BuildContext {
  EdgeInsets get zeroAllPadding => const EdgeInsets.all(0);
  EdgeInsets get minAllPadding => const EdgeInsets.all(8.0);
  EdgeInsets get minMidAllPadding => const EdgeInsets.all(10.0);

  EdgeInsets get midAllPadding => const EdgeInsets.all(16.0);
  EdgeInsets get largeAllPadding => const EdgeInsets.all(20.0);
}

extension Size on BuildContext {
  double get height => MediaQuery.of(this).size.height;
  double get width => MediaQuery.of(this).size.width;
}

extension SymetricPadding on BuildContext {
```

```
  EdgeInsets get minHorzPadding => const EdgeInsets.symmetric(horizontal: 8.0);
  EdgeInsets get minVertPadding => const EdgeInsets.symmetric(vertical: 8.0);

  EdgeInsets get midVerticalPadding =>
      const EdgeInsets.symmetric(vertical: 16.0);
  EdgeInsets get midLargeVerticalPadding =>
      const EdgeInsets.symmetric(vertical: 18.0);

  EdgeInsets get largeHorizontalPadding =>
      const EdgeInsets.symmetric(horizontal: 45.0);

  EdgeInsets get symVertPadding =>
      const EdgeInsets.symmetric(horizontal: 12, vertical: 4);
  EdgeInsets get symVertHorzPadding =>
      const EdgeInsets.symmetric(horizontal: 12, vertical: 12);
  EdgeInsets get minsymVertHorzPadding =>
      const EdgeInsets.symmetric(horizontal: 4, vertical: 4);
}

extension OnlyPadding on BuildContext {
  EdgeInsets get minLeft => const EdgeInsets.only(left: 8.0);
  EdgeInsets get midLeft => const EdgeInsets.only(left: 12.0);
  EdgeInsets get minTopBtm =>
```

```
    const EdgeInsets.only(top: 20.0, right: 8, left: 8);

  EdgeInsets get midTop => const EdgeInsets.only(top: 70);
  EdgeInsets get riBottom => const EdgeInsets.only(right: 10.0, bottom: 5);
  EdgeInsets get minTopBottom => const EdgeInsets.only(top: 32.0, bottom: 5);
  EdgeInsets get midTopBottom => const EdgeInsets.only(top: 120, bottom: 10);
}

extension LtrbPadding on BuildContext {
  EdgeInsets get minLtrb => const EdgeInsets.fromLTRB(8.0, 10.0, 8.0, 5.0);
  EdgeInsets get minMidLtrb => const EdgeInsets.fromLTRB(16.0, 0.0, 16.0, 5.0);
  EdgeInsets get minLargeLtrb =>
      const EdgeInsets.fromLTRB(16.0, 10.0, 16.0, 5.0);

  EdgeInsets get midLtrb => const EdgeInsets.fromLTRB(20.0, 5.0, 20.0, 20.0);
  EdgeInsets get midLargeLtrb =>
      const EdgeInsets.fromLTRB(25.0, 10.0, 25.0, 10.0);
  EdgeInsets get largeLtrb => const EdgeInsets.fromLTRB(16.0, 16.0, 20.0, 16.0);
  EdgeInsets get xLargeLtrb => const EdgeInsets.fromLTRB(16.0, 50.0, 16.0, 5.0);
}
```

constTheme.dart

```
import 'package:fistness_app_firebase/views/views_shelf.dart';


extension ThemeExtension on BuildContext {
  ThemeData get theme => Theme.of(this);
  Color get scfBackColor => theme.scaffoldBackgroundColor;
  Color get mainColor => theme.errorColor;
  Color get textColor => theme.hintColor;
  // Color get errColor => const Color.fromRGBO(183, 28, 28, 1);
  Color get scndTxtColor => theme.primaryColor;
  Color get shadeGreyColor => theme.shadowColor;
  Color get greenColor => theme.canvasColor;


  TextStyle? subtitle1(context) => theme.textTheme.subtitle1;


  TextStyle? subtitle2(context) => theme.textTheme.subtitle2;


  TextStyle? headline4(context) => theme.textTheme.headline4;


  TextStyle? headline6(context) => theme.textTheme.headline6;


  TextStyle? bdSmall(context) => theme.textTheme.bodySmall;
```

```
  // TextStyle? headline2(context) => theme.textTheme.headline2;
}
```

AuthService.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:fistness_app_firebase/views/home/bottomNavigateBar/navigare_bar.dart';
import 'package:google_sign_in/google_sign_in.dart';

import '../../src/texts.dart';
import '../../views/home/view/home_page.dart';
import '../../views/views_shelf.dart';

class AuthService {
  final FirebaseAuth auth = FirebaseAuth.instance;
  FirebaseFirestore firestore = FirebaseFirestore.instance;
  String collectionName = "users";
  bool isLoading = false;

  Future signOut() async {
    final _google = GoogleSignIn();
```

```
  await _google.disconnect();
  await _google.signOut();
  await auth.signOut();
}

Future<void> SignOut() async {
  final GoogleSignIn googleSignIn = GoogleSignIn();
  FirebaseAuth.instance.currentUser == null;
  if (googleSignIn.currentUser != null) {
    await googleSignIn.disconnect();
    await FirebaseAuth.instance.signOut();
  } else {
    await FirebaseAuth.instance.signOut();
  }
}

Future<DocumentSnapshot<Map<String, dynamic>>> fetchCurrentUserDoc() async {
  return await FirebaseFirestore.instance
      .collection('users')
      .doc(auth.currentUser!.uid)
      .get();
}
```

```
Future<bool?> createPerson(
    String username,
    String email,
    String password,
    String uid,
    String name,
    String gender,
    int age,
    String mobility,
    int height,
    int weight,
    int userRightPoint) async {
  var user = await auth.createUserWithEmailAndPassword(
      email: email, password: password);
  final userInfo = <String, String>{
    "username": username,
    "email": email,
    "name": name,
    "gender": gender,
    "age": age.toString(),
    "mobility": mobility,
    "height": height.toString(),
    "weight": weight.toString(),
```

```
    "userRightPoint": userRightPoint.toString()
  };
  await firestore
      .collection("users")
      .doc(auth.currentUser!.uid)
      .set(userInfo);
  return true;
}

Future<bool?> createPersonEmail(
    String username,
    String email,
    String password,
    String uid,
    String name,
    String gender,
    String age,
    String height,
    String weight) async {
  await firestore.collection(collectionName).doc(uid).set({
    "username": username,
    "email": email,
    "name": name,
```

```
    "gender": gender,
    "age": age,
    "height": height,
    "weight": weight,
  }).catchError((error) => print("Failed to set data: $error"));
  return true;
}

Future sendEmailVerfied() async {
  final user = FirebaseAuth.instance.currentUser!;
  await user.sendEmailVerification();
}

void checkUid() {
  final User? userr = FirebaseAuth.instance.currentUser;
  final uid = userr?.uid;
  return null;
}

Future<void> signInWithGoogle(context) async {
  final _google = GoogleSignIn();

  final googleAccount = await _google.signIn();
```

```
    if (googleAccount != null) {
      final googleAuth = await googleAccount.authentication;
      if (googleAuth.accessToken != null && googleAuth.idToken != null) {
        try {
          await auth.signInWithCredential(GoogleAuthProvider.credential(
              idToken: googleAuth.idToken,
              accessToken: googleAuth.accessToken));
          Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => const MainPage(),
              ));
        } on FirebaseException catch (e) {
          print(e);
        }
      }
    }
//     // Obtain the auth details from the request
  }

  Future<bool> signInWithEmailandPassword(String email, String password) async {
    try {
      final credential = await FirebaseAuth.instance.signInWithEmailAndPassword(
```

```
        email: email,
        password: password,
      );
      if (credential.user!.emailVerified) {
        return true;
      } else {
        return false;
      }
    } on FirebaseAuthException catch (e) {
      if (e.code == 'weak-password') {
        print('The password provided is too weak.');
      } else if (e.code == 'email-already-in-use') {
        print('The account already exists for that email.');
      }
      return false;
    } catch (e) {
      print(e);
      return false;
    }
  }
}
```

ThemeControl.dart

```
import 'package:fistness_app_firebase/views/views_shelf.dart';
import 'package:google_fonts/google_fonts.dart';

class ThemeNotifier extends ChangeNotifier {
  bool isLight = false;

  void changeTheme() {
    isLight = !isLight;
    notifyListeners();
  }

  ThemeData get currentTheme => isLight
      ? ThemeData.light().copyWith(
          scaffoldBackgroundColor: const Color(
            (0xFF19282F),
          ),
          //listTile theme
          listTileTheme: const ListTileThemeData(
            iconColor: Color.fromARGB(255, 96, 25, 211),
            textColor: Color.fromARGB(255, 16, 16, 16),
          ),
```

```
  //textThee=me
  textTheme: TextTheme(
    subtitle1: GoogleFonts.poppins(
      color: const Color.fromARGB(255, 3, 2, 2),
      fontWeight: FontWeight.w400,
      fontStyle: FontStyle.normal,
    ),
    headline4: GoogleFonts.poppins(
      color: const Color.fromARGB(255, 15, 11, 11),
    ),
    headline6: GoogleFonts.poppins(
        color: const Color.fromARGB(255, 19, 18, 18)),
    subtitle2: GoogleFonts.poppins(
      color: const Color(0xffFFFFFF),
      fontWeight: FontWeight.w500,
      fontStyle: FontStyle.normal,
    ),
  ),
  //iconTheme
  iconTheme: const IconThemeData(
    color: Color.fromARGB(255, 96, 25, 211),
  ),
)
```

```
: ThemeData(
    //colors
    canvasColor: Colors.green,
    errorColor: const Color.fromRGBO(183, 28, 28, 1),
    primaryColor: Colors.grey,
    shadowColor: Colors.grey.shade200,
    hintColor: Colors.white,
    scaffoldBackgroundColor: const Color(
      (0xFF19282F),
    ),
    //appbar theme
    appBarTheme: const AppBarTheme(
        color: Colors.transparent,
        elevation: 0,
        iconTheme: IconThemeData(color: Color(0xFFC4FB6D))),
    //listTile theme
    listTileTheme: const ListTileThemeData(
      iconColor: Color(0xFFC4FB6D),
      textColor: Colors.white,
    ),
    //inputDecorationTheme
    inputDecorationTheme: InputDecorationTheme(
      enabledBorder: OutlineInputBorder(
```

```
    borderRadius: BorderRadius.circular(10),
    borderSide: BorderSide(color: Colors.grey.shade200),
  ),
  focusedBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(10),
      borderSide: BorderSide(color: Colors.red.shade900)),
),
//iconTheme
iconTheme: const IconThemeData(color: Color(0xFFC4FB6D)),
bottomNavigationBarTheme: const BottomNavigationBarThemeData(
    backgroundColor: Colors.black,
    selectedIconTheme: IconThemeData(color: Color(0xFFC4FB6D)),
    unselectedIconTheme: IconThemeData(color: Colors.red)),

//textTheme
textTheme: TextTheme(
  subtitle1: GoogleFonts.poppins(
    color: const Color(0xffFFFFFF),
    fontWeight: FontWeight.w400,
    fontStyle: FontStyle.normal,
  ),
  headline4: GoogleFonts.poppins(
    color: const Color(0xffFFFFFF),
```

```
        ),
        headline6: GoogleFonts.poppins(color: const Color(0xffFFFFFF)),
        subtitle2: GoogleFonts.poppins(
          color: const Color(0xffFFFFFF),
          fontWeight: FontWeight.w500,
          fontStyle: FontStyle.normal,
        ),
      ),
    );
}
```

StaticTexts.dart

```
import 'package:fistness_app_firebase/core/service/auth_service.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';

class MyText {
  static final AuthService authService = AuthService();
  static String usernameText = "User name";
  static String bnFirstText = "Home";
  static String bnSecondText = "Search";
  static String bnThirdText = "Profile";
  static String homeWelcomeText = "Good Morning ";
```

```
  static String continueText = "continue";
  static String nextText = "Next";
  static String exerciseText = "Exercises";
  static String fitText = "Fitness Is The Key";
  static dynamic currentUser;
  MyText._();
}

class RegisterText {
  static String registerSuccesfully =
      "Your registration has been completed successfully.";
  static String signEmailText = 'Sign in with Email ';
  static String googleText = "Google";
  static String emailText = "Email";
  static String verifyEmailText = "Verify Email";
  static String faceText = "Facebook";
  static String emailAssociated =
      "Enter the Email address associated with your account";
  static String womanText = "Female";
  static String manText = "Male";
  static String cmText = "cm";
  static String kgText = "kg";
  static String orLogText = "Or log in with:";
```

```
  static String orSignText = "Or sign-up with:";
  static String createText = "Create Account";
  static String passwordText = "Password";
}

class QuestionsText {
  static String sexText = "What is your sex?";
  static String ageText = "What is your birth of date?";
  static String heightText = "How much is your height?";
  static String accountText = "Don't you have an account?";
  static String forgotPassText = "Forgot password?";
  static String nameText = "Your Name?";
  static String weightText = "How much is your weight?";
}

class WarningText {
  static String registerEmptyUsername = "Username can't be empty";
  static String registerEmptyEmail = "Mail can't be emoty";
  static String registerEmptyPassword = "Password can't be empty";
  static String sexWarningText = "You should choice your sex!";
  static String nameWarningText = "You should enter your name!";
  static String loginWrongEmailText = "Invalid mail!";
  static String loginNoAccountText = "User not found!";
```

```
  static String loginWrongPasswordText = "Wrong password!";
  static String errorText = "An unexpected error has occurred...";
  static String registerInvalidPassword =
      "Password must be at least 6 character";
  static String registerUniqueMail =
      "Mail is already in use! Please register with difference mail ";
}

class AllColors {
  static Color gradColor1 = const Color(0xFF19282F);
  static Color gradColor2 = const Color(0xFF3d444e);
  static Color gradColor3 = const Color(0xFF2c2f37);
  static Color gradColor4 = const Color(0xFF000000);
  static Color gradColor5 = const Color(0xFF2a2d32);
  static Color gradColor6 = const Color(0xFF4b4f5b);
}
```

LaunchPageView.dart

```
import 'package:fistness_app_firebase/core/const/const_shelf.dart';
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
```

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

  @override
  _LaunchPageState createState() => _LaunchPageState();
}

class _LaunchPageState extends State<LaunchPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Container(
          margin: context.midTopBottom,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const LogoBody(),
              const SizedBox(
```

```
          height: 15,
        ),
        _headText(context),
        _signEmailButton(context, context.height, context.width),
        const SizedBox(
          height: 25,
        ),
        _orText(context),
        const SizedBox(
          height: 10.0,
        ),
        const LaunchPageButtons(),
        const SizedBox(
          height: 20.0,
        ),
        Padding(
          padding: context.largeHorizontalPadding,
          child: _bottomText(context),
        )
      ],
    ),
  ),
),
```

```
  );
}

Row _bottomText(BuildContext context) {
  return Row(
    crossAxisAlignment: CrossAxisAlignment.center,
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      Expanded(child: Container()),
      Text(
        QuestionsText.accountText,
        style: context.subtitle2(context),
      ),
      const SizedBox(
        width: 15,
      ),
      InkWell(
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => const RegisterPage()),
          );
        },
```

```
        child: Text(
          RegisterText.createText,
          style: context.bdSmall(context)?.copyWith(color: context.mainColor),
        ),
      ),
    ],
  );
}

Padding _orText(BuildContext context) {
  return Padding(
    padding: context.largeHorizontalPadding,
    child: Text(
      RegisterText.orLogText,
      textAlign: TextAlign.left,
      style: context.bdSmall(context)?.copyWith(color: context.textColor),
    ),
  );
}

Column _signEmailButton(
    BuildContext context, double mediaQueryHeight, double mediaQueryWidth) {
  return Column(
```

```
children: [
  Center(
    child: Container(
      margin: context.midTop,
      height: mediaQueryHeight * 0.06,
      width: mediaQueryWidth * 0.8,
      child: TextButton.icon(
        style: TextButton.styleFrom(
          backgroundColor: context.mainColor,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(8.0),
          ),
        ),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => const LoginPage()),
          );
        },
        icon: Icon(
          Icons.email_outlined,
          color: context.textColor,
        ),
```

```
            label: Align(
              alignment: Alignment.center,
              child: Text(
                RegisterText.signEmailText,
                style: context.subtitle1(context),
              ),
            ),
          ),
        ),
      ),
    ],
  );
 }

 Center _headText(BuildContext context) {
  return Center(
    child: Text(
      MyText.fitText,
      style: context.headline4(context),
    ),
  );
 }
}
```

LaunchPageButton.dart

```
// ignore_for_file: avoid_print, use_build_context_synchronously, dead_code

import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';

class LaunchPageButtons extends StatefulWidget {
  const LaunchPageButtons({
    Key? key,
  }) : super(key: key);

  @override
  _LaunchPageButtonsState createState() => _LaunchPageButtonsState();
}

class _LaunchPageButtonsState extends State<LaunchPageButtons> {
  bool isLoading = false;

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

```
return !isLoading
    ? Center(
        child: SizedBox(
          height: context.height * 0.06,
          width: context.width * 0.82,
          child: SizedBox(
            height: context.height * 0.06,
            width: context.width * 0.4,
            child: OutlinedButton.icon(
              icon: ImagePaths.google.googletoWidget(),
              onPressed: () async {
                await MyText.authService.signInWithGoogle(context);


              },
              label: Text(
                RegisterText.googleText,
                style: context.subtitle1(context)?.copyWith(
                    color: context.scndTxtColor,
                    fontWeight: FontWeight.bold),
              ),
              style: ButtonStyle(
                backgroundColor: MaterialStateProperty.all<Color>(
                    context.shadeGreyColor),
```

```
                    side: MaterialStateProperty.all<BorderSide>(
                        BorderSide.none)),
                ),
              ),
            ),
          )
        : const CircularProgressIndicator();
  }


}

enum ImagePaths { google, facebook }

extension ImageExtension on ImagePaths {
  String googlePath() {
    return "assets/${ImagePaths.google.name}.png";
  }

  String facePath() {
    return "assets/${ImagePaths.facebook.name}.png";
  }
```

```
  Widget facetoWidget() {
    return Image.asset(
      facePath(),
      height: 23,
    ); //
  }

  Widget googletoWidget() {
    return Image.asset(
      googlePath(),
      height: 23,
    ); //
  }
}
```

LoginPage.dart

```
// ignore_for_file: use_build_context_synchronously

import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/home/bottomNavigateBar/navigare_bar.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
```

```
import '../../core/const/const_shelf.dart';


class LoginPage extends StatefulWidget {
  const LoginPage({
    Key? key,
  }) : super(key: key);


  @override
  _LoginPageState createState() => _LoginPageState();
}


class _LoginPageState extends State<LoginPage> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();


  bool _isVisible = true;
  bool isLoading = false;


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: const CommonAppBar(),
      body: _body(context),
```

```
  );
}

Stack _body(BuildContext context) {
  return Stack(
    children: [
      Padding(
        padding: context.minLargeLtrb,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            _bodyContainer(),
            const SizedBox(
              height: 10,
            ),
          ],
        ),
      ),
      if (isLoading) const LoadingPage()
    ],
  );
}
```

```
Flexible _bodyContainer() {
  return Flexible(
      child: Container(
    decoration: const BoxDecoration(
        borderRadius: BorderRadius.all(Radius.circular(16.0))),
    child: Padding(
      padding: context.largeAllPadding,
      child: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.end,
          children: [
            const LogoBody(),
            const SizedBox(
              height: 50,
            ),
            _emailTextfield(),
            const SizedBox(
              height: 20,
            ),
            _passwordTextfield(),
            const SizedBox(
              height: 5,
```

```
),
_forgotPassword(),
const SizedBox(
  height: 40,
),
CommonButton(
    text: MyText.continueText,
    onPressed: () async {
      bool? isSucces = await MyText.authService
          .signInWithEmailandPassword(
              _emailController.text.trim().toString(),
              _passwordController.text.trim());

      if (isSucces) {
        MyText.authService.checkUid();
        Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => const MainPage(),
            ));
      } else {
        warningToast(
            context, "Wrong Pass/Email! or verify your email!");
```

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

TextField _emailTextfield() {
  return TextField(
      style: context.subtitle1(context),
      controller: _emailController,
      cursorColor: Colors.black,
      keyboardType: TextInputType.emailAddress,
      decoration: InputDecoration(
        prefixIcon: Icon(
          Icons.mail,
          color: context.mainColor,
        ),
        hintText: RegisterText.emailText,
      ));
}
```

```
TextField _passwordTextfield() {
  return TextField(
    style: context.subtitle1(context),
    controller: _passwordController,
    obscureText: _isVisible ? true : false,
    cursorColor: context.textColor,
    decoration: InputDecoration(
      suffixIcon: InkWell(
        onTap: () {
          if (_isVisible) {
            setState(() {
              _isVisible = false;
            });
          } else {
            setState(() {
              _isVisible = false;
            });
            _isVisible = true;
          }
        },
        child: _isVisible
            ? Icon(
```

```
              Icons.remove_red_eye,
              color: context.mainColor,
            )
          : Icon(
              Icons.remove_red_eye_outlined,
              color: context.mainColor,
            ),
      ),
      prefixIcon: Icon(
        Icons.vpn_key,
        color: context.mainColor,
      ),
      hintText: RegisterText.passwordText,
      hintStyle: context.subtitle1(context),
    ),
  );
}

// Future _logInWithEmail() async {
//   if (_emailController.text.isNotEmpty &&
//       _passwordController.text.isNotEmpty) {
//     setState(() {
//       isLoading = true;
```

```
//     });

//     try {
//       MyText.currentUser = await MyText.authService.auth
//           .signInWithEmailAndPassword(
//               email: _emailController.text.trim(),
//               password: _passwordController.text.trim());

//       if (MyText.currentUser != null) {
//         setState(() {
//           isLoading = false;
//         });

//         Navigator.pushAndRemoveUntil(
//             context,
//             MaterialPageRoute(builder: (context) => const HomePage()),
//             (route) => false);
//       } else {
//         setState(() {
//           isLoading = false;
//         });
//       }
//     } catch (error) {
```

```
//       setState(() {
//         isLoading = false;
//       });

//       if (error.toString().contains('invalid-email')) {
//         await warningToast(context, WarningText.loginWrongEmailText);
//       } else if (error.toString().contains('user-not-found')) {
//         await warningToast(context, WarningText.loginNoAccountText);
//       } else if (error.toString().contains('wrong-password')) {
//         await warningToast(context, WarningText.loginWrongPasswordText);
//       } else {
//         await warningToast(context, WarningText.errorText);
//       }
//     }
//   } else {
//     setState(() {
//       isLoading = false;
//     });

//     warningToast(context, WarningText.errorText);
//   }
// }
```

```
  _forgotPassword() {
    return InkWell(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => const ForgotPasswordPage()),
        );
      },
      child: Text(
        QuestionsText.forgotPassText,
        style: context.subtitle2(context),
      ),
    );
  }
}
```

LoginToken.dart

```
class TokenModel {
  String? token;

  TokenModel({this.token});
```

```
  TokenModel.fromJson(Map<String, dynamic> json) {
    token = json['token'];
  }

  Map<String, dynamic> toJson() {
    final Map<String, dynamic> data = <String, dynamic>{};
    data['token'] = token;
    return data;
  }
}
```

ForgotPasswordPage.dart

```
import 'package:fistness_app_firebase/core/const/warning_toast.dart';
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/service/foods_exercises_service.dart';
import 'package:fistness_app_firebase/views/service/project_network.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import '../../core/const/const_logo_body.dart';

class ForgotPasswordPage extends StatefulWidget {
```

```
  const ForgotPasswordPage({Key? key}) : super(key: key);

  @override
  _ForgotPasswordPageState createState() => _ForgotPasswordPageState();
}

class _ForgotPasswordPageState extends State<ForgotPasswordPage> {
  final TextEditingController _emailController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: IconButton(
          icon: const Icon(
            Icons.arrow_back_ios,
          ),
          onPressed: () => Navigator.of(context).pop(),
        ),
      ),
      body: _body(context),
    );
  }
```

```
Stack _body(BuildContext context) {
  return Stack(
    children: [
      Padding(
        padding: context.minLargeLtrb,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            _bodyContainer(),
            const SizedBox(
              height: 10,
            ),
          ],
        ),
      )
    ],
  );
}

Flexible _bodyContainer() {
  return Flexible(
```

```
  child: Container(
decoration: const BoxDecoration(
    borderRadius: BorderRadius.all(Radius.circular(16.0))),
child: Padding(
  padding: context.largeAllPadding,
  child: SingleChildScrollView(
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.end,
      children: [
        const LogoBody(),
        const SizedBox(
          height: 40,
        ),
        _forgotText(),
        const SizedBox(
          height: 30,
        ),
        _emailTextfield(),
        const SizedBox(
          height: 20,
        ),
        const SizedBox(
          height: 40,
```

```
            ),
            myButton(),
          ],
        ),
      ),
    ),
  ));
}

TextField _emailTextfield() {
  return TextField(
    style: context.subtitle1(context),
    controller: _emailController,
    cursorColor: context.textColor,
    keyboardType: TextInputType.emailAddress,
    decoration: InputDecoration(
      prefixIcon: Icon(Icons.mail, color: context.mainColor),
      hintText: RegisterText.emailText,
      hintStyle: context.subtitle1(context),
    ),
  );
}
```

```
myButton() {
  return Center(
    child: SizedBox(
      height: context.height * 0.06,
      width: context.width * 0.81,
      child: OutlinedButton(
        style: ButtonStyle(
          shape: MaterialStateProperty.all(RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(8.0))),
          foregroundColor:
              MaterialStateProperty.all<Color>(context.textColor),
          backgroundColor: MaterialStateProperty.all(context.mainColor),
        ),
        onPressed: () async {
          final response = await GeneralService(
                ProjectNetworkManager.instance.service, "reset password")
            .resetPasswordLink(_emailController.text);
          if (response!) {
            await warningToast(context, "Reset password email has been sent",
                color: context.greenColor);
          } else {
            await warningToast(
              context,
```

```
            "Reset password email couldnt sent please try again ",
          );
        }
      },
      child: Text(RegisterText.verifyEmailText,
          style: context.headline6(context)),
    ),
  ),
 );
}

_forgotText() {
 return Column(
  children: [
   Center(
    child: Text(
     QuestionsText.forgotPassText,
     style: context.headline4(context)?.copyWith(
         fontWeight: FontWeight.bold, color: context.scndTxtColor),
    ),
   ),
   const SizedBox(
    height: 10,
```

```
        ),
        Text(
          RegisterText.emailAssociated,
          textAlign: TextAlign.center,
          style: context.subtitle2(context),
        )
      ],
    );
  }
}
```

RegisterPage.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import '../../core/const/const_shelf.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import '../../core/service/auth_service.dart';

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

  @override
```

```
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  final TextEditingController _usernameController = TextEditingController();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();

  bool _isVisible = true;
  bool isLoading = false;
  final AuthService authService = AuthService();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: const CommonAppBar(),
      body: _body(context),
    );
  }

  Stack _body(BuildContext context) {
    return Stack(
      children: [
```

```
      Padding(
        padding: context.minMidLtrb,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            _bodyContainer(),
            const SizedBox(
              height: 10,
            ),
          ],
        ),
      )
    ],
  );
}

Flexible _bodyContainer() {
  return Flexible(
      child: Container(
    decoration: const BoxDecoration(
        borderRadius: BorderRadius.all(Radius.circular(16.0))),
    child: Padding(
```

```
padding: context.midLtrb,
child: SingleChildScrollView(
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      const LogoBody(),
      const SizedBox(
        height: 40,
      ),
      _usernameTextfield(),
      const SizedBox(
        height: 20,
      ),
      _emailTextfield(),
      const SizedBox(
        height: 20,
      ),
      _passwordTextfield(),
      const SizedBox(
        height: 5,
      ),
      _orText(),
```

```
            const SizedBox(
              height: 30,
            ),
            CommonButton(
                text: MyText.continueText, onPressed: _registerOnTap),
          ],
        ),
      ),
    ),
  ));
}

TextField _usernameTextfield() {
  return TextField(
    style: context.subtitle1(context),
    textInputAction: TextInputAction.next,
    controller: _usernameController,
    cursorColor: context.textColor,
    decoration: InputDecoration(
      prefixIcon: Icon(
        Icons.account_circle,
        color: context.mainColor,
      ),
```

```
      hintText: MyText.usernameText,
    ),
  );
}

TextField _emailTextfield() {
  return TextField(
    textInputAction: TextInputAction.next,
    style: context.subtitle1(context),
    controller: _emailController,
    cursorColor: context.textColor,
    keyboardType: TextInputType.emailAddress,
    decoration: InputDecoration(
      prefixIcon: Icon(
        Icons.mail,
        color: context.mainColor,
      ),
      hintText: RegisterText.emailText,
    ),
  );
}

TextField _passwordTextfield() {
```

```
return TextField(
  style: context.subtitle1(context),
  controller: _passwordController,
  obscureText: _isVisible ? true : false,
  cursorColor: context.textColor,
  decoration: InputDecoration(
    suffixIcon: InkWell(
      onTap: () {
        if (_isVisible) {
          setState(() {
            _isVisible = false;
          });
        } else {
          setState(() {
            _isVisible = false;
          });
          _isVisible = true;
        }
      },
      child: _isVisible
          ? Icon(
              Icons.remove_red_eye,
              color: context.mainColor,
```

```
            )
          : Icon(
              Icons.remove_red_eye_outlined,
              color: context.mainColor,
            ),
      ),
      prefixIcon: Icon(
        Icons.vpn_key,
        color: context.mainColor,
      ),
      hintText: RegisterText.passwordText,
    ),
  );
}

_orText() {
  return Padding(
    padding: context.minLeft,
    child: Text(
      RegisterText.orSignText,
      style: context.bdSmall(context)?.copyWith(color: context.textColor),
    ),
  );
```

```
}

void _registerOnTap() {
  if (_usernameController.text.isNotEmpty &&
      _emailController.text.isNotEmpty &&
      _emailController.text.contains("@") &&
      _emailController.text.contains(".com") &&
      _passwordController.text.isNotEmpty &&
      _passwordController.text.length >= 6) {
    setState(() {
      isLoading = true;
    });

    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => RegisterNamePage(
                username: _usernameController.text,
                mail: _emailController.text,
                password: _passwordController.text,
              )));
  } else if (_usernameController.text.isEmpty) {
    warningToast(context, WarningText.registerEmptyUsername);
```

```
    } else if (_emailController.text.toString().isEmpty) {
      warningToast(context, WarningText.registerEmptyEmail);
    } else if (_passwordController.text.toString().isEmpty) {
      warningToast(context, WarningText.registerInvalidPassword);
    } else if (_passwordController.text.length < 6) {
      warningToast(context, WarningText.registerInvalidPassword);
    } else {
      warningToast(context, WarningText.loginWrongEmailText);
    }
  }
}
```

RegisterNamePage.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';

import '../../core/const/const_shelf.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';

class RegisterNamePage extends StatefulWidget {
  final String? username;
  final String? mail;
```

```
  final String? password;
  final String? uid;

  const RegisterNamePage(
      {Key? key, this.username, this.mail, this.password, this.uid})
      : super(key: key);

  @override
  _RegisterNamePageState createState() => _RegisterNamePageState();
}

class _RegisterNamePageState extends State<RegisterNamePage> {
  final TextEditingController _nameController = TextEditingController();
  bool isLoading = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: const CommonAppBar(),
      body: Padding(
        padding: context.xLargeLtrb,
        child: SingleChildScrollView(
          child: Column(
            children: [
```

```
            myText(),
            SizedBox(
              height: context.height * 0.01,
            ),
            _nameField(),
            SizedBox(
              height: context.height * 0.3,
            ),
            CommonButton(text: MyText.nextText, onPressed: _registerOnTap)
          ],
        ),
      ),
    ),
  );
}

myText() {
  return SizedBox(
      height: context.height * 0.09,
      width: context.width * 0.87,
      child: ConstText(
        text: QuestionsText.nameText,
      ));
```

```
}

_nameField() {
  return TextField(
      style: context.subtitle1(context),
      controller: _nameController,
      cursorColor: context.textColor,
      keyboardType: TextInputType.emailAddress,
      decoration: InputDecoration(
        hintText: QuestionsText.nameText,
        hintStyle: context.headline6(context),
      ));
}

void _registerOnTap() {
  if (_nameController.text.isNotEmpty) {
    setState(() {
      isLoading = true;
    });

    Navigator.push(
        context,
        MaterialPageRoute(
```

```
            builder: (context) => GenderPage(
                username: widget.username,
                mail: widget.mail,
                uid: widget.uid.toString(),
                password: widget.password,
                name: _nameController.text,
              )));
    } else {
      warningToast(context, WarningText.nameWarningText);
    }
  }
}
```

GenderPage.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import '../../core/const/const_shelf.dart';
import '../../core/service/auth_service.dart';

class GenderPage extends StatefulWidget {
  final String? username;
```

```
  final String? mail;
  final String? name;
  final String? password;
  final String uid;
  const GenderPage(
      {Key? key,
      this.username,
      this.mail,
      this.name,
      this.password,
      required this.uid})
      : super(key: key);

  @override
  _GenderPageState createState() => _GenderPageState();
}

class _GenderPageState extends State<GenderPage> {
  late List<bool> isSelected;
  bool isLoading = false;
  final AuthService authService = AuthService();
  var choice = "";
```

```
@override
void initState() {
  super.initState();
  isSelected = [true, false];
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: const CommonAppBar(),
    body: Padding(
      padding: context.minLtrb,
      child: SingleChildScrollView(
        child: Center(
          child: Column(
            children: [
              const LogoBody(),
              SizedBox(
                height: context.height / 17,
              ),
              ConstText(
                text: QuestionsText.sexText,
              ),
```

```
            SizedBox(
              height: context.height / 15,
            ),
            _toggleButton(),
            SizedBox(height: context.height * 0.05),
            CommonButton(
              text: MyText.continueText,
              onPressed: _registerOnTap,
            ),
          ],
        ),
      ),
    ),
  ),
 );
}


_toggleButton() {
  return ToggleButtons(
    borderRadius: BorderRadius.circular(16.0),
    isSelected: isSelected,
    onPressed: (index) {
      setState(() {
```

```
    for (var i = 0; i < isSelected.length; i++) {
      if (i == index) {
        isSelected[i] = true;
        choiceControl();
      } else {
        isSelected[i] = false;
      }
    }
  });
},
color: context.scndTxtColor,
children: [
  Column(
    children: [
      Icon(Icons.male,
          size: context.height / 6,
          color: isSelected[0] == false
              ? context.textColor
              : context.mainColor),
      Padding(
        padding: context.midVerticalPadding,
        child: Text(
          RegisterText.womanText,
```

```
        style: context.subtitle1(context)?.copyWith(
            color: isSelected[0] == false
                ? context.textColor
                : context.mainColor),
      ),
    )
  ],
),
Column(
  children: [
    Icon(Icons.female,
        size: context.height / 6,
        color: isSelected[0] == true
            ? context.textColor
            : Colors.blue.shade900),
    Padding(
      padding: context.midVerticalPadding,
      child: Text(
        RegisterText.manText,
        style: context.subtitle1(context)?.copyWith(
            color: isSelected[0] == true
                ? context.textColor
                : Colors.blue),
```

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

void _registerOnTap() {
  if (isSelected.isNotEmpty) {
    setState(() {
      isLoading = true;
    });

    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => AgePage(
                  mail: widget.mail,
                  username: widget.username,
                  password: widget.password,
                  uid: widget.uid,
                  name: widget.name,
```

```
                gender: choiceControl().toString(),
              )));
    } else {
      warningToast(context, WarningText.sexWarningText);
    }
  }

  choiceControl() {
    if (isSelected[0] == true) {
      choice = RegisterText.womanText;
    } else {
      choice = RegisterText.manText;
    }
    return choice;
  }
}
```

AgePage.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/exerciseMobility/exercise_mobility.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
```

```
import '../../core/const/const_appbar.dart';
import '../../core/const/const_logo_body.dart';


class AgePage extends StatefulWidget {
  final String? username;
  final String? mail;
  final String? password;
  final String uid;
  final String? name;
  final String? gender;
  const AgePage(
      {Key? key,
      this.username,
      this.mail,
      this.password,
      required this.uid,
      this.name,
      this.gender})
      : super(key: key);

  @override
  _AgePageState createState() => _AgePageState();
}
```

```
class _AgePageState extends State<AgePage> {
  int _currentValue = 2004;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: const CommonAppBar(),
      body: Padding(
        padding: context.minLtrb,
        child: SingleChildScrollView(
          child: Column(
            children: [
              const LogoBody(),
              SizedBox(
                height: context.height / 17,
              ),
              Text(
                QuestionsText.ageText,
                style: context.headline6(context),
              ),
              SizedBox(
                height: context.height / 15,
```

```
            ),
            _pickerBody(),
            SizedBox(height: context.height * 0.1),
            myButton(),
          ],
        ),
      ),
    ),
  );
}

myButton() {
  return Center(
    child: SizedBox(
      height: context.height * 0.06,
      width: context.width * 0.81,
      child: OutlinedButton(
        style: ButtonStyle(
          shape: MaterialStateProperty.all(RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(8.0))),
          foregroundColor:
              MaterialStateProperty.all<Color>(context.textColor),
          backgroundColor: MaterialStateProperty.all(context.mainColor),
```

```
        ),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) => DailyMobilityView(
                      username: widget.username,
                      mail: widget.mail,
                      password: widget.password,
                      uid: widget.uid,
                      name: widget.name,
                      gender: widget.gender,
                      age: 2023 - _currentValue,
                    )),
          );
        },
        child: Text(MyText.continueText, style: context.headline6(context)),
      ),
    ),
  );
}

_pickerBody() {
```

```
return NumberPicker(
  selectedTextStyle:
      context.headline6(context)?.copyWith(color: context.mainColor),
  textStyle: context.subtitle2(context),
  decoration: const BoxDecoration(
    border: Border(
      top: BorderSide(
        //                <--- left side
        color: Color(0xFFC4FB6D),
        width: 3.0,
      ),
      bottom: BorderSide(
        //                 <--- top side
        color: Color(0xFFC4FB6D),
        width: 3.0,
      ),
    ),
  ),
  value: _currentValue,
  minValue: 1950,
  maxValue: 2004,
  onChanged: (value) => setState(() => _currentValue = value),
);
```

```
  }
}
```

ExercisesMobilityPage.dart

```
import 'package:fistness_app_firebase/core/const/const_shelf.dart';
import 'package:fistness_app_firebase/core/extensions/edge_insets.dart';
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/src/texts.dart';
import 'package:fistness_app_firebase/views/height/height_page.dart';
import 'package:flutter/material.dart';

List<String> list = <String>[
  'Desk job or sedentary',
  'Both at the table and standing',
  'Those on the move during the day',
  'Active employees'
];

class DailyMobilityView extends StatefulWidget {
  final String? username;
  final String? mail;
```

```
  final String? password;
  final String? name;
  final String? gender;
  final int? age;
  final String uid;
  const DailyMobilityView(
      {Key? key,
      this.username,
      this.mail,
      this.password,
      required this.uid,
      this.name,
      this.gender,
      this.age})
      : super(key: key);
  @override
  State<DailyMobilityView> createState() => _DailyMobilityViewState();
}

class _DailyMobilityViewState extends State<DailyMobilityView> {
  String selectedValue = list.first;
  String dropdownValue = list.first;
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(),
    body: Padding(
      padding: context.minAllPadding,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          const LogoBody(),
          Text("Select your daily  Exercise and your Mobility",
              textAlign: TextAlign.center, style: context.headline6(context)),
          Text(selectedValue.toString(), style: context.headline4(context)),
          DropdownButton<String>(
            dropdownColor: Colors.deepPurpleAccent,
            isExpanded: true,
            value: dropdownValue,
            icon: const Icon(Icons.arrow_downward),
            elevation: 5,
            style: context.headline6(context),
            underline: Container(
              width: context.width,
              height: 5,
```

```
    color: Colors.deepPurpleAccent,
  ),
  onChanged: (String? value) {
    setState(() {
      dropdownValue = value!;
      selectedValue = value;
    });
  },
  items: list.map<DropdownMenuItem<String>>((String value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value),
    );
  }).toList(),
),
CommonButton(
    text: MyText.nextText,
    onPressed: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => HeightPage(
            username: widget.username,
```

```
                    mail: widget.mail,
                    password: widget.password,
                    uid: widget.uid,
                    name: widget.name,
                    gender: widget.gender,
                    age: widget.age,
                    mobility: selectedValue,
                  ),
                ),
              );
            })
        ],
      ),
    ),
  );
 }
}
```

HeightPage.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import '../../core/const/const_shelf.dart';
```

```
class HeightPage extends StatefulWidget {
  final String? username;
  final String? mail;
  final String? password;
  final String? name;
  final String? gender;
  final int? age;
  final String uid;
  final String? mobility;
  const HeightPage({
    Key? key,
    this.username,
    this.mail,
    this.password,
    required this.uid,
    this.name,
    this.gender,
    this.age,
    this.mobility,
  }) : super(key: key);

  @override
```

```
  _HeightPageState createState() => _HeightPageState();
}

class _HeightPageState extends State<HeightPage> {
  int _currentValue = 160;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: const CommonAppBar(),
      body: Padding(
        padding: context.minLtrb,
        child: SingleChildScrollView(
          child: Column(
            children: [
              const LogoBody(),
              SizedBox(
                height: context.height / 17,
              ),
              ConstText(
                text: QuestionsText.heightText,
              ),
              SizedBox(
```

```
            height: context.height / 12,
          ),
          _pickerBody(),
          SizedBox(height: context.height * 0.1),
          CommonButton(
            text: MyText.nextText,
            onPressed: _onPressed,
          )
        ],
      ),
    ),
  ),
 );
}

void _onPressed() {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => WeightPage(
          username: widget.username,
          mail: widget.mail,
          password: widget.password,
```

```
            uid: widget.uid,
            name: widget.name,
            gender: widget.gender,
            age: widget.age,
            mobility: widget.mobility,
            height: _currentValue,
          )),
  );
}

_pickerBody() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.center,
    children: [
      NumberPicker(
        selectedTextStyle:
            context.headline6(context)?.copyWith(color: context.mainColor),
        textStyle: context.subtitle2(context),
        decoration: const BoxDecoration(
          border: Border(
            top: BorderSide(
              color: Color(0xFFC4FB6D),
```

```
            width: 3.0,
          ),
          bottom: BorderSide(
            color: Color(0xFFC4FB6D),
            width: 3.0,
          ),
        ),
      ),
      value: (_currentValue),
      minValue: 100,
      maxValue: 220,
      onChanged: (value) => setState(() => _currentValue = value),
    ),
    Container(
      margin: context.midLeft,
      child: Text(
        RegisterText.cmText,
        style: context.subtitle2(context),
      ),
    )
  ],
 );
}
```

}

WeightPage.dart

```
// ignore_for_file: avoid_print, invalid_return_type_for_catch_error, use_build_context_synchronously

import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/service/foods_exercises_service.dart';
import 'package:fistness_app_firebase/views/service/project_network.dart';
import '../../core/const/const_shelf.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';

class WeightPage extends StatefulWidget {
  final String? username;
  final String? mail;
  final String? password;
  final String? name;
  final String? gender;
  final int? age;
  final String? mobility;
  final int? height;
```

```
  final String uid;

  const WeightPage(
      {Key? key,
      this.username,
      this.mail,
      this.password,
      required this.uid,
      this.name,
      this.gender,
      this.age,
      this.height,
      this.mobility})
      : super(key: key);

  @override
  _WeightPageState createState() => _WeightPageState();
}

class _WeightPageState extends State<WeightPage> {
  int _currentValue = 65;
  int totalPoint = 0;
```

```
bool isLoading = false;


@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: const CommonAppBar(),
    body: Padding(
      padding: context.minLtrb,
      child: SingleChildScrollView(
        child: Column(
          children: [
            const LogoBody(),
            SizedBox(
              height: context.height / 17,
            ),
            ConstText(text: QuestionsText.weightText),
            SizedBox(
              height: context.height / 10,
            ),
            _pickerBody(),
            if (isLoading) const LoadingPage(),
            SizedBox(height: context.height * 0.1),
            CommonButton(
```

```
            text: MyText.nextText,
            onPressed: () async {
              setState(() {
                isLoading = true;
              });

              await registerTheUser();
              setState(() {
                isLoading = false;
              });
            },
          ),
        ],
      ),
    ),
  ),
 );
}

_pickerBody() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.center,
```

```
children: [
  NumberPicker(
    selectedTextStyle:
        context.headline6(context)?.copyWith(color: context.mainColor),
    textStyle: context.subtitle2(context),
    decoration: const BoxDecoration(
      border: Border(
        top: BorderSide(
          color: Color(0xFFC4FB6D),
          width: 3.0,
        ),
        bottom: BorderSide(
          color: Color(0xFFC4FB6D),
          width: 3.0,
        ),
      ),
    ),
    value: _currentValue,
    minValue: 30,
    maxValue: 200,
    onChanged: (value) => setState(() => _currentValue = value),
  ),
  Container(
```

```
        margin: context.midLeft,
        child: Text(
          RegisterText.kgText,
          style: context.subtitle2(context),
        ),
      )
    ],
  );
}

Future registerTheUser() async {
  try {
    if (widget.age! <= 20) {
      totalPoint += 5;
    } else if (widget.age! >= 21 && widget.age! <= 35) {
      totalPoint += 4;
    } else if (widget.age! >= 36 && widget.age! <= 50) {
      totalPoint += 3;
    } else if (widget.age! >= 51 && widget.age! <= 65) {
      totalPoint += 2;
    }

    if (_currentValue >= 40 && _currentValue <= 49) {
```

```
  totalPoint += 4;
} else if (_currentValue >= 50 && _currentValue <= 59) {
  totalPoint += 5;
} else if (_currentValue >= 60 && _currentValue <= 69) {
  totalPoint += 6;
} else if (_currentValue >= 70 && _currentValue <= 79) {
  totalPoint += 7;
} else if (_currentValue >= 80 && _currentValue <= 89) {
  totalPoint += 8;
} else if (_currentValue >= 90 && _currentValue <= 99) {
  totalPoint += 9;
} else if (_currentValue >= 100 && _currentValue <= 109) {
  totalPoint += 10;
} else if (_currentValue >= 110 && _currentValue <= 119) {
  totalPoint += 11;
} else if (_currentValue >= 120 && _currentValue <= 129) {
  totalPoint += 12;
} else if (_currentValue >= 130 && _currentValue <= 139) {
  totalPoint += 13;
} else if (_currentValue >= 140 && _currentValue <= 149) {
  totalPoint += 14;
}
```

```
if (widget.height! >= 160) {
  totalPoint += 2;
} else {
  totalPoint += 1;
}

bool? userGender = widget.gender!.contains("fe");
if (userGender) {
  totalPoint += 7;
} else {
  totalPoint += 15;
}

String mobilityControl = widget.mobility![0];
if (mobilityControl == "D") {
  totalPoint += 0;
} else if (mobilityControl == "B") {
  totalPoint += 2;
} else if (mobilityControl == "T") {
  totalPoint += 4;
} else if (mobilityControl == "A") {
  totalPoint += 6;
}
```

```
print("||||||||||||||?????????|||||||$mobilityControl");


print("******????????????******$totalPoint");


Future.delayed(Duration(seconds: 2));


bool? isSucces = await await MyText.authService.createPerson(
    widget.username!,
    widget.mail!,
    widget.password!,
    widget.uid,
    widget.name!,
    widget.gender!,
    widget.age!,
    widget.mobility!,
    widget.height!,
    _currentValue,
    totalPoint);



if (isSucces!) {
  await warningToast(context, RegisterText.registerSuccesfully,
      color: context.greenColor);
```

```
        await MyText.authService.sendEmailVerfied();
        Navigator.pushAndRemoveUntil(
            context,
            MaterialPageRoute(builder: (context) => const LoginPage()),
            (route) => false);
      } else {
        setState(() {
          isLoading = false;
        });
        setState(() {
          isLoading = false;
        });
        await warningToast(context, WarningText.registerUniqueMail);
      }
    } catch (error) {
      setState(() {
        isLoading = false;
      });
      await warningToast(context, error.toString());
      print(error);
    }
  }
}
```

MainPage.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';

import '../../exercises/view/initial/initial_page.dart';
import '../../views_shelf.dart';

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

  @override
  State<MainPage> createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {
  List pages = [const InitialPage(), const ProfilePage()];
  int selectedIndex = 0;
  void changeIndex(int index) {
    setState(() {
      selectedIndex = index;
    });
```

```
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: pages[selectedIndex],
    bottomNavigationBar: BottomNavigationBar(
      onTap: changeIndex,
      currentIndex: selectedIndex,
      unselectedItemColor: context.textColor,
      selectedItemColor: context.scndTxtColor,
      items: <BottomNavigationBarItem>[
        BottomNavigationBarItem(
          icon: const Icon(
            Icons.home,
          ),
          label: MyText.bnFirstText,
        ),
        BottomNavigationBarItem(
          icon: const Icon(
            Icons.account_circle,
          ),
          label: MyText.bnThirdText,
```

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

ExercisePageView.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/exercises/model/exercises_model.dart';
import
'package:fistness_app_firebase/views/exercises/view/detailPages/detail_exercises_page.dart';
import 'package:fistness_app_firebase/views/exercises/viewModel/exercises_view_model.dart';
import 'package:fistness_app_firebase/views/service/project_network.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import '../../../../core/const/const_deco.dart';
import '../../../service/foods_exercises_service.dart';

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

```
  @override
  State<ExercisesPage> createState() => _ExercisesPageState();
}

class _ExercisesPageState extends State<ExercisesPage> {
  String topImgPath = 'largeExrc';

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) {
        String item = "exercises";
        return ExercisesViewModel(
            GeneralService(ProjectNetworkManager.instance.service, item));
      },
      builder: (context, child) {
        return Scaffold(
```

```
      body: ListView(
        children: [
          topImgField(context),
          listTileBody(
              context, context.watch<ExercisesViewModel>().exercises),
        ],
      ),
    );
  },
 );
}

Stack topImgField(BuildContext context) {
 return Stack(
  children: [
    topImg(),
    IconButton(
      onPressed: () {
        Navigator.pop(context);
      },
      icon: const Icon(Icons.chevron_left_rounded),
    ),
    Positioned(
```

```
        bottom: 10,
        left: context.width * 0.03,
        child: Text(
          "Exercises",
          style: context.headline4(context),
        ))
    ],
  );
}

Image topImg() {
  return Image.asset("assets/$topImgPath.jpg");
}

Container listTileBody(BuildContext context, List<Exercise> items) {
  return Container(
    decoration: commonBoxDec(
        context.scfBackColor, context.scfBackColor, context.scfBackColor),
    child: ListView.builder(
      shrinkWrap: true,
      physics: const NeverScrollableScrollPhysics(),
      itemCount: items.length,
      itemBuilder: (BuildContext ctxt, int i) {
```

```
return Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Container(
      margin: context.minAllPadding,
      decoration: BoxDecoration(
        color: Colors.purple,
        borderRadius: BorderRadius.circular(12)),
      child: ListTile(
        onTap: () {
          Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => DetailExercisesPage(
                  items: items[i].categoryData!,
                  images: items[i],
                ),
              ));
        },
        leading: _listTileLeading(context, items, i),
        title: _listTileTitle(items, i, context),
        subtitle: _listTileSubT(items, i, context),
        trailing: const Icon(Icons.chevron_right_outlined),
```

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

Text _listTileSubT(List<Exercise> items, int i, BuildContext context) {
 return Text(
  "See ${items[i].categoryName.toString()} workouts",
  style: context
      .bdSmall(context)
      ?.copyWith(fontSize: 14, color: context.textColor),
 );
}

Text _listTileTitle(List<Exercise> items, int i, BuildContext context) {
 return Text(
  items[i].categoryName.toString(),
  style: context.subtitle2(context)?.copyWith(fontWeight: FontWeight.bold),
 );
```

```
  }

  Container _listTileLeading(
      BuildContext context, List<Exercise> items, int i) {
    return Container(
        decoration: BoxDecoration(
          color: context.greenColor,
          borderRadius: BorderRadius.circular(40),
        ),
        child: _listTileImg(items, i));
  }

  ClipRRect _listTileImg(List<Exercise> items, int i) {
    return ClipRRect(
      borderRadius: BorderRadius.circular(8.0),
      child: Image.network(
        items[i].imgUrl.toString(),
      ),
    );
  }
}
```

ExerciseModel.dart

```
import 'package:json_annotation/json_annotation.dart';

part "exercises_model.g.dart";


@JsonSerializable()

class ExercisesModel {

  List<Exercise>? exercise;


  ExercisesModel({this.exercise});


  factory ExercisesModel.fromJson(Map<String, dynamic> json) {

    return _$ExercisesModelFromJson(json);

  }


  Map<String, dynamic> toJson() {

    return _$ExercisesModelToJson(this);

  }

}


@JsonSerializable()

class Exercise {

  String? categoryName;
```

```
  String? imgUrl;
  List<CategoryData>? categoryData;

  Exercise({this.categoryName, this.imgUrl, this.categoryData});

  factory Exercise.fromJson(Map<String, dynamic> json) {
    return _$ExerciseFromJson(json);
  }

  Map<String, dynamic> toJson() {
    return _$ExerciseToJson(this);
  }
}

@JsonSerializable()
class CategoryData {
  String? contentImage;
  String? type;
  String? equipment;
  String? mechanic;
  String? exerciseLevel;
  String? exerciseName;
  VideoPageData? videoPageData;
```

```
  CategoryData(
      {this.contentImage,
      this.type,
      this.equipment,
      this.mechanic,
      this.exerciseLevel,
      this.exerciseName,
      this.videoPageData});

  factory CategoryData.fromJson(Map<String, dynamic> json) {
    return _$CategoryDataFromJson(json);
  }

  Map<String, dynamic> toJson() {
    return _$CategoryDataToJson(this);
  }
}

@JsonSerializable()
class VideoPageData {
  String? videoUrl;
  String? fullBodyImage;
```

```
  String? firstTitle;
  String? firstContent;
  String? secondTitle;

  VideoPageData(
      {this.videoUrl,
      this.fullBodyImage,
      this.firstTitle,
      this.firstContent,
      this.secondTitle});

  factory VideoPageData.fromJson(Map<String, dynamic> json) {
    return _$VideoPageDataFromJson(json);
  }

  Map<String, dynamic> toJson() {
    return _$VideoPageDataToJson(this);
  }
}
```

ExerciseViewModel.dart

```
import 'package:fistness_app_firebase/views/exercises/model/exercises_model.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import '../../service/foods_exercises_service.dart';

class ExercisesViewModel extends ChangeNotifier {
  List<Exercise> exercises = [];
  final IGeneralService generalService;
  bool isLoading = false;

  ExercisesViewModel(this.generalService) {
    fetchExercisesData();
  }

  void changeLoading() {
    isLoading = !isLoading;
    notifyListeners();
  }

  Future<void> fetchExercisesData() async {
    changeLoading();
    exercises = (await generalService.fetchExercisesItem())?.exercise ?? [];
    changeLoading();
  }
```

```
}
```

DetailExerciseView.dart

```
// ignore_for_file: public_member_api_docs, sort_constructors_first
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/exercises/model/exercises_model.dart';
import
'package:fistness_app_firebase/views/exercises/view/detailVideoPages/detail_video_page.dart';
import 'package:fistness_app_firebase/views/exercises/viewModel/exercises_view_model.dart';
import 'package:fistness_app_firebase/views/service/project_network.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

import '../../../service/foods_exercises_service.dart';

class DetailExercisesPage extends StatefulWidget {
  List<CategoryData> items;
  Exercise images;
  DetailExercisesPage({
    Key? key,
    required this.items,
```

```
    required this.images,
  }) : super(key: key);

  @override
  State<DetailExercisesPage> createState() => _DetailExercisesPageState();
}

class _DetailExercisesPageState extends State<DetailExercisesPage> {
  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) {
        String item = "exercises";
        return ExercisesViewModel(
            GeneralService(ProjectNetworkManager.instance.service, item));
      },
      builder: (context, child) {
        return Scaffold(
```

```
      body: _listViewBody(context),
    );
  },
 );
}

ListView _listViewBody(BuildContext context) {
 return ListView(children: [
  Stack(
   children: [
    ClipRRect(
      borderRadius: BorderRadius.circular(12),
      child: Image.network(
       widget.images.imgUrl.toString(),
      )),
    Positioned(
     left: 0,
     child: IconButton(
      onPressed: () {
       Navigator.pop(context);
      },
      icon: const Icon(
       Icons.chevron_left_sharp,
```

```
        ),
      ),
    ),
    Positioned(
      bottom: 0,
      left: context.width * 0.03,
      child: Text(
        widget.images.categoryName.toString(),
        style: context
            .headline4(context)
            ?.copyWith(color: context.mainColor),
      ),
    )
  ],
),
ListView.builder(
  itemCount: widget.items.length,
  shrinkWrap: true,
  physics: const NeverScrollableScrollPhysics(),
  itemBuilder: (BuildContext ctxt, int i) {
    return _navigateToPage(context, i);
  },
)
```

```
  ]);
}

InkWell _navigateToPage(BuildContext context, int i) {
  return InkWell(
    onTap: () {
      Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) =>
                DetailVideoPage(items: widget.items[i].videoPageData!),
          ));
    },
    child: Container(
      margin: context.minAllPadding,
      decoration: const BoxDecoration(
        color: Colors.purple,
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(24.0),
          topRight: Radius.circular(8.0),
          bottomLeft: Radius.circular(16.0),
        ),
      ),
```

```
child: Center(
  child: Column(
    children: [
      Container(
        decoration: BoxDecoration(
            color: Colors.purple,
            borderRadius: BorderRadius.circular(12)),
        child: ListTile(
          leading: _imageField(i),
          title: _exercisesTitle(context, i),
          subtitle: Container(
            decoration: const BoxDecoration(
                color: Colors.green,
                borderRadius: BorderRadius.only(
                    bottomRight: Radius.circular(10),
                    topLeft: Radius.circular(10))),
            child: Row(
              children: [
                _movementInfo(context, i, "Type", widget.items[i].type),
                _movementInfo(
                    context, i, "Equipment", widget.items[i].equipment),
                _movementInfo(
                    context, i, "Mechanics", widget.items[i].mechanic),
```

```
                    _movementInfo(context, i, "Exp Level",
                        widget.items[i].exerciseLevel),
                  ],
                ),
              ),
              trailing: const Icon(Icons.play_circle_fill_rounded),
            ),
          ),
        ],
      ),
    ),
  ),
);
}

_imageField(int i) => ClipRRect(
    borderRadius: BorderRadius.circular(8.0),
    child: Image.network(widget.items[i].contentImage.toString()));

_exercisesTitle(BuildContext context, int i) {
  return Text(
    widget.items[i].exerciseName.toString(),
    style: context.subtitle1(context),
```

```
  );
}

_movementInfo(BuildContext context, int i, String title, var info) {
  return Expanded(
    child: Padding(
      padding: context.minVertPadding,
      child: Column(
        children: [
          Text(
            title,
            style: context.subtitle1(context)?.copyWith(
                fontSize: 8,
                color: context.scfBackColor,
              ),
          ),
          Text(
            info.toString(),
            style: context.subtitle1(context)?.copyWith(fontSize: 6.0),
          ),
        ],
      ),
    ),
```

```
    );
  }
}
```

DetailExerciseVideoView.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/exercises/model/exercises_model.dart';
import
'package:fistness_app_firebase/views/exercises/view/detailVideoPages/video_player_widget.dart
';
import 'package:fistness_app_firebase/views/exercises/viewModel/exercises_view_model.dart';
import 'package:fistness_app_firebase/views/service/project_network.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:video_player/video_player.dart';
import '../../../service/foods_exercises_service.dart';
import 'package:youtube_player_flutter/youtube_player_flutter.dart';

class DetailVideoPage extends StatefulWidget {
  final VideoPageData items;
  const DetailVideoPage({
```

```
    Key? key,
    required this.items,
  }) : super(key: key);

  @override
  State<DetailVideoPage> createState() => _DetailVideoPageState();
}

class _DetailVideoPageState extends State<DetailVideoPage> {
  final String titleText = 'Exercise Information';
  late YoutubePlayerController _controller;
  late VideoPlayerController videoController;
  int currentIndex = 0;
  String url = "";
  late bool checkUrl;

  @override
  void initState() {
    super.initState();
    checkVideoType();
    _playNormalVideo();
    _playYoutubeVideo();
  }
```

```
@override
void dispose() {
  videoController.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return ChangeNotifierProvider(
    create: (context) {
      String item = "exercises";
      return ExercisesViewModel(
          GeneralService(ProjectNetworkManager.instance.service, item));
    },
    builder: (context, child) {
      return Scaffold(
        body: ListView(children: [
          Stack(
            children: [
              Column(children: [
                checkUrl
                    ? _videoPlayer()
```

```
          : VideoPlayerWidget(controller: videoController)
    ]),
    Positioned(
      top: 0,
      left: 0,
      child: IconButton(
          onPressed: () {
            Navigator.pop(context);
          },
          icon: const Icon(Icons.chevron_left_outlined)),
    ),
  ],
),
ListView.builder(
  itemCount: 1,
  shrinkWrap: true,
  physics: const NeverScrollableScrollPhysics(),
  itemBuilder: (BuildContext ctxt, int i) {
    return Container(
      margin: context.minAllPadding,
      decoration: const BoxDecoration(
        color: Colors.cyan,
        borderRadius: BorderRadius.only(
```

```
                topLeft: Radius.circular(24.0),
                topRight: Radius.circular(8.0),
                bottomLeft: Radius.circular(16.0),
              ),
            ),
            child: Column(
              children: [
                _titleText(context),
                _overviewTextTitle(context),
                _overviewContent(context),
                _instrctTitle(context),
              ],
            ),
          );
        },
      )
    ]),
  );
 },
);
}

void _playNormalVideo() {
```

```
  videoController = VideoPlayerController.network(url)
    ..addListener(() => setState(() {}))
    ..setLooping(true)
    ..initialize().then((_) => videoController.play());
}

void _playYoutubeVideo() {
  _controller = YoutubePlayerController(
      initialVideoId: YoutubePlayer.convertUrlToId(url) ?? "",
      flags: const YoutubePlayerFlags(mute: true, autoPlay: false));
}

void checkVideoType() {
  url = widget.items.videoUrl.toString();
  checkUrl = url.contains("youtube");
}

Container _instrctTitle(BuildContext context) {
  return Container(
    margin: context.minsymVertHorzPadding,
    alignment: Alignment.center,
    height: context.height * 0.05,
    child: Text(
```

```
      widget.items.secondTitle.toString(),
      style: context.subtitle1(context),
    ),
  );
}

Container _overviewContent(BuildContext context) {
  return Container(
    margin: context.minsymVertHorzPadding,
    alignment: Alignment.center,
    child: Text(
      widget.items.firstContent.toString(),
      style: context.subtitle1(context),
    ),
  );
}

Container _overviewTextTitle(BuildContext context) {
  return Container(
    margin: context.minsymVertHorzPadding,
    alignment: Alignment.center,
    height: context.height * 0.05,
    child: Text(
```

```
      widget.items.firstTitle.toString(),
      style: context.subtitle1(context),
    ),
  );
}

YoutubePlayer _videoPlayer() {
  return YoutubePlayer(
    controller: _controller,
    showVideoProgressIndicator: true,
    progressIndicatorColor: Colors.red,
    onReady: () {},
    bottomActions: [
      CurrentPosition(),
      ProgressBar(
        isExpanded: true,
        colors: const ProgressBarColors(
            bufferedColor: Colors.red,
            playedColor: Colors.purple,
            handleColor: Colors.green),
      ),
      const PlaybackSpeedButton(),
      FullScreenButton(),
```

```
      RemainingDuration()
    ],
  );
 }

 Container _titleText(BuildContext context) {
  return Container(
    margin: context.minsymVertHorzPadding,
    alignment: Alignment.centerLeft,
    height: context.height * 0.05,
    child: Text(
      titleText,
      style: context.subtitle1(context)?.copyWith(
          fontWeight: FontWeight.bold, color: context.scfBackColor),
    ),
  );
 }
}
```

VideoPlayer.dart

```
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
```

```
import 'package:flutter/material.dart';
import 'package:video_player/video_player.dart';

class VideoPlayerWidget extends StatelessWidget {
  final VideoPlayerController controller;
  const VideoPlayerWidget({super.key, required this.controller});

  @override
  Widget build(BuildContext context) => controller.value.isInitialized
      ? Column(
          children: [
            Container(
              height: 200,
              alignment: Alignment.topCenter,
              child: buildVideo(),
            ),
            const SizedBox(
              height: 10,
            ),
            Row(
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [buildPlayButton(), buildVideoIndicator(context)],
            )
```

```
        ],
      )
    : const SizedBox(
        height: 200,
        child: Center(child: CircularProgressIndicator()),
      );

Expanded buildVideoIndicator(BuildContext context) {
  return Expanded(
    child: SizedBox(
      height: 20,
      child: VideoProgressIndicator(
        controller,
        allowScrubbing: true,
        padding: context.minAllPadding,
      ),
    ),
  );
}

IconButton buildPlayButton() {
  return IconButton(
      onPressed: () =>
```

```
            controller.value.isPlaying ? controller.pause() : controller.play(),
        icon:
            Icon(controller.value.isPlaying ? Icons.pause : Icons.play_arrow));
  }

  Widget buildVideo() => buildVideoPlayer();

  Widget buildVideoPlayer() => AspectRatio(
      aspectRatio: controller.value.aspectRatio,
      child: VideoPlayer(controller));
}
```

DietListView.dart

```
// ignore_for_file: prefer_const_constructors, avoid_init_to_null, use_build_context_synchronously

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:fistness_app_firebase/core/const/const_shelf.dart';
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/service/foods_exercises_service.dart';
import 'package:fistness_app_firebase/views/home/viewModel/hp_view_mode.dart';
```

```
import 'package:fistness_app_firebase/views/service/project_network.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import 'package:provider/provider.dart';
import 'package:shared_preferences/shared_preferences.dart';

import '../model/foods_model.dart';

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

  @override
  State<HomeView> createState() => _HomeViewState();
}

class _HomeViewState extends State<HomeView> {
  late SharedPreferences prefs;
  late double lastSavedPoint = 0.0;

  @override
  void initState() {
    super.initState();
    definePref();
  }
```

```
Future<void> definePref() async {
  prefs = await SharedPreferences.getInstance();
  lastSavedPoint = prefs.getDouble("point") ?? 0.0;
}

@override
Widget build(BuildContext context) {
  return ChangeNotifierProvider(
    create: (context) {
      String item = "foods";
      return HomeViewModel(
        GeneralService(ProjectNetworkManager.instance.service, item),
      );
    },
    builder: (context, child) {
      return SafeArea(
        child: Scaffold(
          appBar: AppBar(
              leading: IconButton(
                icon: Icon(
                  Icons.arrow_back_outlined,
                  color: context.mainColor,
```

```
      ),
      onPressed: () => Navigator.of(context).pop(),
    ),
    title: Text("Saved daily point is: $lastSavedPoint")),
body: SingleChildScrollView(
  child: Padding(
    padding: const EdgeInsets.all(8.0),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        SizedBox(
          height: context.height * 0.1,
          child: FutureBuilder(
              future: MyText.authService.fetchCurrentUserDoc(),
              builder: (context,
                  AsyncSnapshot<DocumentSnapshot> snapshot) {
                if (snapshot.connectionState !=
                    ConnectionState.waiting) {
                  if (snapshot.hasData) {
                    return ListView.builder(
                      itemCount: 1,
                      itemBuilder: (context, index) {
                        return Container(
```

```
height: context.height * 0.1,
decoration: BoxDecoration(
   color: Colors.purple,
   borderRadius:
      BorderRadius.circular(12)),
child: Column(
 crossAxisAlignment:
    CrossAxisAlignment.center,
 children: [
   Expanded(
     child: Text(
       "Initial point depends on your information: ",
       style: context.subtitle1(context),
     ),
   ),
   Expanded(
     child: Text(
       snapshot.data?[
            "userRightPoint"] ??
         "no dataa",
       style: context.headline4(context),
     ),
   ),
```

```
                ],
              ),
            );
          },
        );
      }
    }
    return Center(
      child: CircularProgressIndicator(),
    );
  }),
),
Container(
  height: context.height * 0.70,
  width: double.infinity,
  decoration: commonBoxDec(context.scfBackColor,
      context.scfBackColor, context.scfBackColor),
  child: ListView(
    children: [
      _allFoodsTitles(
          context, context.watch<HomeViewModel>().foods),
    ],
  ),
```

```
),
Row(
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,
 children: [
  Container(
   height: context.height * 0.05,
   alignment: Alignment.center,
   decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(14.0),
    color: Colors.purple,
   ),
   child: TextButton.icon(
    onPressed: () async {
     var lastPoint = prefs.getDouble("point") ?? 0.0;
     await prefs.setDouble(
        "point",
        lastPoint +
            context.read<HomeViewModel>().totalPoint);

     var savedPoint = prefs.getDouble("point") ?? 0.0;
     setState(() {
      lastSavedPoint = savedPoint;
     });
```

```
    },
    icon: Icon(Icons.save_alt_outlined,
        color: context.textColor),
    label: Text("Save Daily Point",
        style: context.subtitle2(context)),
  ),
),
Container(
  height: context.height * 0.05,
  alignment: Alignment.center,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(14.0),
    color: Colors.purple,
  ),
  child: TextButton.icon(
    onPressed: () async {
      await prefs.remove("point");

      setState(() {
        lastSavedPoint = 0.0;
      });
    },
    icon: Icon(Icons.remove_circle_outline,
```

```
                        color: context.textColor),
                    label: Text("Reset Daily Point",
                        style: context.subtitle2(context)),
                  ),
                ),
              ],
            ),
            SizedBox(
              height: 5,
            ),
            _totalPointText(context),
          ],
        ),
      ),
    ),
  );
 },
 );
}

Container _allFoodsTitles(BuildContext context, List<Kategori> items) {
  return Container(
```

```
padding: context.zeroAllPadding,
child: ListView.builder(
  shrinkWrap: true,
  physics: NeverScrollableScrollPhysics(),
  itemCount: items.length,
  itemBuilder: (BuildContext ctxt, int i) {
    return Column(
      children: [
        Container(
          width: double.infinity,
          margin: context.minTopBtm,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(14.0),
            color: Colors.purple,
          ),
          child: Text(
            items[i].name.toString(),
            style: context.headline6(context),
            textAlign: TextAlign.center,
          ),
        ),
        _allFoodsInfos(context, items, i),
      ],
```

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

Container _allFoodsInfos(BuildContext context, List<Kategori> items, int i) {
  return Container(
    padding: context.minHorzPadding,
    child: ListView.builder(
      shrinkWrap: true,
      itemCount: items[i].icerik!.length,
      itemBuilder: (BuildContext ctx, int j) {
        return Card(
          color: context.scfBackColor,
          child: Row(children: [
            Text(
              items[i].icerik![j].isim.toString(),
              style: context.subtitle2(context),
            ),
            Expanded(child: Container()),
            Text(
              "${items[i].icerik![j].puan!.toDouble()} puan",
```

```
            style: context.subtitle2(context),
          ),
          _checkBox(items, i, j, context),
        ]),
      );
    },
  ),
 );
}

Checkbox _checkBox(List<Kategori> items, int i, int j, BuildContext context) {
 return Checkbox(
  hoverColor: Colors.pink,
  checkColor: context.textColor,
  activeColor: context.mainColor,
  value: items[i].icerik![j].kontrol,
  onChanged: (value) {
   setState(() {
    items[i].icerik![j].kontrol = value!;
    if (items[i].icerik![j].kontrol!) {
     context.read<HomeViewModel>().totalPoint +=
       items[i].icerik![j].puan!;
    } else {
```

```
            context.read<HomeViewModel>().totalPoint -=
                items[i].icerik![j].puan!;
          }
        });
      },
    );
  }

  _totalPointText(BuildContext context) {
    return Container(
      height: context.height * 0.05,
      alignment: Alignment.center,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(14.0),
        color: Colors.purple,
      ),
      child: Text(
        "Current point is: ${context.watch<HomeViewModel>().totalPoint}",
        style: context.headline6(context),
      ),
    );
  }
}
```

DietListModel.dart

```
import 'package:json_annotation/json_annotation.dart';

part "foods_model.g.dart";

@JsonSerializable()
class FoodsModel {
  List<Kategori>? kategori;

  FoodsModel({this.kategori});

  factory FoodsModel.fromJson(Map<String, dynamic> json) {
    return _$FoodsModelFromJson(json);
  }

  Map<String, dynamic> toJson() {
    return _$FoodsModelToJson(this);
  }
}
```

```
@JsonSerializable()
class Kategori {
  final int? id;
  final String? name;
  final List<Icerik>? icerik;

  Kategori({this.id, this.name, this.icerik});

  factory Kategori.fromJson(Map<String, dynamic> json) {
    return _$KategoriFromJson(json);
  }

  Map<String, dynamic> toJson() {
    return _$KategoriToJson(this);
  }
}

@JsonSerializable()
class Icerik {
  final String? isim;
  final dynamic puan;
  bool? kontrol;
```

```
  Icerik({this.isim, this.puan, this.kontrol});

  factory Icerik.fromJson(Map<String, dynamic> json) {
    return _$IcerikFromJson(json);
  }

  Map<String, dynamic> toJson() {
    return _$IcerikToJson(this);
  }
}
```

DietListViewModel.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import '../model/foods_model.dart';
import '../../service/foods_exercises_service.dart';

class HomeViewModel extends ChangeNotifier {
  List<Kategori> foods = [];
  final IGeneralService generalService;
  double totalPoint = 0.0;
```

```
  bool isLoading = false;

  HomeViewModel(this.generalService) {
    fetch();
  }

  void changeLoading() {
    isLoading = !isLoading;
    notifyListeners();
  }

  Future<void> fetch() async {
    changeLoading();
    foods = (await generalService.fetchFoodsItem())?.kategori ?? [];
    changeLoading();
  }
}
```

ProfilePageView.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
```

```
import 'package:fistness_app_firebase/product/global/theme_control.dart';
import 'package:fistness_app_firebase/views/bmi/bmi_page.dart';
import 'package:fistness_app_firebase/views/bmi/future_bmi.dart';
import 'package:fistness_app_firebase/views/profile/edit_profile.dart';
import 'package:fistness_app_firebase/views/views_shelf.dart';
import 'package:provider/provider.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../../core/const/const_shelf.dart';

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

  @override
  State<ProfilePage> createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  Future deleteToken() async {
    final prefs = await SharedPreferences.getInstance();
    prefs.remove("token");
  }

  @override
```

```
Widget build(BuildContext context) {
  return Scaffold(
    body: FutureBuilder(
        future: MyText.authService.fetchCurrentUserDoc(),
        builder: (context, AsyncSnapshot<DocumentSnapshot> snapshot) {
          if (snapshot.connectionState != ConnectionState.waiting) {
            if (snapshot.hasData) {
              return ListView.builder(
                itemCount: 1,
                itemBuilder: (context, index) {
                  return Container(
                    padding: context.minMidAllPadding,
                    decoration: commonBoxDec(
                      context.watch<ThemeNotifier>().isLight
                          ? Colors.grey
                          : const Color(0xFF19282F),
                      context.watch<ThemeNotifier>().isLight
                          ? Colors.white
                          : const Color(0xFF3d444e),
                      context.watch<ThemeNotifier>().isLight
                          ? Colors.white
                          : const Color(0xFF2c2f37),
                    ),
```

```
child: Column(
  children: [
    Row(
      crossAxisAlignment: CrossAxisAlignment.start,
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        GestureDetector(
          child: Container(
            height: 50,
            width: 50,
            decoration: commonBoxDec(
              context.watch<ThemeNotifier>().isLight
                  ? Colors.grey
                  : const Color(0xFF19282F),
              context.watch<ThemeNotifier>().isLight
                  ? Colors.grey
                  : const Color(0xFF19282F),
              context.watch<ThemeNotifier>().isLight
                  ? Colors.grey
                  : const Color(0xFF000000),
            ),
            child: const Icon(
              Icons.arrow_back_ios,
```

```
    ),
  ),
  onTap: () {
    Navigator.pop(context);
  },
),
GestureDetector(
  child: Container(
    height: 50,
    width: 50,
    decoration: commonBoxDec(
      context.watch<ThemeNotifier>().isLight
          ? Colors.grey
          : const Color(0xFF19282F),
      context.watch<ThemeNotifier>().isLight
          ? Colors.grey
          : const Color(0xFF19282F),
      context.watch<ThemeNotifier>().isLight
          ? Colors.grey
          : const Color(0xFF000000),
    ),
    child: const Icon(
      Icons.receipt_outlined,
```

```
        ),
      ),
      onTap: () {
        Navigator.push(
            context,
            (MaterialPageRoute(
              builder: (context) =>
                  const LaunchPage(),
            )));
      },
    ),
  ],
),
Container(
    alignment: Alignment.center,
    child: Text(
      snapshot.data!["name"].toString(),
      style: context
          .headline6(context)
          ?.copyWith(color: context.textColor),
    )),
Container(
  decoration: commonBoxDec(
```

```
    const Color(0xFF0B0B0C),
    const Color(0xFF19282F),
    const Color(0xFF000000),
  ),
),
const SizedBox(
  height: 60,
),
Column(
  children: <Widget>[
    Column(
      children: <Widget>[
        ...ListTile.divideTiles(
          color: context.mainColor,
          tiles: [
            ListTile(
              leading: Icon(
                Icons.email,
              ),
              title: Text("Email"),
              subtitle: Text(
                  snapshot.data!["email"].toString()),
            ),
```

```
GestureDetector(
  onTap: () => context
      .read<ThemeNotifier>()
      .changeTheme(),
  child: const ListTile(
    leading: Icon(
      Icons.light_mode_outlined,
    ),
    title: Text("Theme Light"),
    subtitle: Text("Change theme mode"),
  ),
),
InkWell(
  onTap: () => Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
            BmiCalculator(),
      )),
  child: const ListTile(
    leading: Icon(
      Icons.leave_bags_at_home_outlined,
    ),
```

```
    title: Text("BMI"),
    subtitle: Text("See bmi result"),
  ),
),
GestureDetector(
  child: const ListTile(
    leading: Icon(
      Icons.update,
    ),
    title: Text("Update Profile"),
    subtitle: Text(
        "Update your weight, height..."),
  ),
  onTap: () async {
    Navigator.push(
        context,
        (MaterialPageRoute(
          builder: (context) =>
              const UpdateInfosView(),
        )));
  },
),
GestureDetector(
```

```
            child: const ListTile(
              leading: Icon(
                Icons.logout,
              ),
              title: Text("Logout"),
              subtitle: Text("Have a good day"),
            ),
            onTap: () async {
              await deleteToken();
              MyText.authService.SignOut();
              Navigator.push(
                  context,
                  (MaterialPageRoute(
                    builder: (context) =>
                        const LaunchPage(),
                  )));
            },
          ),
        ],
      ),
    ],
  )
],
```

```
                    )
                  ],
                ),
              );
            },
          );
        }
      }
      return const Center(
        child: CircularProgressIndicator(),
      );
    }),
  );
 }
}
```

UpdateProfilePage.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:fistness_app_firebase/core/const/const_shelf.dart';
import 'package:fistness_app_firebase/core/extensions/edge_insets.dart';
import 'package:fistness_app_firebase/core/extensions/extensions_shelf.dart';
import 'package:fistness_app_firebase/views/bmi/bmi_page.dart';
```

```
import 'package:fistness_app_firebase/views/views_shelf.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';

class UpdateInfosView extends StatefulWidget {
  const UpdateInfosView({Key? key}) : super(key: key);
  @override
  State<UpdateInfosView> createState() => _UpdateInfosViewState();
}

class _UpdateInfosViewState extends State<UpdateInfosView> {
  TextEditingController weightController = TextEditingController();
  TextEditingController heightController = TextEditingController();
  TextEditingController mobilityController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: FutureBuilder(
        future: MyText.authService.fetchCurrentUserDoc(),
        builder: (context, AsyncSnapshot<DocumentSnapshot> snapshot) {
          if (snapshot.connectionState != ConnectionState.waiting) {
```

```
if (snapshot.hasData) {
  return Padding(
    padding: context.minAllPadding,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        TextFormField(
          keyboardType: TextInputType.number,
          maxLength: 3,
          controller: weightController,
          decoration: InputDecoration(
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(14),
              ),
              labelText: "weight -> ${snapshot.data?["weight"]}"),
        ),
        TextFormField(
          keyboardType: TextInputType.number,
          maxLength: 3,
          controller: heightController,
          decoration: InputDecoration(
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(14),
```

```
      ),
      labelText: "height -> ${snapshot.data?["height"]}"),
),
CommonButton(
   text: "Update",
   onPressed: () {
     Map<String, String> dataToUpdate = {
       "weight": (weightController.text.isNotEmpty)
           ? weightController.text
           : snapshot.data?["weight"],
       "height": (heightController.text.isNotEmpty)
           ? heightController.text
           : snapshot.data?["height"],
     };
     CollectionReference collection =
        FirebaseFirestore.instance.collection("users");
     DocumentReference doc = collection
        .doc(FirebaseAuth.instance.currentUser!.uid);
     doc.update(dataToUpdate);
     warningToast(
        context, "Your update completed successfully",
        color: context.greenColor);
     Future.delayed(Duration(seconds: 2));
```

```
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => BmiCalculator(),
                      ));
                  })
              ],
            ),
          );
        }
      }
      return const Center(
        child: CircularProgressIndicator(),
      );
    },
  ),
 );
 }
}
```