**CMR** **University**

**P R O J E C T R E P O R T**

**WNNDERPC**

**An Application to buy Computer parts to build your own custom PC, We can also buy readymade PC’s and Laptops.**

***Submitted in partial fulfillment of the requirements***

***For the award of the degree***

**Bachelor of Computer Application**

**(VI Semester)**

**DEVELOPED BY**

**Name: Sandeep.M**

**Reg No: 19DBCAG126**

**GUIDED BY**

**Prof. Aurangjeb Khan**

Assistant Professor,

IT Department, CMRIMS, Bangalore

**2020 – 2021**

**School of Science Studies**

*#*5, Bhuvanagiri, OMBR Layout, Bangalore- 560 043, Karnataka - India

**CMR** **University**



Certificate

This is to certify that ……………….., and **……………………..**Belonging to **VI Semester, BCA** course has satisfactorily completed the project titled ………………….in partial fulfillment of Practical prescribed by the School of Science Studies for the subject **------------------** during the academic year 2020 - 2021

Prof.AurangjebKhan Dr. Chitra Ravi

Project Guide Director(SOSS)

**Name: Reg No:** 

**Date of Practical Examination:**

**School of Science Studies**

*#*5, Bhuvanagiri, OMBR Layout, Bangalore- 560 043, Karnataka - India

**CMR** **University**

Declaration

The project titled **--------------------------**Developed by us in the partial fulfillment of **VI Semester, BCA** course, is an authentic work carried out by us under the guidance of **Prof.Mohan Reddy.Y, Assistant Professor** School of Science Studies, CMR University, Bangalore.

We declare that the project has not been submitted to any degree or diploma to the above said university or any other university.

Signature: ……………………………………… Signature: ………………………………………

**Name: Name:**

**Reg No: Reg No:**

I certify that all the above statements given by the candidate are true to the best of my knowledge and belief.

Signature: ………………………………………

**Prof. Mohan Reddy.Y**

**Project Guide**

**School of Science Studies**

*#*5, Bhuvanagiri, OMBR Layout, Bangalore- 560 043, Karnataka - India

**ACKNOWLEDGEMENT**

We take this opportunity to express our gratitude to all those who have given their moral support during our entire project.

Firstly, we wish to express our profound thanks to **Dr.Sabita Ramamurthy**, President, CMRJT, Bangalore, for providing us all the facilities required in completion of our project.

We are greatly indebted to **Prof.Jayanthi.M,** Director, School of Science Studies, Bangalore, for her encouragement and suggestion at every step of our project work.

Our Sincere thanks to **Prof.Mohan Reddy.Y,** project guide, without whom this project is unimaginable, for guiding us with keen interest and constant encouragement at every stage during the course of our project work.

Finally, yet importantly, we want to thank all our friends and our family members who have helped us directly or indirectly in the successful completion of this project.

**Name: Sandeep.M**

**Reg No: 19DBCAG126**

**CONTENTS**



**1.Introduction**

**2.Objectives**

**3.SRS (Software Requirement Specification)**

**4.System analysis**

**5. Identification of the Need**

**5.1 Preliminary Investigation**

**5.2 Pert Chart**

**5.3 Gantt Chart**

**5.4 Feasibility Study**

1. **Technical Feasibility**
2. **Economical Feasibility**
3. **Operational Feasibility**

**6. SDLC(Software Development Life Cycle)**

**7. Module Description**

**8. Design**

**8.1 DFD**

**8.2 ER Diagram**

**8.3 Form Design/Snapshot**

**9. Coding**

**10. Testing**

**11.Data validation checks**

**12.Implementation**

**13.Security measures taken**

**14.Cost estimation of the project**

**15.Maintenance**

**16.Limitation and future enhancement**

**17.Bibliography**

**INTRODUCTION**

The app WNNDERPC provides custom-built PC’s and helps you find the perfect parts for building your PC. WNNDERPC desires to fill the void in the PC building industry in India as there are no apps which help build a PC in India. WNNDERPC provides convenience for building a beautiful PC from scratch at affordable prices.

There are options to even buy your laptops, Built -PC’s as to reduce the hassle of building PC’s from scratch.

**OBJECTIVES**

Build your own PC! This app helps you to find PC build ideas for your gaming PC or workstation. You can pick parts and generate builds based on budget. We are always working for more improvement & features.

Parts category: CPU, Motherboard, Ram, SSD, HDD, GPU, PSU, Case, Cooler, Monitor, Speaker/headset, Mouse, Keyboard, Capture Cards, Sound Cards, Game Controller, Gaming Chairs, Monitor Accessories, Adapters.

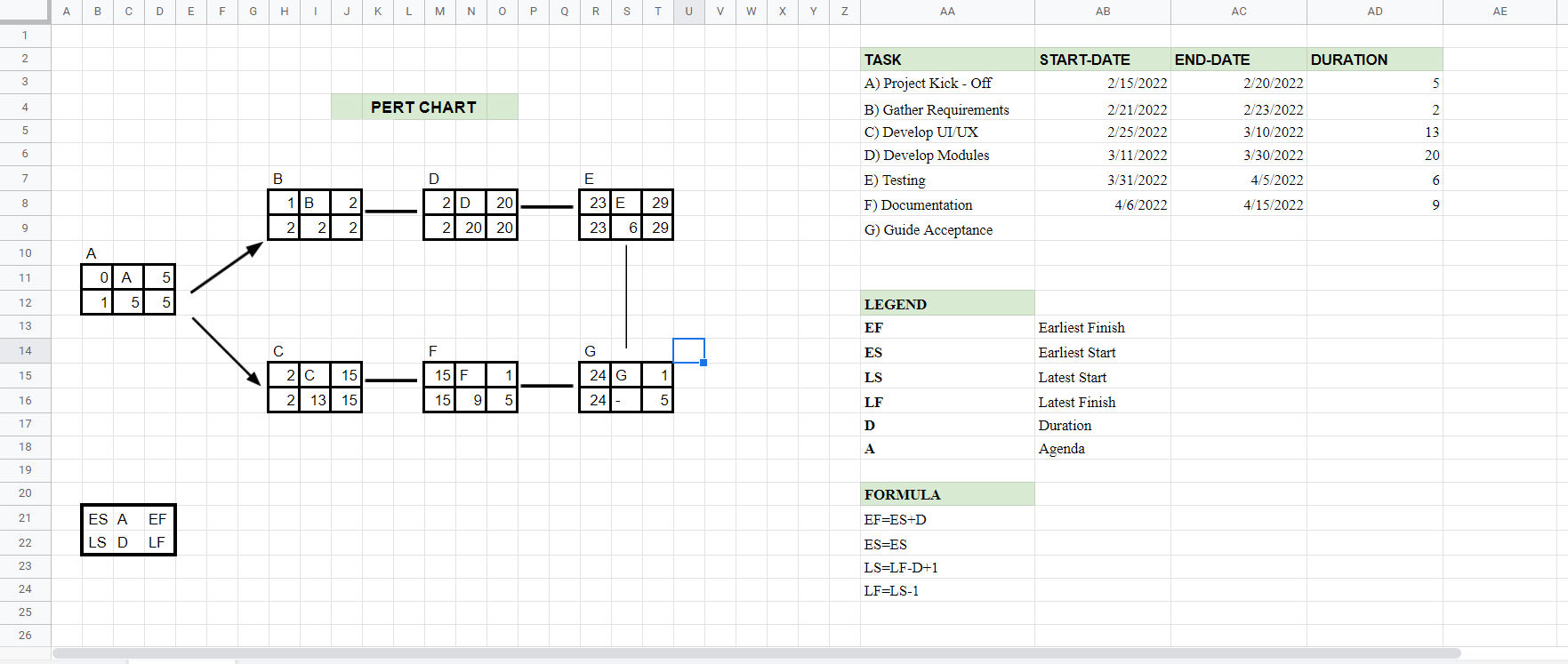
We also provide Laptops and Custom Built PC’s.

**Identification of the Need**

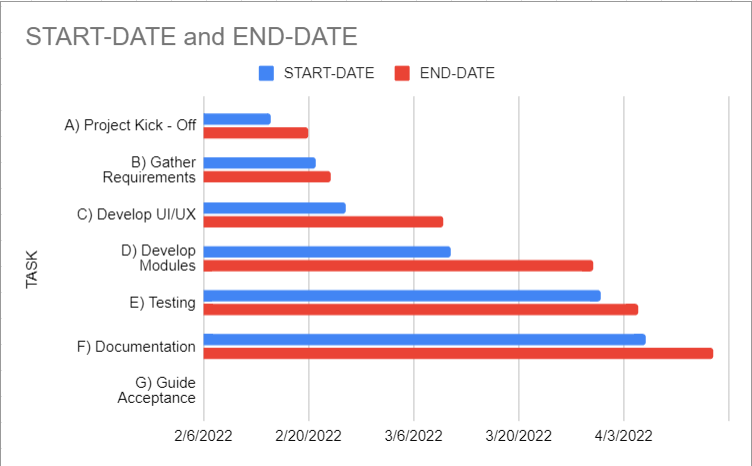
This project aims to fill the void in the PC Building industry in India. Almost all the PC building apps on the playstore are usually just projects or they’re developed in other countries, which means the pricing is too high for the Indian market.

Although the app isn’t complete, this project has given me a chance to develop an application which I have always wanted to build. The app mainly focuses on gaming laptops and PC’s and is designed to exactly show what the customer receives when he places an order in the application.

**PERT CHART**



**GANTT CHART**

****

**FEASIBILITY STUDY**

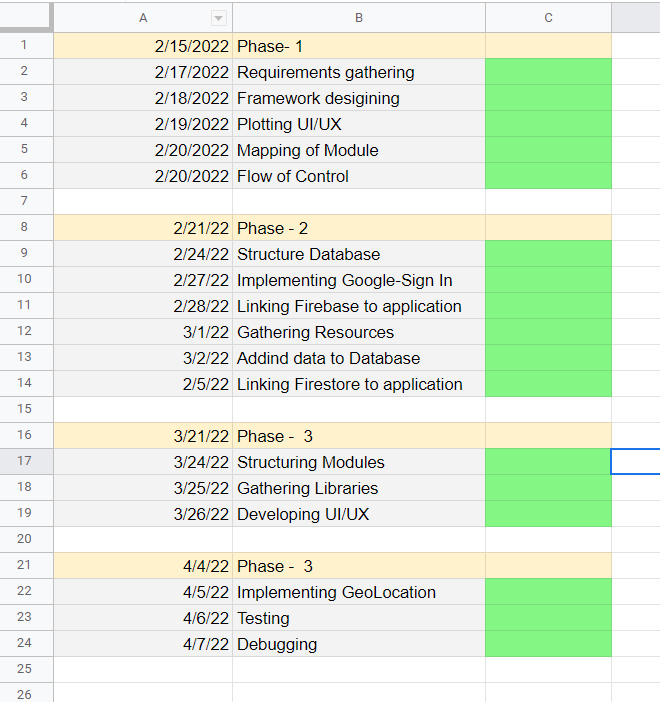
* **TECHNICAL FEASIBILITY**

The development of the application did not have a lot of technical requirements. To run the app we would need a basic android device with any Android 6.0 or more. The application currently does not support IOS devices as the application would have to rebuild its entire framework to support IOS. The application was built using Flutter and Android Studio.

* **ECONOMIC FEASIBILITY**

The application was developed using open-source softwares like Flutter and Android Studio and hence is very feasible on an economic standpoint. The application is free to download.

**Software Development Lifecycle(SDLC)**

****

**MODULE DESCRIPTION**

* **Welcome Page**:

The Welcome Page mainly aims to provide a good impression to the user by displaying a variety of animations and to show the user that the app provides a user-friendly and a techy UI. The animations are loaded using a special library called Video Widget. This page leads to the Category Page of the application if not signed in, else takes the user to the sign in page.

* **Sign In Page**:

The homepage is redirected to this page if the user is not signed in, The app provides only Google-Sign In and the user needs to sign in to the app only once and from the next time the app automatically signs in for the user. This functionality is added to make the app convenient to use as the user doesn't have to login multiple times.

* **Select Category:**

The Google Sign In page redirects the user to this page. This page provides the user to select from 2 categories, Laptops and Custom Built PC’s. On Click of the Laptops Button the user is redirected to the Laptops.dart Page and provides the user multiple options to buy laptops. Similarly On Click of the Custom Built PC’s the user is redirected to the PC.dart and shows a list of Custom Built PC’s for the user to purchase.

* **Laptops Page:**

On Click of the laptops button, the laptops page opens providing the user a navigation bar to choose from the home page and the cart page. The user can also access his profile details. The laptops homepage provides the user a wide range of laptops to choose from and also shows the price of the product, model name, and an image of the laptop to be bought. On clicking on the product the user is redirected to the product page. This page is linked to the database end, and passes the details of the product to the product\_page.dart. This page uses a StreamBuilder to receive data from the backend which is linked to the app.

* **Product Page:**

On click of any laptop in the home page, the user is redirected to this page. Here the user is shown a bigger picture of the product, the model name, the price, ratings, Graphics Card, Memory, Processor, RAM, Refresh Rate. The top left corner includes a back button which takes the user back to the home page and the top left corner button takes the user to the cart. The Add to Cart Button adds the product to the cart and shows a snackbar saying “Added To Cart '' and provides the option to view the cart after adding the item. The compare button takes the user to the Compare page where two products are compared for the convenience of the user as to which to purchase according to their needs.

* **Compare Page:**

The Compare page is specially designed for the user to compare two products, this page is made only for laptops and specifies all the details of the two laptops. The first laptop is added when clicked on the button from the product page and the second laptop is added when the second compare button is clicked and a product is selected.

* **Cart Page:**

The Cart Page is mainly designed to show the user the items added in the cart. The data is added to the cart when the user clicks the add to cart button in the product page and the data is added to Firebase, our backend software. And the data is retrieved from this collection to the Cart Page and is displayed in a ListView attached to a StreamBuilder which has the Data Retrieval Commands. The Cart Page displays the price of the product and an image of the product. It also provides the option to remove individual items from the cart by clicking the delete button icon. The data is updated and the item is even removed from the database. The total price displayed on the cart Page is realtime and keeps updating according to the items added and removed from the cart.

* **Enter Details Module:**

The enter details module takes input from the user about the user, Ex: Their address, phone number, pincode, name, delivery time. This page uses a stepper which combines 3 pages in one. The first page contains 3 text boxes taking input from the user. The application asks for the address of the user and also uses the GeoLocation Widget which takes the exact latitude and longitude of the user's location or where the product wants to be delivered to. The second stepper contains 3 textboxes which ask the user to input their personal details like name, phone number, and also provides an option for the user to schedule their delivery time. The user can click on the cancel button to go back and change any details entered. The final stepper displays the final confirmation for placing the order and the user can still make changes by clicking the cancel button and moving backwards.

* **Profile Page:**

The profile page module displays the details of the user, retrieved from the google servers. The page displays the name of the user, email-id, and if the account is verified or not. This page is also used in the PC buying module.

* **PC’s Page:**

On Click of the PC’s button, the PC’s home page opens providing the user a navigation bar to choose from the home page and the cart page. The user can also access his profile details. The PC’s homepage provides the user a wide range of PC’s to choose from and also shows the price of the product, model name, and an image of the laptop to be bought. On clicking on the product the user is redirected to the product page. This page is linked to the database end, and passes the details of the product to the product\_page.dart. This page uses a StreamBuilder to receive data from the backend which is linked to the app.

* **Product Page:**

On click of any PC in the home page, the user is redirected to this page. Here the user is shown a bigger picture of the product, the model name, the price, ratings, Graphics Card, Memory, Processor, RAM. The top left corner includes a back button which takes the user back to the home page and the top left corner button takes the user to the cart. The Add to Cart Button adds the product to the cart and shows a snackbar saying “Added To Cart '' and provides the option to view the cart after adding the item. The compare button takes the user to the Compare page where two products are compared for the convenience of the user as to which to purchase according to their needs.

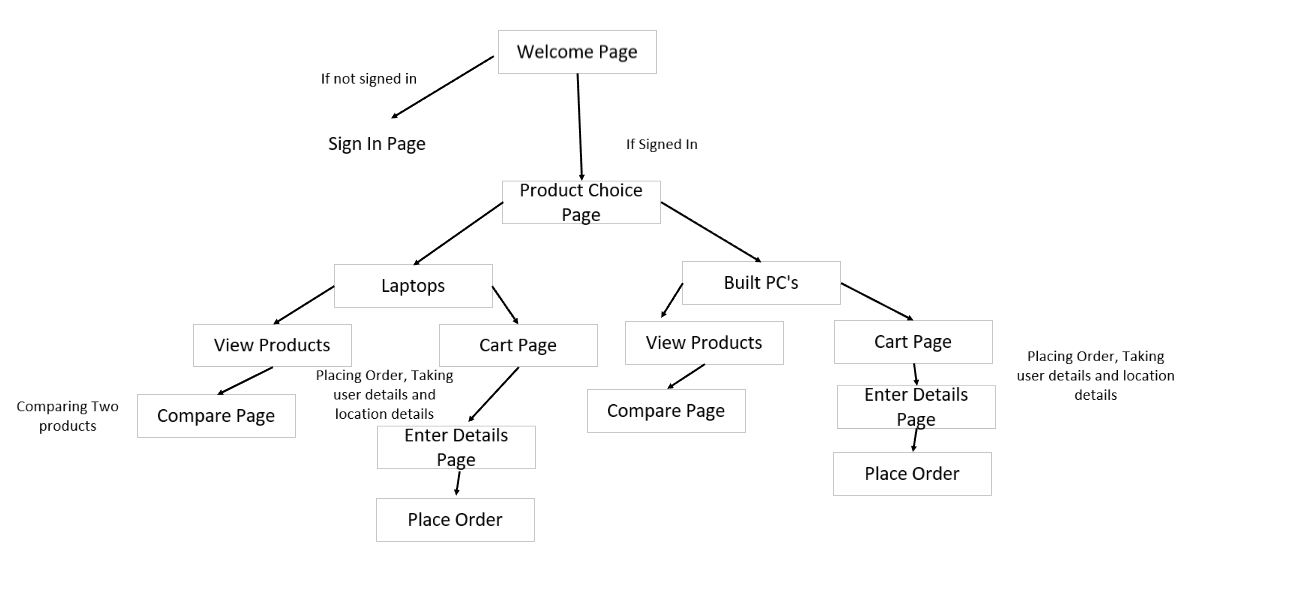
* **Cart Page:**

The Cart Page is mainly designed to show the user the PC’s added in the cart. The data is added to the cart when the user clicks the add to cart button in the product page and the data is added to Firebase, our backend software. And the data is retrieved from this collection to the Cart Page and is displayed in a ListView attached to a StreamBuilder which has the Data Retrieval Commands. The Cart Page displays the price of the product and an image of the product. It also provides the option to remove individual items from the cart by clicking the delete button icon. The data is updated and the item is even removed from the database. The total price displayed on the cart Page is realtime and keeps updating according to the items added and removed from the cart.

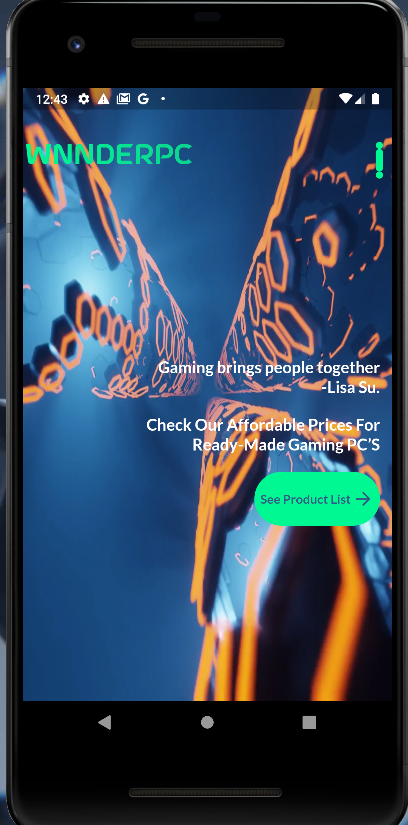
* **Enter Details Module:**

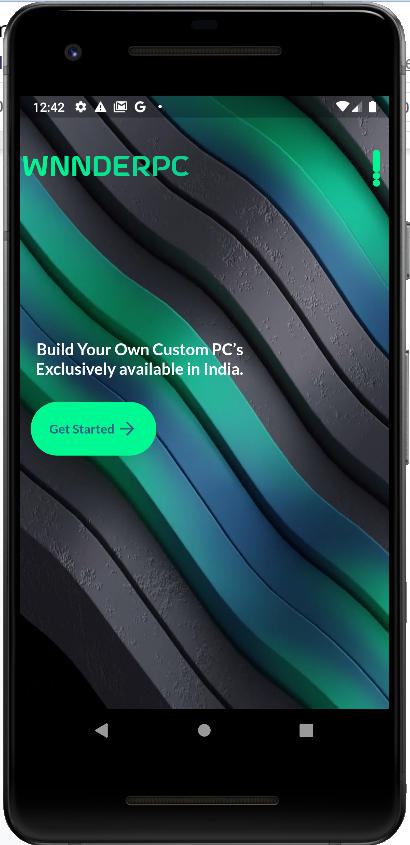
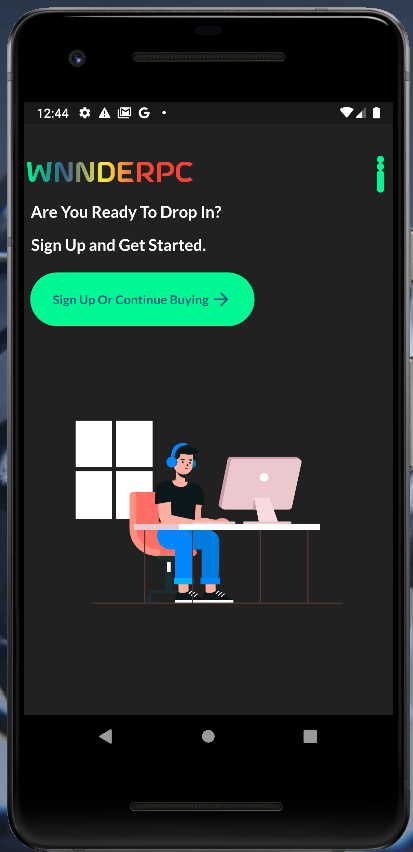
The enter details module takes input from the user about the user, Ex: Their address, phone number, pincode, name, delivery time. This page uses a stepper which combines 3 pages in one. The first page contains 3 text boxes taking input from the user. The application asks for the address of the user and also uses the GeoLocation Widget which takes the exact latitude and longitude of the user's location or where the product wants to be delivered to. The second stepper contains 3 textboxes which ask the user to input their personal details like name, phone number, and also provides an option for the user to schedule their delivery time. The user can click on the cancel button to go back and change any details entered. The final stepper displays the final confirmation for placing the order and the user can still make changes by clicking the cancel button and moving backwards.

**Data Flow Diagram**

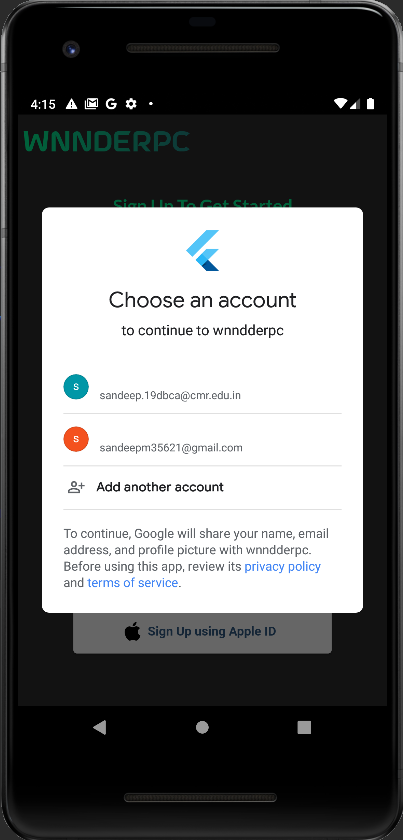
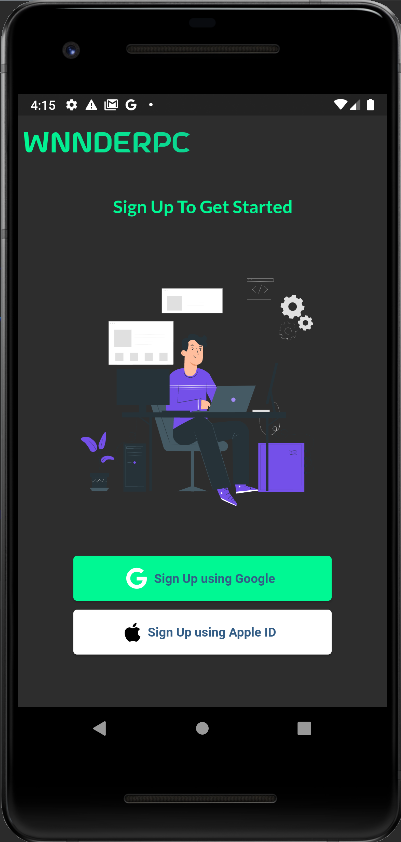


**Form Design and Snapshot**

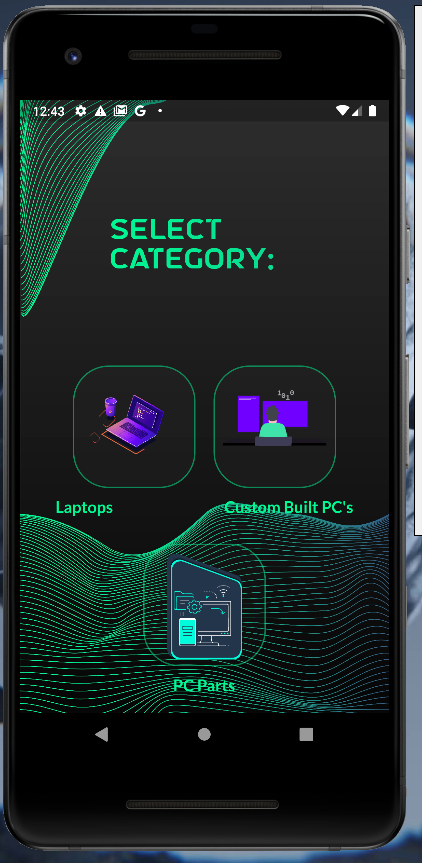
**Welcome Page:** 

**** 

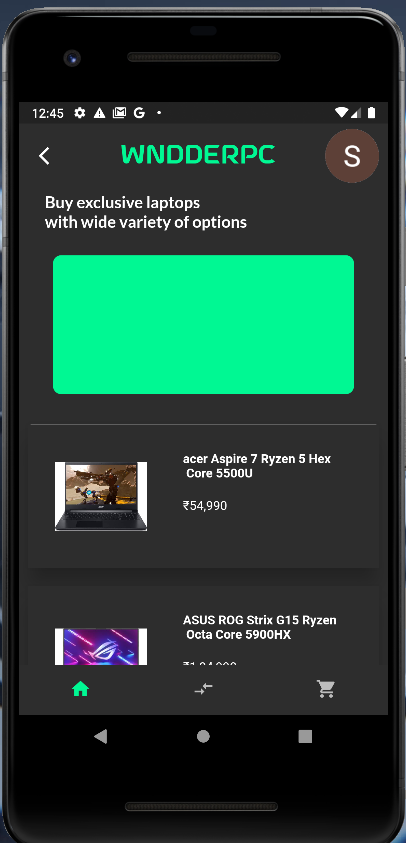
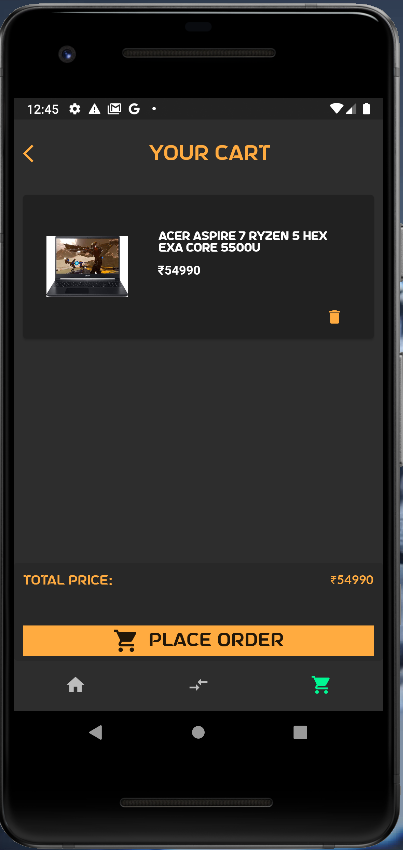
**Sign In Page:**

****

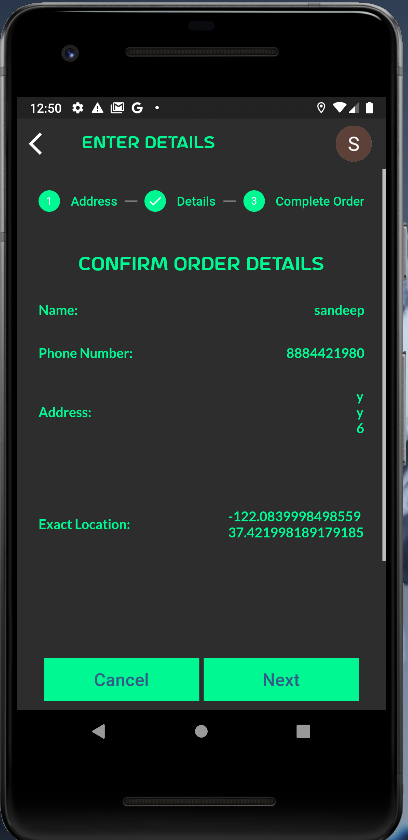
**Categories Page:**

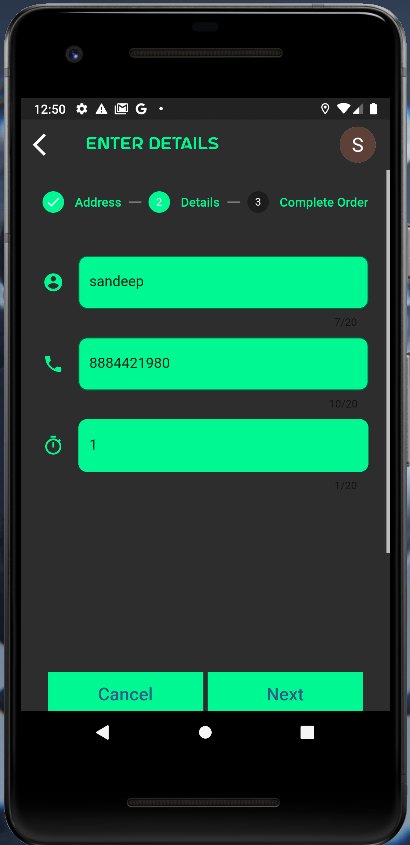
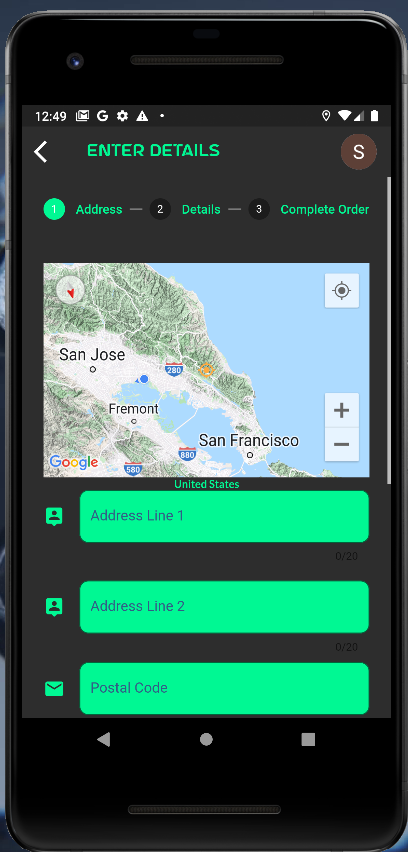
****

**Laptops page:**

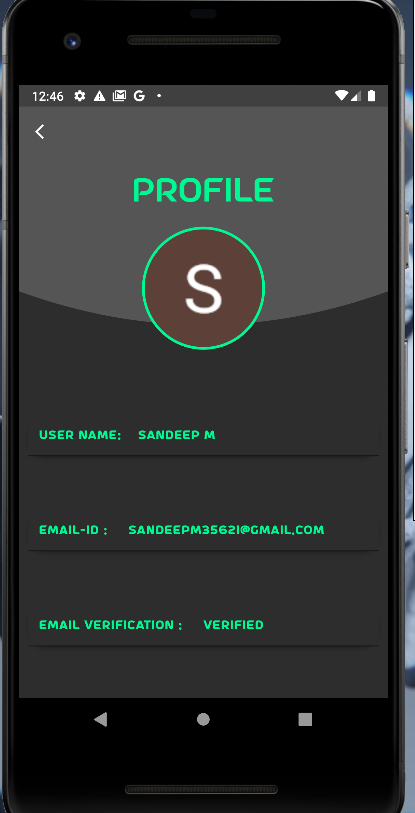
** ** 

**Enter Details Page:**

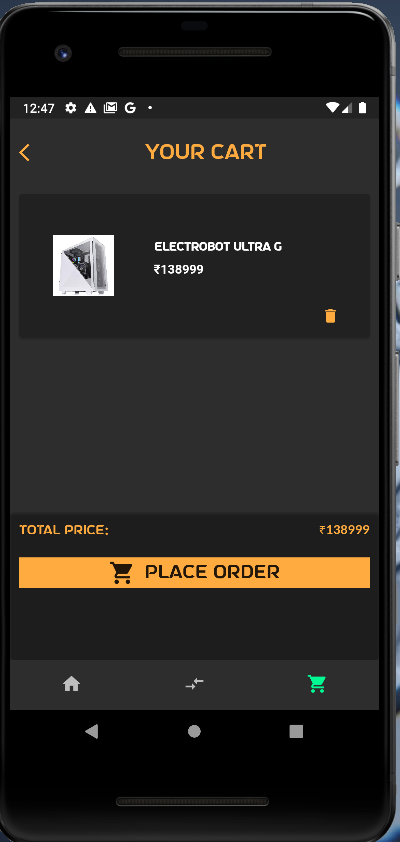


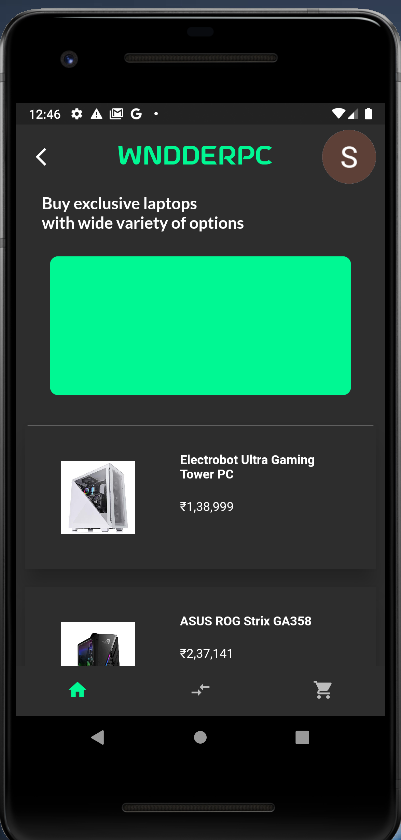
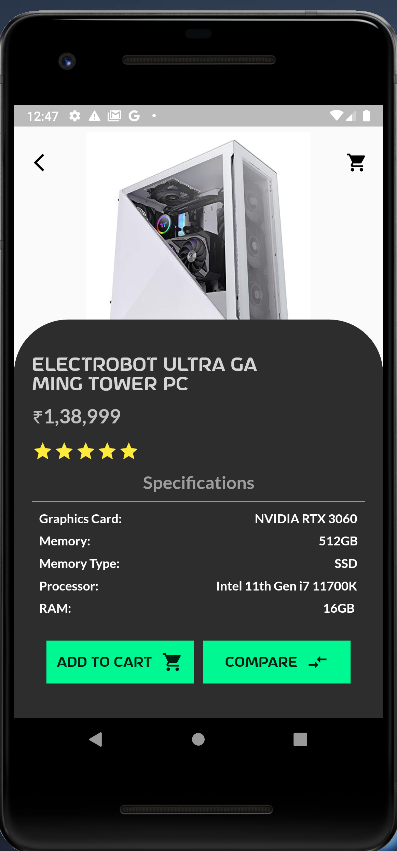
****

**Profile Page:**

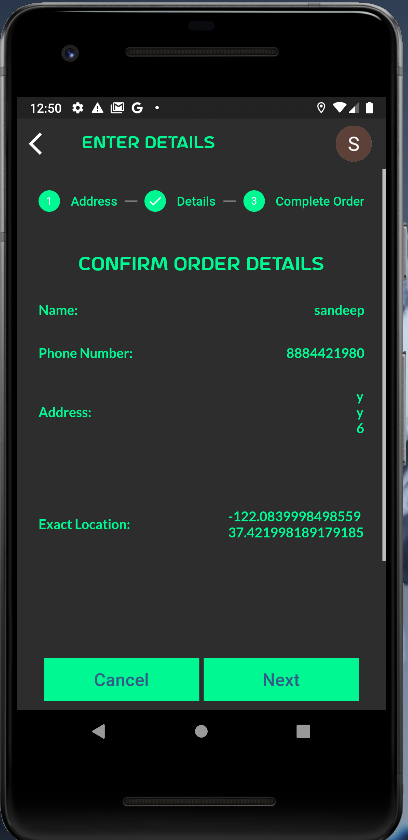
****

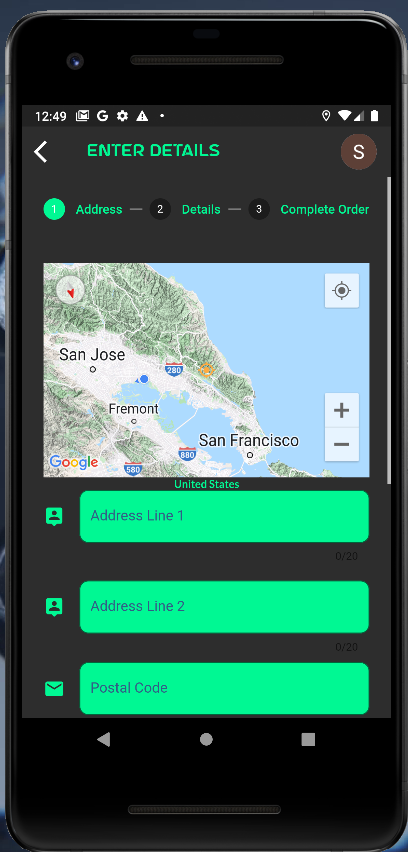
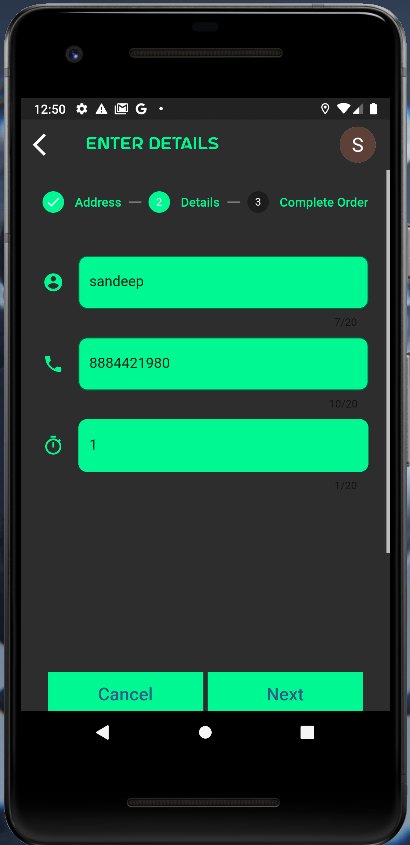
**PC’s Page:**



** **

**Enter Details Page:**



** **

**CODING**

**Welcome\_page.dart:**

import 'package:animated\_text\_kit/animated\_text\_kit.dart';

import 'package:flutter/material.dart';

import 'package:lottie/lottie.dart';

import 'package:wnndderpc/pages/homepage.dart';

import 'package:wnndderpc/widgets/video\_widget.dart';

import '../colors.dart';

import '../widgets/AppLargeText.dart';

import '../widgets/login\_page.dart';

import '../widgets/responsive\_button.dart';

class Welcome\_page extends StatefulWidget {

const Welcome\_page({Key? key}) : super(key: key);

@override

\_Welcome\_pageState createState() => \_Welcome\_pageState();

}

class \_Welcome\_pageState extends State<Welcome\_page> {

static const *colorizeColors* = [

AppColours.*green*,

AppColours.*blue*,

Colors.*yellow*,

Colors.*red*,

];

static const *colorizeTextStyle* = TextStyle(

fontSize: 30.0,

fontFamily: 'Astro',

);

@override

Widget build(BuildContext context) {

return Scaffold(

body: PageView.builder(

scrollDirection: Axis.vertical,

itemCount: 3,

itemBuilder: (\_, index) {

return index == 0

? Stack(

children: [

VideoApp(

url: "videos/back1.mov",

),

Container(

margin: const EdgeInsets.only(left: 2.0, top: 60),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

AnimatedTextKit(

animatedTexts: [

ColorizeAnimatedText("WNNDERPC",

textStyle: *colorizeTextStyle*,

colors: *colorizeColors*),

],

repeatForever: true,

isRepeatingAnimation: true,

),

Column(

children: List.generate(3, (index) {

return Container(

margin: const EdgeInsets.only(right: 10),

width: 8,

height: index == 0 ? 25 : 8,

decoration: BoxDecoration(

borderRadius:

BorderRadius.circular(8),

color: AppColours.*green*),

);

}),

),

],

),

const SizedBox(

height: 50,

),

Column(

children: [

Stack(

children: [

Padding(

padding: const EdgeInsets.only(

top: 120, left: 15),

child: app\_large\_text(

text: "Build Your Own Custom PC’s\nExclusively available in India.",

size: 18,

color: Colors.*white*,

),

),

Padding(

padding: const EdgeInsets.only(

left: 10,

top: 190.0,

),

child: GestureDetector(

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) =>

const LoginPage()),

);

},

child: GestureDetector(

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) =>

const HomePage()),

);

},

child: ResponsiveButton(

text: "Get Started",

width: 140,

),

),

),

),

],

)

],

)

],

),

),

],

)

: index == 1

? Stack(

children: [

VideoApp(

url: "videos/back3.mov",

),

Stack(

children: [

Row(

crossAxisAlignment: CrossAxisAlignment.start,

mainAxisAlignment:

MainAxisAlignment.spaceBetween,

children: [

Padding(

padding: const EdgeInsets.only(

left: 2.0, top: 60),

child: AnimatedTextKit(

animatedTexts: [

ColorizeAnimatedText("WNNDERPC",

textStyle: *colorizeTextStyle*,

colors: *colorizeColors*),

],

repeatForever: true,

isRepeatingAnimation: true,

),

),

Padding(

padding: const EdgeInsets.only(

left: 2.0, top: 60),

child: Column(

children: List.generate(3, (index) {

return Container(

margin:

const EdgeInsets.only(right: 10),

width: 8,

height: index == 1 ? 25 : 8,

decoration: BoxDecoration(

borderRadius:

BorderRadius.circular(8),

color: AppColours.*green*),

);

}),

),

),

],

Column(

crossAxisAlignment: CrossAxisAlignment.end,

children: [

const Padding(

padding: EdgeInsets.only(

left: 150, top: 300, right: 10),

child: Text(

"Gaming brings people together\n-Lisa Su.",

textAlign: TextAlign.end,

style: TextStyle(

color: Colors.*white*,

fontSize: 18,

fontFamily: "LatoBold",

))),

const SizedBox(

height: 20,

),

const Padding(

padding: EdgeInsets.only(right: 10.0),

child: Text(

"Check Our Affordable Prices For\nReady-Made Gaming PC’S ",

textAlign: TextAlign.end,

style: TextStyle(

color: Colors.*white*,

fontSize: 18,

fontFamily: "LatoBold",

)),

),

const SizedBox(

height: 20,

),

Padding(

padding: const EdgeInsets.only(right: 10.0),

child: GestureDetector(

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) =>

const HomePage()),

);

},

child: ResponsiveButton(

text: "See Product List", width: 140),

),

)

],

)

],

),

],

)

: Container(

width: double.*maxFinite*,

height: double.*maxFinite*,

color: Colors.*black87*,

child: Container(

margin: const EdgeInsets.only(left: 2.0, top: 60),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Row(

mainAxisAlignment:

MainAxisAlignment.spaceBetween,

children: [

AnimatedTextKit(

animatedTexts: [

ColorizeAnimatedText("WNNDERPC",

textStyle: *colorizeTextStyle*,

colors: *colorizeColors*),

],

repeatForever: true,

isRepeatingAnimation: true,

),

Column(

children: List.generate(3, (index) {

return Container(

margin:

const EdgeInsets.only(right: 10),

width: 8,

height: index == 2 ? 25 : 8,

decoration: BoxDecoration(

borderRadius:

BorderRadius.circular(8),

color: AppColours.*green*),

);

}),

),

],

),

const SizedBox(

height: 10,

),

Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Padding(

padding: const EdgeInsets.only(left: 5),

child: app\_large\_text(

text: "Are You Ready To Drop In?",

size: 18,

color: Colors.*white*,

),

),

const SizedBox(

height: 15,

),

Padding(

padding: const EdgeInsets.only(left: 5.0),

child: app\_large\_text(

text: "Sign Up and Get Started.",

size: 18,

color: Colors.*white*,

),

),

const SizedBox(

height: 20,

),

Padding(

padding: const EdgeInsets.only(left: 5.0),

child: GestureDetector(

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) =>

const HomePage()),

);

},

child: ResponsiveButton(

text: "Sign Up Or Continue Buying",

width: 250,

),

),

),

Container(

alignment: Alignment.*centerLeft*,

height: MediaQuery.*of*(context).size.height \*

0.6,

width:

MediaQuery.*of*(context).size.width \* 1,

child:

Lottie.*asset*("assets/animation.json"),

),

], )],),),

**Google Sign In page:**

import 'package:animated\_text\_kit/animated\_text\_kit.dart';

import 'package:flutter/material.dart';

import 'package:lottie/lottie.dart';

import 'package:wnndderpc/pages/mainLaptop.dart';

import 'package:wnndderpc/ContentPages/pc.dart';

import 'package:wnndderpc/ContentPages/pc\_parts.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/widgets/app\_medtext.dart';

import 'package:wnndderpc/widgets/content\_button.dart';

class LoggedIn extends StatefulWidget {

const LoggedIn({Key? key}) : super(key: key);

@override

\_LoggedInState createState() => \_LoggedInState();

}

class \_LoggedInState extends State<LoggedIn> {

static const *colorizeColors* = [

AppColours.*green*,

AppColours.*blue*,

Colors.*yellow*,

Colors.*red*,

];

static const *colorizeTextStyle* = TextStyle(

fontSize: 30.0,

letterSpacing: 0.5,

fontFamily: 'Astro',

);

bool \_isBlur = false;

@override

Widget build(BuildContext context) {

return Scaffold(

body: Container(

width: double.*maxFinite*,

decoration: const BoxDecoration(

image: DecorationImage(

image: AssetImage("images/BackGround-min.jpg"),

fit: BoxFit.cover,

)),

height: double.*maxFinite*,

child: Column(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Padding(

padding: const EdgeInsets.only(top: 130.0,left: 100),

child: AnimatedTextKit(

animatedTexts: [

ColorizeAnimatedText("Select Category: ",

textStyle: *colorizeTextStyle*,

colors: *colorizeColors*),

],

),

),

const SizedBox(

height: 100,

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

SizedBox(

height: MediaQuery.*of*(context).size.height \* 0.2,

width: MediaQuery.*of*(context).size.height \* 0.2,

child: GestureDetector(

onTap: () {

setState(() {

\_isBlur = !\_isBlur;

});

Navigator.*push*(

context,

MaterialPageRoute(builder: (context) => const MainLaptop()));

},

child: GlassButton(

blur: \_isBlur ? 28 : 0,

opacity: 0.25,

child: Lottie.*network*(

"https://assets1.lottiefiles.com/packages/lf20\_1elvfrxr.json"),

),

),

),

const SizedBox(

width: 20,

),

SizedBox(

height: MediaQuery.*of*(context).size.height \* 0.2,

width: MediaQuery.*of*(context).size.height \* 0.2,

child: GestureDetector(

onTap: () {

setState(() {

\_isBlur = !\_isBlur;

});

Navigator.*push*(

context,

MaterialPageRoute(builder: (context) => const MainPC()));

},

child: GlassButton(

blur: \_isBlur ? 28 : 0,

opacity: 0.25,

child: Lottie.*network*(

"https://assets4.lottiefiles.com/packages/lf20\_lpsLVC.json",

width: MediaQuery.*of*(context).size.width \* 0.35,

height: MediaQuery.*of*(context).size.width \* 0.35,

),

),

),

),

],

),

const SizedBox(

height: 10,

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

app\_med\_text(

text: "Laptops",

color: AppColours.*green*,

size: 18,

),

SizedBox(

width: MediaQuery.*of*(context).size.width \* 0.3,

),

app\_med\_text(

text: "Custom Built PC's",

color: AppColours.*green*,

size: 18,

)

],

),

const SizedBox(

height: 30,

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

SizedBox(

height: MediaQuery.*of*(context).size.height \* 0.2,

width: MediaQuery.*of*(context).size.height \* 0.2,

child: GestureDetector(

onTap: () {

setState(() {

\_isBlur = !\_isBlur;

});

Navigator.*push*(

context,

MaterialPageRoute(builder: (context) => const PC\_Parts()));

},

child: GlassButton(

blur: \_isBlur ? 15 : 0,

opacity: 0.1,

child: Center(

child: SizedBox(

width: MediaQuery.*of*(context).size.width \* 0.3,

height: 500,

child: Lottie.*network*(

"https://assets10.lottiefiles.com/packages/lf20\_anvjxyyq.json",

),

),

),

),

),

),

],

),

const SizedBox(

height: 10,

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

app\_med\_text(

text: "PC Parts",

color: AppColours.*green*,

size: 18,

),

],

)]),),);}}

**GoogleSignIn.dart:**

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/cupertino.dart';

import 'package:google\_sign\_in/google\_sign\_in.dart';

class GoogleSignInProvider extends ChangeNotifier{

final googleSignIn = GoogleSignIn();

GoogleSignInAccount? \_user;

GoogleSignInAccount get user => \_user!;

Future googleLogin() async

{

final googleUser = await googleSignIn.signIn();

if(googleUser == null) return;

\_user = googleUser;

final googleAuth = await googleUser.authentication;

final credential = GoogleAuthProvider.*credential*(

accessToken: googleAuth.accessToken,

idToken: googleAuth.idToken,

);

await FirebaseAuth.*instance*.signInWithCredential(credential);

notifyListeners();

}}

**MainLaptop.dart:**

import 'package:flutter/material.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/navpages/cart.dart';

import 'package:wnndderpc/navpages/laptops.dart';

import '../navpages/compare\_page.dart';

class MainLaptop extends StatefulWidget {

const MainLaptop({Key? key}) : super(key: key);

@override

\_LaptopsState createState() => \_LaptopsState();

}

class \_LaptopsState extends State<MainLaptop> {

List pages = [

const Laptops(),

const Compare(mem: '0', memtype: '0', price: '0', ram: '0', name: 'Nonw', grcard: '0', proccesor: '0', rrate: '0', stars: '5', image: '0',),

const CartPage(),

];

int currentIndex = 0;

void onTap(int index)

{

setState(() {

currentIndex = index;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: AppColours.*black*,

body: pages[currentIndex],

bottomNavigationBar: BottomNavigationBar(

currentIndex: currentIndex,

onTap: onTap,

unselectedFontSize: 0,

backgroundColor: AppColours.*black*,

selectedItemColor: AppColours.*green*,

unselectedItemColor: Colors.*grey*[400],

showSelectedLabels: false,

showUnselectedLabels: false,

elevation: 0,

items: const [

BottomNavigationBarItem(icon: Icon(Icons.*home*), label: "Home"),

BottomNavigationBarItem(icon: Icon(Icons.*compare\_arrows*), label: "Compare"),

BottomNavigationBarItem(icon: Icon(Icons.*shopping\_cart*), label: "Cart"),

],

),

);

}

}

**Laptops.dart:**

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:wnndderpc/colors.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:wnndderpc/pages/homepage.dart';

import 'package:wnndderpc/pages/product\_page.dart';

import 'package:wnndderpc/pages/profile\_page.dart';

import 'package:wnndderpc/widgets/app\_medtext.dart';

class Laptops extends StatefulWidget {

const Laptops({Key? key}) : super(key: key);

@override

\_LaptopsState createState() => \_LaptopsState();

}

class \_LaptopsState extends State<Laptops> {

int compare = 0;

ScrollController controller = ScrollController();

bool closeTopContainer = false;

final user = FirebaseAuth.*instance*.currentUser!;

final CollectionReference \_firebaseFirestore =

FirebaseFirestore.*instance*.collection("laptops");

@override

void initState() {

super.initState();

controller.addListener(() {

setState(() {

closeTopContainer = controller.offset > 10;

});

});

}

@override

Widget build(BuildContext context) {

final Size size = MediaQuery.*of*(context).size;

return Scaffold(

backgroundColor: AppColours.*black*,

body: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Padding(

padding: const EdgeInsets.only(top: 30.0, left: 10, right: 10),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

IconButton(onPressed: (){

Navigator.*push*(context, MaterialPageRoute(builder: (context) => const HomePage()));

}, icon: const Icon(Icons.*arrow\_back\_ios*),color: Colors.*white*,),

const Text(

"WNDDERPC",

style: TextStyle(

fontSize: 28,

fontFamily: "Astro",

color: AppColours.*green*,

),

),

GestureDetector(

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) => const ProfilePage()));

},

child: CircleAvatar(

radius: 30,

backgroundImage: NetworkImage(user.photoURL!),

),

),

],

),*//AppBar*

),

const SizedBox(

height: 10,

),

AnimatedOpacity(

duration: const Duration(milliseconds: 350),

opacity: closeTopContainer ? 0 : 1,

child: AnimatedContainer(

duration: const Duration(milliseconds: 350),

alignment: Alignment.*topCenter*,

width: size.width,

height: closeTopContainer ? 0 : size.height \* 0.33,

child: FittedBox(

fit: BoxFit.fill,

alignment: Alignment.*topCenter*,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Padding(

padding: const EdgeInsets.only(left: 10.0),

child: app\_med\_text(

text:

"Buy exclusive laptops\nwith wide variety of options",

color: Colors.*white*,

size: 20,

),

),

const SizedBox(

height: 30,

),

Container(

margin: const EdgeInsets.only(left: 20, right: 20),

alignment: Alignment.*center*,

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10),

color: AppColours.*green*,

),

height: size.height \* 0.25,

width: size.width \* 0.9,

)

],

),

),

),

),*//Image Container*

StreamBuilder(

stream: \_firebaseFirestore.snapshots().asBroadcastStream(),

builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {

if(!snapshot.hasData)

{

return const Center(child: CircularProgressIndicator(

color: Colors.*orangeAccent*,

),);

}

else {

return Expanded(

child: ListView.builder(

shrinkWrap: true,

itemCount: snapshot.data?.size,

physics: const BouncingScrollPhysics(),

cacheExtent: 200.0,

itemExtent: 180,

controller: controller,

scrollDirection: Axis.vertical,

itemBuilder: (context, index) {

DocumentSnapshot lap = snapshot.data?.docs[index] as DocumentSnapshot<Object?>;

return GestureDetector(

onTap: (){

Navigator.*push*(context,MaterialPageRoute(builder: (context) => ProductPage(

image: lap['image'],

grcard: lap['graphic'],

mem: lap['mem'],

memtype: lap['memtype'],

proccesor: lap['proccesor'],

ram: lap['ram'],

rrate: lap['refresh'], stars:

lap['ratings'], name: lap['name'], price: lap['price'],)));

},

child: Card(

elevation: 20,

shadowColor: Colors.*black*,

margin: const EdgeInsets.all(10),

child: Container(

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(0),

color: AppColours.*black*,

),

child: Row(

children: [

Container(

margin: const EdgeInsets.only(left: 30),

height: size.height \* 0.12,

width: size.width \* 0.25,

decoration: BoxDecoration(

image: DecorationImage(

image: NetworkImage(

lap['image']

)),

),

),

const SizedBox(

width: 40,

),

Padding(

padding: const EdgeInsets.only(top: 30.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment

.start,

children: [

Text(

lap['name'].toString().substring(0,25) + "\n" + lap['name'].toString().substring(26),

textAlign: TextAlign.start,

style: const TextStyle(

fontFamily: "Lato",

fontWeight: FontWeight.*bold*,

color: Colors.*white*),

),

const SizedBox(

height: 20,

),

*//'\u{20B9}'*

Text(lap['price'].toString().length > 5? '\u{20B9}' + lap['price'].toString()[0] + "," + lap['price'].toString().substring(1,3) + "," + lap['price'].toString().substring(3)

: '\u{20B9}' + lap['price'].toString().substring(0,2) + "," + lap['price'].toString().substring(2,),

style: const TextStyle(

color: Colors.*white*,

),

),

],

),

),

],

),

),

),);

}),);}},

),*//Cards*

],),);}}

**Cart.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:flutter/rendering.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/pages/mainLaptop.dart';

import 'package:wnndderpc/pages/placeorder.dart';

import 'package:wnndderpc/widgets/app\_medtext.dart';

import '../widgets/AppLargeText.dart';

class CartPage extends StatefulWidget {

const CartPage({

Key? key,

}) : super(key: key);

@override

\_CartPageState createState() => \_CartPageState();

}

class \_CartPageState extends State<CartPage> {

var total;

late int length;

@override

Widget build(BuildContext context) {

final Size size = MediaQuery.*of*(context).size;

return Scaffold(

backgroundColor: AppColours.*black*,

body: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Padding(

padding: const EdgeInsets.only(top: 50, left: 10, right: 125),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

crossAxisAlignment: CrossAxisAlignment.end,

children: [

GestureDetector(

child: const Icon(

Icons.*arrow\_back\_ios*,

color: Colors.*orangeAccent*,

),

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) => const MainLaptop()));

},

),

Center(

child: app\_large\_text(

text: "Your Cart",

size: 22,

color: Colors.*orangeAccent*,

FontFamily: "Astro",

)),

],

),

),

StreamBuilder(

stream: FirebaseFirestore.*instance*

.collection("users-cart-items")

.doc(FirebaseAuth.*instance*.currentUser!.email)

.collection("items")

.snapshots(),

builder: (BuildContext context,

AsyncSnapshot<QuerySnapshot> snapshot) {

if (!snapshot.hasData) {

return const Center(

child: CircularProgressIndicator(),

);

} else {

length = snapshot.data!.docs.length;

return Expanded(

child: ListView.builder(

scrollDirection: Axis.vertical,

shrinkWrap: true,

itemCount: snapshot.data!.docs.length,

physics: const BouncingScrollPhysics(),

cacheExtent: 200.0,

itemExtent: 180,

itemBuilder: (context, index) {

DocumentSnapshot \_documentSnapshot =

snapshot.data!.docs[index];

return Card(

margin: const EdgeInsets.only(

left: 10, right: 10, top: 10, bottom: 10),

color: Colors.*grey*[900],

elevation: 2,

child: Container(

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10),

),

child: Row(

children: [

Container(

margin: const EdgeInsets.only(left: 10),

width: size.width \* 0.3,

height: size.height \* 0.1,

decoration: const BoxDecoration(),

child: Image(

fit: BoxFit.scaleDown,

image: NetworkImage(

\_documentSnapshot['images']),

),

),

const SizedBox(

width: 10,

),

Column(

crossAxisAlignment:

CrossAxisAlignment.start,

mainAxisAlignment: MainAxisAlignment.end,

children: [

Padding(

padding:

const EdgeInsets.only(left: 8.0),

child: Text(

\_documentSnapshot['name']

.toString()

.substring(0, 25) +

"\n" +

\_documentSnapshot['name']

.toString()

.substring(23),

textAlign: TextAlign.start,

style: const TextStyle(

fontFamily: "Astro",

fontSize: 12,

fontWeight: FontWeight.*bold*,

color: Colors.*white*),

),

),

const SizedBox(

height: 10,

),

Text(

" \u{20B9}" +

\_documentSnapshot['price'],

textAlign: TextAlign.start,

style: const TextStyle(

fontFamily: "Lato",

fontWeight: FontWeight.*bold*,

color: Colors.*white*),

),

Padding(

padding: const EdgeInsets.only(

left: 180.0,

top: 20,

),

child: IconButton(

iconSize: 20,

icon: const Icon(Icons.*delete*),

color: Colors.*orangeAccent*,

onPressed: () {

final snackbar = SnackBar(

duration:

const Duration(seconds: 2),

behavior:

SnackBarBehavior.floating,

margin: const EdgeInsets.only(

bottom: 10),

backgroundColor: Colors.*white*,

content: const Text(

"Removed From Cart!",

style: TextStyle(

color: AppColours.*blue*),

),

action: SnackBarAction(

label: "Dismiss",

onPressed: () {

ScaffoldMessenger.*of*(context)

.hideCurrentSnackBar();

},

),

);

FirebaseFirestore.*instance*

.collection("users-cart-items")

.doc(FirebaseAuth.*instance*

.currentUser!.email)

.collection("items")

.doc(\_documentSnapshot.id)

.delete();

ScaffoldMessenger.*of*(context)

.showSnackBar(snackbar);

},

),

)

],

),

],

),

),

);

}),

);

}

}),

SizedBox(

height: size.height \* 0.16,

width: size.width,

child: Card(

color: Colors.*black12*,

margin: EdgeInsets.*zero*,

elevation: 0,

child: Padding(

padding: const EdgeInsets.only(

left: 10, right: 10, top: 10),

child: Column(

children: [

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

app\_large\_text(

text: "Total Price:",

color: Colors.*orangeAccent*,

FontFamily: "Astro",

size: 14,

),

StreamBuilder(

stream: FirebaseFirestore.*instance*

.collection("users-cart-items")

.doc(FirebaseAuth.*instance*.currentUser!.email)

.collection("items")

.snapshots(),

builder: (BuildContext context,

AsyncSnapshot<QuerySnapshot> snapshot) {

if (snapshot.hasError) {

return const Text("Something went Wrong");

}

if (snapshot.connectionState ==

ConnectionState.waiting) {

return const CircularProgressIndicator();

} else {

total = 0;

snapshot.data?.docs.forEach((element) {

total += int.*parse*(element['price']);

});

return app\_med\_text(

text: '\u{20B9}' + total.toString(),

color: Colors.*orangeAccent*,

size: 14,

);

}

},

),

],

),

const SizedBox(

height: 40,

),

GestureDetector(

onTap: (){

Navigator.*push*(context, MaterialPageRoute(builder: (context) => const PlaceOrders()));

},

child: Container(

width: size.width,

height: size.height \* 0.05,

color: Colors.*orangeAccent*,

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Icon(

Icons.*shopping\_cart*,

color: Colors.*black87*,

size: 30,

),

const SizedBox(

width: 10,

),

app\_med\_text(

text: "Place Order",

color: Colors.*black87*,

FontFamily: "Astro",

size: 20,

)

],

),

),

)

],

),

),

),

)

],

),

);

}

}

**Compare.dart:**

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/navpages/find\_pc.dart';

import 'package:wnndderpc/pages/mainLaptop.dart';

import 'package:wnndderpc/widgets/AppLargeText.dart';

import 'package:wnndderpc/widgets/app\_medtext.dart';

import '../pages/profile\_page.dart';

class Compare extends StatefulWidget {

final String image, grcard, mem, memtype, proccesor, ram, rrate, name, stars, price;

const Compare({Key? key,

required this.name,

required this.stars,

required this.image,

required this.grcard,

required this.mem,

required this.memtype,

required this.proccesor,

required this.ram,

required this.rrate,

required this.price, image1, grcard1, mem1, memtype1, proccesor1, ram1, rrate1, stars1, price1, name1

}) : super(key: key);

@override

\_CompareState createState() => \_CompareState();

}

class \_CompareState extends State<Compare> {

final user = FirebaseAuth.*instance*.currentUser!;

@override

Widget build(BuildContext context) {

final Size size = MediaQuery

.*of*(context)

.size;

return Scaffold(

appBar: AppBar(

elevation: 0,

centerTitle: true,

leading: IconButton(

icon: const Icon(Icons.*arrow\_back\_ios*), onPressed: () {

Navigator.*push*(context, MaterialPageRoute(builder: (context) => const MainLaptop()));

},

),

backgroundColor: AppColours.*black*,

title: Padding(

padding: const EdgeInsets.only(top: 6.0,),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

app\_large\_text(text: "Compare", color: AppColours.*green*,FontFamily: "Astro",),

GestureDetector(

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) => const ProfilePage()));

},

child: CircleAvatar(

radius: 20,

backgroundImage: NetworkImage(user.photoURL!)

,

),

),

],

),

),

),

backgroundColor: AppColours.*black*,

body: SizedBox(

width: size.width,

height: size.height,

child: Row(

crossAxisAlignment: CrossAxisAlignment.center,

mainAxisAlignment: MainAxisAlignment.start,

children: [

Padding(

padding: const EdgeInsets.only(top: 130.0,left: 7),

child: SizedBox(

width: size.width/4,

child: Column(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.start,

children: [

app\_med\_text(text: "Name:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "Ratings:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "Graphics\nCard:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "Memory:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "Memory\ntype:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "Price:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "Processor:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "Refresh Rate:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

app\_med\_text(text: "RAM:", color: AppColours.*green*,size: 18,FontFamily: "LatoBold",),

const SizedBox(height: 30,),

],

),

),

),

Container(

decoration: const BoxDecoration(

color: AppColours.*blue*,

boxShadow: [

BoxShadow(

color: Colors.*black54*,

blurRadius: 1.0,

spreadRadius: 0.0,

offset: Offset(3.0, 2.0),

)

]

),

margin: EdgeInsets.*zero*,

width: size.width/3,

height: size.height / 1.1,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(

margin: const EdgeInsets.all(5),

width: size.width/3,

height: size.height / 9,

decoration: BoxDecoration(

image: DecorationImage(

fit: BoxFit.contain,

image: NetworkImage(widget.image),

)

),

),

const SizedBox(height: 5,),

Padding(

padding: const EdgeInsets.only(left: 3.0),

child: app\_med\_text(text: widget.name.isEmpty? "None" : widget.name, color: AppColours.*green*,size: 15,),

),

const SizedBox(height: 10,),

Wrap(

children: List.generate(5, (index) {

return Icon(Icons.*star*,

color: index < int.*parse*(widget.stars)

? Colors.*yellow*

: Colors.*grey*);

}),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.grcard, color: AppColours.*green*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.mem, color: AppColours.*green*,size: 15,),

),

const SizedBox(height: 55,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.memtype, color: AppColours.*green*,size: 15,),

),

const SizedBox(height: 50,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.price, color: AppColours.*green*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.proccesor, color: AppColours.*green*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.rrate, color: AppColours.*green*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.ram, color: AppColours.*green*,size: 15,),

),

],

),

),

const SizedBox(width: 10,),

Container(

decoration: const BoxDecoration(

color: AppColours.*green*,

boxShadow: [

BoxShadow(

color: Colors.*black54*,

blurRadius: 1.0,

spreadRadius: 0.0,

offset: Offset(3.0, 2.0),

)

]

),

margin: EdgeInsets.*zero*,

width: size.width/3,

height: size.height / 1.1,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(

margin: const EdgeInsets.all(5),

width: size.width/3,

height: size.height / 9,

decoration: BoxDecoration(

image: DecorationImage(

fit: BoxFit.contain,

image: NetworkImage(widget.image),

)

),

),

const SizedBox(height: 5,),

Padding(

padding: const EdgeInsets.only(left: 3.0),

child: app\_med\_text(text: widget.name.substring(0,20) + "\n" + widget.name.substring(20,), color: AppColours.*blue*,size: 15,),

),

const SizedBox(height: 10,),

Wrap(

children: List.generate(5, (index) {

return Icon(Icons.*star*,

color: index < int.*parse*(widget.stars)

? Colors.*yellow*

: Colors.*grey*);

}),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.grcard, color: AppColours.*blue*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.mem, color: AppColours.*blue*,size: 15,),

),

const SizedBox(height: 55,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.memtype, color: AppColours.*blue*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.price.toString().length > 5? '\u{20B9}' + widget.price.toString()[0] + "," + widget.price.toString().substring(1,3) + "," + widget.price.toString().substring(3)

: '\u{20B9}' + widget.price.toString().substring(0,2) + "," + widget.price.toString().substring(2,), color: AppColours.*blue*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.proccesor, color: AppColours.*blue*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.rrate, color: AppColours.*blue*,size: 15,),

),

const SizedBox(height: 35,),

Padding(

padding: const EdgeInsets.only(left:5.0),

child: app\_med\_text(text: widget.ram, color: AppColours.*blue*,size: 15,),

),

Padding(

padding: const EdgeInsets.only(top: 10.0,left: 5),

child: RaisedButton.icon(onPressed: (){

Navigator.*push*(context, MaterialPageRoute(builder: (context)=> SelectLaptop(

map : map

)));

}, icon: const Icon(Icons.*compare\_arrows*,color: AppColours.*green*,), label: const Text("Compare", style: TextStyle(color: AppColours.*green*),), color: AppColours.*blue*,),

)

],

),

),

],

),

),);}}

**Enter\_Details.dart:**

import 'dart:async';

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:fluttertoast/fluttertoast.dart';

import 'package:geocoding/geocoding.dart' as geocoding;

import 'package:location/location.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/cupertino.dart';

import 'package:flutter/material.dart';

import 'package:google\_maps\_flutter/google\_maps\_flutter.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/pages/profile\_page.dart';

import 'package:wnndderpc/widgets/AppLargeText.dart';

import 'package:wnndderpc/widgets/app\_medtext.dart';

class PlaceOrders extends StatefulWidget {

const PlaceOrders({Key? key}) : super(key: key);

@override

State<PlaceOrders> createState() => \_PlaceOrdersState();

}

class \_PlaceOrdersState extends State<PlaceOrders> {

final Completer<GoogleMapController> \_controller = Completer();

LocationData? currentPos;

LatLng? \_latLng;

bool locating = false;

geocoding.Placemark? place;

int currentStep = 0;

bool isCompleted = false;

bool textFlag = false;

TextEditingController myController = TextEditingController();

TextEditingController myController1 = TextEditingController();

TextEditingController myController2 = TextEditingController();

TextEditingController myController3 = TextEditingController();

TextEditingController myController4 = TextEditingController();

TextEditingController myController5 = TextEditingController();

@override

void dispose() {

myController.dispose();

myController1.dispose();

myController2.dispose();

myController3.dispose();

myController4.dispose();

myController5.dispose();

super.dispose();

}

static const CameraPosition *\_kGooglePlex* = CameraPosition(

target: LatLng(28.644800, 77.216721),

zoom: 14.476,

);

@override

void initState() {

\_getUserLocation();

super.initState();

}

Future<LocationData> \_getLocationPermission() async {

Location location = Location();

bool \_serviceEnabled;

PermissionStatus \_permissionGranted;

LocationData \_locationData;

\_serviceEnabled = await location.serviceEnabled();

if (!\_serviceEnabled) {

\_serviceEnabled = await location.requestService();

if (!\_serviceEnabled) {

return Future.error("Service Not Enabled");

}

}

\_permissionGranted = await location.hasPermission();

if (\_permissionGranted == PermissionStatus.denied) {

\_permissionGranted = await location.requestPermission();

if (\_permissionGranted != PermissionStatus.granted) {

return Future.error("Permission Denied");

}

}

\_locationData = await location.getLocation();

return \_locationData;

}

\_getUserLocation() async {

currentPos = await \_getLocationPermission();

\_goToCurrentPos(LatLng(currentPos!.latitude!, currentPos!.longitude!));

}*//address Location convert*

getUserAddress()

async {

List<geocoding.Placemark> placemarks = await geocoding.placemarkFromCoordinates(\_latLng!.latitude, \_latLng!.longitude);

setState(() {

place = placemarks.first;

});

}*//latitude,longitude*

final user = FirebaseAuth.*instance*.currentUser!;

@override

Widget build(BuildContext context) {

final Size size = MediaQuery.*of*(context).size;

return Scaffold(

backgroundColor: AppColours.*black*,

appBar: AppBar(

leading: GestureDetector(

onTap: () {

Navigator.*of*(context).pop();

},

child: const Icon(

Icons.*arrow\_back\_ios*,

color: Colors.*white*,

size: 30,

)),

elevation: 0,

backgroundColor: AppColours.*black*,

centerTitle: true,

title: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

Center(

child: app\_large\_text(

text: "Enter Details",

size: 18,

color: AppColours.*green*,

FontFamily: "Astro",

)),

GestureDetector(

onTap: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) => const ProfilePage()));

},

child: CircleAvatar(

radius: 20,

backgroundImage: NetworkImage(user.photoURL!),

),

),

],

),

),

body: Theme(

data: ThemeData(

canvasColor: AppColours.*black*,

accentColor: AppColours.*green*,

backgroundColor: AppColours.*green*,

colorScheme: const ColorScheme.light(

primary: AppColours.*green*,

secondary: AppColours.*black*

),

),

child: Scrollbar(

isAlwaysShown: true,

child: Stepper(

elevation: 0,

type: StepperType.horizontal,

steps: getSteps(),

controlsBuilder:(BuildContext context, ControlsDetails controls) {

return Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

const SizedBox(height: 250,),

if(currentStep!= 0)

Container(

color: AppColours.*green*,

width: size.width\*0.42,

child: TextButton(

onPressed: controls.onStepCancel,

child: const Text('Cancel',style: TextStyle(fontSize: 20,color: AppColours.*blue*)),

),

),

const SizedBox(width: 5,),

Container(

color: AppColours.*green*,

width: currentStep==0? size.width \* 0.8 : size.width\*0.42,

child: TextButton(

onPressed:controls.onStepContinue,

child: const Text('Next',style: TextStyle(fontSize: 20,color: AppColours.*blue*),),

),

),

],

);

},

currentStep: currentStep,

onStepContinue: (){

final isLastStep = currentStep ==getSteps().length - 1;

if(isLastStep)

{

setState(() => isCompleted = true);

}

else {

setState(() {

myController1.text.isNotEmpty || myController.text.isNotEmpty || myController3.text.isNotEmpty||

myController4.text.isNotEmpty|| myController5.text.isNotEmpty? currentStep += 1:

Fluttertoast.*showToast*(

msg: "Enter Address To Proceed",

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.BOTTOM,

backgroundColor: Colors.*white*,

textColor: AppColours.*black*,

fontSize: 16.0

);

});

}

},

onStepCancel: (){

setState(() {

currentStep == 0? null: currentStep -= 1;

});

},

),

),

)

);

}

List<Step> getSteps() =>[

Step(

state: currentStep == 1? StepState.complete:StepState.indexed,

isActive: currentStep >= 0,

content: Column(

children: [

SizedBox(

width: MediaQuery.*of*(context).size.width,

height: MediaQuery.*of*(context).size.height \* 0.35,

child: Stack(

children: [

GoogleMap(

myLocationButtonEnabled: true,

myLocationEnabled: true,

mapType: MapType.terrain,

initialCameraPosition: *\_kGooglePlex*,

onMapCreated: (GoogleMapController controller) {

\_controller.complete(controller);

},

onCameraIdle: (){

locating = false;

},

onCameraMove: (CameraPosition position) {

setState(() {

locating = true;

\_latLng = position.target;

});

getUserAddress();

}),

const Align(

alignment: Alignment.*center*,

child: Icon(

Icons.*my\_location*,

size: 20,

color: Colors.*orangeAccent*,

)),

],

),

),

app\_large\_text(text: \_latLng == null? "Locating...." : place == null? " " : place!.country! , color: AppColours.*green*,size: 12,),

*//Text(locating ?'locating....': place!.locality!),*

*// Text(place?.name == null ? "" : place!.name!),*

*// Text(place?.thoroughfare == null? "" : place!.thoroughfare!),*

*// Text(\_latLng?.latitude == null?"" : \_latLng!.latitude.toString()),*

*// Text(\_latLng?.longitude == null?"" : \_latLng!.longitude.toString())*

TextFormField(

controller: myController,

maxLength: 20,

decoration: const InputDecoration(

icon: Icon(Icons.*location\_history\_rounded*,color: AppColours.*green*,),

hintText: "Address Line 1",

fillColor: AppColours.*green*,

filled: true,

hintStyle: TextStyle(color: AppColours.*blue*),

border: OutlineInputBorder(

borderSide: BorderSide(width: 2),

borderRadius: BorderRadius.all(

Radius.circular(10),

)

)

),

),

const SizedBox(height: 20,),

TextFormField(

controller: myController1,

maxLength: 20,

decoration: const InputDecoration(

icon: Icon(Icons.*location\_history\_rounded*,color: AppColours.*green*,),

hintText: "Address Line 2",

fillColor: AppColours.*green*,

filled: true,

hintStyle: TextStyle(color: AppColours.*blue*),

border: OutlineInputBorder(

borderSide: BorderSide(width: 2),

borderRadius: BorderRadius.all(

Radius.circular(10),

)

)

),

),

const SizedBox(height: 10,),

TextFormField(

controller: myController2,

keyboardType: const TextInputType.numberWithOptions(),

maxLength: 20,

decoration: const InputDecoration(

icon: Icon(Icons.*local\_post\_office*,color: AppColours.*green*,),

hintText: "Postal Code",

fillColor: AppColours.*green*,

filled: true,

hintStyle: TextStyle(color: AppColours.*blue*),

border: OutlineInputBorder(

borderSide: BorderSide(width: 2),

borderRadius: BorderRadius.all(

Radius.circular(10),

)

)

),

),

],

), title: const Text("Address",style: TextStyle(color: AppColours.*green*),),),*//gmaps*

Step(

state: currentStep == 2? StepState.complete:StepState.indexed,

isActive: currentStep >= 1,

title: const Text("Details",style: TextStyle(color: AppColours.*green*)), content: Column(

children: [

TextFormField(

controller: myController3,

keyboardType: TextInputType.*name*,

maxLength: 20,

decoration: const InputDecoration(

icon: Icon(Icons.*account\_circle\_rounded*,color: AppColours.*green*,),

hintText: "Name Of Receiver",

fillColor: AppColours.*green*,

filled: true,

hintStyle: TextStyle(color: AppColours.*blue*),

border: OutlineInputBorder(

borderSide: BorderSide(width: 2),

borderRadius: BorderRadius.all(

Radius.circular(10),

)

)

),

),

const SizedBox(height: 10,),

TextFormField(

controller: myController4,

keyboardType: const TextInputType.numberWithOptions(),

maxLength: 20,

decoration: const InputDecoration(

icon: Icon(Icons.*phone*,color: AppColours.*green*,),

hintText: "Phone Number",

fillColor: AppColours.*green*,

filled: true,

hintStyle: TextStyle(color: AppColours.*blue*),

border: OutlineInputBorder(

borderSide: BorderSide(width: 2),

borderRadius: BorderRadius.all(

Radius.circular(10),

)

)

),

),

const SizedBox(height: 10,),

TextFormField(

controller: myController5,

keyboardType: const TextInputType.numberWithOptions(),

maxLength: 20,

decoration: const InputDecoration(

icon: Icon(Icons.*timer*,color: AppColours.*green*,),

hintText: "Schedule Delivery Time",

fillColor: AppColours.*green*,

filled: true,

hintStyle: TextStyle(color: AppColours.*blue*),

border: OutlineInputBorder(

borderSide: BorderSide(width: 2),

borderRadius: BorderRadius.all(

Radius.circular(10),

))),),

SizedBox(height: 100,)

],

)),

Step(

state: currentStep == 3? StepState.complete:StepState.indexed,

isActive: currentStep >= 2,

title: const Text("Complete Order",style: TextStyle(color: AppColours.*green*)), content: Column(

children: [

app\_large\_text(text: "Confirm Order Details", color: AppColours.*green*, FontFamily: "Astro",size: 20,),

const SizedBox(height: 30,),

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

app\_med\_text(text: "Name: ", color: AppColours.*green*,size: 15,),

app\_med\_text(text: myController3.text, color: AppColours.*green*,size: 15,),

],),

const SizedBox(height: 30,),

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

app\_med\_text(text: "Phone Number: ", color: AppColours.*green*,size: 15,),

app\_med\_text(text: myController4.text, color: AppColours.*green*,size: 15,),

],),

const SizedBox(height: 30,),

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

app\_med\_text(text: "Address: ", color: AppColours.*green*,size: 15,),

app\_med\_text(text: myController.text+"\n"+myController1.text+"\n"+myController2.text, color: AppColours.*green*,size: 15,),],),

const SizedBox(height: 80,),

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

app\_med\_text(text: "Exact Location: ", color: AppColours.*green*,size: 15,),

app\_med\_text(text: \_latLng == null? "0":\_latLng!.longitude.toString()+"\n"+\_latLng!.latitude.toString(), color: AppColours.*green*,size: 15,),

],),

const SizedBox(height: 30,),

], )) ];

Future<void> \_goToCurrentPos(LatLng latLng) async {

final GoogleMapController controller = await \_controller.future;

controller.animateCamera(CameraUpdate.*newCameraPosition*(CameraPosition(

bearing: 192.8334901395799,

target: LatLng(latLng.latitude, latLng.longitude),

*//tilt: 59.440717697143555,*

zoom: 18)));

}

}

**pc\_prod.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/pcnavpages/cartpc.dart';

import 'package:wnndderpc/widgets/AppLargeText.dart';

import '../ContentPages/pc.dart';

import '../pcnavpages/comparepc.dart';

class PCProductPage extends StatefulWidget {

String image, grcard, mem, memtype, proccesor, ram, name, stars, price;

PCProductPage(

{Key? key,

required this.name,

required this.stars,

required this.image,

required this.grcard,

required this.mem,

required this.memtype,

required this.proccesor,

required this.ram,

required this.price})

: super(key: key);

@override

\_ProductPageState createState() => \_ProductPageState();

}

const Color transparent = Color(0x00000000);

class \_ProductPageState extends State<PCProductPage> {

Future addCart() async {

final FirebaseAuth \_auth = FirebaseAuth.*instance*;

var currentUser = \_auth.currentUser;

CollectionReference \_collectionRef =

FirebaseFirestore.*instance*.collection("pc-users-cart-items");

return \_collectionRef

.doc(currentUser!.email)

.collection("items")

.doc()

.set({

"name": widget.name,

"price": widget.price,

"images": widget.image,

});

}

@override

Widget build(BuildContext context) {

Size size = MediaQuery.*of*(context).size;

return Scaffold(

body: SizedBox(

width: double.*maxFinite*,

height: double.*maxFinite*,

child: Stack(

children: [

Positioned(

top: -20,

left: 0,

right: 0,

child: Container(

width: size.width \* 0.5,

height: 350,

decoration: BoxDecoration(

image: DecorationImage(

scale: 2.0,

image: NetworkImage(widget.image),

fit: BoxFit.scaleDown),

),

)),

Positioned(

left: 10,

top: 40,

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

IconButton(

onPressed: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) => const MainPC()));

},

icon: const Icon(Icons.*arrow\_back\_ios*),

color: Colors.*black*,

),

SizedBox(

width: size.width \* 0.73,

),

IconButton(

onPressed: () {

Navigator.*push*(

context,

MaterialPageRoute(

builder: (context) => const CartPC()));

},

icon: const Icon(Icons.*shopping\_cart*),

color: Colors.*black*,

),

],

)),

Positioned(

top: size.height \* 0.35,

child: Container(

padding: const EdgeInsets.only(left: 20, right: 20, top: 40),

width: MediaQuery.*of*(context).size.width,

height: 700,

decoration: const BoxDecoration(

color: AppColours.black,

borderRadius: BorderRadius.only(

topLeft: Radius.circular(60),

topRight: Radius.circular(60))),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

Text(

widget.name.substring(0, 19) +

" \n" +

widget.name.substring(19),

textAlign: TextAlign.start,

style: TextStyle(

fontFamily: "Astro",

fontSize: 20,

color: Colors.white.withOpacity(0.8)),

),

],

),

const SizedBox(

height: 10,

),

app\_large\_text(

text: widget.price.toString().length > 5? '\u{20B9}' + widget.price.toString()[0] + "," + widget.price.toString().substring(1,3) + "," + widget.price.toString().substring(3)

: '\u{20B9}' + widget.price.toString().substring(0,2) + "," + widget.price.toString().substring(2,),

color: Colors.white70,

size: 22,

),

const SizedBox(

height: 10,

),

Wrap(

children: List.generate(5, (index) {

return Icon(Icons.star,

color: index < int.parse(widget.stars)

? Colors.yellow

: Colors.grey);

}),

),

const SizedBox(

height: 10,

),

Center(

child: app\_large\_text(

text: "Specifications",

color: Colors.grey,

size: 20,

)),

const Divider(

color: Colors.white70,

height: 20,

),

Padding(

padding: const EdgeInsets.only(left: 8.0, right: 8),

child: Center(

child: Column(

children: [

Row(

mainAxisAlignment:

MainAxisAlignment.spaceBetween,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Text(

"Graphics Card: ",

textAlign: TextAlign.start,

style: TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

Text(

widget.grcard,

textAlign: TextAlign.start,

style: const TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

],

),

const SizedBox(

height: 8,

),

Row(

mainAxisAlignment:

MainAxisAlignment.spaceBetween,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Text(

"Memory: ",

textAlign: TextAlign.start,

style: TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

Text(

widget.mem,

textAlign: TextAlign.end,

style: const TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

],

),

const SizedBox(

height: 8,

),

Row(

mainAxisAlignment:

MainAxisAlignment.spaceBetween,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Text(

"Memory Type: ",

textAlign: TextAlign.start,

style: TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

Text(

widget.memtype,

textAlign: TextAlign.start,

style: const TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

],

),

const SizedBox(

height: 8,

),

Row(

mainAxisAlignment:

MainAxisAlignment.spaceBetween,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Text(

"Processor: ",

textAlign: TextAlign.start,

style: TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

Text(

widget.proccesor,

textAlign: TextAlign.start,

style: const TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

],

),

const SizedBox(

height: 8,

),

Row(

mainAxisAlignment:

MainAxisAlignment.spaceBetween,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Text(

"RAM: ",

textAlign: TextAlign.start,

style: TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

Text(

widget.ram,

textAlign: TextAlign.start,

style: const TextStyle(

color: Colors.white,

fontFamily: "LatoBold"),

),

],

),

const SizedBox(

height: 8,

),

const SizedBox(

height: 20,

),

],

),

),

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

GestureDetector(

onTap: () {

addCart();

final snackbar = SnackBar(

behavior: SnackBarBehavior.floating,

margin: const EdgeInsets.only(bottom: 10),

backgroundColor: Colors.white,

content: const Text(

"Added To Cart!",

style: TextStyle(color: AppColours.blue),

),

action: SnackBarAction(

label: "View Cart",

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => const CartPC()));

},

),

);

ScaffoldMessenger.of(context)

.showSnackBar(snackbar);

},

child: Container(

margin: EdgeInsets.zero,

height: size.height \* 0.07,

width: size.width \* 0.4,

decoration: const BoxDecoration(

color: AppColours.green,

),

child: Row(

crossAxisAlignment: CrossAxisAlignment.center,

mainAxisAlignment: MainAxisAlignment.center,

children: const [

Text(

"Add To Cart",

style: TextStyle(

fontFamily: "Astro", fontSize: 15),

),

SizedBox(

width: 10,

),

Icon(

Icons.shopping\_cart,

)

],

),

),

),

const SizedBox(width: 10,),

GestureDetector(

onTap: (){

Navigator.push(context, MaterialPageRoute(builder: (context) => ComparePC(

image: widget.image,

grcard: widget.grcard,

mem: widget.mem,

memtype: widget.memtype,

proccesor: widget.proccesor,

ram: widget.ram,

stars:widget.stars,

name: widget.name,

price: widget.price.toString().length > 5? '\u{20B9}' + widget.price.toString()[0] + "," + widget.price.toString().substring(1,3) + "," + widget.price.toString().substring(3)

: '\u{20B9}' + widget.price.toString().substring(0,2) + "," + widget.price.toString().substring(2,),

)));

},

child: Container(

margin: EdgeInsets.zero,

height: size.height \* 0.07,

width: size.width \* 0.4,

decoration: const BoxDecoration(

color: AppColours.green,

),

child: Row(

crossAxisAlignment: CrossAxisAlignment.center,

mainAxisAlignment: MainAxisAlignment.center,

children: const [

Text(

"Compare",

style: TextStyle(

fontFamily: "Astro", fontSize: 15),

),

SizedBox(

width: 10,

),

Icon(

Icons.compare\_arrows,

)

],

),

),

),

],

)

],

),

)),

],

),

),

);

}}

**cartpc.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/pcnavpages/pcs.dart';

import 'package:wnndderpc/widgets/app\_medtext.dart';

import '../ContentPages/pc.dart';

import '../pages/placeorder.dart';

import '../widgets/AppLargeText.dart';

class CartPC extends StatefulWidget {

const CartPC({Key? key,}) : super(key: key);

@override

\_CartPageState createState() => \_CartPageState();

}

class \_CartPageState extends State<CartPC> {

var total;

@override

Widget build(BuildContext context) {

final Size size = MediaQuery

.*of*(context)

.size;

return Scaffold(

backgroundColor: AppColours.*black*,

body: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Padding(

padding: const EdgeInsets.only(top: 50, left: 10, right: 125),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

crossAxisAlignment: CrossAxisAlignment.end,

children: [

GestureDetector(

child: const Icon(

Icons.*arrow\_back\_ios*, color: Colors.*orangeAccent*,),

onTap: () {

Navigator.*push*(context, MaterialPageRoute(

builder: (context) => const MainPC()));

},

),

Center(child: app\_large\_text(text: "Your Cart",

size: 22,

color: Colors.*orangeAccent*,

FontFamily: "Astro",)),

],

),

),

StreamBuilder(

stream: FirebaseFirestore.*instance*.collection("pc-users-cart-items")

.doc(FirebaseAuth.*instance*.currentUser!.email).collection(

"items")

.snapshots(),

builder: (BuildContext context,

AsyncSnapshot<QuerySnapshot> snapshot) {

if (!snapshot.hasData) {

return const Center(child: CircularProgressIndicator(),);

}

else {

return Expanded(

child: ListView.builder(

scrollDirection: Axis.vertical,

shrinkWrap: true,

itemCount: snapshot.data!.docs.length,

physics: const BouncingScrollPhysics(),

cacheExtent: 200.0,

itemExtent: 180,

itemBuilder: (context, index) {

DocumentSnapshot \_documentSnapshot = snapshot.data!

.docs[index];

return Card(

margin: const EdgeInsets.only(

left: 10, right: 10, top: 10, bottom: 10),

color: Colors.*grey*[900],

elevation: 2,

child: Container(

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10),

),

child: Row(

children: [

Container(

margin: const EdgeInsets.only(left: 10),

width: size.width \* 0.3,

height: size.height \* 0.1,

decoration: const BoxDecoration(

),

child: Image(

fit: BoxFit.scaleDown,

image: NetworkImage(

\_documentSnapshot['images']),

),

),

const SizedBox(width: 10,),

Column(

crossAxisAlignment: CrossAxisAlignment

.start,

mainAxisAlignment: MainAxisAlignment.end,

children: [

Padding(

padding: const EdgeInsets.only(

left: 8.0),

child: Text(

\_documentSnapshot['name']

.toString().substring(0,18),

textAlign: TextAlign.start,

style: const TextStyle(

fontFamily: "Astro",

fontSize: 12,

fontWeight: FontWeight.*bold*,

color: Colors.*white*),

),

),

const SizedBox(height: 10,),

Text(

" \u{20B9}" + \_documentSnapshot['price'],

textAlign: TextAlign.start,

style: const TextStyle(

fontFamily: "Lato",

fontWeight: FontWeight.*bold*,

color: Colors.*white*),

),

Padding(

padding: const EdgeInsets.only(

left: 180.0, top: 20,),

child: IconButton(iconSize: 20,

icon: const Icon(Icons.*delete*),

color: Colors.*orangeAccent*,

onPressed: () {

final snackbar = SnackBar(

duration: const Duration(seconds: 2),

behavior: SnackBarBehavior

.floating,

margin: const EdgeInsets.only(

bottom: 10),

backgroundColor: Colors.*white*,

content: const Text(

"Removed From Cart!",

style: TextStyle(

color: AppColours.*blue*),),

action: SnackBarAction(

label: "Dismiss",

onPressed: () {

ScaffoldMessenger.*of*(context)

.hideCurrentSnackBar();

},

),);

FirebaseFirestore.*instance*

.collection("pc-users-cart-items")

.doc(FirebaseAuth.*instance*

.currentUser!.email).collection(

"items").doc(

\_documentSnapshot.id)

.delete();

ScaffoldMessenger.*of*(context)

.showSnackBar(snackbar);

},),

)

],

),

],

),

),

);

}),

);

}

}

),

Card(

color: Colors.*black12*,

margin: EdgeInsets.*zero*,

elevation: 2,

child: Padding(

padding: const EdgeInsets.only(

bottom: 80.0, left: 10, right: 10, top: 10),

child: Column(

children: [

Row(mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

app\_large\_text(text: "Total Price:",

color: Colors.*orangeAccent*,

FontFamily: "Astro",

size: 14,),

StreamBuilder(

stream: FirebaseFirestore.*instance*.collection("pc-users-cart-items").

doc(FirebaseAuth.*instance*.currentUser!.email).collection("items").snapshots(),

builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {

if (snapshot.hasError) {

return const Text("Somethong went Wrong");

}

if (snapshot.connectionState == ConnectionState.waiting) {

return const CircularProgressIndicator();

} else {

total = 0;

snapshot.data?.docs.forEach((element) {

total += int.*parse*(element['price']);

});

return app\_med\_text(

text: '\u{20B9}'+ total.toString(),

color: Colors.*orangeAccent*,

size: 14,);

}

},

),

],

),

const SizedBox(

height: 20,

),

GestureDetector(

onTap: (){

Navigator.*push*(context, MaterialPageRoute(builder: (context) => const PlaceOrders()));

},

child: Container(

width: size.width,

height: size.height \* 0.05,

color: Colors.*orangeAccent*,

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Icon(

Icons.*shopping\_cart*,

color: Colors.*black87*,

size: 30,

),

const SizedBox(

width: 10,

),

app\_med\_text(

text: "Place Order",

color: Colors.*black87*,

FontFamily: "Astro",

size: 20,

)

],

),

),

)],

),

),

),

],

),

);

}

}

**profilepage.dart:**

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:wnndderpc/colors.dart';

class ProfilePage extends StatelessWidget {

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

Widget textField({required String hintText, required String gfield}){

return Material(

elevation:4,

color: Colors.black45,

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),

child: TextField(

decoration: InputDecoration(

hintText: hintText + " " + gfield,

hintStyle: const TextStyle(

letterSpacing: 1,

color: AppColours.green,

fontWeight: FontWeight.bold,

fontFamily: "Astro",

fontSize: 12

),

fillColor: AppColours.black,

filled: true ,

),

),

);

}

@override

Widget build(BuildContext context) {

final Size size = MediaQuery.of(context).size;

final user = FirebaseAuth.instance.currentUser!;

return Scaffold(

backgroundColor: AppColours.black,

appBar: AppBar(

elevation: 0,

backgroundColor: const Color(0xff555555),

leading: IconButton(

icon: const Icon(Icons.arrow\_back\_ios, color: Colors.white,),

iconSize: 20

,onPressed: () {

Navigator.of(context).pop();

},

),

),

body: Stack(

alignment: Alignment.center,

children: [

Column(

mainAxisAlignment: MainAxisAlignment.end,

children: [

Container(

height: size.height \* .55,

width: double.infinity,

margin: const EdgeInsets.symmetric(horizontal: 10),

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

textField(hintText: "User Name:" , gfield: user.displayName!),

textField(hintText: "Email-ID : ", gfield: user.email!),

textField(hintText: "Email Verification : ", gfield: user.emailVerified? "Verified" : "Not Verified")

],

),

)

],

),

CustomPaint(

child: SizedBox(

width: size.width,

height: size.height,

),

painter: HeaderCurvedCOntainer()

),

Column(

crossAxisAlignment: CrossAxisAlignment.center,

children: [

const Padding(padding: EdgeInsets.all(20),

child: Text(

"Profile",

style: TextStyle(

fontSize: 35,fontFamily: "Astro",

color: AppColours.green

),

),),

Container(

width: size.width / 3,

height: size.width/3,

decoration: BoxDecoration(

border: Border.all(color: AppColours.green,width: 3 ),

shape: BoxShape.circle,

image: DecorationImage(

image: NetworkImage(user.photoURL!),

fit: BoxFit.fitWidth

)

),

)

],

)

],

),

);

}

}

class HeaderCurvedCOntainer extends CustomPainter{

@override

void paint(Canvas canvas, Size size) {

Paint paint = Paint()..color = const Color(0xff555555);

Path path = Path()..relativeLineTo(0, 150)

..quadraticBezierTo(size.width/2, 225, size.width, 150)

..relativeLineTo(0, -150)

..close();

canvas.drawPath(path, paint);

}

@override

bool shouldRepaint(covariant CustomPainter oldDelegate)=> false;

}

**main.dart:**

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import 'package:wnndderpc/colors.dart';

import 'package:wnndderpc/google\_sign\_in.dart';

import 'package:wnndderpc/pages/welcome\_page.dart';

import 'package:firebase\_core/firebase\_core.dart';

Future<void> main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp();

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

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

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) {

return ChangeNotifierProvider(

create: (context) => GoogleSignInProvider(),

child: MaterialApp(

title: "WNNDERPC",

debugShowCheckedModeBanner: false,

home: const Welcome\_page(),

theme: ThemeData(

primaryColor: AppColours.green),),); }}

**TESTING**

| **Errors found while testing:** | **Status** |
| --- | --- |
| 1.Animations getting paused while long pressing in home screen | Fixed |
| 2.Scrollbar very laggy and glitchy | Fixed |
| 3.Loading animation speed too slow | Fixed |
| 4.Compare Page start up errors | Not fixed |
| 5.Sign Up Page loading problems | Fixed |
| 6.Firebase images retrieval | Fixed |
| 7.Compare page doesn't send second laptop information | Not fixed |
| 8.Geolocation widget latitude longitude errors | Fixed |
| 9.App start up problems | Fixed |
| 10. UI/UX elements misaligned | Fixed |

**Security measures taken**

**Account Verification:**

The application uses Firebase Google Sign in methods to verify the user’s identity. There is no password authentication as it requires the user to keep logging in with their email-id and password every time they open the application which is redundant and time consuming.

The android device already has their google accounts signed in. So the user can easily just sign in and verify their identity without any extra hassle.

**Limitations and Future Enhancements**

**Limitations of the applications are:**

* The prices of the products in the app are not realtime. Meaning prices will have to be manually updated in the database.
* The data in the database isn’t very safe as we have changed a setting where users can both read and write data into the database.
* The application’s UI/UX elements might be misaligned on phones with different dimensions.
* The application isn’t built for IOS applications.
* The application does not store the addresses of the user while placing an order.

**Future Enhancements:**

* One of the main things the application needs to enhance would be the PC Parts module.
* Working on clean and reusable code to reduce the length of the code, which inturn makes the code more efficient.
* Working on reducing the size of the app as it requires around 78 MB, which is a lot for an ecommerce application.

**Bibliography**

**Links referred to:**

* geeksforgeeks.org
* stackoverflow.com

**Libraries imported from:**

* pub.dev
* stackoverflow.com

**Youtube Channels referred to:**

* freeCodeCamp.org
* The Net Ninja
* Flutter
* Johannes Mike