

# MAD and PWA Lab

## Experiment - 6

**Name: Anuj Chitari**

**Class: D15A Roll no:04**

**Aim: To Connect Flutter UI with FireBase database**

**Theory:**

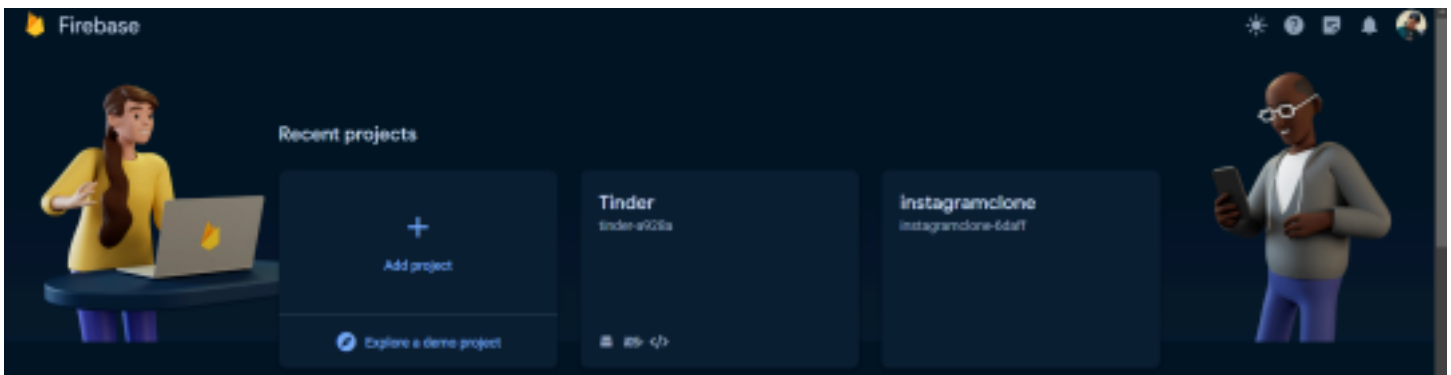
### Prerequisites

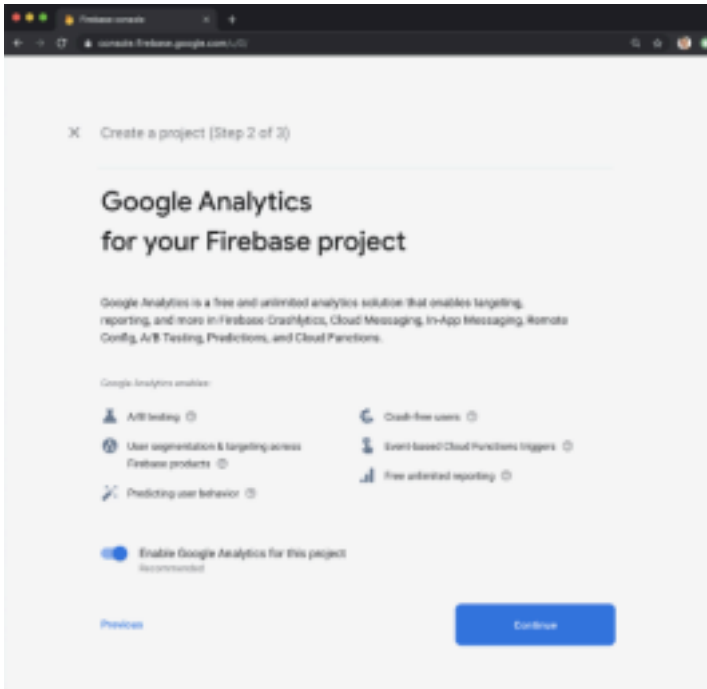
To complete this tutorial, you will need:

- A Google account to use Firebase.
- Developing for iOS will require XCode.
- To download and install Flutter.
- To download and install Android Studio and Visual Studio Code.
- It is recommended to install plugins for your code editor:
  - [Flutter](#) and [Dart](#) plugins installed for Android Studio.
  - [Flutter](#) extension installed for Visual Studio Code.

**Create a Firebase Project:**

First, log in with your Google account to manage your Firebase projects. From within the Firebase dashboard, select the Create new project button and give it a name:





Go to the Firebase Console and create a new project.  
Add your Flutter app to the Firebase project:

Register your app in the Firebase project, and follow the instructions to download the configuration files (google-services.json for Android, GoogleService-Info.plist for iOS).

The most important thing here is to match up the Android package name that you choose here with the one inside of our application.

The structure consists of at least two segments. A common pattern is to use a domain name, a company name, and the application name:

com.example.flutterfirebaseexample

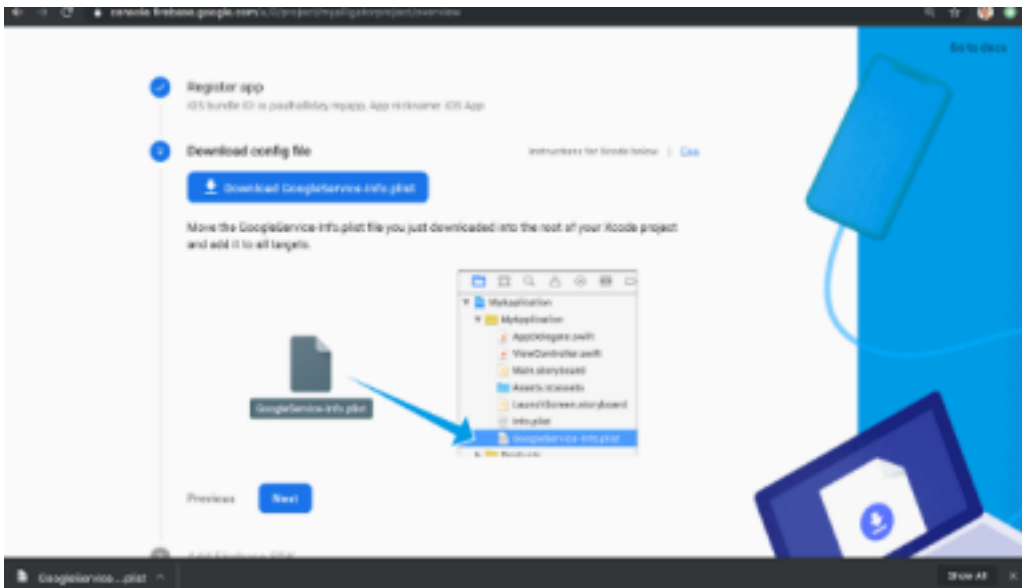
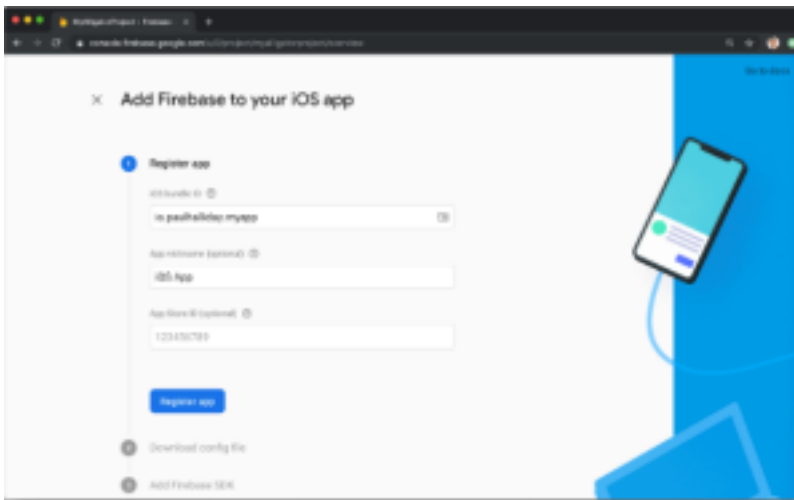
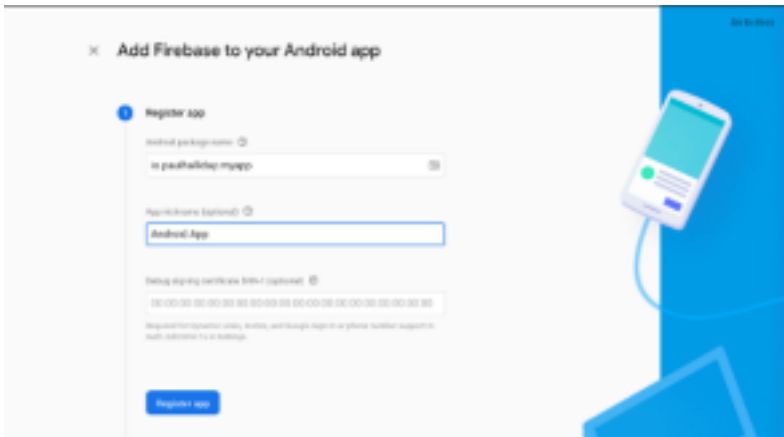
Once you've decided on a name, open android/app/build.gradle in your code editor and update the applicationId to match the Android package name:

android/app/build.gradle

```
...
defaultConfig {
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
    applicationId 'com.example.flutterfirebaseexample'
    ...
}
...
```

## Downloading the Config File

The next step is to add the Firebase configuration file into our Flutter project. This is important as it contains the API keys and other critical information for Firebase to use.  
Select Download `google-services.json` from this page:



2. Add Firebase to your Flutter project:  
Add Dependencies:

Open your pubspec.yaml file and add the necessary dependencies:  
yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  curved_navigation_bar: ^1.0.3 #latest version  
  # The following adds the Cupertino Icons font to your application.  
  # Use with the CupertinoIcons class for iOS style icons.  
  cupertino_icons: ^1.0.2  
  firebase_core: ^2.25.4  
  firebase_auth: ^4.17.4  
  cloud_firestore: ^4.15.4  
  image_picker: ^1.0.7  
  firebase_storage: ^11.6.5  
  shared_preferences: ^2.2.2  
  random_string: ^2.3.1
```

Code:

Signup.dart

```
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:food_panda/pages/bottomnav.dart';  
import 'package:food_panda/pages/login.dart';  
import 'package:food_panda/widget/widget_support.dart';
```

```
class SignUp extends StatefulWidget {  
  const SignUp({super.key});  
  
  @override  
  State<SignUp> createState() => _SignUpState();  
}
```

```
class _SignUpState extends State<SignUp> {  
  String email = "", password = "", name = "";
```

```
  TextEditingController namecontroller = new TextEditingController();
```

```
TextEditingController passwordcontroller = new TextEditingController();
```

```
TextEditingController mailcontroller = new TextEditingController();
```

```
final _formkey = GlobalKey<FormState>();
```

```
registration() async {  
  if (password != null) {  
    try {  
      UserCredential userCredential = await FirebaseAuth.instance  
        .createUserWithEmailAndPassword(email: email, password: password);
```

```
      ScaffoldMessenger.of(context).showSnackBar((SnackBar(  
        backgroundColor: Colors.redAccent,  
        content: Text(  
          "Registered Successfully",  
          style: TextStyle(fontSize: 20.0),  
        ))));
```

```
      // ignore: use_build_context_synchronously
```

```
      Navigator.pushReplacement(  
        context, MaterialPageRoute(builder: (context) => BottomNav()));
```

```
    } on FirebaseException catch (e) {
```

```
      if (e.code == 'weak-password') {  
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(  
          backgroundColor: Colors.orangeAccent,  
          content: Text(  
            "Password Provided is too Weak",  
            style: TextStyle(fontSize: 18.0),  
          )));
```

```
      } else if (e.code == "email-already-in-use") {  
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(  
          backgroundColor: Colors.orangeAccent,  
          content: Text(  
            "Account Already exists",  
            style: TextStyle(fontSize: 18.0),  
          )));
```

```
      }
```

```
    }
```

```
  }
```

```
}
```

```
@override
```

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

```

return Scaffold(
  body: Container(
    child: Stack(
      children: [
        Container(
          width: MediaQuery.of(context).size.width,
          height: MediaQuery.of(context).size.height / 2.5,
          decoration: BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topLeft,
              end: Alignment.bottomRight,
              colors: [
                Color(0xFF000000),
                Color(0xFF000000),
              ])),
        ),
        Container(
          margin:
            EdgeInsets.only(top: MediaQuery.of(context).size.height / 3),
          height: MediaQuery.of(context).size.height / 2,
          width: MediaQuery.of(context).size.width,
          decoration: BoxDecoration(
            color: Color.fromARGB(255, 18, 17, 17),
            borderRadius: BorderRadius.only(
              topLeft: Radius.circular(40),
              topRight: Radius.circular(40))),
          child: Text(""),
        ),
        Container(
          margin: EdgeInsets.only(top: 60.0, left: 20.0, right: 20.0),
          child: Column(
            children: [
              Center(
                child: Image.asset(
                  "images/logo copy.png",
                  width: MediaQuery.of(context).size.width / 1.5,
                  fit: BoxFit.cover,
                )),
              SizedBox(
                height: 50.0,
              ),
              Material(
                elevation: 5.0,
                borderRadius: BorderRadius.circular(20),
                child: Container(

```

```

padding: EdgeInsets.only(left: 20.0, right: 20.0),
width: MediaQuery.of(context).size.width,
height: MediaQuery.of(context).size.height / 1.8,
decoration: BoxDecoration(
  color: const Color.fromARGB(255, 255, 255, 255),
  borderRadius: BorderRadius.circular(20)),
child: Form(
  key: _formkey,
  child: Column(
    children: [
      SizedBox(
        height: 30.0,
      ),
      Text(
        "Sign up",
        style: AppWidget.HeadlineTextFieldStyle(),
      ),
      SizedBox(
        height: 30.0,
      ),
      TextFormField(
        controller: namecontroller,
        validator: (value) {
          if (value == null || value.isEmpty) {
            return 'Please Enter Name';
          }
          return null;
        },
        decoration: InputDecoration(
          hintText: 'Name',
          hintStyle: AppWidget.semiBoldTextFieldStyle(),
          prefixIcon: Icon(Icons.person_outlined)),
      ),
      SizedBox(
        height: 30.0,
      ),
      TextFormField(
        controller: mailcontroller,
        validator: (value) {
          if (value == null || value.isEmpty) {
            return 'Please Enter E-mail';
          }
          return null;
        },
        decoration: InputDecoration(

```

```

        hintText: 'Email',
        hintStyle: AppWidget.semiBoldTextFieldStyle(),
        prefixIcon: Icon(Icons.email_outlined)),
    ),
    SizedBox(
      height: 30.0,
    ),
    TextFormField(
      controller: passwordcontroller,
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please Enter Password';
        }
        return null;
      },
      obscureText: true,
      decoration: InputDecoration(
        hintText: 'Password',
        hintStyle: AppWidget.semiBoldTextFieldStyle(),
        prefixIcon: Icon(Icons.password_outlined)),
    ),
    SizedBox(
      height: 80.0,
    ),
    GestureDetector(
      onTap: () async {
        if (_formkey.currentState!.validate()) {
          setState(() {
            email = mailcontroller.text;
            name = namecontroller.text;
            password = passwordcontroller.text;
          });
        }
        registration();
      },
      child: Material(
        elevation: 5.0,
        borderRadius: BorderRadius.circular(20),
        child: Container(
          padding: EdgeInsets.symmetric(vertical: 8.0),
          width: 200,
          decoration: BoxDecoration(
            color: Color.fromARGB(255, 26, 14, 10),
            borderRadius: BorderRadius.circular(20)),
          child: Center(

```






# Uber Eats

## Sign up

 Name

 Email

\*\*\* Password

**SIGN UP**

Already have an account? [Login](#)



# Uber Eats

## Login

 Email

\*\*\* Password

[Forgot Password?](#)

**LOGIN**

Don't have an account? [Sign up](#)



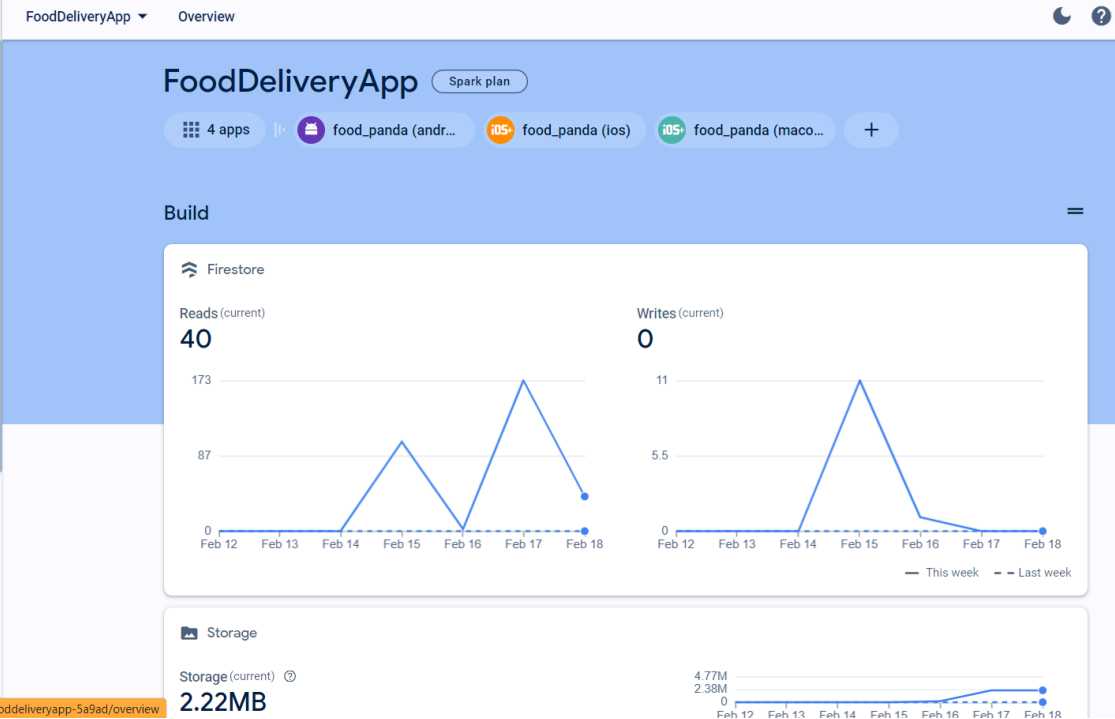
# Authentication

Search by email address, phone number, or user UID

Add user

| Identifier                 | Providers | Created ↓    | Signed In    | User UID                     |
|----------------------------|-----------|--------------|--------------|------------------------------|
| aishwariyachitari@gmail... |           | Feb 17, 2024 | Feb 17, 2024 | imENPunnWTeZC9FSamEf2yz...   |
| hgharsh2003@gmail.co...    |           | Feb 13, 2024 | Feb 13, 2024 | OGDw1MDgWGR0urelMRI0cO...    |
| 2021.sarvadnya.awagh...    |           | Feb 13, 2024 | Feb 13, 2024 | McQPhrK7HIY5VLtCcME0Is3...   |
| abhishekfatate5@gmail...   |           | Feb 13, 2024 | Feb 13, 2024 | VBo64CK52tgFescrILJZh7Aiz... |
| atharvachavan089@gm...     |           | Feb 12, 2024 | Feb 12, 2024 | OylcFmXaBFceIYJmoJSI5wee...  |
| chitarianuj@gmail.com      |           | Feb 11, 2024 | Feb 19, 2024 | m047z7r9wPfofxD0AMGjnUa...   |

Rows per page: 50 1 - 6 of 6 < >



Storage

Storage (current) ⓘ  
2.22MB

4.77M

2.38M

0

Feb 12

Feb 13

Feb 14

Feb 15

Feb 16

Feb 17

Feb 18

bddeliveryapp-5a9ad/overview

## 2nd DB part

### add\_food.dart

```
import 'dart:io';

import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/material.dart';
import 'package:food_panda/service/database.dart';
import 'package:food_panda/widget/widget_support.dart';
import 'package:image_picker/image_picker.dart';
import 'package:random_string/random_string.dart';
// import 'package:sample_flutter/widget/widget_support.dart';

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

  @override
  State<AddFood> createState() => _AddFoodState();
}

class _AddFoodState extends State<AddFood> {
  final List<String> fooditems = ['Ice-cream', 'Burger', 'Salad', 'Pizza'];
  String? value;
  TextEditingController namecontroller = new TextEditingController();
  TextEditingController pricecontroller = new TextEditingController();
  TextEditingController detailcontroller = new TextEditingController();
  final ImagePicker _picker = ImagePicker();
  File? selectedImage;

  Future getImage() async {
    var image = await _picker.pickImage(source: ImageSource.gallery);
    selectedImage = File(image!.path);
    setState(() {});
  }

  uploadItem() async {
    if (selectedImage != null &&
        namecontroller.text != "" &&
        pricecontroller.text != "" &&
        detailcontroller.text != "") {
      String addId = randomAlphaNumeric(10);
      Reference firebaseStorageRef =
```

```

        FirebaseStorage.instance.ref().child("blogImages").child(addId);
final UploadTask task = firebaseStorageRef.putFile(selectedImage!);

var downloadUrl = await (await task).ref.getDownloadURL();

Map<String, dynamic> addItem = {
  "Image": downloadUrl,
  "Name": namecontroller.text,
  "Price": pricecontroller.text,
  "Detail": detailcontroller.text
};
await DatabaseMethods().addFoodItem(addItem, value!).then((value) {
  ScaffoldMessenger.of(context).showSnackBar((SnackBar(
    backgroundColor: Colors.redAccent,
    content: Text(
      "Food item added successfully",
      style: TextStyle(fontSize: 20.0),
    )),
  ));
});
}
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: GestureDetector(
        onTap: () {
          Navigator.pop(context);
        },
        child: Icon(
          Icons.arrow_back_ios_new_outlined,
          color: Color(0xFF373866),
        )),
    centerTitle: true,
    title: Text(
      "Add Item",
      style: AppWidget.HeadlineTextFieldStyle(),
    ),
  ),
  body: SingleChildScrollView(
    child: Container(
      margin:
        EdgeInsets.only(left: 20.0, right: 20.0, top: 20.0, bottom: 50.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(

```

```

"Upload the Item Picture",
style: AppWidget.semiBoldTextFieldStyle(),
),
SizedBox(
  height: 20.0,
),
selectedImage == null
  ? GestureDetector(
    onTap: () {
      getImage();
    },
    child: Center(
      child: Material(
        elevation: 4.0,
        borderRadius: BorderRadius.circular(20),
        child: Container(
          width: 150,
          height: 150,
          decoration: BoxDecoration(
            border:
              Border.all(color: Colors.black, width: 1.5),
            borderRadius: BorderRadius.circular(20),
          ),
          child: Icon(
            Icons.camera_alt_outlined,
            color: Colors.black,
          ),
        ),
      ),
    ),
  ),
)
: Center(
  child: Material(
    elevation: 4.0,
    borderRadius: BorderRadius.circular(20),
    child: Container(
      width: 150,
      height: 150,
      decoration: BoxDecoration(
        border: Border.all(color: Colors.black, width: 1.5),
        borderRadius: BorderRadius.circular(20),
      ),
      child: Image.file(
        selectedImage!,
        fit: BoxFit.cover,
      ),
    ),
  ),
),
),

```

```

    ),
    SizedBox(
      height: 30.0,
    ),
    Text(
      "Item Name",
      style: AppWidget.semiBoldTextFieldStyle(),
    ),
    SizedBox(
      height: 10.0,
    ),
    Container(
      padding: EdgeInsets.symmetric(horizontal: 20.0),
      width: MediaQuery.of(context).size.width,
      decoration: BoxDecoration(
        color: Color(0xFFecec8),
        borderRadius: BorderRadius.circular(10)),
      child: TextField(
        controller: namecontroller,
        decoration: InputDecoration(
          border: InputBorder.none,
          hintText: "Enter Item Name",
          hintStyle: AppWidget.LightTextFieldStyle()),
      ),
    ),
    SizedBox(
      height: 30.0,
    ),
    Text(
      "Item Price",
      style: AppWidget.semiBoldTextFieldStyle(),
    ),
    SizedBox(
      height: 10.0,
    ),
    Container(
      padding: EdgeInsets.symmetric(horizontal: 20.0),
      width: MediaQuery.of(context).size.width,
      decoration: BoxDecoration(
        color: Color(0xFFecec8),
        borderRadius: BorderRadius.circular(10)),
      child: TextField(
        controller: pricecontroller,
        decoration: InputDecoration(
          border: InputBorder.none,
          hintText: "Enter Item Price",
          hintStyle: AppWidget.LightTextFieldStyle()),
      ),

```

```

),
 SizedBox(
   height: 30.0,
 ),
 Text(
   "Item Detail",
   style: AppWidget.semiBoldTextFieldStyle(),
 ),
 SizedBox(
   height: 10.0,
 ),
 Container(
   padding: EdgeInsets.symmetric(horizontal: 20.0),
   width: MediaQuery.of(context).size.width,
   decoration: BoxDecoration(
     color: Color(0xFFececfc),
     borderRadius: BorderRadius.circular(10)),
   child: TextField(
     maxLines: 6,
     controller: detailcontroller,
     decoration: InputDecoration(
       border: InputBorder.none,
       hintText: "Enter Item Detail",
       hintStyle: AppWidget.LightTextFieldStyle()),
   ),
 ),
 SizedBox(
   height: 20.0,
 ),
 Text(
   "Select Category",
   style: AppWidget.semiBoldTextFieldStyle(),
 ),
 SizedBox(
   height: 20.0,
 ),
 Container(
   padding: EdgeInsets.symmetric(horizontal: 10.0),
   width: MediaQuery.of(context).size.width,
   decoration: BoxDecoration(
     color: Color(0xFFececfc),
     borderRadius: BorderRadius.circular(10)),
   child: DropdownButtonHideUnderline(
     child: DropdownButton<String>(
       items: fooditems
         .map((item) => DropdownMenuItem<String>(
           value: item,
           child: Text(

```



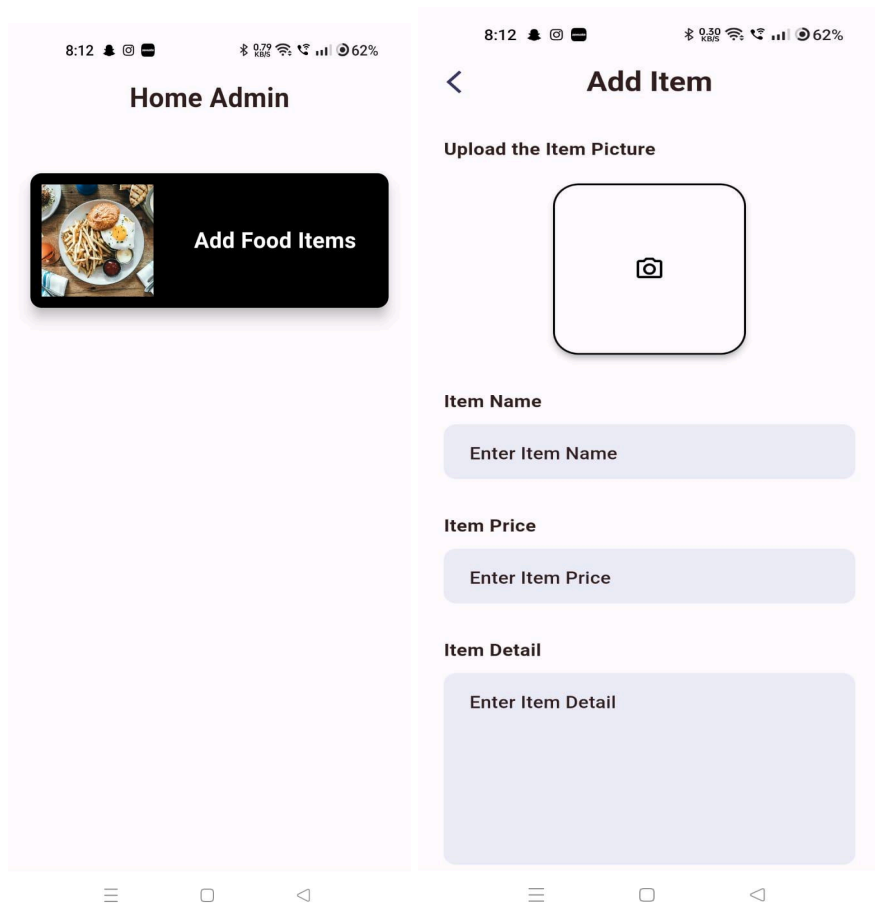
```

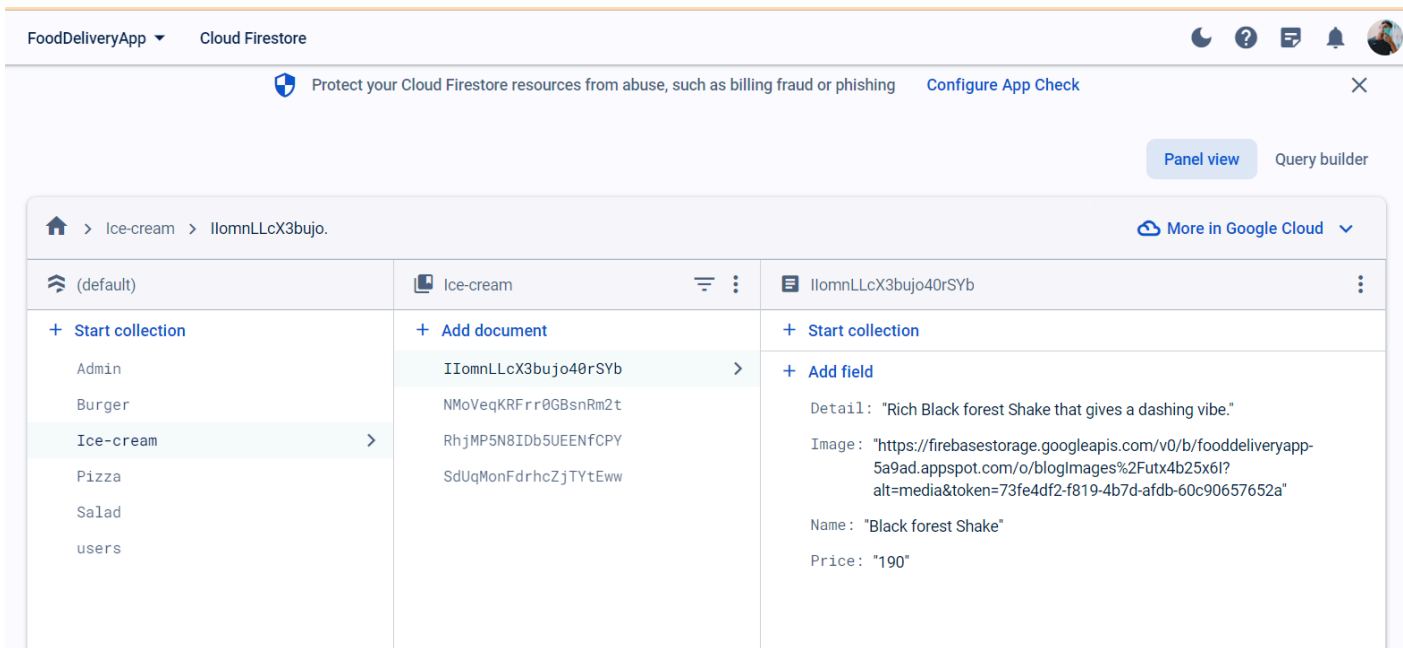
        item,
        style:
            TextStyle(fontSize: 18.0, color: Colors.black),
    )))
    .toList(),
    onChanged: ((value) => setState(() {
      this.value = value;
    })),
    dropdownColor: Colors.white,
    hint: Text("Select Category"),
    iconSize: 36,
    icon: Icon(
      Icons.arrow_drop_down,
      color: Colors.black,
    ),
    value: value,
  )),
),
SizedBox(
  height: 30.0,
),
GestureDetector(
  onTap: () {
    uploadItem();
  },
  child: Center(
    child: Material(
      elevation: 5.0,
      borderRadius: BorderRadius.circular(10),
      child: Container(
        padding: EdgeInsets.symmetric(vertical: 5.0),
        width: 150,
        decoration: BoxDecoration(
          color: Colors.black,
          borderRadius: BorderRadius.circular(10)),
        child: Center(
          child: Text(
            "Add",
            style: TextStyle(
              color: Colors.white,
              fontSize: 22.0,
              fontWeight: FontWeight.bold),
          ),
        ),
      ),
    ),
  ),
),
),
),
),
)

```

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

## OUTPUT:





## home.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:food_panda/pages/details.dart';
import 'package:food_panda/service/database.dart';
import 'package:food_panda/widget/widget_support.dart';
```

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

```
  @override
  State<Home> createState() => _HomeState();
}
```

```
class _HomeState extends State<Home> {
  bool icecream = false, pizza = false, salad = false, burger = false;
  //Syream is used to get all the items in the form of a stream which can be used to handle it easily
  Stream? fooditemStream;
```

```
  ontheLoad() async {
    fooditemStream = await DatabaseMethods().getFoodItem("Pizza");
    setState(() {});
  }
```

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

```

Widget allItemsVertically() {
  return StreamBuilder(
    stream: fooditemStream,
    builder: (context, AsyncSnapshot snapshot) {
      return snapshot.hasData
        ? ListView.builder(
            padding: EdgeInsets.zero,
            itemCount: snapshot.data.docs.length,
            shrinkWrap: true,
            scrollDirection: Axis.vertical,
            itemBuilder: (context, index) {
              DocumentSnapshot ds = snapshot.data.docs[index];
              return GestureDetector(
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) => Details(
                        detail: ds["Detail"],
                        name: ds["Name"],
                        price: ds["Price"],
                        image: ds["Image"],
                      )),
                ),
              ),
            child: Container(
              margin: const EdgeInsets.only(right: 10.0, bottom: 20),
              child: Material(
                elevation: 5.0,
                borderRadius: BorderRadius.circular(20),
                child: Container(
                  padding: const EdgeInsets.all(5),
                  child: Row(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      ClipRRect(
                        borderRadius: BorderRadius.circular(20),
                        child: Image.network(
                          ds["Image"],
                          height: 120,
                          width: 120,
                          fit: BoxFit.cover,
                        ),
                      ),
                      const SizedBox(
                        width: 20.0,
                      ),
                      Column(
                        children: [

```

```

        Container(
          width:
            MediaQuery.of(context).size.width /
              2,
          child: Text(
            ds["Name"],
            style: AppWidget
              .semiBoldTextFieldStyle(),
          )),
        const SizedBox(
          height: 5.0,
        ),
        Container(
          width:
            MediaQuery.of(context).size.width /
              2,
          child: Text(
            "Honey goot cheese",
            style:
              AppWidget.LightTextFieldStyle(),
          )),
        const SizedBox(
          height: 5.0,
        ),
        Container(
          width:
            MediaQuery.of(context).size.width /
              2,
          child: Text(
            "\$" + ds["Price"],
            style: AppWidget
              .semiBoldTextFieldStyle(),
          ))
      ],
    ),
  ],
),
),
),
),
),
);
})
: CircularProgressIndicator();
});
}

```

```

Widget allItems() {
  return StreamBuilder(

```

```

stream: fooditemStream,
builder: (context, AsyncSnapshot snapshot) {
  return snapshot.hasData
    ? ListView.builder(
      padding: EdgeInsets.zero,
      itemCount: snapshot.data.docs.length,
      shrinkWrap: true,
      scrollDirection: Axis.horizontal,
      itemBuilder: (context, index) {
        DocumentSnapshot ds = snapshot.data.docs[index];
        return GestureDetector(
          onTap: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => Details(
                  detail: ds["Detail"],
                  name: ds["Name"],
                  price: ds["Price"],
                  image: ds["Image"],
                )),
            );
          },
        );
      },
    );
  child: Container(
    margin: const EdgeInsets.all(4),
    child: Material(
      elevation: 5.0,
      borderRadius: BorderRadius.circular(20),
      child: Container(
        padding: const EdgeInsets.all(14),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            //ClipRRect , This is used to add border radius to the image
            ClipRRect(
              borderRadius: BorderRadius.circular(20),
              //This gets the image stored in the firebase
              child: Image.network(
                ds["Image"],
                height: 150,
                width: 150,
                fit: BoxFit.cover,
              ),
            ),
            Text(ds["Name"],
              style:
                AppWidget.semiBoldTextFieldStyle()),
            const SizedBox(
              height: 5.0,

```

```

    ),
    Text("Fresh and Healthy",
        style: AppWidget.LightTextFieldStyle()),
    const SizedBox(
        height: 5.0,
    ),
    Text(
        "\$" + ds["Price"],
        style: AppWidget.semiBoldTextFieldStyle(),
    )
  ],
),
),
),
);
})
: CircularProgressIndicator();
});
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      scrollDirection: Axis.vertical,
      child: Container(
        margin: const EdgeInsets.only(top: 50.0, left: 20.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                Text("Hello User,", style: AppWidget.boldTextFieldStyle()),
                Container(
                  margin: const EdgeInsets.only(right: 20.0),
                  padding: const EdgeInsets.all(3),
                  decoration: BoxDecoration(
                    color: Colors.black,
                    borderRadius: BorderRadius.circular(8)),
                  child: const Icon(
                    Icons.shopping_cart_outlined,
                    color: Colors.white,
                  ),
                ),
              ],
            ),
          ],
        ),
      ),
    ),
  );
  const SizedBox(

```

```

        height: 20.0,
      ),
      Text("Delicious Food", style: AppWidget.HeadlineTextFieldStyle()),
      Text("Discover and Get Great Food",
        style: AppWidget.LightTextFieldStyle()),
      const SizedBox(
        height: 20.0,
      ),
      Container(
        margin: const EdgeInsets.only(right: 20.0),
        child: showItem(),
      ),
      const SizedBox(
        height: 30.0,
      ),
      Container(height: 270, child: allItems()),
        const SizedBox(
          height: 30.0,
        ),
      ),
      allItemsVertically(),
      const SizedBox(
        height: 30.0,
      ),
    ],
  ),
),
),
);
}

```

```

Widget showItem() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      GestureDetector(
        onTap: () async {
          icecream = true;
          pizza = false;
          salad = false;
          burger = false;
          fooditemStream = await DatabaseMethods().getFoodItem("Ice-cream");
          setState(() {});
        },
        child: Material(
          elevation: 5.0,
          borderRadius: BorderRadius.circular(10),
          child: Container(
            decoration: BoxDecoration(
              color: icecream ? Colors.black : Colors.white,

```



```

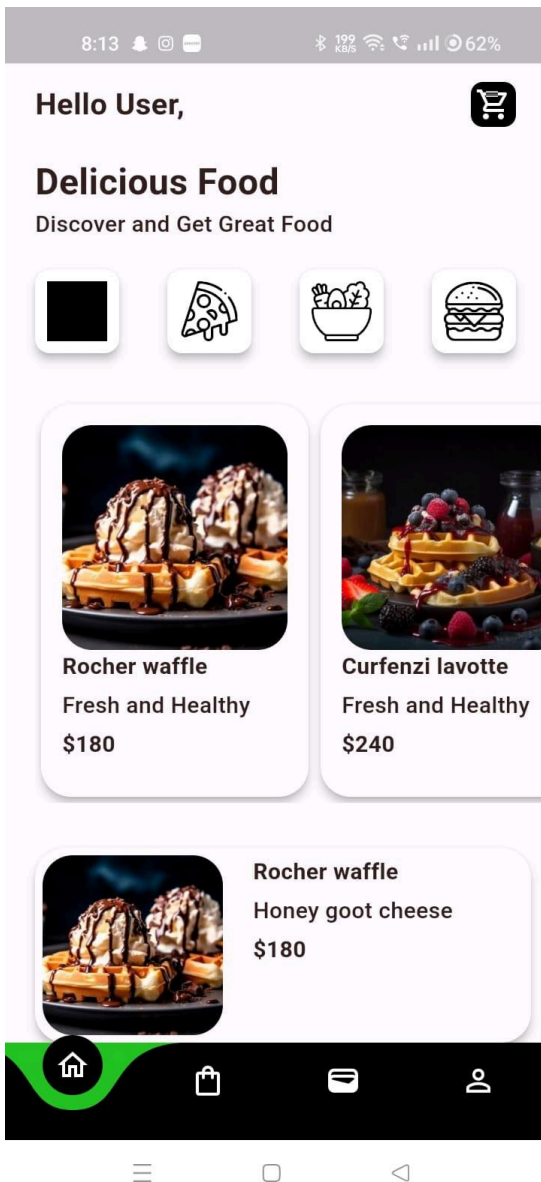
        borderRadius: BorderRadius.circular(10)),
padding: const EdgeInsets.all(8),
child: Image.asset(
  "images/shake.png",
  height: 40,
  width: 40,
  fit: BoxFit.cover,
  color: icecream ? Colors.white : Colors.black,
),
),
),
),
GestureDetector(
  onTap: () async {
    icecream = false;
    pizza = true;
    salad = false;
    burger = false;
    fooditemStream = await DatabaseMethods().getFoodItem("Pizza");
    setState(() {});
  },
  child: Material(
    elevation: 5.0,
    borderRadius: BorderRadius.circular(10),
    child: Container(
      decoration: BoxDecoration(
        color: pizza ? Colors.black : Colors.white,
        borderRadius: BorderRadius.circular(10)),
      padding: const EdgeInsets.all(8),
      child: Image.asset(
        "images/pizza.png",
        height: 40,
        width: 40,
        fit: BoxFit.cover,
        color: pizza ? Colors.white : Colors.black,
      ),
    ),
  ),
),
GestureDetector(
  onTap: () async {
    icecream = false;
    pizza = false;
    salad = true;
    burger = false;
    fooditemStream = await DatabaseMethods().getFoodItem("Salad");
    setState(() {});
  },

```

```

child: Material(
  elevation: 5.0,
  borderRadius: BorderRadius.circular(10),
  child: Container(
    decoration: BoxDecoration(
      color: salad ? Colors.black : Colors.white,
      borderRadius: BorderRadius.circular(10)),
    padding: const EdgeInsets.all(8),
    child: Image.asset(
      "images/salad.png",
      height: 40,
      width: 40,
      fit: BoxFit.cover,
      color: salad ? Colors.white : Colors.black,
    ),
  ),
),
),
GestureDetector(
  onTap: () async {
    icecream = false;
    pizza = false;
    salad = false;
    burger = true;
    fooditemStream = await DatabaseMethods().getFoodItem("Burger");
    setState(() {});
  },
  child: Material(
    elevation: 5.0,
    borderRadius: BorderRadius.circular(10),
    child: Container(
      decoration: BoxDecoration(
        color: burger ? Colors.black : Colors.white,
        borderRadius: BorderRadius.circular(10)),
      padding: const EdgeInsets.all(8),
      child: Image.asset(
        "images/burger.png",
        height: 40,
        width: 40,
        fit: BoxFit.cover,
        color: burger ? Colors.white : Colors.black,
      ),
    ),
  ),
),
],
);

```



## Conclusion:

In this experiment, we have successfully connected firebase database and authenticated using google sign in and email and password without flutter application successfully.