

Name: Devesh Wadhwa

D15A Roll no:68

EXPERIMENT NO: 06

Aim: To Connect Flutter UI with firebase database

Theory:

Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

CODE:

Auth_controller

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:get/get.dart';
import 'package:get/get_state_manager/src/simple/get_controllers.dart';
import 'package:mynta_clone/consts/consts.dart';

class AuthController extends
  GetxController{var isLoading = false.obs;
//text controller
var emailController = TextEditingController();
var passwordController = TextEditingController();

//login method
Future<UserCredential?>loginMethod({context})
  async{UserCredential? userCredential;
    try{
      await auth.signInWithEmailAndPassword(email:emailController.text,password:
passwordController.text);

    } on FirebaseAuthException
      catch(e){ VxToast.show(context,msg:e.toString())
        ;

    }
    return userCredential;
  }
//Signup method
Future<UserCredential?>signupMethod({email,password,context})
  async{UserCredential? userCredential;
    try{
      await auth.createUserWithEmailAndPassword(email:email,password: password);

    } on FirebaseAuthException
      catch(e){ VxToast.show(context,msg:e.toString())
        ;

    }
  }
}
```

```

    }
    return userCredential;

}

//Storing data
storeUserData( {name,password,email}) async {
  DocumentReference store =await
  firestore.collection(userCollection).doc(currentUser!.uid);
  store.set({ 'name':name
    e,
    'password':password,
    'email': email,
    'imageUrl': "",
    'id': currentUser!.uid,
    'cart_count': "00",
    'order_count': "00",
    'wishlist_count': "00",

  });
}

//signout method
signoutMethod(context) async {
  try {
    await auth.signOut();

  } catch(e){ VxToast.show(context,msg:e.toString());
  }
}

```


Edit_profile_screen

```

import 'dart:io';

import 'package:get/get.dart';
import 'package:myntra_clone/controllers/profile_controller.dart';
import 'package:myntra_clone/widgets_common/bg_widget.dart';
import 'package:myntra_clone/widgets_common/custom_textfield.dart';
import 'package:myntra_clone/widgets_common/our_button.dart';

import '../const/const.dart';

class EditProfileScreen extends StatelessWidget {

```

```

final dynamic data;
const EditProfileScreen( {Key ? key,this.data}):super(key:key);

@override
Widget build(BuildContext context) {
  var controller = Get.find<ProfileController>();

  return
    bgWidget(child:
      Scaffold(
        resizeToAvoidBottomInset: false,
        appBar: AppBar(),
        body: Obx(()=>
          Column( mainAxisSize:
            MainAxisSize.min,
            children: [
              //Data Image Url and controller path is empty
              data['imageUrl'] == " " && controller.profileImgPath.isEmpty ?
Image.asset(imgProfile2,width: 100,fit:
BoxFit.cover).box.roundedFull.clip(Clip.antiAlias).make():
              data['imageUrl'] != " " && controller.profileImgPath.isEmpty
? Image.network(data['imageUrl'], width :100,fit: BoxFit.cover,
).box.roundedFull.clip(Clip.antiAlias).make():
              Image.file(File(controller.profileImgPath.value) ,
width :100,
fit: BoxFit.cover,
).box.roundedFull.clip(Clip.antiAlias).make(),

              10.heightBox,
              ourButton(color: buttonColor,onPress: ()
                {controller.changeImage(context);
              },
              textColor: whiteColor,title: "Change"),
              const Divider(),
              20.heightBox,
              customTextField(
                controller: controller.nameController,
                hint: nameHint,title: name,isPass: false),
              customTextField(
                controller: controller.passController,
                hint: password,title: password,isPass: true),
              20.heightBox,
              controller.isloading.value?
                CircularProgressIndicator( valueColor:
                  AlwaysStoppedAnimation(buttonColor) ,
                ):SizedBox( width:context.scre
enWidth-60,
                child: ourButton(color: buttonColor,onPress: ()
                  async {controller.isloading(true);
                  await controller.uploadProfileImage();
                  await controller.updateProfile( imgUrl:
                    controller.profileImageLink,

```

```

        name: controller.nameController.text,
        password: controller.passController.text,
      );
      VxToast.show(context, msg: "Updated");

      ),textColor: whiteColor,title: "Save")),
    ],
  ).box.white.shadowSm.padding(EdgeInsets.all(16)).margin(EdgeInsets.only(top:
50,left: 12,right: 12)).rounded.make(),
  ),
),
);
}
}

```

Profile_Controller

```

import 'dart:io';

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/services.dart';
import 'package:get/get.dart';
import 'package:image_picker/image_picker.dart';
import '../consts/consts.dart';
import 'package:path/path.dart';

class ProfileController extends
  GetxController{var profileImgPath = ".obs;
  var profileImageLink = "";
  var isloading = false.obs;
  var nameController = TextEditingController();
  var passController = TextEditingController();
  changeImage(context)async{
    try {
      final img = await ImagePicker().pickImage( source:
        ImageSource.gallery, imageQuality: 70);
      if(img==null) return;
      profileImgPath.value = img.path;
    } on PlatformException
      catch(e){ VxToast.show(context,msg :
        e.toString());

    }
  }
  uploadProfileImage() async {
    var filename = basename(profileImgPath.value);
    var destination = 'images/${currentUser!.uid}/${filename}';
    Reference ref = FirebaseStorage.instance.ref().child(destination);
    await ref.putFile(File(profileImgPath.value));
  }
}

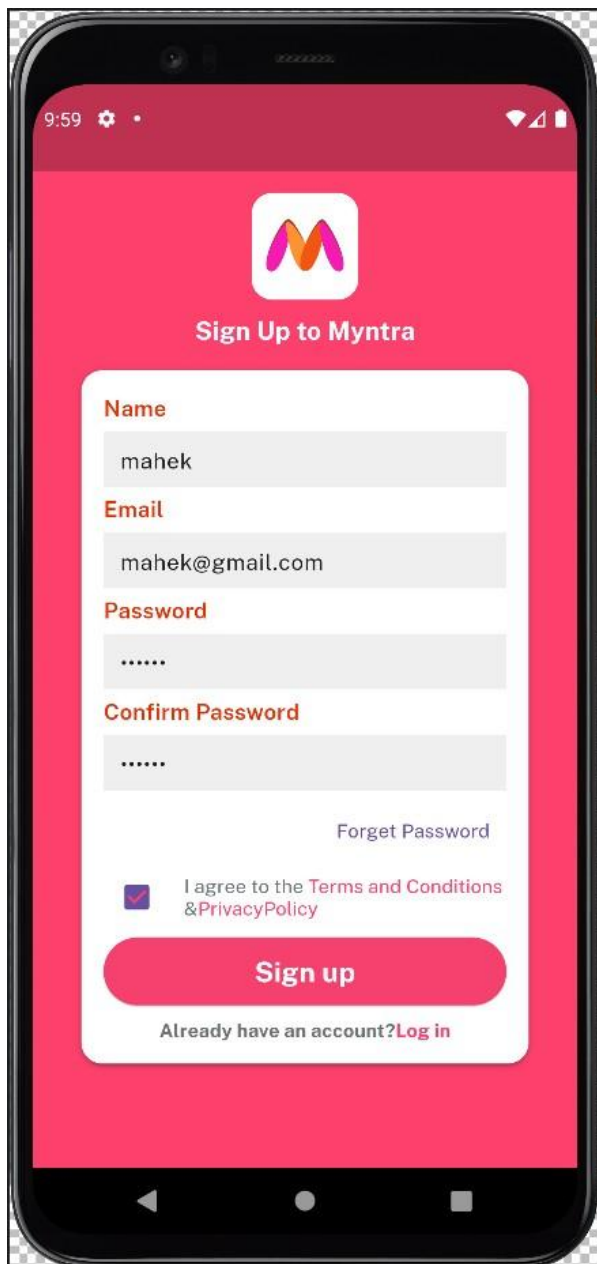
```

```


        profileImageLink = await ref.getDownloadURL();
    }
    updateProfile({name,password,imgUrl}) async{
        var store = firestore.collection(userCollection).doc(currentUser!.uid);
        await store.set({
            'name':name,
            'password':password,
            'imageUrl': imgUrl
        },SetOptions(merge: true));
        isLoading(false);
    }
}

```

Sign up Screen



Login screen



Log in to Meesho

Email

Password




[Forget Password](#)

Log in

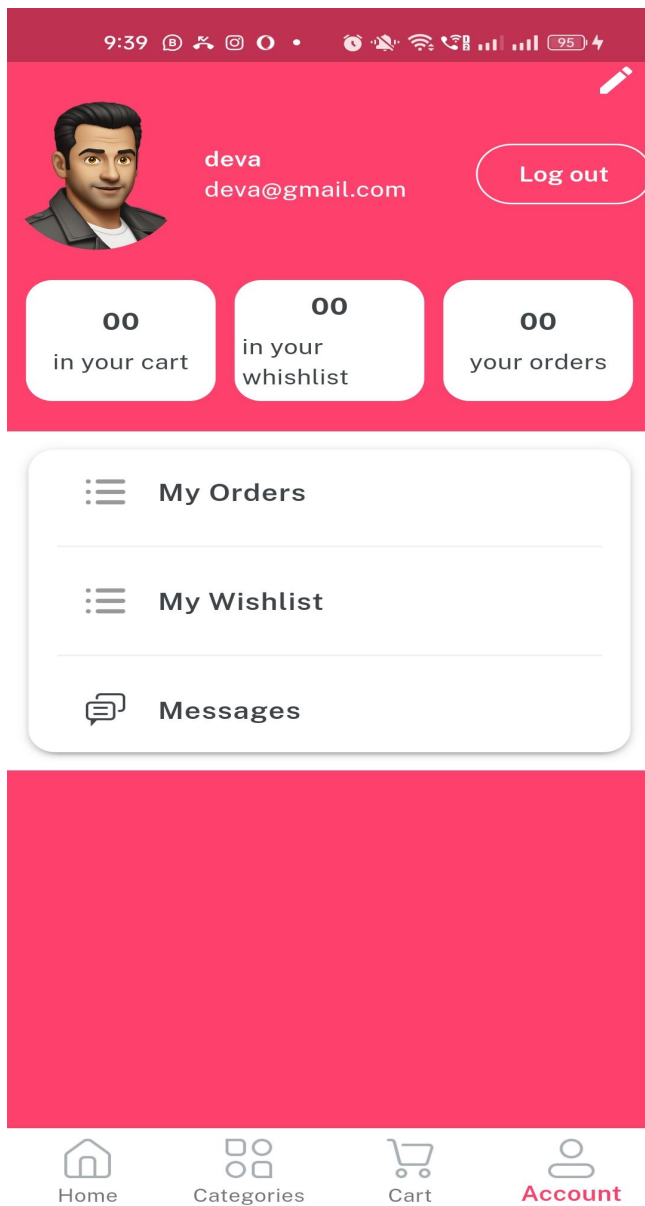
or, create a new account

Sign up

Log in with



Edit Profile Screen



Updated name



Change

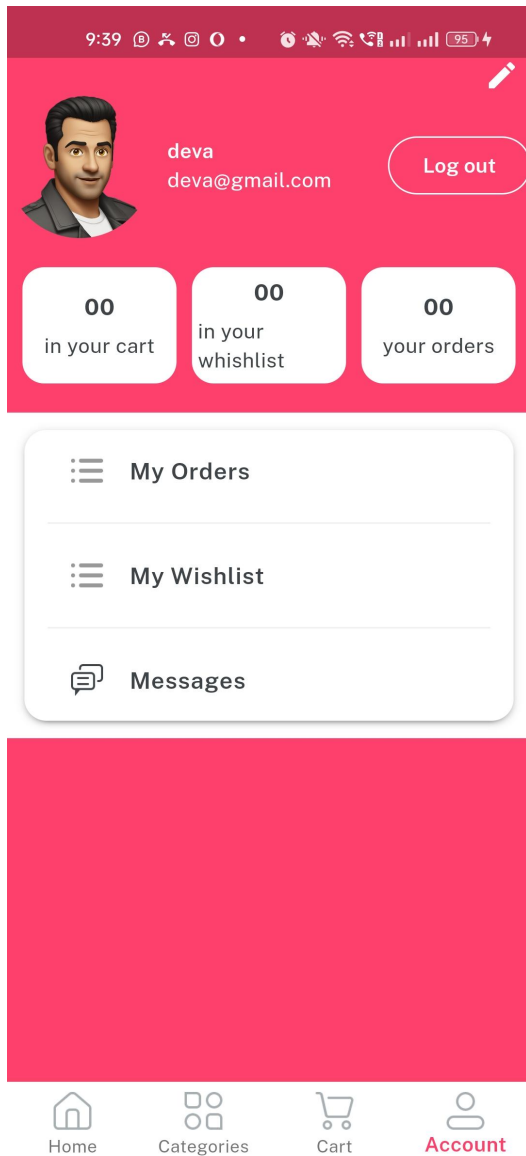
Name

Devesh Wadhwa

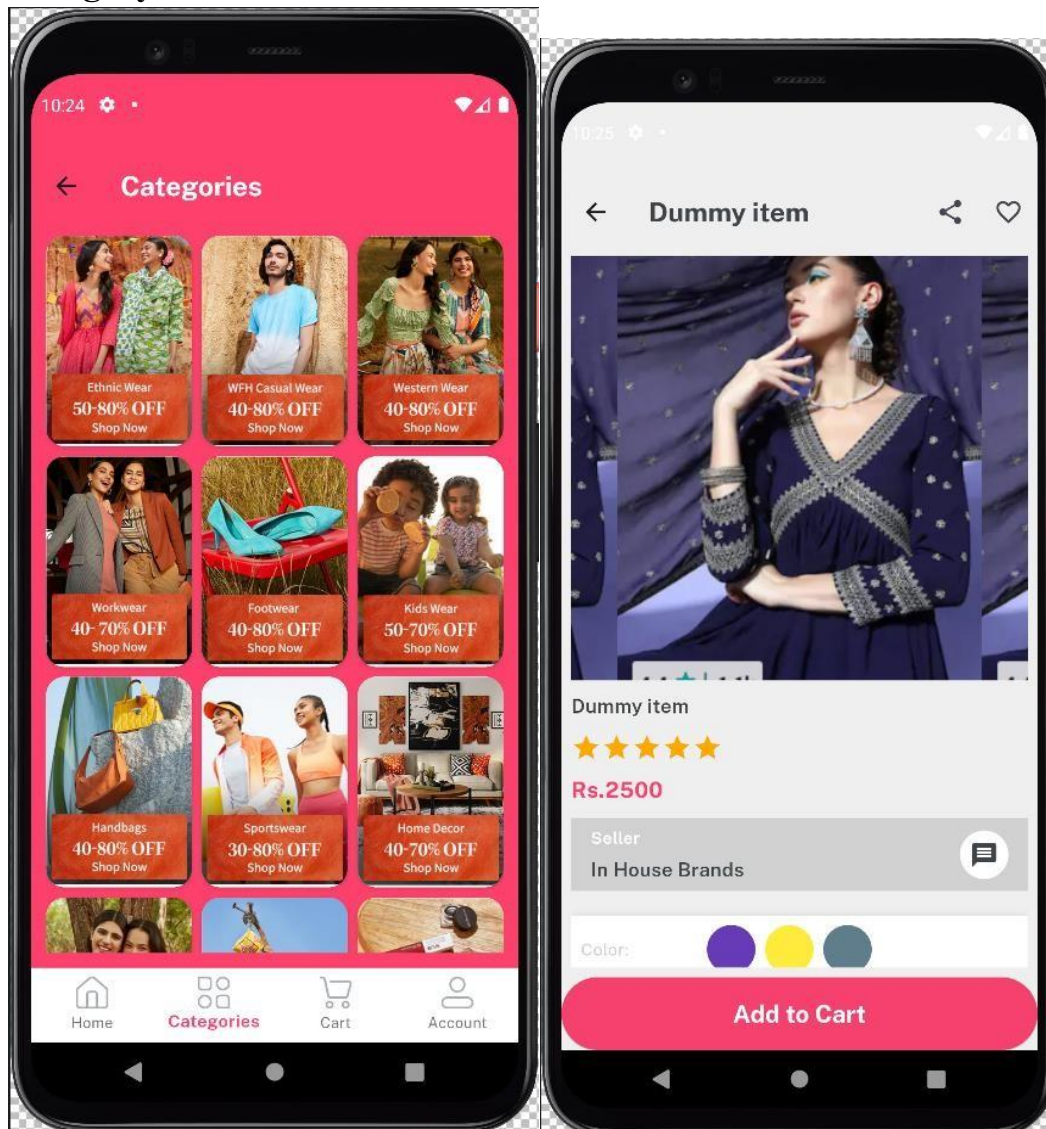
Password

.....

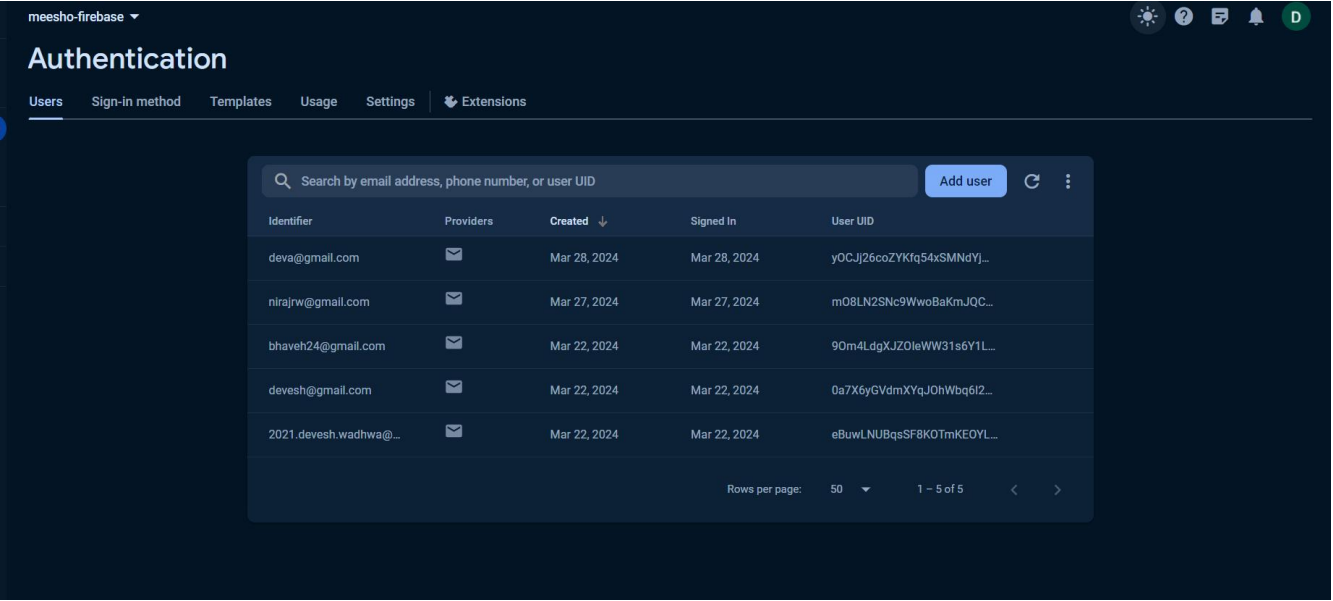
Account is updated



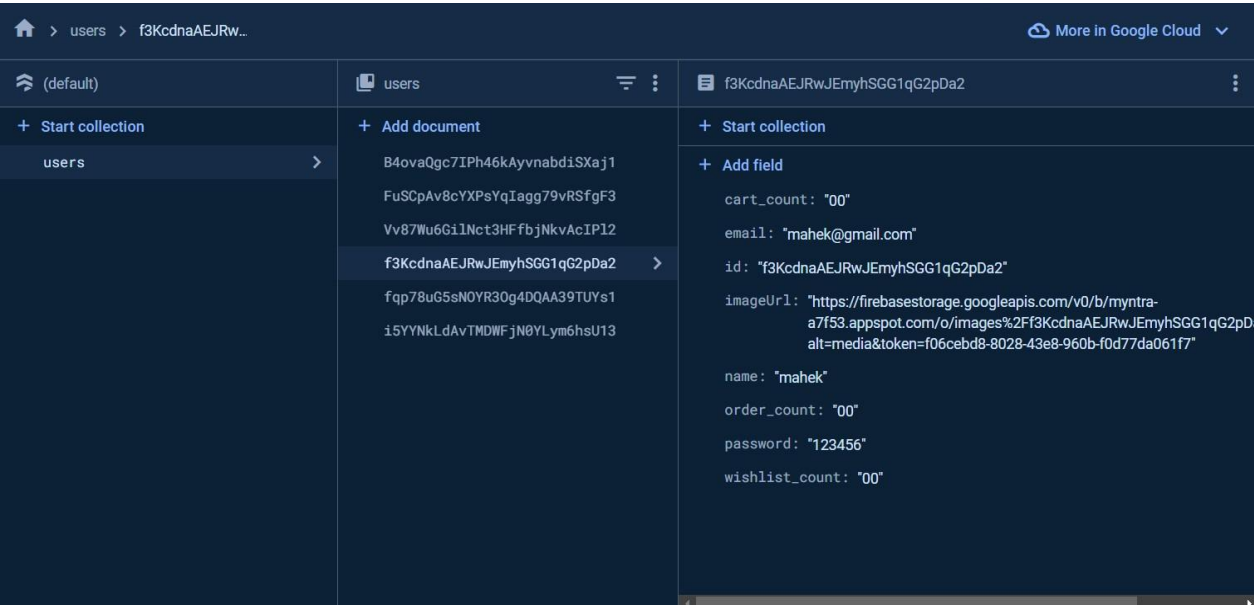
Category



Firebase Authentication



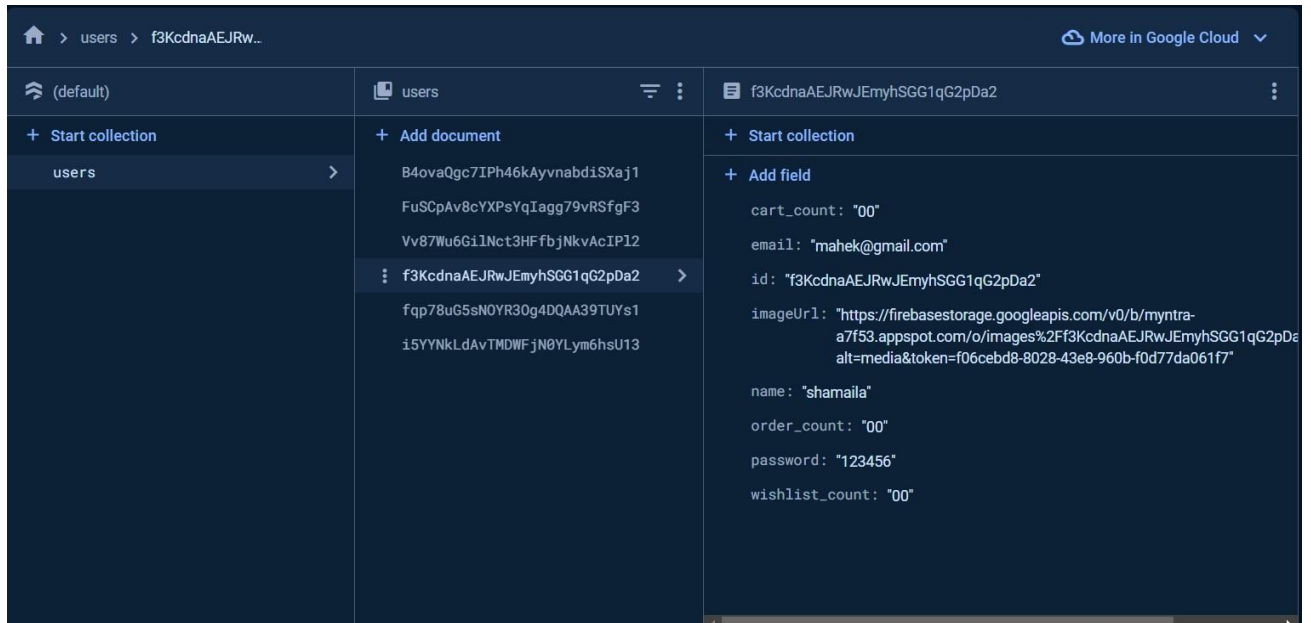
Login Signup Creds



Profile image and name is changed

Mahek → Shamaila

Image url is also updated in Database



Firebase Storage

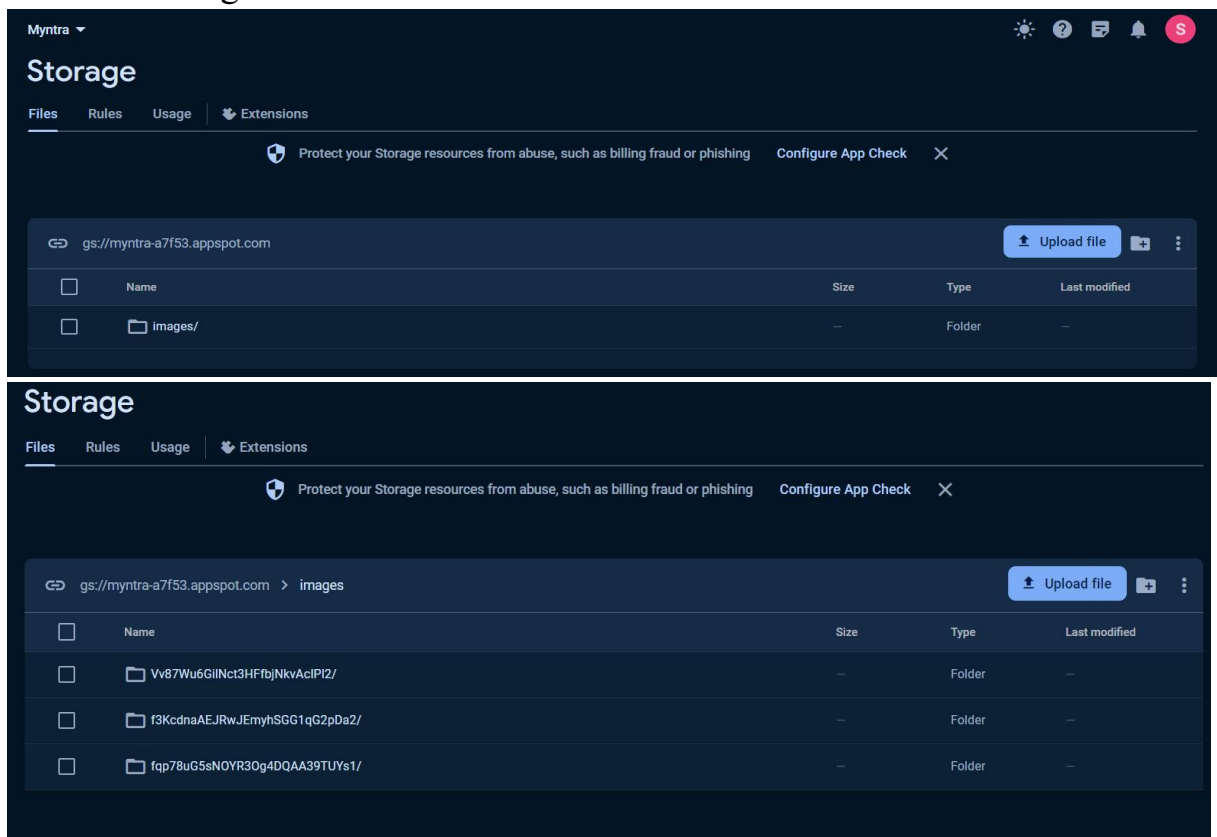
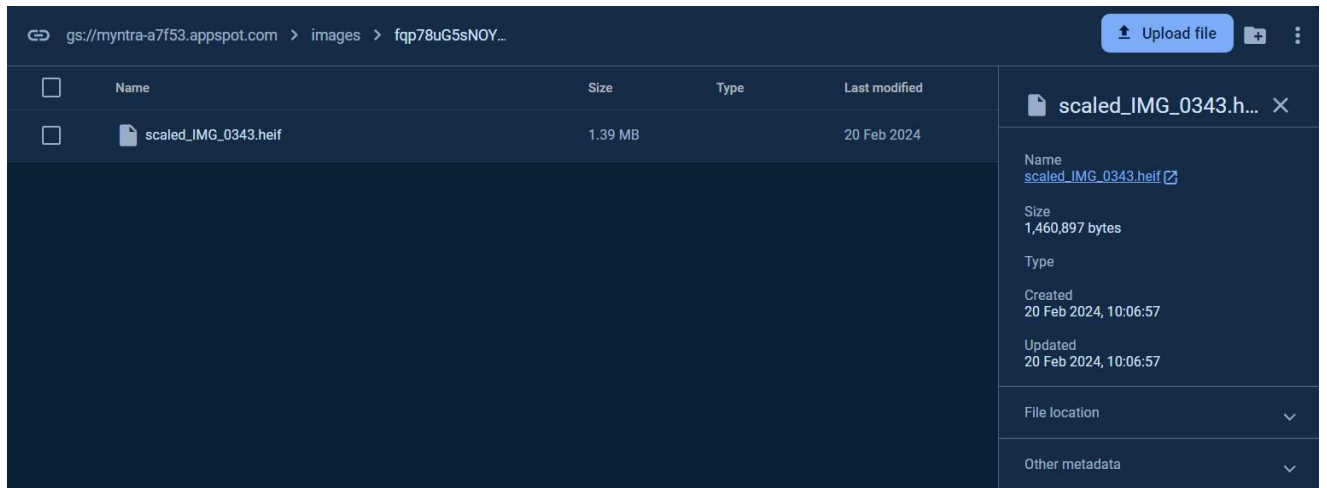
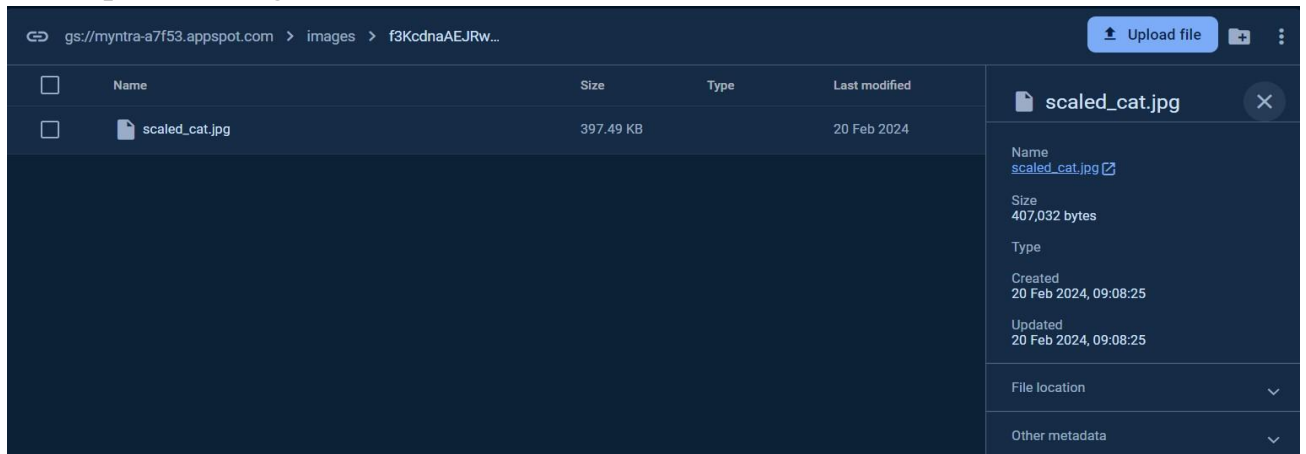


Image is Updated

Cat → profile image of user



Conclusion:

Here, Successfully Connected Flutter UI with Firebase database