Atharva Mulam Batch B
D15A Roll No: 37

Exp No: 6

Aim: To Connect Flutter UI with FireBase database.

Theory:

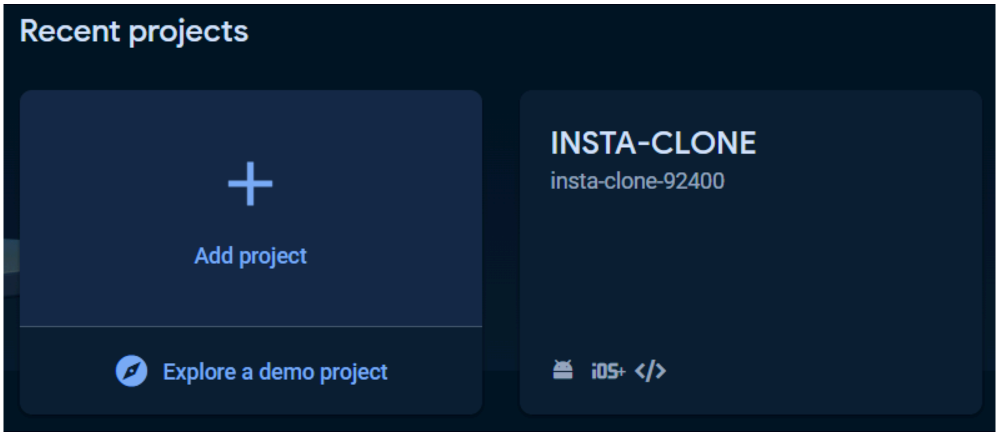
Prerequisites

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.

1)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:

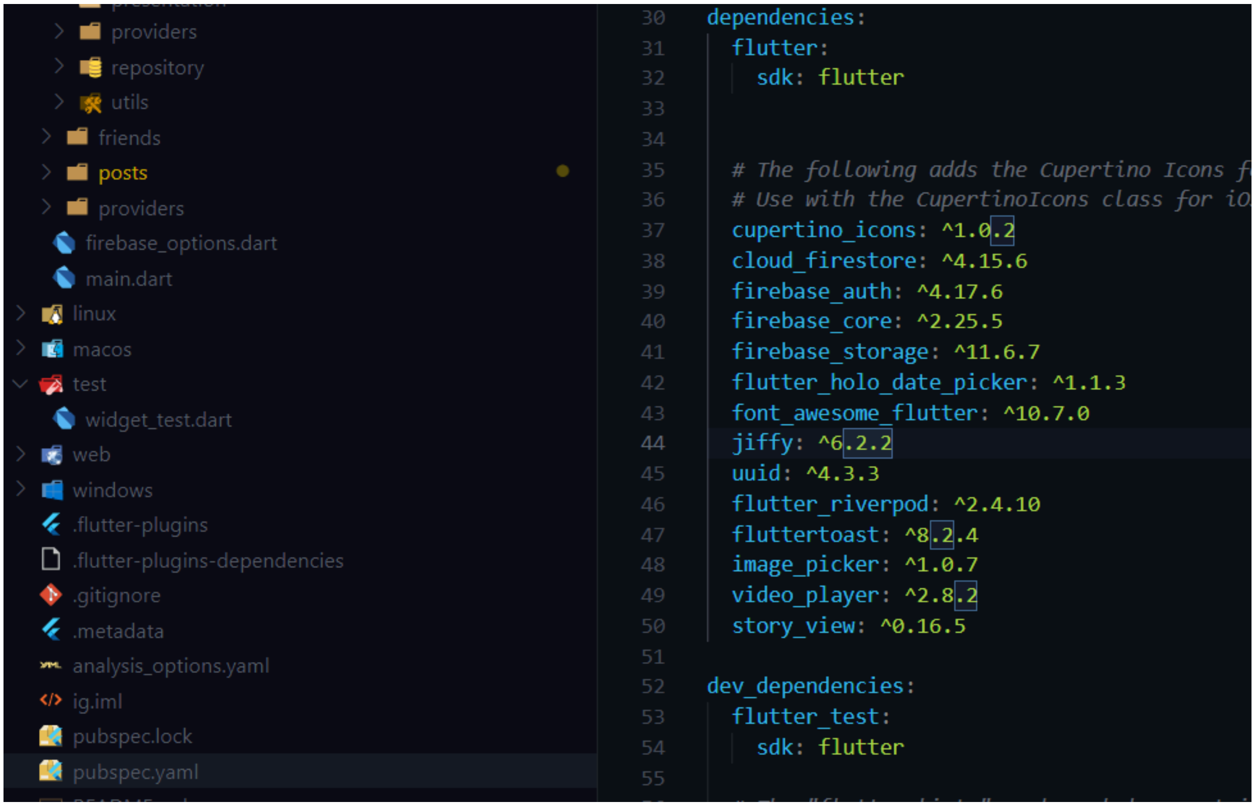


2)Go to the Firebase Console and create a new project.

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.

3) Add Firebase to your Flutter project and add the necessary dependencies:



```
dependencies:
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons f
  # Use with the CupertinoIcons class for io
  cupertino_icons: ^1.0.2
  cloud_firestore: ^4.15.6
  firebase_auth: ^4.17.6
  firebase_core: ^2.25.5
  firebase_storage: ^11.6.7
  flutter_holo_date_picker: ^1.1.3
  font_awesome_flutter: ^10.7.0
  jiffy: ^6.2.2
  uuid: ^4.3.3
  flutter_riverpod: ^2.4.10
  fluttertoast: ^8.2.4
  image_picker: ^1.0.7
  video_player: ^2.8.2
  story_view: ^0.16.5

dev_dependencies:
  flutter_test:
    sdk: flutter
```

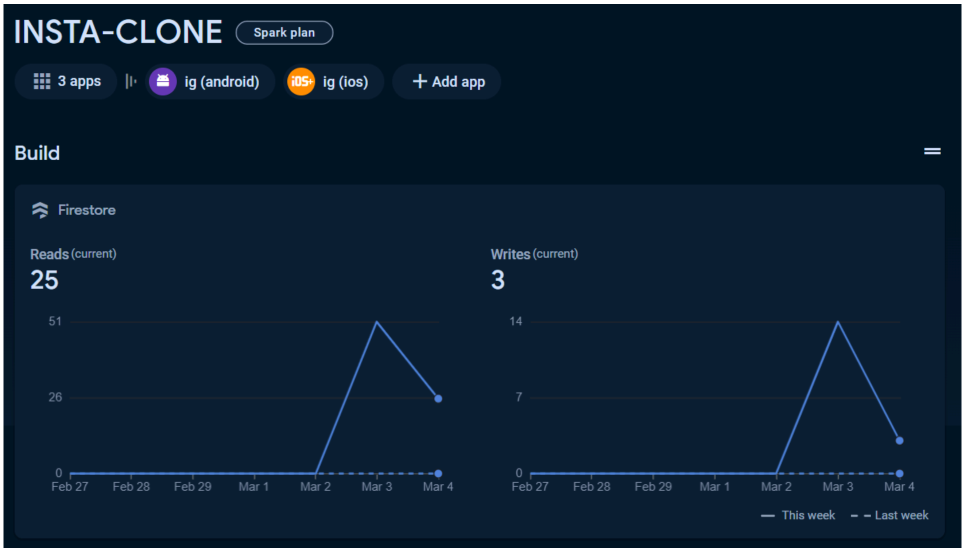
Code:

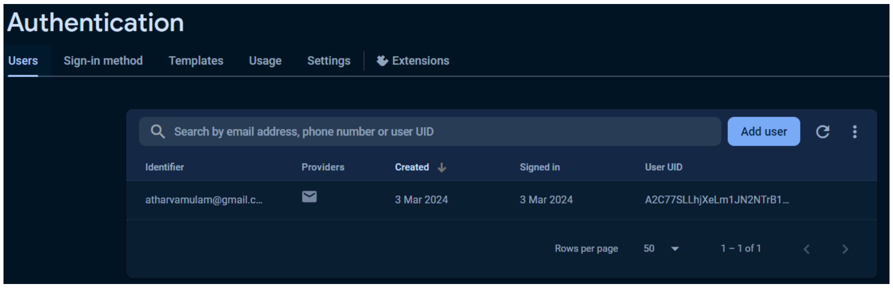
```
import 'dart:io';
import 'package:ig/core/constants/app_colors.dart';
import 'package:ig/core/constants/constants.dart';
import 'package:ig/core/utils/utils.dart';
import 'package:ig/core/widgets/pick_image_widget.dart';
import 'package:ig/core/widgets/round_button.dart';
import 'package:ig/core/widgets/round_text_field.dart';
import 'package:ig/features/auth/presentation/widgets/birthday_picker.dart';
import 'package:ig/features/auth/presentation/widgets/gender_picker.dart';
import 'package:ig/features/auth/providers/auth_provider.dart';
import 'package:ig/features/auth/utils/utils.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
```

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

class CreateAccountScreen extends ConsumerStatefulWidget {
  const CreateAccountScreen({super.key});

  static const routeName = '/create-account';

  @override
  ConsumerState<CreateAccountScreen> createState() =>
      _CreateAccountScreenState();
}

class _CreateAccountScreenState extends
ConsumerState<CreateAccountScreen> {
  File? image;
  DateTime? birthday;
  String gender = 'male';
  bool isLoading = false;

  // controllers
  late final TextEditingController _fNameController;
  late final TextEditingController _lNameController;
  late final TextEditingController _emailController;
  late final TextEditingController _passwordController;

  @override
  void initState() {
    _fNameController = TextEditingController();
    _lNameController = TextEditingController();
    _emailController = TextEditingController();
    _passwordController = TextEditingController();
    super.initState();
  }

  @override
  void dispose() {
```

```
  _fNameController.dispose();
  _lNameController.dispose();
  _emailController.dispose();
  _passwordController.dispose();
  super.dispose();
}

Future<void> createAccount() async {
  if (_formKey.currentState!.validate()) {
    _formKey.currentState!.save();
    setState(() => isLoading = true);
    await ref
        .read(authProvider)
        .createAccount(
          fullName: '${_fNameController.text} ${_lNameController.text}',
          birthday: birthday ?? DateTime.now(),
          gender: gender,
          email: _emailController.text,
          password: _passwordController.text,
          image: image,
        )
        .then((credential) {
      if (!credential!.user!.emailVerified) {
        Navigator.pop(context);
      }
    }).catchError((_) {
      setState(() => isLoading = false);
    });
    setState(() => isLoading = false);
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: AppColors.realWhiteColor,
    appBar: AppBar(),
body: SingleChildScrollView(
  child: Padding(
    padding: Constants.defaultPadding,
    child: Form(
      key: _formKey,
      child: Column(
        children: [
          GestureDetector(
            onTap: () async {
              image = await pickImage();
              setState(() {});
            },
            child: PickImageWidget(image: image),
          ),
          const SizedBox(height: 20),
          Row(
            children: [
              // First Name Text Field
              Expanded(
                child: RoundTextField(
                  controller: _fNameController,
                  hintText: 'First name',
                  textInputAction: TextInputAction.next,
                  validator: validateName,
                ),
              ),
              const SizedBox(width: 10),
              // Last Name Text Field
              Expanded(
                child: RoundTextField(
                  controller: _lNameController,
                  hintText: 'Last name',
                  textInputAction: TextInputAction.next,
                  validator: validateName,
                ),
              ),
            ],
),
const SizedBox(height: 20),
BirthdayPicker(
  dateTime: birthday ?? DateTime.now(),
  onPressed: () async {
    birthday = await pickSimpleDate(
      context: context,
      date: birthday,
    );
    setState(() {});
  },
),
const SizedBox(height: 20),
GenderPicker(
  gender: gender,
  onChanged: (value) {
    gender = value ?? 'male';
    setState(() {});
  },
),
const SizedBox(height: 20),
// Phone number / email text field
RoundTextField(
  controller: _emailController,
  hintText: 'Email',
  textInputAction: TextInputAction.next,
  keyboardType: TextInputType.emailAddress,
  validator: validateEmail,
),
const SizedBox(height: 20),
// Password Text Field
RoundTextField(
  controller: _passwordController,
  hintText: 'Password',
  textInputAction: TextInputAction.done,
  keyboardType: TextInputType.visiblePassword,
  validator: validatePassword,
         isPassword: true,
        ),
        const SizedBox(height: 20),
        isLoading
          ? const Center(child: CircularProgressIndicator())
          : RoundButton(
            onPressed: createAccount,
            label: 'Create Account',
           ),
       ],
      ),
     ),
    ),
   ),
  );
 }
}
```

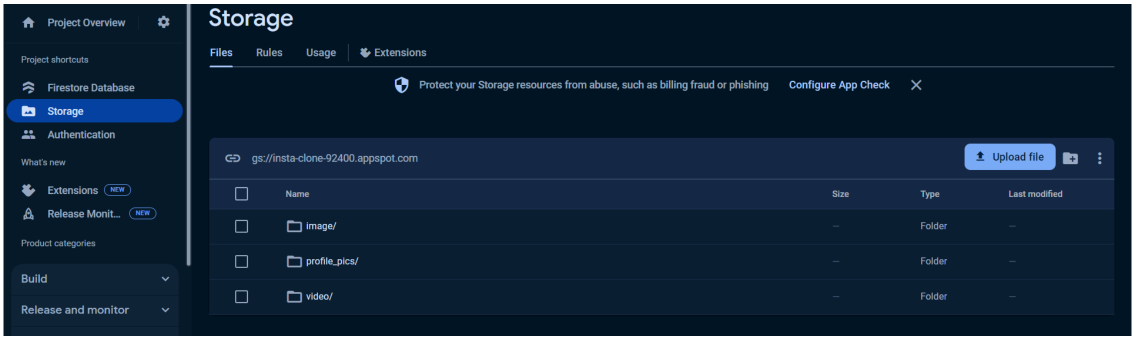
Output:





Identifier	Providers	Created ↓	Signed in	User UID
atharvamulam@gmail.c...	✉	3 Mar 2024	3 Mar 2024	A2C77SLLhjXeLm1JN2NTrB1...

Rows per page 50 ▾ 1 – 1 of 1

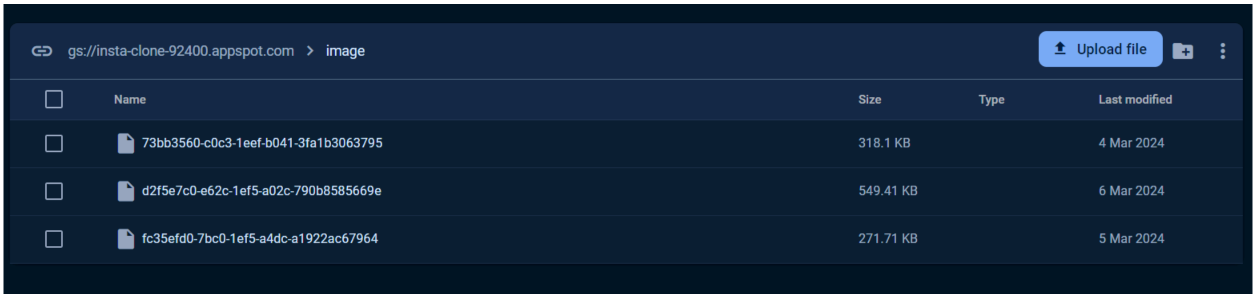


Storage

Files
Rules
Usage
Extensions
Protect your Storage resources from abuse, such as billing fraud or phishing
Configure App Check
Upload file

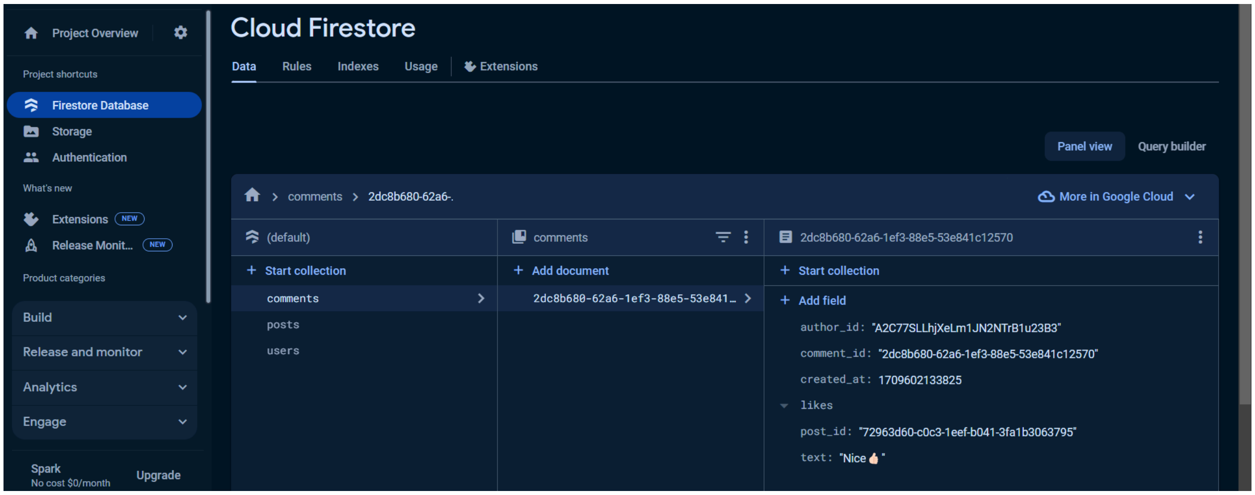
gs://insta-clone-92400.appspot.com

Name	Size	Type	Last modified
image/	–	Folder	–
profile_pics/	–	Folder	–
video/	–	Folder	–



gs://insta-clone-92400.appspot.com > image

Name	Size	Type	Last modified
73bb3560-c0c3-1eef-b041-3fa1b3063795	318.1 KB		4 Mar 2024
d2f5e7c0-e62c-1ef5-a02c-790b8585669e	549.41 KB		6 Mar 2024
fc35efd0-7bc0-1ef5-a4dc-a1922ac67964	271.71 KB		5 Mar 2024



Cloud Firestore

comments > 2dc8b680-62a6-.

(default)
- comments
- posts
- users

comments
- 2dc8b680-62a6-1ef3-88e5-53e841...

2dc8b680-62a6-1ef3-88e5-53e841c12570

```
author_id: "A2C77SLLhjXeLm1JN2NTrB1u23B3"
comment_id: "2dc8b680-62a6-1ef3-88e5-53e841c12570"
created_at: 1709602133825
likes
post_id: "72963d60-c0c3-1eef-b041-3fa1b3063795"
text: "Nice👍"
```

Conclusion: In this experiment, we have successfully connected firebase database and have authenticated login/signup and all the info related to a particular user are successfully stored in our database.