Flutter Course

***Coding Club, IIT Guwahati***

**Head:** *Varenya Tiwari (AficionadoXD#7443)*

**Coordinators:** *Pranav Nair(Pranav#7114) , Soumadip Das(SoumadipDas#4773), Harsh Bhakkad(The\_Lord\_Harsh#4494), Mansi (Mansi#6161), Aman Kumar(Aman Kumar#7963), Pranshu Pandya, Pratyush Ranjan, Shikhar Bind, Vignesh Deshpande(Delta Proxim#7619), Vishwesh S Warrier(Whatever00#8650)*

**WEEK 3 OBJECTIVE**

* Connecting to Firebase, Authentication
* CRUD Operations in Cloud Firestore
* Firebase Storage

From this week onwards we’ll start adding functionality to the clone Instagram UI you prepared in last week's assignment.

In this week we’ll implement:

* User login and Sign Up
* Retrieving logged in user data in profile screen and being able to change it as per need
* Able to upload and display user posts in profile section using Firebase Storage
* Showing posts of all registered users in the explorer screen

By the end of this week, we’ll have a functional authentication system, user profile screen, edit profile option and explorer screen.

**Setting up Firebase**

* **Getting started with Firebase on Flutter**
  + [Getting started with Firebase on Flutter - Firecasts](https://www.youtube.com/watch?v=EXp0gq9kGxI&t=19s)

Set up Firebase only for android, if you are interested you can go for iOS.

Note- Don’t return home in the FutureBuilder as directed in the video. Do so, only if you don’t have your routes set. If you have your initialRoute set, use the following code in main.dart

<https://pastebin.com/MUsyyFrE>

**Firebase Authentication**

Create a new project and implement this. Later after completing the three topics you can do the same in the Instagram App

* **Sign up and Sign in using FirebaseAuth**
  + [Flutter Firebase Authentication - Email and Password](https://www.youtube.com/watch?v=lhZGf0vcG7Y)

After watching this video, you will learn how to sign up and sign in with email and password using Firebase Authentication. Users can also be authenticated with link, without password. Refer the blog or the video below to implement email and passwordless sign in.

[FirebaseAuth with email and link](https://medium.com/firebase-developers/dive-into-firebase-auth-on-flutter-email-and-link-sign-in-e51603eb08f8)

or [Flutter Firebase Email Login (Step 5) - Login Anonymously](https://www.youtube.com/watch?v=zjo91VqILkg)

For getting started just try to implement Auth with email and password only and if interested you can move on to Auth with Google and Facebook.

* + [Flutter Tutorial - Google SignIn [2021] With Firebase Auth - Android, iOS, Flutter Web](https://www.youtube.com/watch?v=1k-gITZA9CI)
  + [FirebaseAuth with facebook](https://medium.com/@karldb/flutter-facebook-sign-in-with-firebase-in-2020-66556a8c3586)
* **Update password in FirebaseAuth**
  + [Verify and Update User password with Firebase and Flutter - Firebase Profile IV](https://youtu.be/2yBiPAS3CZE)

**Note-** FirebaseAuth will store the email address and password of the user. To store the username of the user we need to use Firestore Database. So you need to know about Firestore Database and CRUD operations in Firestore.

**Firestore Database, CRUD operations**

* **Firestore CRUD operations**

Before starting with Firestore, refer this official [documentation](https://firebase.flutter.dev/docs/firestore/usage/) to set up Firestore in your Project. It will help you to resolve any issue in the learning process.

* + [Flutter Firebase complete CRUD | Flutter with Firebase | Firestore | Beginner Tutorial | Codevio](https://youtu.be/PHf7NH2_gQs)

After watching this video you will have a basic idea of the CRUD operations in Firestore. Have a look at the blog below to get more clarity in the CRUD operations.

* + [Adding,updating and deleting data from Firestore](https://medium.com/firebase-tips-tricks/how-to-use-cloud-firestore-in-flutter-9ea80593ca40)
* **Storing User data in firestore**
  + [Add User Data to Firestore in Flutter](https://youtu.be/1xPMbwOFa9I)

This video covers how to store user data like username and other credentials to Firestore. Now you will know how to update user data stored in firestore using Firestore update.

* **Updating User Data in firestore**
  + [Firestore update](https://saveyourtime.medium.com/firebase-cloud-firestore-add-set-update-delete-get-data-6da566513b1b)

To update user data in Firestore we need to use Firestore update. Refer the blog above, to implement Firestore update

**Firebase Storage**

* **Firebase Storing files,images**
  + [Firebase\_storage](https://pub.dev/packages/firebase_storage) this is the plugin you need to add in your dependencies.
  + [Introducing Cloud Storage for Firebase - YouTube](https://www.youtube.com/watch?v=_tyjqozrEPY)

This will introduce you to powers of firebase storage. Now let’s see how you can upload any file on firebase storage, get a link of that and add that to your firebase collection field so that you can use it for retrieving that file in future.

* + [Using Firebase Storage in Flutter | Peter Coding](https://petercoding.com/firebase/2020/05/06/using-firebase-storage-in-flutter/)

Below video will summarise all this:

* + [Flutter Tutorial - Download Files From Firebase Storage - Download Images, Videos, Files [2021]](https://www.youtube.com/watch?v=YA_fHCF_EYc)

**PRACTICE TASK 1:-**

* **Flutter Firestore Notes App**
  + [Flutter Tutorial | Notes App with Flutter and Firestore | Cloud Firestore Tutorial | Beginners](https://www.youtube.com/watch?v=hYKueBVKqoc)

**PRACTICE TASK 2:-**

* **Flutter Chat app**
  + [**Part 1**](https://www.youtube.com/watch?v=O0LHve3I1IY)

This video will cover linking your project to Firebase to be able to access user authentication and

Cloud Firestore.

* + [**Part 2**](https://www.youtube.com/watch?v=D1VFR97zVtM)

This video will cover setting up multi-screen navigation, google Sign-in for user authentication,

uploading user’s data to cloud firestore, retrieve user data from cloud using streambuilder.

Note:

1. Don’t mind the crossing line over widgets like when using FlatButton or RaisedButton, it isn’t an error.

2. In the new firebase update whenever making a firebase instance use

“FirebaseFirestore.instance.......” instead of “Firestore.instance”.

* + [**Part 3**](https://youtu.be/sfqhIk1K8VI)

This video will cover making a chat screen and simultaneously uploading and retrieving

messages data entered by different users to the cloud.

<https://dev.to/kazuhideoki/how-to-get-data-from-firestore-and-show-it-on-flutterbuilder-or-streambuilder-e05>

<https://medium.com/quick-code/reading-lists-from-firestore-using-streambuilder-in-flutter-eda590f461ed>

Refer these blogs for more clarity on StreamBuilder.

Note:

1. Try to understand StreamBuilder, ListView and performing data queries carefully, this is the most tricky part in most development projects.

2. Firestore package was updated recently, hence most of the tutorials may have wrong syntax, but the logic remains the same. So, don’t blindly copy the code without understanding. Refer Firestore documentation to get the most recent syntax

**It’s done! Now, you have to update your Instagram app, you created in Week 2 task, and make it functional. You have to implement :-**

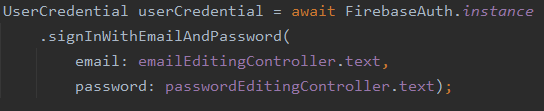
* User login and Sign Up
* Retrieving logged in user data in profile screen and being able to change it as per need
* Able to upload and display user posts in profile section using Firebase Storage
* Showing posts of all registered users in the explorer screen.

After the user signs in, he will be directed to the home screen. On clicking on the (+) icon, in the bottom bar, he will be able to upload images and videos. Clicking on the Camera button at the top left corner he will be able to capture photos, videos and upload them. All the posts of all registered users should be visible in the home screen as well as the explorer screen. In the profile screen, show all the posts by that particular signed in user.

**User login and sign up**

The user should be able to sign in using email. Facebook authentication is

**OPTIONAL**, it can be implemented for **EXTRA MARKS.**



Once the user signs in, direct him to Home page using [Navigator.pushReplacement](https://api.flutter.dev/flutter/widgets/Navigator/pushReplacement.html) so that he can’t come back to sign in page.

Pass the (UserCredential usercredential) of the user to the Home Page as a parameter and from there to the profile screen.

[Passing data between widgets, Flutter](https://medium.com/flutter-community/simple-ways-to-pass-to-and-share-data-with-widgets-pages-f8988534bd5b)

To keep the user signed in locally, use the [SharedPreferences](https://pub.dev/packages/shared_preferences) package which can store data locally. Store the username of the user in sharedpreference. Destroy the instance of the sharedpreference once the user logs out.

[How To Keep User Logged In Using Shared Preferences Flutter | Logout user using firebase](https://youtu.be/AGbTa6zzias)

Store the username, email and uid of the user in Firestore database as a collection 'user’ as explained in the above tutorials.

**Displaying user data in profile screen**

Now you have the userCredential of the user in the profile screen which you sent as a parameter from the signed in screen.

To get email and uniqueid from userCredential use

uid = userCredential.user!.uid;

email = userCredential.user!.email;

Fetch the username of user from Shared Preference in the profile screen. It is an [async](https://dart.dev/codelabs/async-await) task, so implement it in an async method and call that method from the initState [function](https://api.flutter.dev/flutter/widgets/State/initState.html).

(Note- you cannot perform async tasks in initState function)

To prevent late initialization error, initialise the username with a random name and then update its value in the async method inside setState.

Now you have the username, email of the user to be displayed in the profile screen.

User should also be able to change the username and reset his password as per need, so update the username field in Firestore as well as in [SharedPreferences](https://stackoverflow.com/questions/61369292/how-to-update-value-inside-shared-preferences-in-flutter). For this create a **editprofile** page where user can update username, add bio, reset password and change his profile picture.

Changing profile picture is a optional task.

**Upload, display posts**

Now you know how to deal with firebase storage you would like to know how can you upload files like images, videos directly from your devices, so this is the time to know image\_picker and video\_player plugins.

* [image picker documentation](https://pub.dev/packages/image_picker), [Video\_player documentation](https://pub.dev/packages/video_player) this are the official documentation of the plugins.
* [Camera and Photo Gallery Using Image Picker with Flutter | Day 25 - #30DaysOfFlutter - YouTube](https://www.youtube.com/watch?v=7iZAZ0CwFKI)

This video will show you how to use an image picker for getting images from a camera or gallery.

* [Flutter Tutorial - Video Player - Asset, File & Network - YouTube](https://www.youtube.com/watch?v=uz4xRnE-UIw) This video will show you all the things you can do with video player
* [Example app of image and video](https://pub.dev/packages/image_picker/example) You can try to copy paste this example code in a new project and see all the things that image picker and video player can do.
* Now you know how to deal with images and video, let’s learn how we can upload them to firestorage. <https://www.youtube.com/watch?v=pvRpzyBYBbA>
* [Firebase Profile in Flutter - Image Upload with Firebase Storage - Part II - YouTube](https://www.youtube.com/watch?v=KurjrXR5mRg)

Now you have all the knowledge required for making profile pic in your app so watch this video to collaborate all your knowledge.

**Note-** Each post must have the following details:-

* Username of user who posted
* Time of posting
* Image url of the stuff posted
* Profile picture of the user who posted
* Caption of the post(while the user picks a image create a alert dialog to get the caption)

Store all these data as fields in firestore, fetch and display in the feeds page.

**OPTIONAL TASK**

Now you should know how to upload and display posts in Firestore Flutter. So, try to implement uploading profile picture. As the user clicks on the profile icon in the edit profile page, he should be able to upload a picture for his profile. Store the url of the profile picture as the field in the particular document in the users collection using Firestore update.

**Implementing Explorer Screen**

Once you store the images from different users in a specific collection in Firestore, it’s time to extract the images from all the users in the collection and put them in the Explorer Screen of the app in a grid view pattern. For streaming the images from all the users, we need to use a new widget: Stream Builder.

* Stream Builder : [StreamBuilder (Flutter Widget of the Week)](https://www.youtube.com/watch?v=MkKEWHfy99Y)

Now, to create the explorer screen with the help of Stream Builder, refer this video :-

* [3 fetching data from firestore and display it in the flutter using StreamBuilder](https://www.youtube.com/watch?v=4-84lwBta08)

Note the following before jumping to the video:

* You don’t need to store the images in the Storage as shown in the video, because you will already be having the images in the documents.
* Because we want to display the images in a grid view, use Grid View widget instead of List View as demonstrated in the video.

**SUBMISSION**

You have to submit the Github repo link and a screen recording of Instagram Clone App showing all the required functionalities as mentioned above. You can implement the two optional tasks for bonus marks.

[**Link**](https://forms.gle/vtC1v2geEsFAb8za6)

**DEADLINE- 28 July EOD**