# GLA University

(Mathura)

PROJECT REPORT

JoVAC

# The Organizer

App using FLUTTER

**Submitted by:**

Himanshu Sharma (201550062) – B.Tech CS AIML

Akshit Saxena (201550018) – B.Tech CS AIML

Molika Agarwal (201550083) – B.Tech CS AIML

Prateek Kumar (201550102) – B.Tech CS AIML

Sangam Bansal (201550123) – B.Tech CS AIML

Mayank Saraswat (201550081) – B.Tech CS AIML

# Table of Contents

## Declaration

I hereby declare that the work which is being presented in the Project Report **"The Organizer - The Ultimate Entertainment Organizer",** in partial fulfilment of the requirements for Project is an authentic record of my own work carried under the supervision of **Mr. Pankaj Kapoor, Corporate Trainer & Mr. Rahul Pradhan, Assistant Professor, GLA University, Mathura**.

Sign __Himanshu Sharma___
Name of Candidate: Himanshu Sharma
University Roll No.: 201550062

Sign ____Molika Agarwal____
Name of Candidate: Molika Agarwal
University Roll No.: 201550083

Sign _____Akshit Saxena______
Name of Candidate: Akshit Saxena
University Roll No.: 201550018

Sign ___Sangam Bansal ____
Name of Candidate: Sangam Bansal
University Roll No.: 201550123

Sign ___Prateek Kumar____
Name of Candidate: Prateek Kumar
University Roll No.: 201550102

Sign ____Mayank Saraswat____
Name of Candidate: Mayank Saraswat
University Roll No.: 201550081

## Certificate

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

**Project Supervisor:**

________________________

**Mr Pankaj Kapoor**
Corporate Trainer

________________________

**Mr. Rahul Pradhan**
Asst. Prof., GLA University

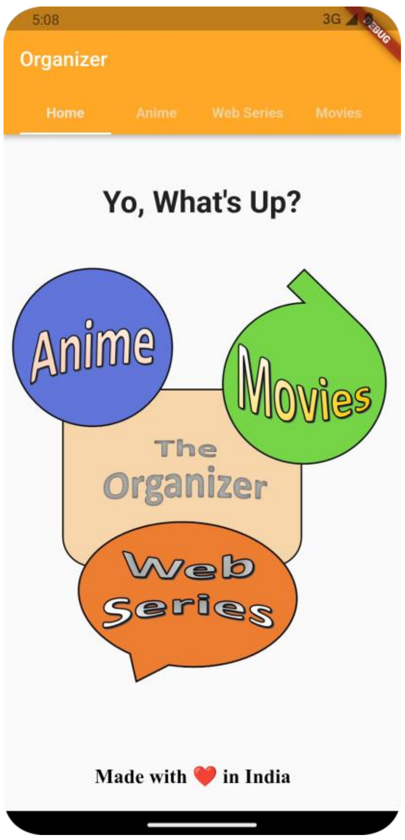
Date: 18/07/202

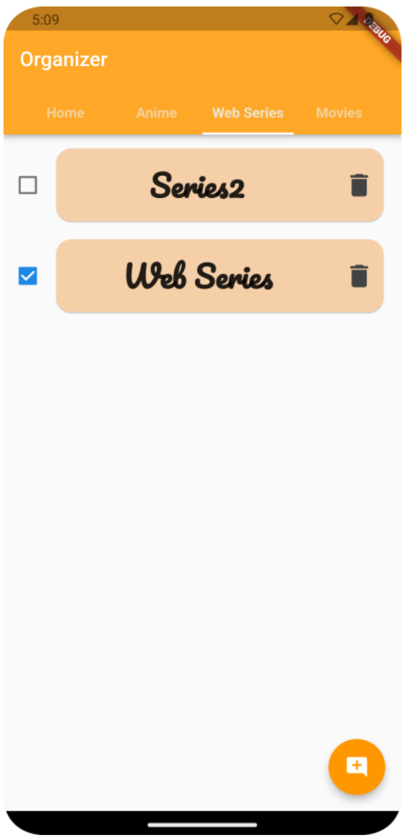
## Introduction

We all know that, nowadays the craze of Anime, Web Series and all other OTT content is gradually increasing. And in our daily routine many of us don't have time to keep track of our 'Watched' and 'Yet to Watch content, because enormous amount of such content exists.
So here we are presenting our app "The Organizer - The Ultimate Entertainment Organizer".
Which helps us to keep track of our desired content such as Anime, Movies and Web Series.

## Executive Summary

We Have Successfully Created an App Which will be proven to be very useful. In this fast pace life, it just provides us some relief from remembering tough names of the anime and movies.





## Project Design

This App Consists a Junction of Various Technologies implemented in Our project. The following table will consist some of the part of the important and cutting-edge tools and technologies that are used and implemented in this very useful project.

Listed below are some of the more important functions which may help one understand the project better.

**ListView**

- ListView is a scrollable list of widgets arranged linearly.
- It displays its children one after another in the scroll direction i.e., vertical or horizontal.

- When there is so much data in single Screen it Provides us the accessibility to scroll.
- Ex- As in our App there are multiple Cards are present so it helps us managing them.

**Alert Dialogs**

- An alert dialog is a useful feature that notifies the user with important information to make a decision or provide the ability to choose a specific action or list of actions.
- It is a pop-up box that appears at the top of the app content and the middle of the screen.

- In our project when the user will delete its data permanently, the alert dialog box will pop up and help the user to enter his confirmation.

**Flutter Buttons**

- We can easily apply themes on buttons, shapes, color, animation, and behavior.
- Buttons can be composed of different child widgets for different characteristics

- Elevated Button and Floating Action Button helped us by allowing user to take certain actions over adding, modifying and deleting content in our app.

## Flutter Stack

- The stack is a widget in Flutter that contains a list of widgets and positions them on top of the other. In other words, the stack allows developers to overlap multiple widgets into a single screen and renders them from bottom to top. Hence, the first widget is the bottommost item, and the last widget is the topmost item.

us in our project and we implemented positioned widget to set the position of our ʒes in this Project.

## Bottom Sheets

- A bottom sheet is positioned at the bottom of the screen.
- Bottom sheet will appear in response to some user action, such as tapping an icon.
- It can be dismissed by any of the following user actions: swiping down, tapping an item within the bottom sheet, tapping the main UI of the app.

- It helped us a lot when using add, delete and update function in the project it is the mini dialog box popped up near to the navigation buttons.

## Flutter CheckBox

- Checkbox in flutter is a material design widget.
- It is always used in the Stateful Widget as it does not maintain a state of its own.
- We can see the checkboxes on the screen as a square box with white space or checked mark.

- It helps the user to know whether He has seen a particular content or not by displaying a Checked Mark.

## Async and Await

- When you are returning from a function and you have code that will take time to finish then use async/await to wait until data is retrieved so it can be returned.

- In our project it helps us to let firebase connection setup be completed just before our app starts and let our database command be executed before returning.

# Source Code

## main.dart

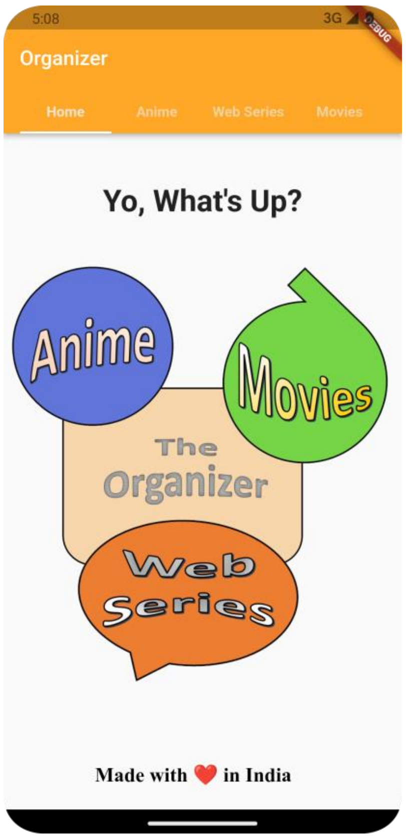
```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'HomePage.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();  //Initializing firebase
  await Firebase.initializeApp(); //--------
  runApp(
      MaterialApp(
        home: HomePage(),        // this will return Home Page of our app
      )
  );
}
```

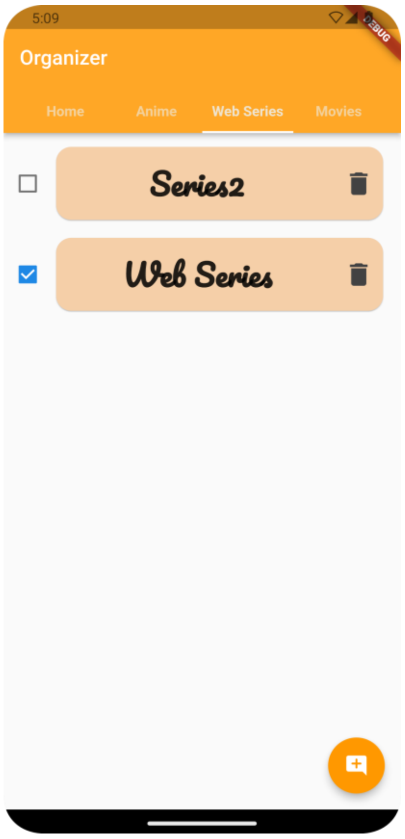
## Tab_Anime.dart

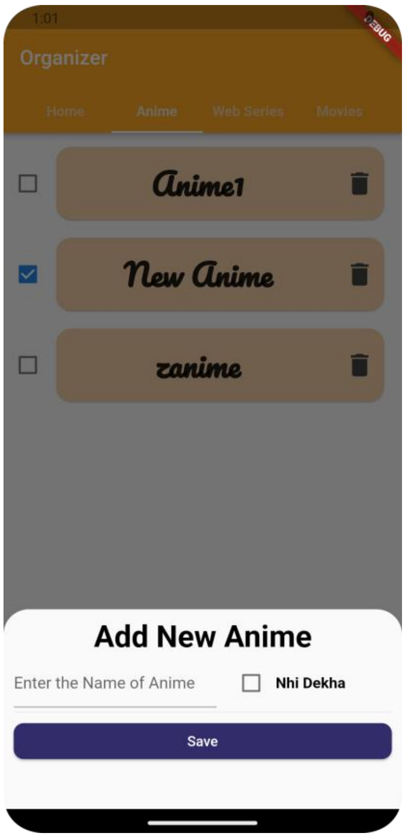
```
import 'package:cloud_firestore/cloud_firestore.dart';  // this is required to use firebase firestore instance.
import 'package:flutter/material.dart';

class Anime extends StatefulWidget {
  const Anime({Key? key}) : super(key: key);

  @override
  State<Anime> createState() => _AnimeState();
}

// We have declared these variables globally so that we can use this in bottom sheet down.
TextEditingController _name =TextEditingController();   // this controller will temporarily save the name
which user will enter.
bool _watched=false;                                    // this will save the value that whether user have watched that
anime or not
final firebase = FirebaseFirestore.instance;            // this is the instance which will help us connect to
firebase

class _AnimeState extends State<Anime> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(      // this button will display bottom sheet
        onPressed: () {
          displayBottomSheet(context);             //here we created a function which will return the bottom sheet
    },
  child: Icon(Icons.add_comment_rounded),
  backgroundColor: Colors.orange.shade500,
),
body: SafeArea(
  child: Container(
    margin: EdgeInsets.symmetric(vertical: 10),
    child: StreamBuilder<QuerySnapshot>(
        stream: firebase.collection("Anime").orderBy("Name").snapshots(),
        builder: (context, snapshot) {
          if (snapshot.hasData) {
            return ListView.builder(
              itemCount: snapshot.data!.docs.length,
              itemBuilder: (context, i) {
                QueryDocumentSnapshot data = snapshot.data!.docs[i];      //here it is retrieving the data from
databse
                return Column(
                  children: [
                    Row(
                      children: [
                        Checkbox(                       // this will display as store in database
                          value: data['Watched'],
                          onChanged: (b) {
                            setState((){
                              update(data['Name'],b!);   // also it will update data in database if clicked
                            });
                          },
                        ),
                        Container(
                          child: Card(
                            color: Color.fromARGB(255, 245, 207, 168),
                            shape: RoundedRectangleBorder(
                              borderRadius: BorderRadius.circular(15.0),
                            ),
                            child: Row(
                              children: [
                                SizedBox(
                                  width: 10,
                                ),
                                Container(
                                  width: 260,
                                  padding: EdgeInsets.all(10),
                                  alignment: Alignment.center,
                                  child: Text(
                                    data['Name'],
                                    softWrap: false,
    overflow: TextOverflow.ellipsis, // this will help us to take care if the name entered is too long to display
    style: TextStyle(fontFamily: 'Pacifico', fontWeight: FontWeight.bold, fontSize: 30),
  ),
),
                                    SizedBox(
                                      width: 5,
                                    ),
                                    IconButton(
                                      icon: Icon(Icons.delete),
                                      color: Colors.grey.shade800,
                                      iconSize: 30,
                                      onPressed: () {
                                        setState((){
                                          delete(data['Name'],data['Watched'],context);             // here we created delete
function to delete the data.
                                        });
                                      },
                                    )
                                  ],
                                ),
                              ),
                            ),
                          ],
                        ),
                        SizedBox(
                          height: 10,
                        )
                      ],
                    );
                  },
                );
              }
              else {
                return Center(
                  child: CircularProgressIndicator(),
                );
              }
            }
        ),
      ),
    ),
  );
}
void displayBottomSheet(BuildContext context) {       // here is that function which will return bottom sheet.
  showModalBottomSheet(
      context: context,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.vertical(top: Radius.circular(25.0)),
      ),
      builder: (BuildContext context){
        return Padding(                                  // we used padding like this so that our sheet will move
          padding: EdgeInsets.only(                 // with our keyboard. lets see...
            bottom: MediaQuery.of(context).viewInsets.bottom,
          ),
        child: Add_Anime(),
      );
    }
  );
 }

 void delete(String name, bool watched, BuildContext context) {                // To delete the data
  var alertDialog = AlertDialog(                                              // Alert dialogue box
   shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(20),
   ),
   backgroundColor: Colors.white,
   title: Container(
    alignment: Alignment.center,
    child: Text(
     'Are You Sure?',
     style: TextStyle(
      color: Color.fromARGB(255, 38, 23, 152),
      fontWeight: FontWeight.bold,
      fontFamily: 'Times New Roman Bold',
      fontSize: 28,
     ),
    ),
   ),
   content: Text(                              // It is displaying the data which we are going to delete
    "Name : $name \nWatched : "+(watched ? "Yes" : "No"),
    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
   ),
   actions: [
    Row(
     mainAxisAlignment: MainAxisAlignment.center,
     children: [
      ElevatedButton(                           // Buttons used
       onPressed: () async {
        try {
         await firebase.collection("Anime").doc(name).delete();          // first the data is deleted
         Navigator.pop(context);                                         // then the alert dialogue box get away
        } catch (e) {
         print(e);
        }
       },
       child: Text("Delete"),
       style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all<Color>(Color.fromARGB(255, 50, 44, 106)),
        shape: MaterialStateProperty.all<OutlinedBorder>(RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10))),
       ),
      ),
      SizedBox(
       width: 10,
          ),
          ElevatedButton(
            onPressed: () {
              Navigator.pop(context);            // here no delete action is performed when clicked on "Cancel"
            },
            child: Text("Cancel"),
            style: ButtonStyle(
              backgroundColor: MaterialStateProperty.all<Color>(Color.fromARGB(255, 50, 44, 106)),
              shape: MaterialStateProperty.all<OutlinedBorder>(RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10))),
            ),
          ),
        ],
      )
    ],
  );

  showDialog(                                // this will further show our dialogue box
      context: context,                      // references: lectures
      builder: (context) {
        return alertDialog;
      });
}

void update(String name, bool watched) async{            // same as add function
  try {
    await firebase.collection("Anime").doc(name).update({"Watched": watched});         //but changing only
bool value
    print("done");
  } catch (e) {
    print(e);
  }
}
}

class Add_Anime extends StatefulWidget {                  // This is the page which will be displayed in
bottom sheet
  const Add_Anime({Key? key}) : super(key: key);

  @override
  State<Add_Anime> createState() => _Add_AnimeState();
}

class _Add_AnimeState extends State<Add_Anime> {
  String? _nameErrorText = null;     // this the error message if user click on save button without giving any
name.
  @override
  Widget build(BuildContext context) {
    return  Container(
      height: 200,
  child: ListView(
    padding: EdgeInsets.all(10),
    children: [
      Container(
        alignment: Alignment.center,
        child: Text(
          "Add New Anime",
          style: TextStyle(color: Colors.black, fontWeight: FontWeight.bold, fontSize: 30),
        ),
      ),
      SizedBox(
        height: 5,
      ),
      Row(
        children: [
          Container(
            width: 200,
            alignment: Alignment.center,
            child: Center(
              child: TextField(
                keyboardType: TextInputType.text,
                controller: _name,          // controller to save data temporarily.
                decoration: InputDecoration(
                  hintText: "Enter the Name of Anime",
                  border: UnderlineInputBorder(),
                  errorText: _nameErrorText,      // error text variable which will change when user click on save
button without giving name.
                ),
              ),
            ),
          ),
          SizedBox(
            width: 10,
          ),
          Checkbox(
            value: _watched,              // check string our globally initialized value
            onChanged: (bool? watch){
              setState((){
                _watched=watch!;                // it is changing our value of checkbox.
              });
            },
          ),
          Text(
            _watched ? "Dekhliya" : "Nhi Dekha",              // this is additional
            style: TextStyle(color: Colors.black, fontSize: 15, fontWeight: FontWeight.bold),
          ),
        ],
      ),
      Divider(
        height: 10,
          ),
          Container(
            width: 250,
            child: ElevatedButton(              // we used elevated button here
              onPressed:
                  () {
                setState((){
                  if(_name.text.isEmpty){                              // this is checking whether user entered any name or
not.
                    _nameErrorText="Name can't be empty.";           // error will display if no name was entered
                  }
                  else{                                              // if user have entered the name then it will add the data to
firebase.
                    _nameErrorText=null;                               // also error text will also removed here
                    add(_name.text,_watched);                          // this is the function which will add the data.
                    Navigator.pop(context);                            // this will takeaway the bottom sheet - as our job is
done.
                    _name.clear();_watched=false;                      // this will clear the data held by _name, and remove
the checkbox value
                  }
                });
              },
              child: Text("Save"),
              style: ButtonStyle(                       // this is our button style.
                backgroundColor: MaterialStateProperty.all<Color>(Color.fromARGB(255, 50, 44, 106)),
                shape: MaterialStateProperty.all<OutlinedBorder>(RoundedRectangleBorder(borderRadius:
BorderRadius.circular(10))),
              ),
            ),
          ),
        ],
      ),
    );
  }

  void add(String name, bool watched) async{
    try {
      await firebase.collection("Anime").doc(name).set({"Name": name, "Watched": watched});   // data
added to frebase here
      print("added");
    } catch (e) {
      print(e);
    }
  }
}
```

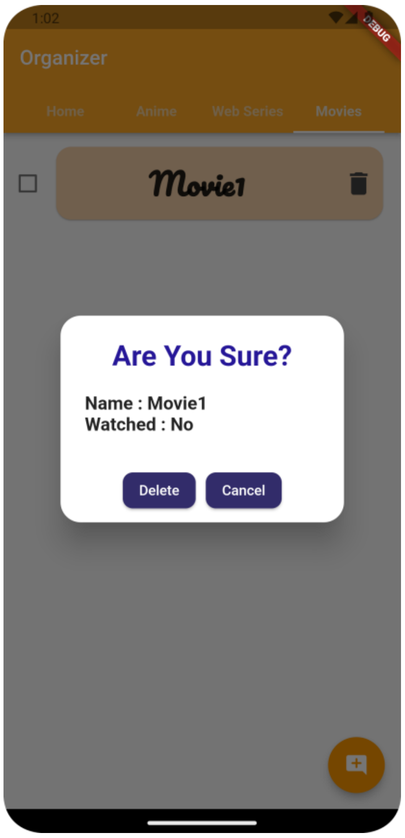
- Likewise, other 2 tabs named Tab_Movies.dart and Tab_WebSeries.dart are there.
- Only difference is that they contain ‘Movies’ and ‘Web Series’ instead of ‘Anime’ in there corresponding tabs.

## User Interface









## Conclusion

Hence, we can create a fully functional and exciting Hybrid Mobile App using Flutter.

We can even customize the app according to our need and can use same code base for multi-platforms

## References

- Notes By Pankaj Kapoor (Corporate Trainer),
- SlideShare,
- Google,
- GeeksforGeeks,
- Flutter Documentation,
- StackOverflow.

Thank you!