

Exp 4**Aim:**

To create an interactive Form using form widget.

Theory:

A form in Flutter is a structured container that collects user input through various fields like text fields, dropdowns, checkboxes, and buttons. It plays a crucial role in applications that require user data entry, such as login pages, registration forms, and feedback submissions.

Flutter provides the Form widget, which works alongside TextFormField and other input elements to manage validation, state handling, and error messages efficiently. By using form validation techniques, developers can ensure data accuracy and enhance user experience. When you create a form, it is necessary to provide the GlobalKey. This key uniquely identifies the form and allows you to do any validation in the form fields. The form widget uses child widget TextFormField to provide the users to enter the text field. This widget renders a material design text field and also allows us to display validation errors when they occur.

Creation of a Form

- While creating a form in Flutter, the Form widget is essential as it acts as a container for grouping multiple form fields and managing validation.
- A GlobalKey is required to uniquely identify the form and enable validation or data retrieval from the form fields.
- The TextFormField widget is used to provide input fields where users can enter data such as names, phone numbers, or email addresses.
- To enhance the appearance and usability of input fields, InputDecoration is used, allowing customization of labels, icons, borders, and hint text.
- Validation plays a crucial role in forms, and the validator property within TextFormField ensures user input meets specific criteria before submission.

Some Properties of Form Widget

- **key:** A GlobalKey that uniquely identifies the Form. You can use this key to interact with the form, such as validating, resetting, or saving its state.
- **child:** The child widget that contains the form fields. Typically, this is a Column, ListView, or another widget that allows you to arrange the form fields vertically.
- **autovalidateMode:** An enum that specifies when the form should automatically validate its fields. Some Methods of Form Widget
- **validate():** This method is used to trigger the validation of all the form fields within the Form. It returns true if all fields are valid, otherwise false. You can use it to check the overall validity of the form before submitting it.
- **save():** This method is used to save the current values of all form fields. It invokes the onSaved callback for each field. Typically, this method is called after validation succeeds.
- **reset():** Resets the form to its initial state, clearing any user-entered data.
- **currentState:** A getter that returns the current FormState associated with the Form

Code:

Home_page.dart

```
// ignore_for_file: invalid_use_of_visible_for_testing_member

import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:path/path.dart';
import 'package:youtube_clone/cores/screens/error_page.dart';
import 'package:youtube_clone/cores/screens/loader.dart';
import 'package:youtube_clone/cores/widgets/image_button.dart';
import 'package:youtube_clone/features/account/account_page.dart';
import 'package:youtube_clone/features/auth/provider/user_provider.dart';
import 'package:youtube_clone/features/content/bottom_navigation.dart';
```

```
import 'package:youtube_clone/features/upload/upload_bottom_sheet.dart';
import 'package:youtube_clone/pages_list.dart';

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

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int currentIndex = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xffFFFFFF),
      body: SafeArea(
        child: Column(
          children: [
            Row(
              children: [
                Image.asset(
                  "assets/images/youtube.jpg",
                  height: 36,
                ),
                const SizedBox(width: 4),
                const Spacer(),
                Padding(
                  padding: const EdgeInsets.only(right: 12),
                  child: SizedBox(
                    height: 42,
                    child: ImageButton(
                      image: "cast.png",
                      onPressed: () {},
                      haveColor: false,
                    ),
                  ),
                const Spacer(),
                const Image.asset(
                  "assets/images/plus.png",
                  height: 36,
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

```

    ),
  ),
),
PreferredSize(
  height: 38,
  child: ImageButton(
    image: "notification.png",
    onPressed: () {},
    haveColor: false,
  ),
),
Padding(
  padding: const EdgeInsets.only(left: 12, right: 15),
  child: SizedBox(
    height: 41.5,
    child: ImageButton(
      image: "search.png",
      onPressed: () {},
      haveColor: false,
    ),
  ),
),
Consumer(
  builder: (context, ref, child) {
    return ref.watch(currentUserProvider).when(
      data: (currentUser) => Padding(
        padding: const EdgeInsets.only(right: 12),
        child: GestureDetector(
          onTap: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => AccountPage(
                  user: currentUser,
                ),
            ),

```

```

        ),
      ),
    },
    child: CircleAvatar(
      radius: 14,
      backgroundColor: Colors.grey,
      backgroundImage:
CachedNetworkImageProvider(
        currentUser.profilePic,
      ),
    ),
  ),
  error: (error, stackTrace) => const ErrorPage(),
  loading: () => const Loader(),
);

    },
  ),
],
),
Expanded(
  child: pages[currentIndex],
),
],
),
),
bottomNavigationBar: BottomNavigation(
  onPressed: (index) {
    if (index != 2) {
      currentIndex = index;
      setState(() {});
    } else {
      showModalBottomSheet(
        context: context,
        builder: (context) => const CreateBottomSheet(),
      );
    }
  },
);

```

```

        }
      },
    ),
  );
}
}

```

Main.dart

```

// ignore_for_file: unnecessary_null_comparison, prefer_const_constructors

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:youtube_clone/cores/screens/loader.dart';
import 'package:youtube_clone/features/auth/pages/login_page.dart';
import 'package:youtube_clone/features/auth/pages/username_page.dart';
import
'package:youtube_clone/features/upload/long_video/video_details_page.dart'
;
import 'package:youtube_clone/firebase_options.dart';
import 'package:youtube_clone/home_page.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(ProviderScope(child: MyApp()));
}

```

```

class MyApp extends ConsumerWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context, WidgetRef ref) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      // we check whether we are signed in or not
      home: StreamBuilder(
        stream: FirebaseAuth.instance.authStateChanges(),
        builder: (context, snapshot) {
          if (!snapshot.hasData) {
            return LoginPage();
          } else if (snapshot.connectionState == ConnectionState.waiting)
{
            return Loader();
          }
          return StreamBuilder(
            stream: FirebaseFirestore.instance
              .collection("users")
              .doc(FirebaseAuth.instance.currentUser!.uid)
              .snapshots(),
            builder: (context, snapshot) {
              final user = FirebaseAuth.instance.currentUser;
              if (!snapshot.hasData || !snapshot.data!.exists) {
                return UsernamePage(
                  displayName: user!.displayName!,
                  profilePic: user.photoURL!,
                  email: user.email!,
                );
              } else if (snapshot.connectionState ==
ConnectionState.waiting) {
                return Loader();
              }
              return HomePage();
            }
          );
        }
      );
    }
  }
}

```

```

        },
      ),
    ),
  ),
);
}
}

```

In folder:

Long_video

|_video_detail.dart

```

// ignore_for_file: public_member_api_docs, sort_constructors_first
import 'dart:io';

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:uuid/uuid.dart';

import 'package:youtube_clone/cores/methods.dart';
import
'package:youtube_clone/features/upload/long_video/video_repository.dart';

class VideoDetailsPage extends ConsumerStatefulWidget {
  final File? video;
  const VideoDetailsPage({
    super.key,
    this.video,
  });

  @override
  ConsumerState<VideoDetailsPage> createState() =>
  _VideoDetailsPageState();
}

```



```
class _VideoDetailsPageState extends ConsumerState<VideoDetailsPage> {
  final titleController = TextEditingController();
  final descriptionController = TextEditingController();
  File? image;
  bool isThumbnailSelected = false;
  String randomNumber = const Uuid().v4();
  String videoId = const Uuid().v4();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.only(
            top: 20,
            left: 10,
            right: 10,
          ),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              const Text(
                "Enter the title",
                style: TextStyle(
                  fontSize: 15,
                  color: Colors.grey,
                ),
              ),
              const SizedBox(height: 5),
              TextField(
                controller: titleController,
                decoration: const InputDecoration(
                  hintText: "Enter the title",
                  prefixIcon: Icon(Icons.title),
                  border: OutlineInputBorder(
                    borderSide: BorderSide(
```

```

        color: Colors.blue,
      ),
    ),
  ),
),
const SizedBox(height: 30),
const Text(
  "Enter the Description",
  style: TextStyle(
    fontSize: 15,
    color: Colors.grey,
  ),
),
const SizedBox(height: 5),
TextField(
  controller: descriptionController,
  maxLines: 5,
  decoration: const InputDecoration(
    hintText: "Enter the Description",
    border: OutlineInputBorder(
      borderSide: BorderSide(
        color: Colors.blue,
      ),
    ),
  ),
),
),
// select thumbnail

Padding(
  padding: const EdgeInsets.only(top: 12),
  child: Container(
    decoration: const BoxDecoration(
      color: Colors.blue,
      borderRadius: BorderRadius.all(

```

```

        Radius.circular(11),
      ),
    ),
    child: TextButton(
      onPressed: () async {
        // pick image
        image = await pickImage();
        isThumbnailSelected = true;
        setState(() {});
      },
      child: const Text(
        "SELECT THUMBNAIL",
        style: TextStyle(
          color: Colors.white,
        ),
      ),
    ),
  ),
),
isThumbnailSelected
  ? Padding(
    padding: const EdgeInsets.only(top: 12, bottom: 12),
    child: Image.file(
      image!,
      cacheHeight: 160,
      cacheWidth: 400,
    ),
  )
: const SizedBox(),

isThumbnailSelected
  ? Padding(
    padding: const EdgeInsets.only(top: 12),
    child: Container(
      decoration: const BoxDecoration(

```

```

        color: Colors.green,
        borderRadius: BorderRadius.all(
            Radius.circular(11),
        ),
    ),
    child: TextButton(
        onPressed: () async {
            // publish video

            String thumbnail = await putFileInStorage(
                image, randomNumber, "image");

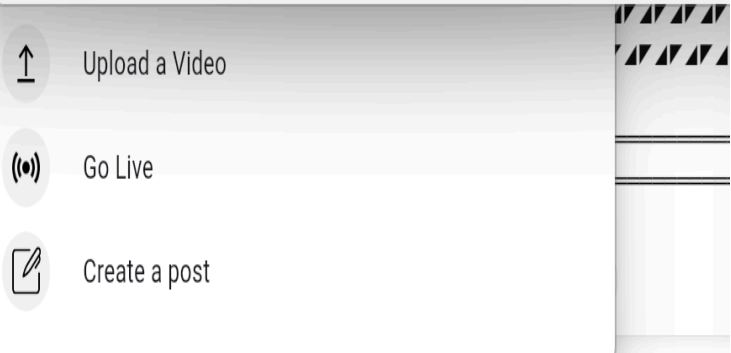
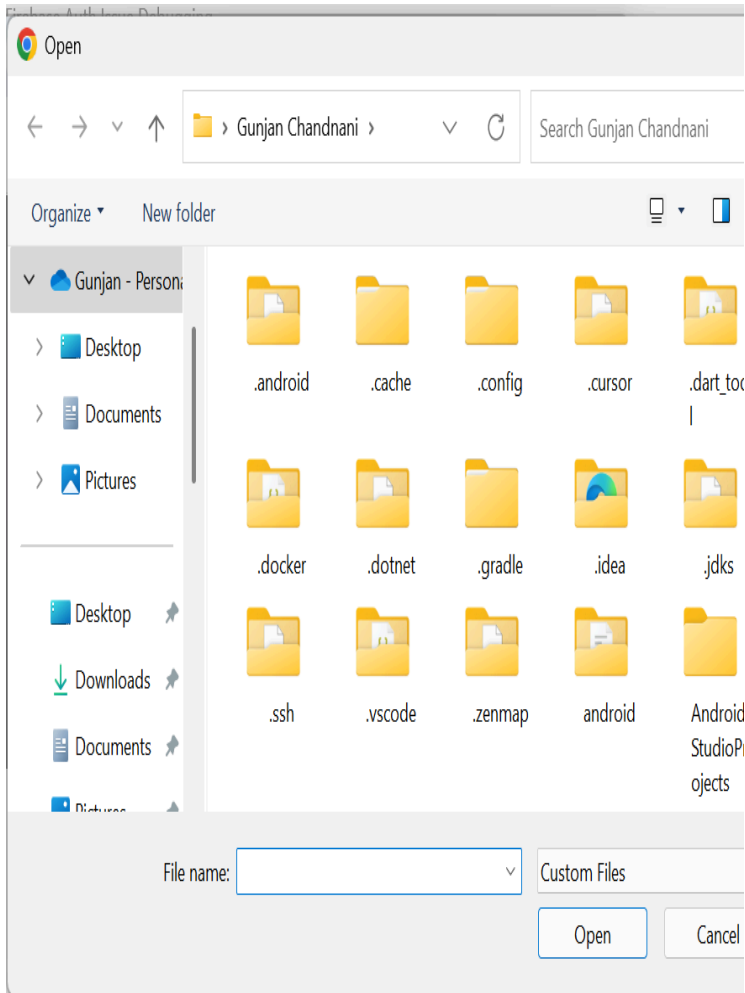
            String videoUrl = await putFileInStorage(
                widget.video, randomNumber, "video");

ref.watch(longVideoProvider).uploadVideoToFirestore(
            videoUrl: videoUrl,
            thumbnail: thumbnail,
            title: titleController.text,
            videoId: videoId,
            datePublished: DateTime.now(),
            userId:
FirebaseAuth.instance.currentUser!.uid,
        );
    },
    child: const Text(
        "Publish",
        style: TextStyle(
            color: Colors.white,
        ),
    ),
),
),
),
)

```

```
        : const SizedBox(),  
        ],  
      ),  
    ),  
  ),  
);  
}  
}
```

OutPut:



Enter the title

T this project by Gunjan - 06[D15A]

Enter the Description

My first yt vdo

SELECT THUMBNAIL



Something went wrong



Create



Create a Short



Upload a Video



Go Live



Create a post