

EXPERIMENT NO: - 03

Name:- Gunjan Chandnani

Class:- D15A

Roll:No: - 06

AIM: - To include icons, images, fonts in Flutter app.

Theory: -

Flutter is a versatile open-source UI framework , which allows developers to build natively compiled applications for mobile, web, and desktop platforms from a single codebase.

One of the key strengths of Flutter is its flexibility in creating highly customizable UIs. This practical focuses on incorporating essential visual elements — icons, images, and custom fonts—into a Flutter application.

These elements enhance the visual appeal and usability of the app, providing an engaging experience for users.

A flutter app when built has both assets (resources) and code. Assets are available and deployed during runtime. The asset is a file that can include static data, configuration files, icons, and images.

The Flutter app supports many image formats, such as JPEG, WebP, PNG, GIF, animated WebP/GIF, BMP, and WBMP.

Visual elements play a significant role in app development.

- Enhanced User Experience: Images and icons make your app visually appealing and userfriendly.
- Information Conveyance: They convey information quickly and intuitively. A wellchosen icon can replace lengthy text.
- Branding: Custom icons and images reinforce your app's branding, making it memorable.

➤ Adding Icons in Flutter Flutter provides built-in material design icons through the Icons class. Custom icons can also be added using third-party packages such as flutter_launcher_icons and font_awesome_flutter. Icon(Icons.home, size: 40,)

➤ Adding Images in Flutter

Flutter supports images from three sources:

1. Assets (Stored locally in the project)

- Place the image inside the assets/images folder in the project.
- Declare the image in pubspec.yaml flutter: assets: - assets/images/sample.png • Display the image in the app

```
Image.asset('assets/images/sample.png');
```

2. Network (Fetched from the internet)

Displaying images from the internet or network is very simple. Flutter provides a built-in method `Image.network` to work with images from a URL. The `Image.network` method also allows you to use some optional properties, such as height, width, color, fit, and many more.

```
Image.network('https://example.com/sample.jpg');
```

3. Memory or File (Stored on the device)

➤ Adding Custom Fonts in Flutter By default, Flutter uses the Roboto font, but custom fonts can be added for a unique UI.

- Download the font and place it in the assets/fonts/ folder.
- Declare the font in pubspec.yaml
- Use the font in the app `Text('Custom Font Example',`

```
style: TextStyle(fontFamily: 'CustomFont', fontSize: 24), );
```

Code:

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

```

upload_bottom_sheet.dart:

```

// ignore_for_file: use_build_context_synchronously

import 'package:flutter/material.dart';
import 'package:youtube_clone/cores/methods.dart';
import 'package:youtube_clone/cores/widgets/image_item.dart';

class CreateBottomSheet extends StatelessWidget {
  const CreateBottomSheet({super.key});

  @override
  Widget build(BuildContext context) {
    return Container(
      color: const Color(0xffFFFFFF),
      child: Padding(
        padding: const EdgeInsets.only(left: 7, top: 12),
        child: SizedBox(
          height: 270,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const Padding(
                padding: EdgeInsets.only(left: 10),
                child: Text(
                  "Create",
                  style: TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.w500,
                  ),
                ),
              ),
              const SizedBox(height: 28),
            ],
          ),
        ),
      ),
    );
  }
}

```

```

        SizedBox(
          height: 38,
          child: ImageItem(
            itemText: "Create a Short",
            itemClicked: () async {
              await pickShortVideo(context);
            },
            imageName: "short-video.png",
            haveColor: true,
          ),
        ),
        const SizedBox(height: 12),
        SizedBox(
          height: 38,
          child: ImageItem(
            itemText: "Upload a Video",
            itemClicked: () async {
              await pickVideo(context);
            },
            imageName: "upload.png",
            haveColor: true,
          ),
        ),
        const SizedBox(height: 12),
        SizedBox(
          height: 38,
          child: ImageItem(
            itemText: "Go Live",
            itemClicked: () {},
            imageName: "go-live.png",
            haveColor: true,
          ),
        ),
        const SizedBox(height: 12),
        SizedBox(
          height: 38,
          child: ImageItem(
            itemText: "Create a post",
            itemClicked: () {},
            imageName: "create-post.png",
            haveColor: true,
          ),
        ),
      ],
    ),
  ),
),
);
}

```

video_model.dart:

```
// ignore_for_file: public_member_api_docs, sort_constructors_first
import 'package:cloud_firestore/cloud_firestore.dart';

class VideoModel {
  final String videoUrl;
  final String thumbnail;
  final String title;
  final DateTime datePublished;
  final int views;
  final String videoId;
  final String userId;
  final List likes;
  final String type;
  VideoModel({
    required this.videoUrl,
    required this.thumbnail,
    required this.title,
    required this.datePublished,
    required this.views,
    required this.videoId,
    required this.userId,
    required this.likes,
    required this.type,
  });

  Map<String, dynamic> toMap() {
    return <String, dynamic>{
      'videoUrl': videoUrl,
      'thumbnail': thumbnail,
      'title': title,
      'datePublished': datePublished,
      'views': views,
      'videoId': videoId,
      'userId': userId,
      'likes': likes,
      'type': type,
    };
  }
}

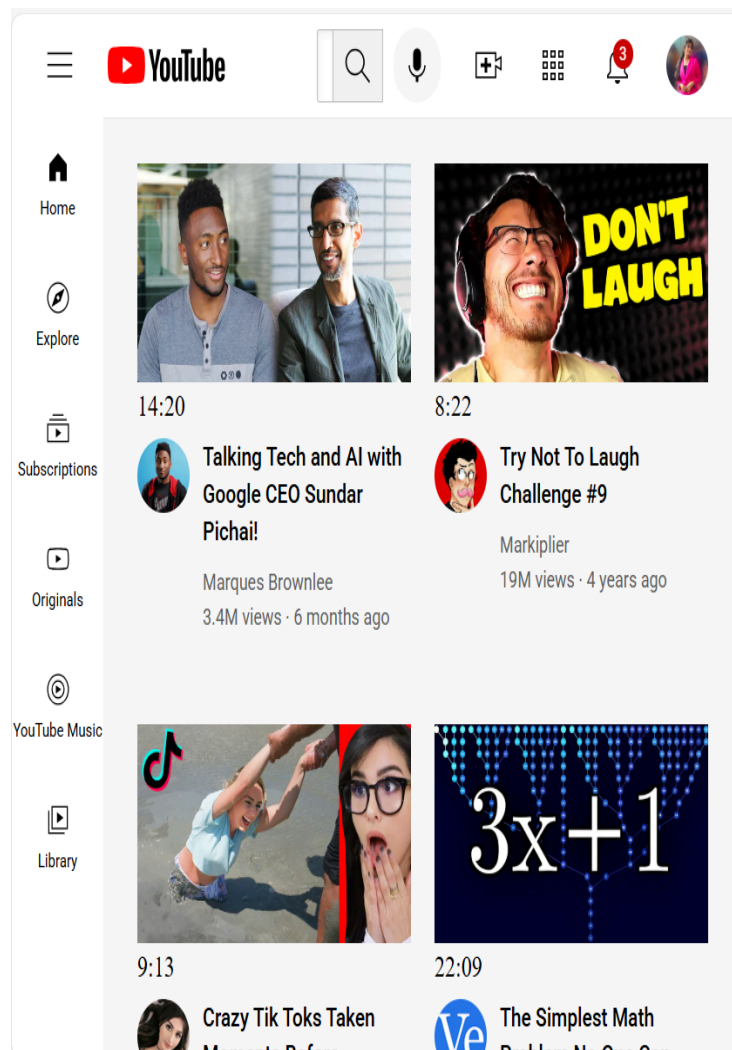
factory VideoModel.fromMap(Map<String, dynamic> map) {
  return VideoModel(
    videoUrl: map['videoUrl'] as String,
    thumbnail: map['thumbnail'] as String,
    title: map['title'] as String,
    datePublished: map["datePublished"] is Timestamp
      ? (map["datePublished"] as Timestamp).toDate()
      : DateTime.fromMillisecondsSinceEpoch(
        map["datepublished"] as int,
```

```

    ),
    views: map['views'] as Int,
    videoId: map['videoId'] as String,
    userId: map['userId'] as String,
    likes: List.from(
        (map['likes'] as List),
    ),
    type: map['type'] as String,
);
}
}

```

OutPut:





Something went wrong



Create



Create a Short



Upload a Video



Go Live



Create a post

Enter the title

T this project by Gunjan - 06[D15A]

Enter the Description

My first yt vdo

SELECT THUMBNAIL