Name: Gunjan Chandnani Roll No: 06 Div: D15A

Experiment 02

Aim:

To design Flutter UI by including common widgets.

Theory:

Flutter provides a wide range of pre-built widgets that help developers create visually appealing and responsive user interfaces. Widgets in Flutter can be classified into two main types: Stateless Widgets, which remain unchanged once built, and Stateful Widgets, which can dynamically update based on user interactions.

To structure a UI layout, Container is commonly used as a fundamental widget that holds other widgets while allowing customization with padding, margins, and colors. Row and Column widgets help arrange elements horizontally and vertically, respectively, ensuring a flexible and responsive design. For overlapping elements, Stack is used, allowing widgets to be layered on top of each other.

Flutter also provides interactive widgets such as TextField for user input, Button widgets like ElevatedButton and FloatingActionButton for user interactions, and ListView for displaying scrollable lists. Additionally, Card is a useful widget for creating material design-like structured components with shadows and elevation effects.

For navigation, Flutter includes Navigator and Drawer, which help in managing multiple screens and creating side menus. Styling and customization can be done using ThemeData, which enables consistent design throughout the app. With these common widgets, developers can build intuitive, responsive, and visually rich applications efficiently.

Types of widgets:

1. Based on State Management

1.1 Stateless Widgets

Stateless widgets are immutable, meaning their properties do not change once they are built. They are used for displaying static content that does not depend on user interaction or external data. Examples include `Text`, `Icon`, and `Container`.

1.2 Stateful Widgets

Stateful widgets maintain an internal state that can change dynamically based on user interactions or data updates. They require a `State` object to manage changes. Examples include `TextField`, `Checkbox`, and `ListView`.

---

2. Based on Functionality

2.1 Layout Widgets

Layout widgets help structure the UI by arranging elements in a specific order.

- Container: A flexible box that holds child widgets with padding, margins, and styling.
- Row & Column: Arrange widgets horizontally (`Row`) or vertically (`Column`).
- Stack: Overlays widgets on top of each other for layered UI designs.
- Expanded & Flexible: Adjust child widget sizes dynamically based on available space.

2.2 Interactive Widgets

These widgets respond to user actions like clicks, typing, or gestures.

- TextField: Allows user input.
- ElevatedButton & TextButton: Buttons for user interaction.
- FloatingActionButton (FAB): A circular button used for quick actions.
- GestureDetector: Detects taps, swipes, and other gestures.

2.3 Scrolling Widgets

These widgets help display content in a scrollable format.

- ListView: Displays a scrollable list of items.
- GridView: Shows items in a grid format.
- SingleChildScrollView: Enables scrolling for a single child widget.

2.4 Styling & Theming Widgets

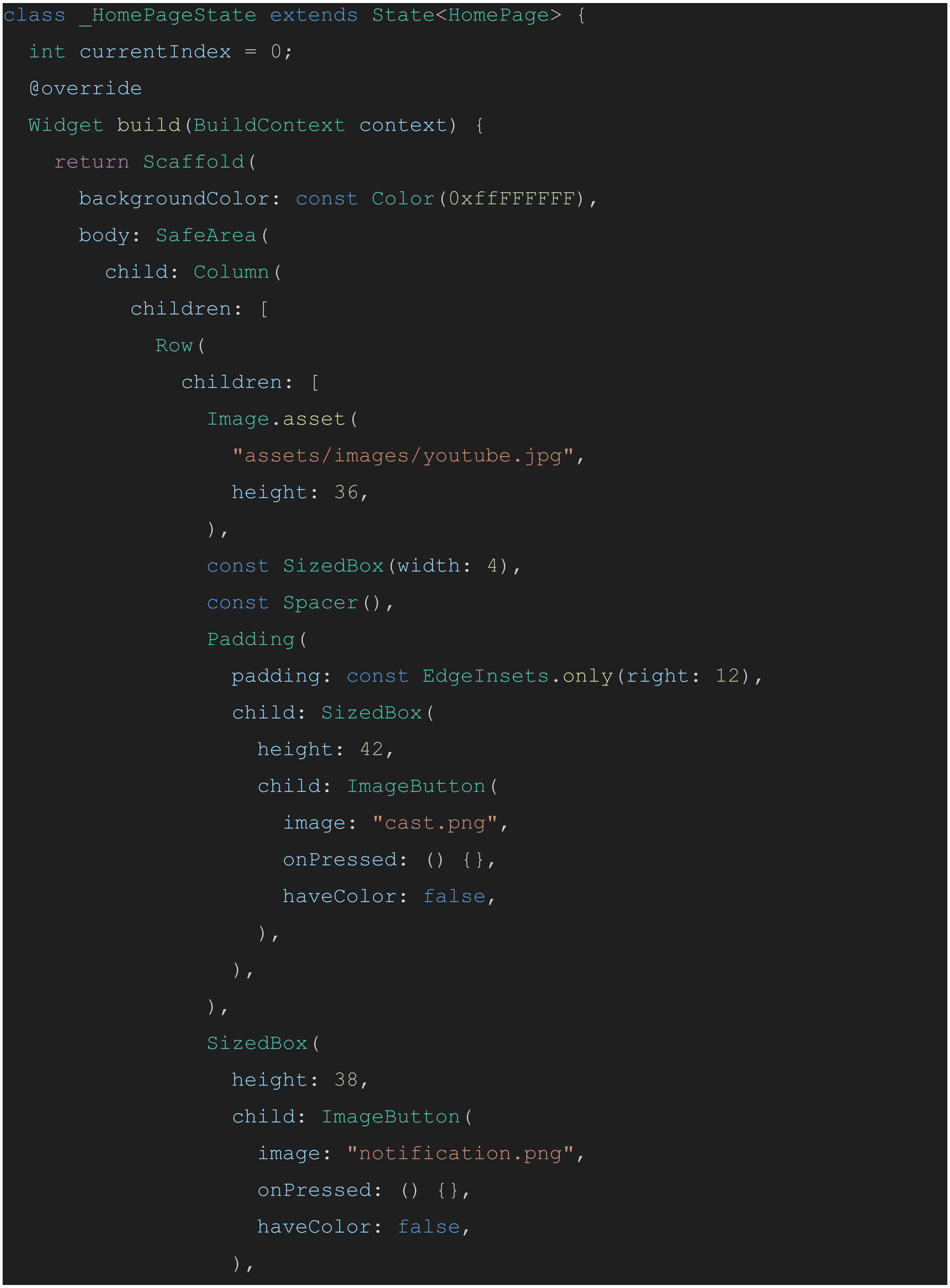
These widgets help in customizing the UI.

- Padding & Margin: Add spacing around widgets.
- DecoratedBox: Provides styling like background color, border, and shadows.
- Theme & ThemeData: Defines app-wide themes and styles.

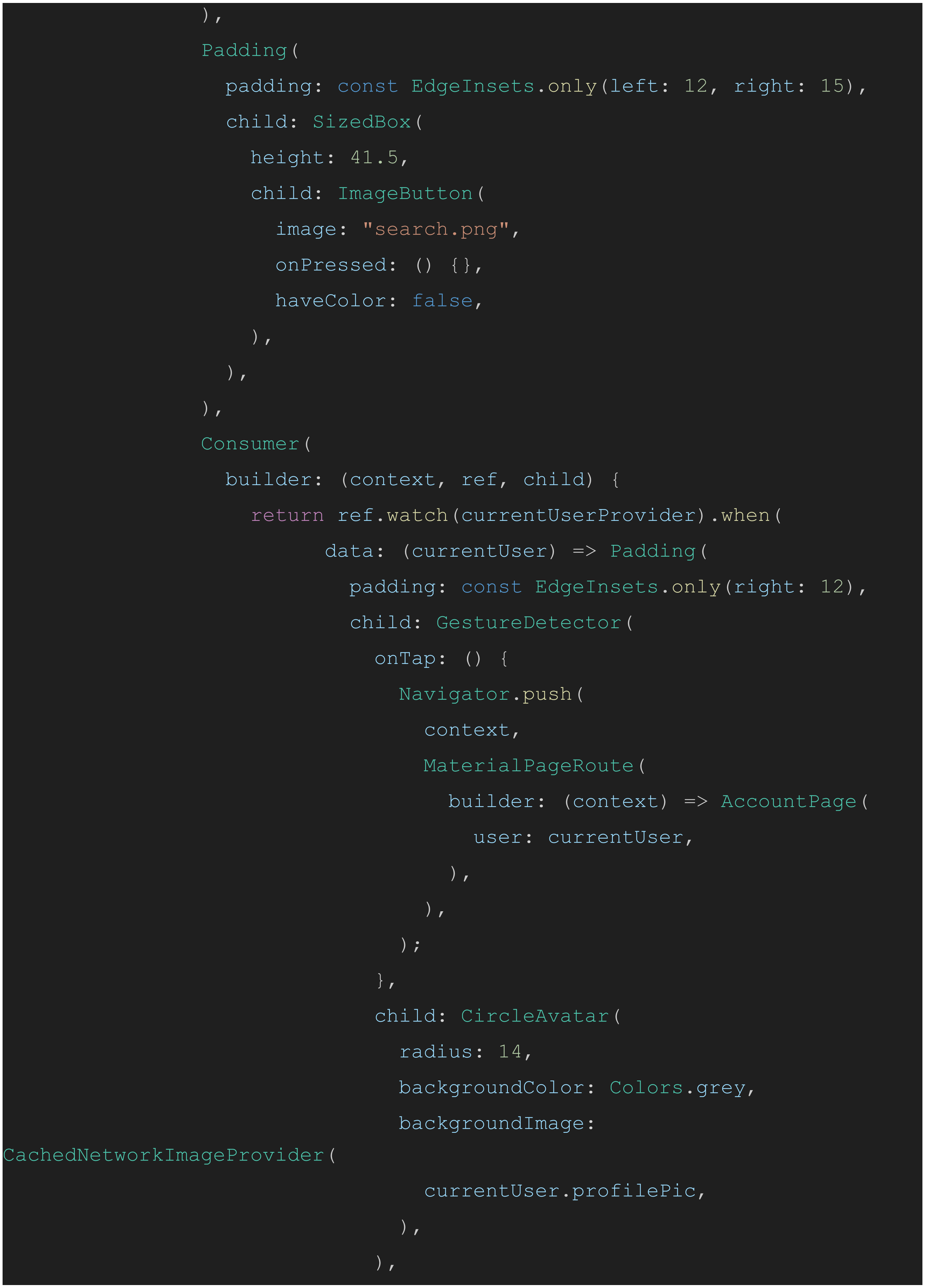
Code:

Home_page.dart

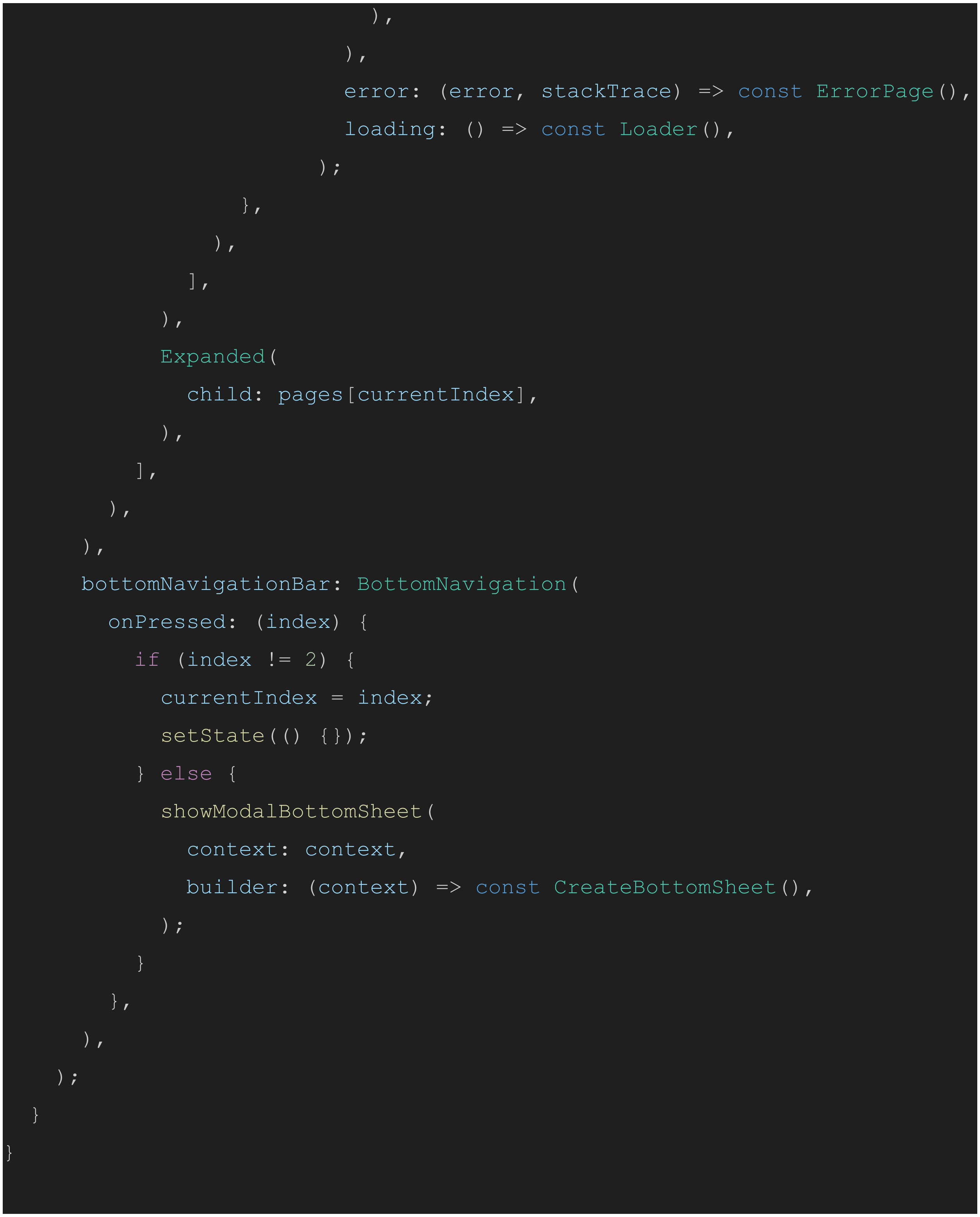
```dart
// ignore_for_file: invalid_use_of_visible_for_testing_member

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();
}
```



```dart
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,
                    ),
                  ),
                ),
                SizedBox(
                  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,
                                ),
                              ),
```



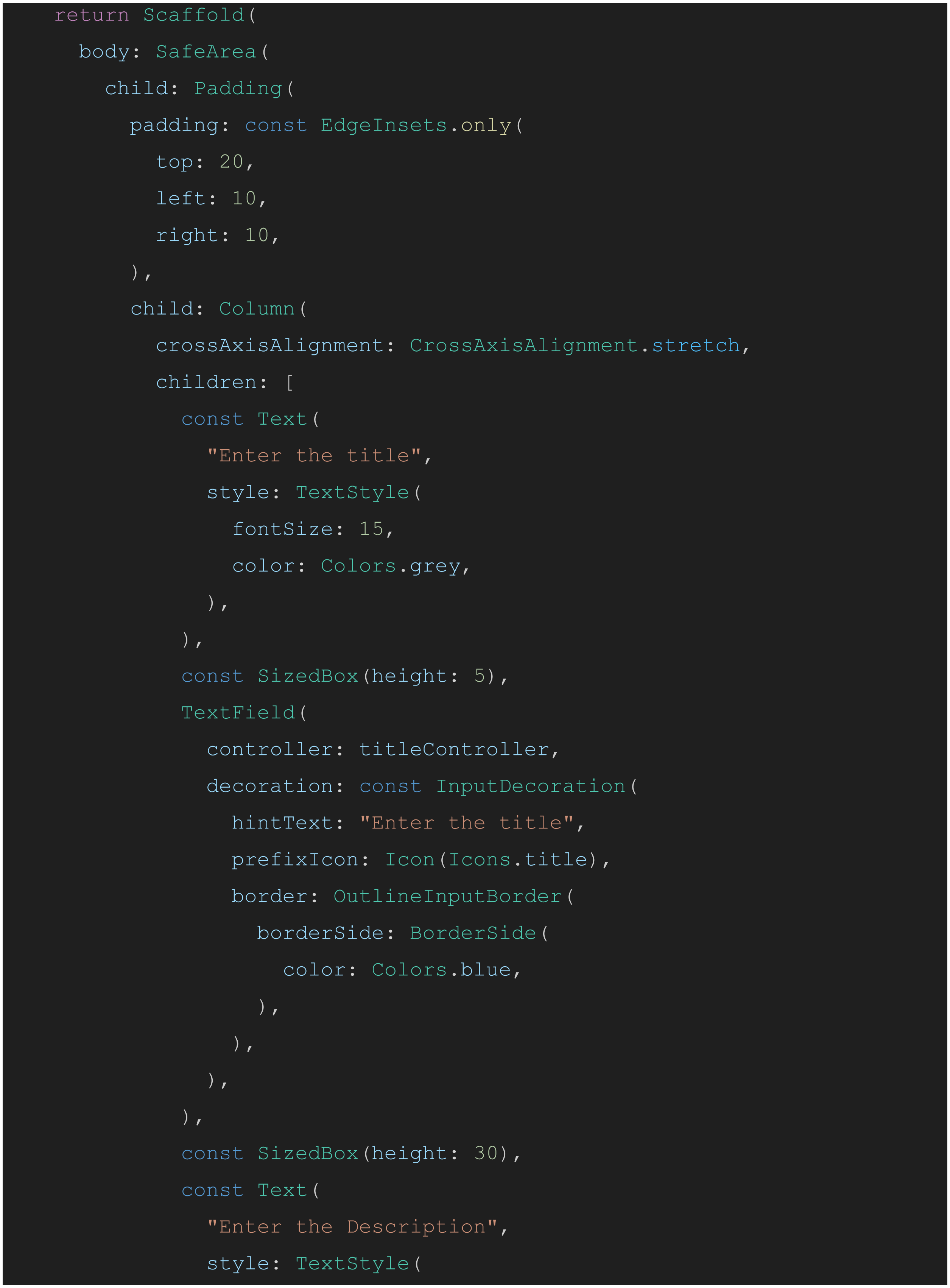
```
                            ),
                          ),
                          error: (error, stackTrace) => const ErrorPage(),
                          loading: () => const Loader(),
                        );
                  },
                ),
              ],
            ),
     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

Main.dart

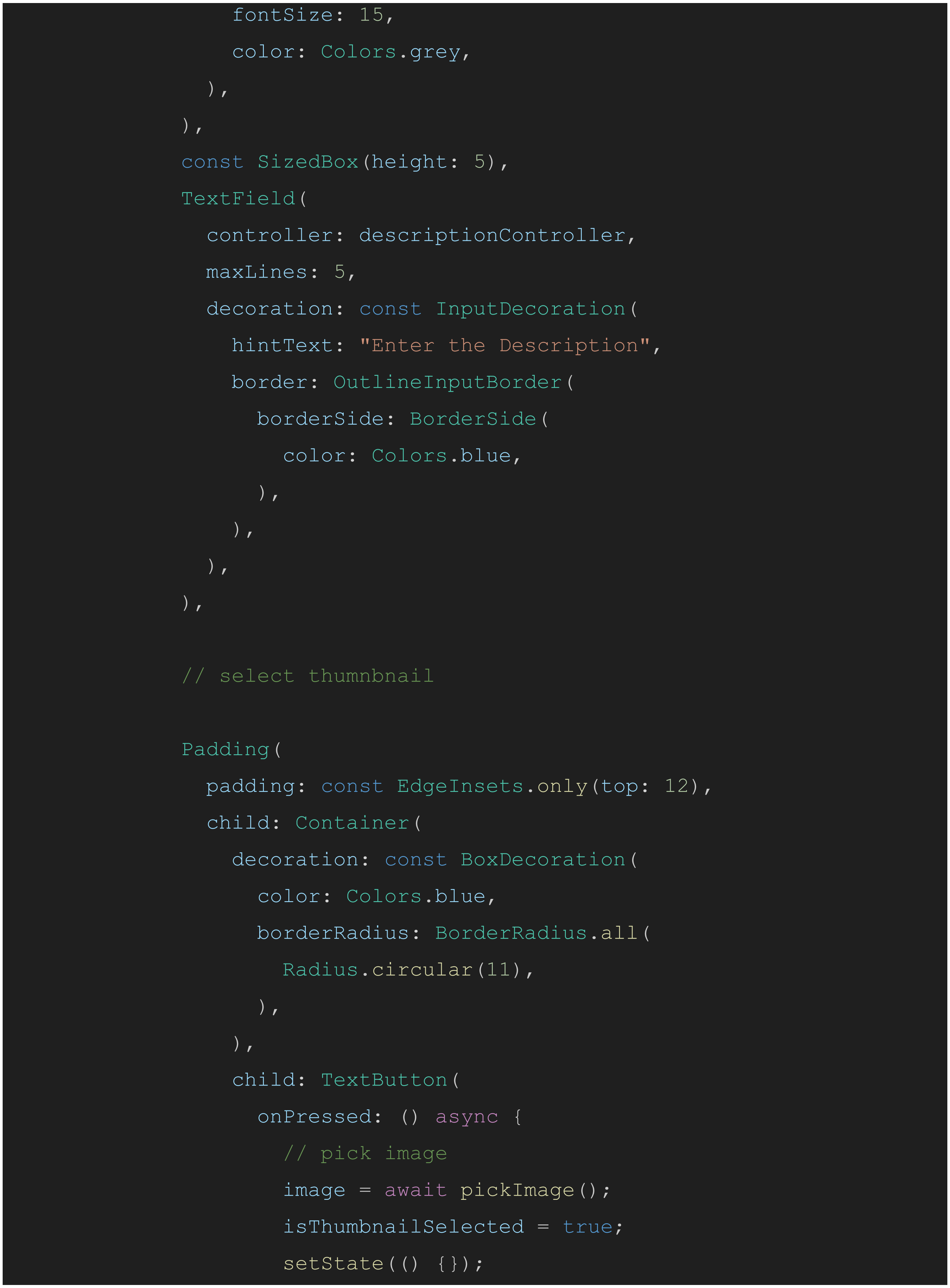
```
// ignore_for_file: unnecessary_null_comparison, prefer_const_constructors

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 wthether 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

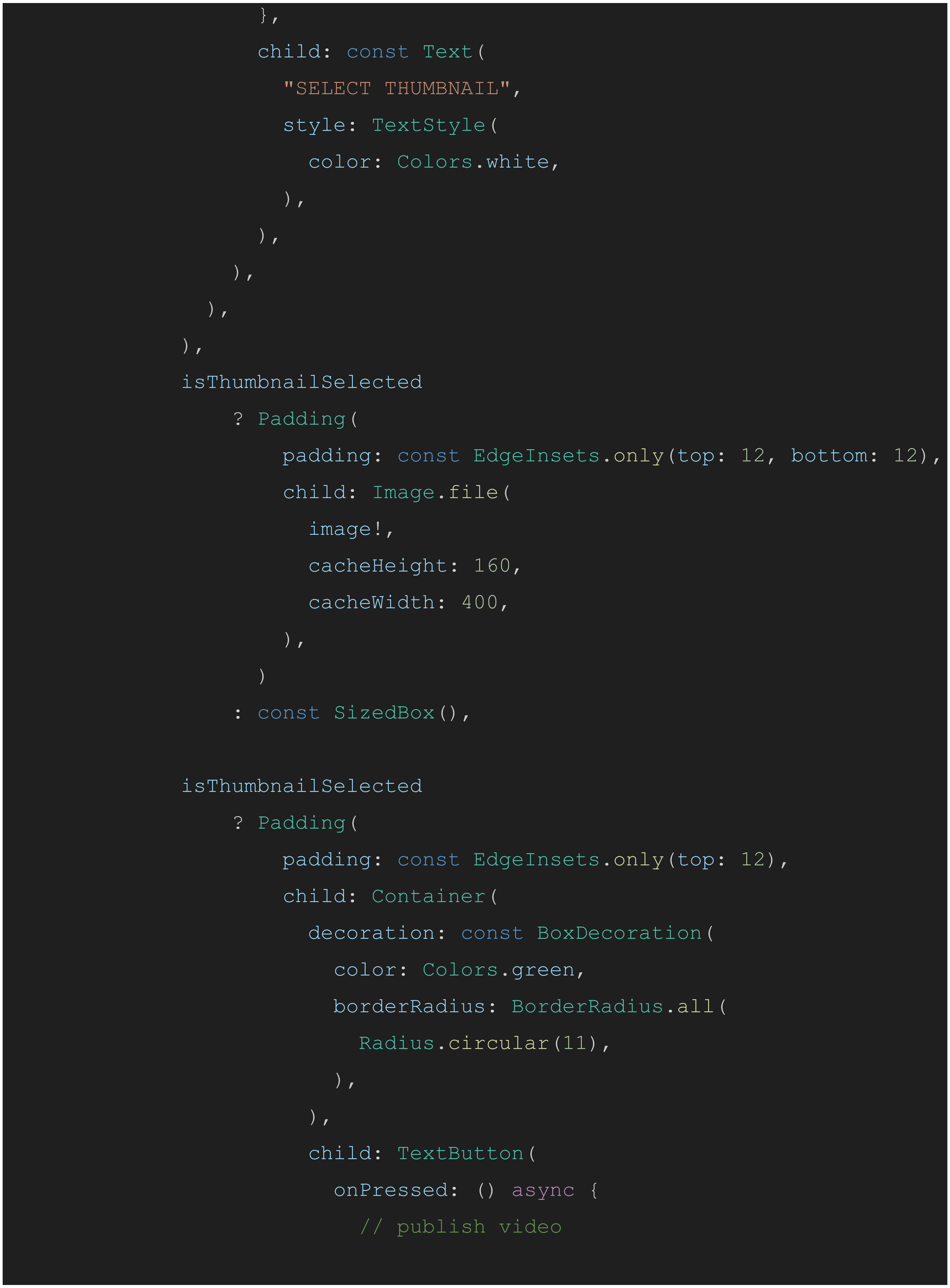
```
// ignore_for_file: public_member_api_docs, sort_constructors_first
import 'dart:io';

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) {
```



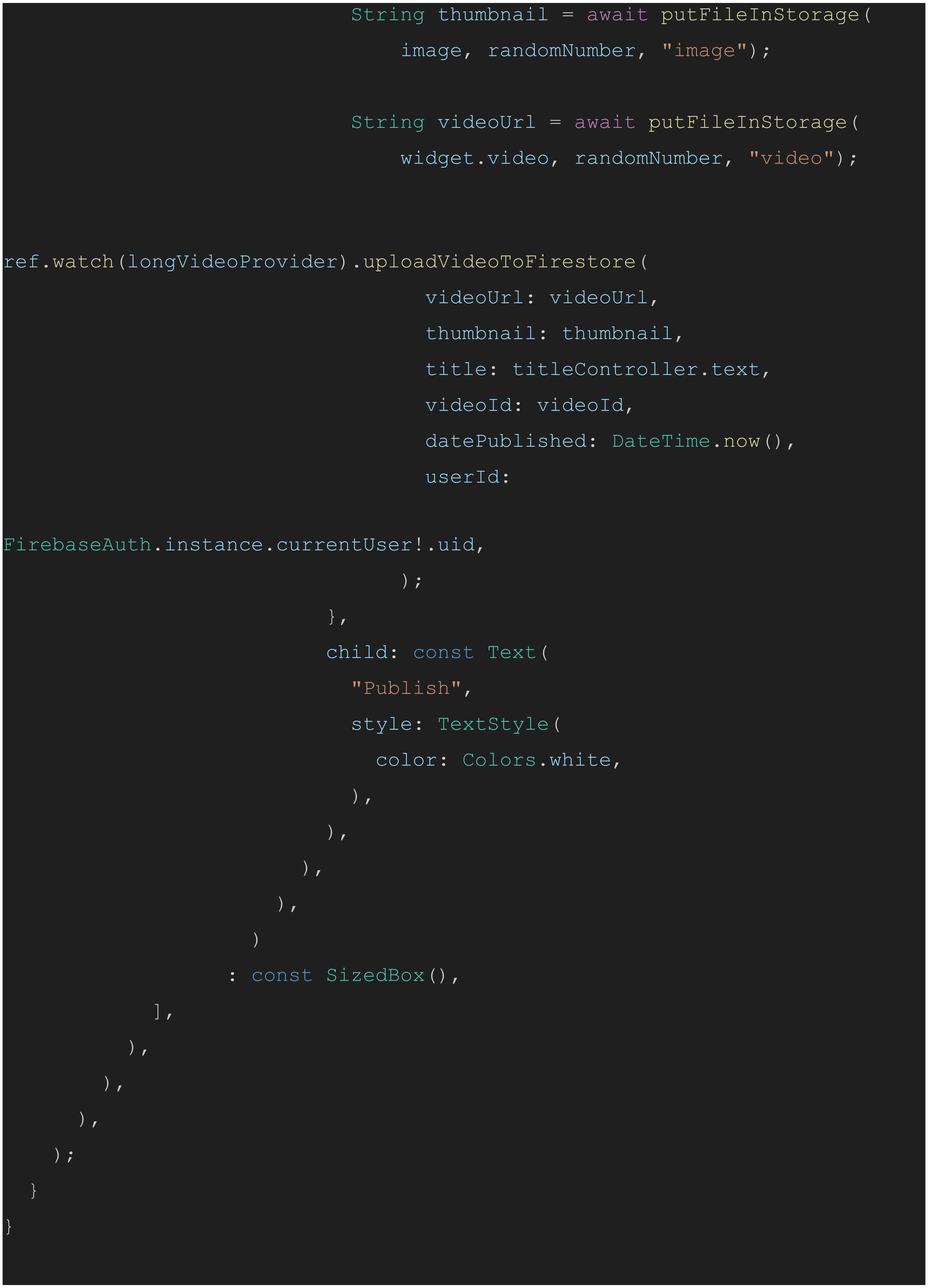
```
    return Scaffold(
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.only(
            top: 20,
            left: 10,
            right: 10,
          ),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              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 thumnbnail

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(() {});
```

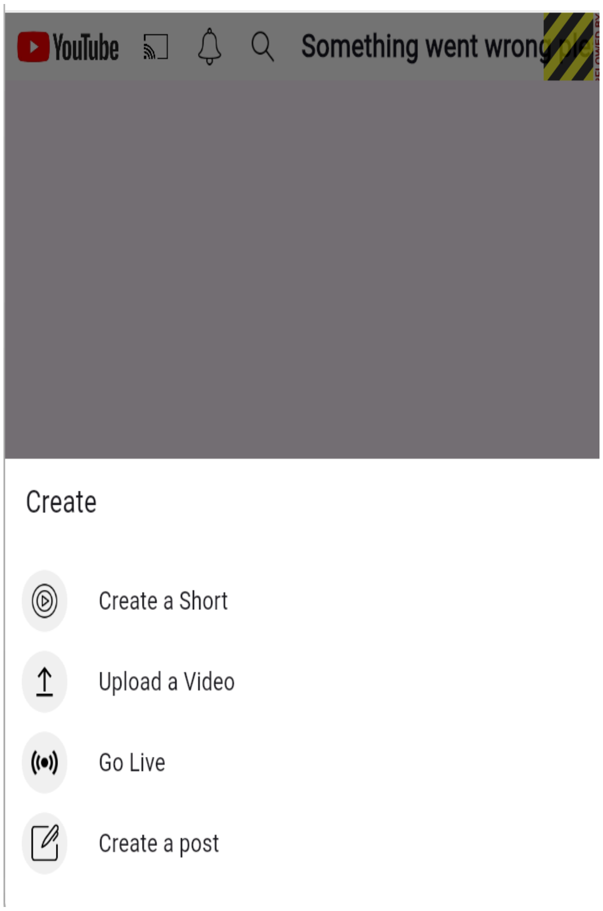


```
      },
      child: const Text(
        "SELECT THUMBNAIL",
        style: TextStyle(
          color: Colors.white,
        ),
      ),
    ),
  ),
),
isThumbnailSelected
    ? Padding(
        padding: const EdgeInsets.only(top: 12, bottom: 12),
        child: Image.file(
          image!,
          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
```

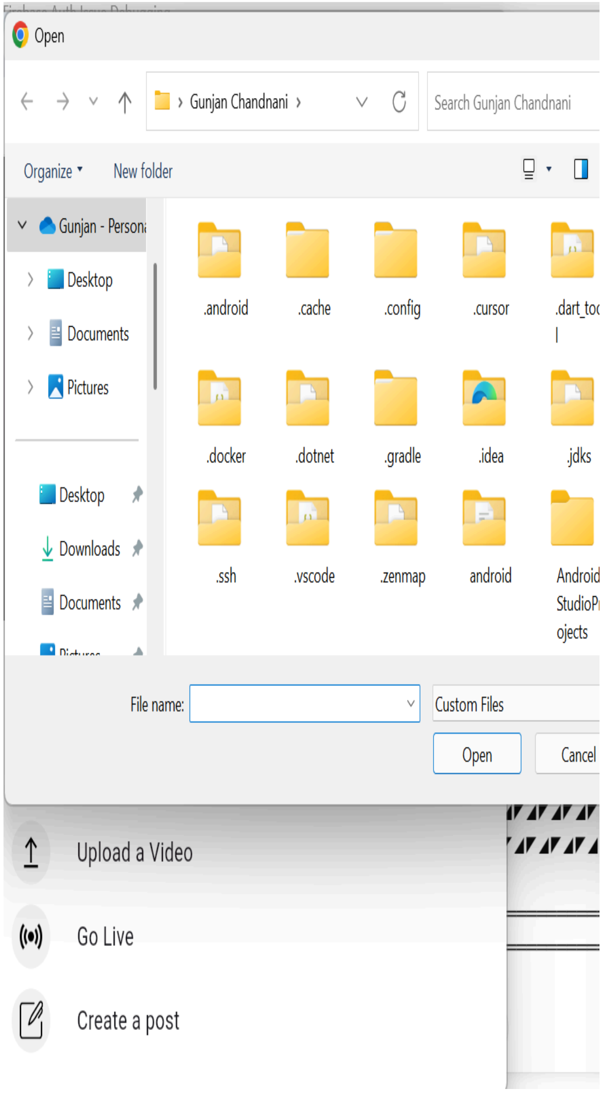


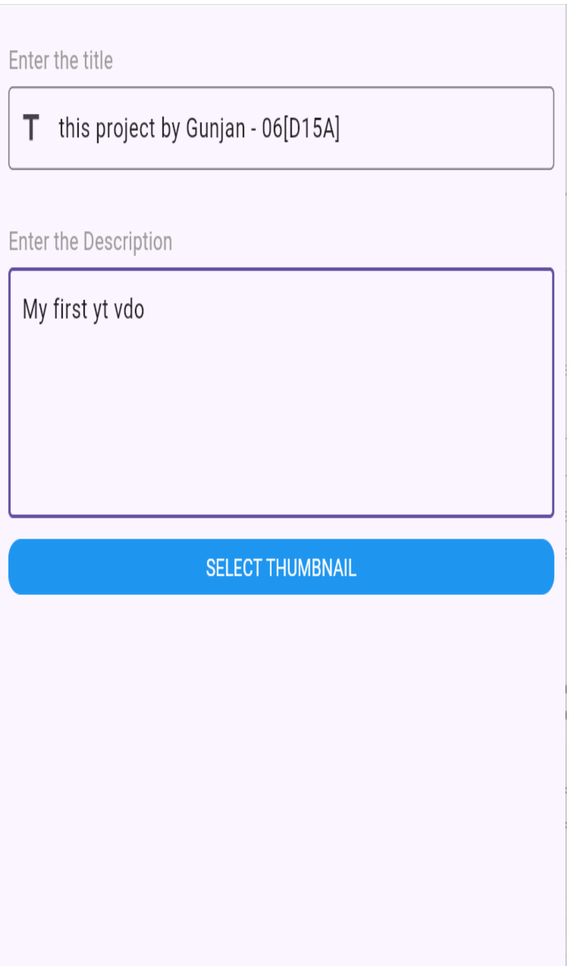
```
                            String thumbnail = await putFileInStorage(
                                image, randomNumber, "image");

                            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:



Something went wrong



 Enter the title this project by Gunjan - 06[D15A] Enter the Description My first yt vdo 	