Name: Vedang Khandagale
Division: D15A
Roll No: 29
Batch: B

Experiment No 6

Aim: To Connect Flutter UI with Firebase

Theory:

FlutterFire is a set of Flutter plugins that enable Flutter developers to integrate their applications with various Firebase services. Firebase is a comprehensive mobile and web application development platform provided by Google. FlutterFire is specifically designed to provide Flutter developers with a seamless way to interact with Firebase services.

Key features of FlutterFire include:

1. Firebase Authentication: FlutterFire provides plugins to easily integrate Firebase Authentication, allowing developers to implement user sign-up, sign-in, and password recovery features in their Flutter applications. Firebase supports various authentication methods, including email/password, Google Sign-In, Facebook Sign-In, and more.

2. Cloud Firestore and Realtime Database: FlutterFire supports both Cloud Firestore and Firebase Realtime Database, enabling developers to store and retrieve data in real-time. Firestore is a NoSQL document database, while Realtime Database is a JSON-based database.

3. Cloud Functions: Developers can deploy serverless functions using Cloud Functions for Firebase, and FlutterFire allows Flutter apps to trigger and interact with these functions.

4. Cloud Storage: FlutterFire supports Firebase Cloud Storage, allowing developers to upload, download, and manage files in the cloud. This is useful for handling user-generated content, such as images or videos.

5. Firebase Cloud Messaging (FCM): FCM enables developers to send push notifications to their Flutter applications. FlutterFire provides plugins for integrating FCM and handling push notifications.

6. Firebase Performance Monitoring: Developers can monitor the performance of their Flutter applications using Firebase Performance Monitoring. This includes measuring app startup time, screen rendering, and network performance.

7. Firebase Analytics: FlutterFire includes plugins for integrating Firebase Analytics, enabling developers to gain insights into user behavior and app usage.

8. Firebase Remote Config: FlutterFire supports Firebase Remote Config, allowing developers to remotely configure app behavior without publishing updates. This is useful for A/B testing and feature toggling.

9. Firebase Crashlytics: FlutterFire includes support for Firebase Crashlytics, providing real-time crash reporting to help developers identify and fix issues quickly.

10. Firebase AdMob: FlutterFire includes AdMob plugins for integrating advertisements into Flutter applications using Firebase AdMob.

Firebase API code:

```dart
// File generated by FlutterFire CLI.
// ignore_for_file: lines_longer_than_80_chars,
avoid_classes_with_only_static_members
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
import 'package:flutter/foundation.dart'
    show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.
///
/// Example:
/// ```dart
/// import 'firebase_options.dart';
/// // ...
/// await Firebase.initializeApp(
///   options: DefaultFirebaseOptions.currentPlatform,
/// );
/// ```
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      return web;
    }
    switch (defaultTargetPlatform) {
      case TargetPlatform.android:
        return android;
      case TargetPlatform.iOS:
        return ios;
      case TargetPlatform.macOS:
        throw UnsupportedError(
```

```dart
          'DefaultFirebaseOptions have not been configured for macos - '
          'you can reconfigure this by running the FlutterFire CLI
again.',
        );
      case TargetPlatform.windows:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for windows - '
          'you can reconfigure this by running the FlutterFire CLI
again.',
        );
      case TargetPlatform.linux:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for linux - '
          'you can reconfigure this by running the FlutterFire CLI
again.',
        );
      default:
        throw UnsupportedError(
          'DefaultFirebaseOptions are not supported for this platform.',
        );
    }
  }

  static const FirebaseOptions web = FirebaseOptions(
    apiKey: 'AIzaSyBg9-swGAvdOLn4vhuIu2PM70VW5ew7wqU',
    appId: '1:466675301682:web:61da688dacd6dccf32ab2b',
    messagingSenderId: '466675301682',
    projectId: 'facebook-clone-5f3aa',
    authDomain: 'facebook-clone-5f3aa.firebaseapp.com',
    storageBucket: 'facebook-clone-5f3aa.appspot.com',
    measurementId: 'G-HZCQMPK1MZ',
  );

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyAbozls46qv0OvzLeNxnuq-nQRpiz_7gVw',
    appId: '1:466675301682:android:40a4a7be8fb3eba732ab2b',
    messagingSenderId: '466675301682',
    projectId: 'facebook-clone-5f3aa',
    storageBucket: 'facebook-clone-5f3aa.appspot.com',
  );
  static const FirebaseOptions ios = FirebaseOptions(
    apiKey: 'AIzaSyB1gwDnGzEv1MZ3VglmVCGgzrzKnXHabSQ',
    appId: '1:466675301682:ios:6283eb03b0d3702832ab2b',
    messagingSenderId: '466675301682',
    projectId: 'facebook-clone-5f3aa',
    storageBucket: 'facebook-clone-5f3aa.appspot.com',
    iosBundleId: 'com.example.facebookClone',
  );
}
```

Login Screen Code:

```
import
'package:facebook_clone/features/auth/presentation/screens/create_account_
screee.dart';
import
'package:facebook_clone/features/auth/providers/auth_provider.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';

import '/core/constants/constants.dart';
import '/core/widgets/round_button.dart';
import '/core/widgets/round_text_field.dart';
import '/features/auth/utils/utils.dart';

final _formKey = GlobalKey<FormState>();

class LoginScreen extends ConsumerStatefulWidget {
  const LoginScreen({super.key});

  @override
  ConsumerState<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends ConsumerState<LoginScreen> {
  late final TextEditingController _emailController;
  late final TextEditingController _passwordController;
```

```dart
bool isLoading = false;

@override
void initState() {
  _emailController = TextEditingController();
  _passwordController = TextEditingController();
  super.initState();
}

@override
void dispose() {
  _emailController.dispose();
  _passwordController.dispose();
  super.dispose();
}

Future<void> login() async {
  if (_formKey.currentState!.validate()) {
    _formKey.currentState!.save();
    setState(() => isLoading = true);
    await ref.read(authProvider).signIn(
          email: _emailController.text,
          password: _passwordController.text,
        );
    setState(() => isLoading = false);
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(),
    body: Padding(
      padding: Constants.defaultPadding,
      child: Column(
        mainAxisSize: MainAxisSize.max,
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          Image.asset(
            'assets/icons/fb_logo.png',
            width: 60,
          ),
          Form(
            key: _formKey,
            child: Column(
              children: [
                RoundTextField(
                  controller: _emailController,
                  hintText: 'Email',
                  keyboardType: TextInputType.emailAddress,
                  textInputAction: TextInputAction.next,
                  validator: validateEmail,
                ),
                const SizedBox(height: 15),
                RoundTextField(
                  controller: _passwordController,
                  hintText: 'Password',
                  keyboardType: TextInputType.visiblePassword,
                  textInputAction: TextInputAction.done,
                  isPassword: true,
                  validator: validatePassword,
                ),
                const SizedBox(height: 15),
                RoundButton(onPressed: login, label: 'Login'),
                const SizedBox(height: 15),
                const Text(
                  'Forget Password',
                  style: TextStyle(fontSize: 18),
                ),
              ],
            ),
          ),
          Column(
            children: [
              RoundButton(
                onPressed: () {
                  Navigator.of(context).pushNamed(
                    CreateAccountScreen.routeName,
                  );
                },
                  label: 'Create new account',
                  color: Colors.transparent,
                ),
                Image.asset(
                  'assets/icons/meta.png',
                  height: 50,
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

Create Account Screen Code:

```
import 'dart:io';

import 'package:facebook_clone/core/constants/app_colors.dart';
import 'package:facebook_clone/core/constants/constants.dart';
import 'package:facebook_clone/core/utils/utils.dart';
import 'package:facebook_clone/core/widgets/pick_image_widget.dart';
import 'package:facebook_clone/core/widgets/round_button.dart';
import 'package:facebook_clone/core/widgets/round_text_field.dart';
import
'package:facebook_clone/features/auth/presentation/widgets/birthday_picker
.dart';
import
'package:facebook_clone/features/auth/presentation/widgets/gender_picker.d
art';
import
'package:facebook_clone/features/auth/providers/auth_provider.dart';
import 'package:facebook_clone/features/auth/utils/utils.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';

final _formKey = GlobalKey<FormState>();
```

```dart
class CreateAccountScreen extends ConsumerStatefulWidget {
  const CreateAccountScreen({super.key});

  static const routeName = '/create-account';

  @override
  ConsumerState<CreateAccountScreen> createState() =>
      _CreateAccountScreenState();
}

class _CreateAccountScreenState extends ConsumerState<CreateAccountScreen>
{
  File? image;
  DateTime? birthday;
  String gender = 'male';
  bool isLoading = false;

  // controllers
  late final TextEditingController _fNameController;
  late final TextEditingController _lNameController;
  late final TextEditingController _emailController;
  late final TextEditingController _passwordController;

  @override
  void initState() {
    _fNameController = TextEditingController();
    _lNameController = TextEditingController();
    _emailController = TextEditingController();
    _passwordController = TextEditingController();
    super.initState();
  }

  @override
  void dispose() {
    _fNameController.dispose();
    _lNameController.dispose();
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }
```

```dart
Future<void> createAccount() async {
  if (_formKey.currentState!.validate()) {
    _formKey.currentState!.save();
    setState(() => isLoading = true);
    await ref
        .read(authProvider)
        .createAccount(
          fullName: '${_fNameController.text} ${_lNameController.text}',
          birthday: birthday ?? DateTime.now(),
          gender: gender,
          email: _emailController.text,
          password: _passwordController.text,
          image: image,
        )
        .then((credential) {
      if (!credential!.user!.emailVerified) {
        Navigator.pop(context);
      }
    }).catchError((_) {
      setState(() => isLoading = false);
    });
    setState(() => isLoading = false);
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: AppColors.realWhiteColor,
    appBar: AppBar(),
    body: SingleChildScrollView(
      child: Padding(
        padding: Constants.defaultPadding,
        child: Form(
          key: _formKey,
          child: Column(
            children: [
              GestureDetector(
                onTap: () async {
```

```dart
    image = await pickImage();
    setState(() {});
  },
  child: PickImageWidget(image: image),
),
const SizedBox(height: 20),
Row(
  children: [
    // First Name Text Field
    Expanded(
      child: RoundTextField(
        controller: _fNameController,
        hintText: 'First name',
        textInputAction: TextInputAction.next,
        validator: validateName,
      ),
    ),
    const SizedBox(width: 10),
    // Last Name Text Field
    Expanded(
      child: RoundTextField(
        controller: _lNameController,
        hintText: 'Last name',
        textInputAction: TextInputAction.next,
        validator: validateName,
      ),
    ),
  ],
),
const SizedBox(height: 20),
BirthdayPicker(
  dateTime: birthday ?? DateTime.now(),
  onPressed: () async {
    birthday = await pickSimpleDate(
      context: context,
      date: birthday,
    );
    setState(() {});
  },
),
            const SizedBox(height: 20),
            GenderPicker(
              gender: gender,
              onChanged: (value) {
                gender = value ?? 'male';
                setState(() {});
              },
            ),
            const SizedBox(height: 20),
            // Phone number / email text field
            RoundTextField(
              controller: _emailController,
              hintText: 'Email',
              textInputAction: TextInputAction.next,
              keyboardType: TextInputType.emailAddress,
              validator: validateEmail,
            ),
            const SizedBox(height: 20),
            // Password Text Field
            RoundTextField(
              controller: _passwordController,
              hintText: 'Password',
              textInputAction: TextInputAction.done,
              keyboardType: TextInputType.visiblePassword,
              validator: validatePassword,
              isPassword: true,
            ),
            const SizedBox(height: 20),
            isLoading
                ? const Center(child: CircularProgressIndicator())
                : RoundButton(
                    onPressed: createAccount,
                    label: 'Create Account',
                  ),
          ],
        ),
      ),
    ),
  ),
);
```

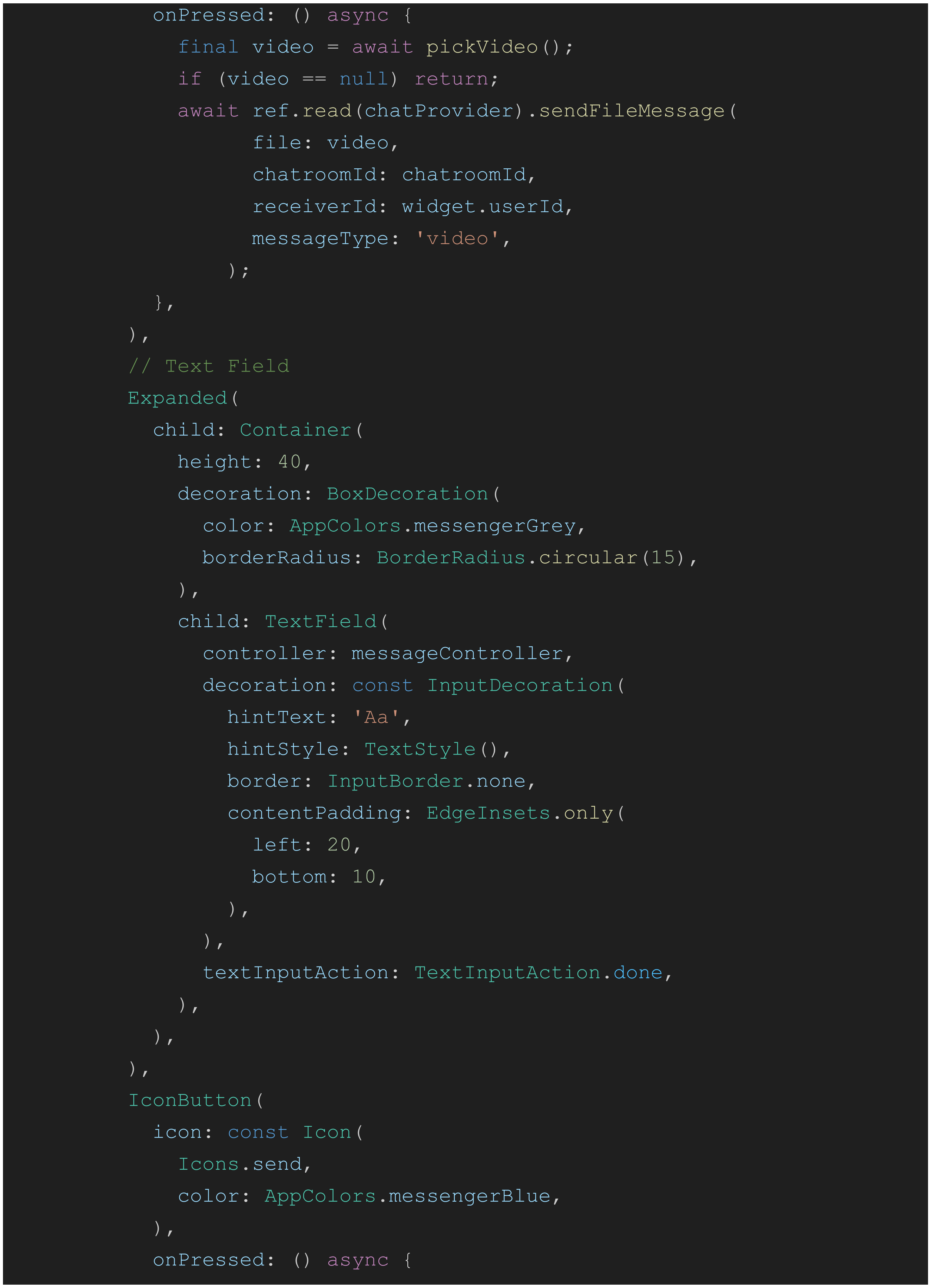
```
  }
}

```

Code to Fetch All Posts:

```
import 'dart:async';

import 'package:cloud_firestore/cloud_firestore.dart';
import
'package:facebook_clone/core/constants/firebaes_collection_names.dart';
import 'package:facebook_clone/core/constants/firebase_field_names.dart';
import 'package:facebook_clone/features/posts/models/post.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';

final getAllPostsProvider =
StreamProvider.autoDispose<Iterable<Post>>((ref) {
  final controller = StreamController<Iterable<Post>>();

  final sub = FirebaseFirestore.instance
      .collection(FirebaseCollectionNames.posts)
      .orderBy(FirebaseFieldNames.datePublished, descending: true)
      .snapshots()
      .listen((snapshot) {
    final posts = snapshot.docs.map(
      (postData) => Post.fromMap(
        postData.data(),
      ),
    );
    controller.sink.add(posts);
  });

  ref.onDispose(() {
    sub.cancel();
    controller.close();
  });

  return controller.stream;
});

```

Chat Screen Code:

```
import 'package:facebook_clone/core/constants/app_colors.dart';
import 'package:facebook_clone/core/screens/loader.dart';
import 'package:facebook_clone/core/utils/utils.dart';
import
'package:facebook_clone/features/chat/presentation/widgets/chats_user_info
.dart';
import
'package:facebook_clone/features/chat/presentation/widgets/messages_list.d
art';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';

import
'package:facebook_clone/features/chat/providers/chat_provider.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:image_picker/image_picker.dart';

class ChatScreen extends ConsumerStatefulWidget {
  const ChatScreen({
    super.key,
    required this.userId,
  });

  final String userId;

  static const routeName = '/chat-screen';

  @override
  ConsumerState<ConsumerStatefulWidget> createState() =>
_ChatScreenState();
}

class _ChatScreenState extends ConsumerState<ChatScreen> {
  late final TextEditingController messageController;
  late final String chatroomId;

  @override
  void initState() {
    messageController = TextEditingController();
    super.initState();
  }

  @override
  void dispose() {
    messageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
      future: ref.watch(chatProvider).createChatroom(userId:
widget.userId),
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Loader();
        }

        chatroomId = snapshot.data ?? 'No chatroom Id';

        return Scaffold(
          backgroundColor: AppColors.realWhiteColor,
          appBar: AppBar(
            leading: IconButton(
              onPressed: Navigator.of(context).pop,
              icon: const Icon(
                Icons.arrow_back_ios,
                color: AppColors.messengerBlue,
              ),
            ),
            titleSpacing: 0,
            title: ChatUserInfo(
              userId: widget.userId,
            ),
          ),
          body: Column(
            children: [
              Expanded(
                child: MessagesList(
                  chatroomId: chatroomId,
                ),
              ),
              const Divider(),
              _buildMessageInput(),
            ],
          ),
        );
      },
    );
  }

  // Chat Text Field
  Widget _buildMessageInput() {
    return Container(
      padding: const EdgeInsets.all(8.0),
      child: Row(
        children: [
          IconButton(
            icon: const Icon(
              Icons.image,
              color: AppColors.messengerDarkGrey,
            ),
            onPressed: () async {
              final image = await pickImage();
              if (image == null) return;
              await ref.read(chatProvider).sendFileMessage(
                    file: image,
                    chatroomId: chatroomId,
                    receiverId: widget.userId,
                    messageType: 'image',
                  );
            },
          ),
          IconButton(
            icon: const Icon(
              FontAwesomeIcons.video,
              color: AppColors.messengerDarkGrey,
              size: 20,
            ),
```


```dart
  onPressed: () async {
    final video = await pickVideo();
    if (video == null) return;
    await ref.read(chatProvider).sendFileMessage(
          file: video,
          chatroomId: chatroomId,
          receiverId: widget.userId,
          messageType: 'video',
        );
  },
),
// Text Field
Expanded(
  child: Container(
    height: 40,
    decoration: BoxDecoration(
      color: AppColors.messengerGrey,
      borderRadius: BorderRadius.circular(15),
    ),
    child: TextField(
      controller: messageController,
      decoration: const InputDecoration(
        hintText: 'Aa',
        hintStyle: TextStyle(),
        border: InputBorder.none,
        contentPadding: EdgeInsets.only(
          left: 20,
          bottom: 10,
        ),
      ),
      textInputAction: TextInputAction.done,
    ),
  ),
),
IconButton(
  icon: const Icon(
    Icons.send,
    color: AppColors.messengerBlue,
  ),
  onPressed: () async {
```

```
              // Add functionality to handle send button press
              await ref.read(chatProvider).sendMessage(
                    message: messageController.text,
                    chatroomId: chatroomId,
                    receiverId: widget.userId,
                  );
              messageController.clear();
            },
          ),
        ],
      ),
    );
  }
}
```

Create Post Code:

```
import 'dart:io';

import 'package:facebook_clone/core/constants/app_colors.dart';
import 'package:facebook_clone/core/constants/constants.dart';
import 'package:facebook_clone/core/utils/utils.dart';
import 'package:facebook_clone/core/widgets/round_button.dart';
import
'package:facebook_clone/features/posts/presentation/widgets/image_video_vi
ew.dart';
import
'package:facebook_clone/features/posts/presentation/widgets/profile_info.d
art';
import
'package:facebook_clone/features/posts/providers/posts_provider.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';

class CreatePostScreen extends ConsumerStatefulWidget {
  const CreatePostScreen({super.key});

  static const routeName = '/create-post';
```

```dart
  @override
  ConsumerState<CreatePostScreen> createState() =>
_CreatePostScreenState();
}

class _CreatePostScreenState extends ConsumerState<CreatePostScreen> {
  late final TextEditingController _postController;
  File? file;
  String fileType = 'image';
  bool isLoading = false;

  @override
  void initState() {
    _postController = TextEditingController();
    super.initState();
  }

  @override
  void dispose() {
    _postController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        actions: [
          TextButton(
            onPressed: makePost,
            child: const Text('Post'),
          ),
        ],
      ),
      body: SingleChildScrollView(
        child: Padding(
          padding: Constants.defaultPadding,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const ProfileInfo(),
              // post text field
              TextField(
                controller: _postController,
                decoration: const InputDecoration(
                  border: InputBorder.none,
                  hintText: 'What\'s on your mind?',
                  hintStyle: TextStyle(
                    fontSize: 18,
                    color: AppColors.darkGreyColor,
                  ),
                ),
                keyboardType: TextInputType.multiline,
                minLines: 1,
                maxLines: 10,
              ),
              const SizedBox(height: 20),
              file != null
                  ? ImageVideoView(
                      file: file!,
                      fileType: fileType,
                    )
                  : PickFileWidget(
                      pickImage: () async {
                        fileType = 'image';
                        file = await pickImage();
                        setState(() {});
                      },
                      pickVideo: () async {
                        fileType = 'video';
                        file = await pickVideo();
                        setState(() {});
                      },
                    ),
              const SizedBox(height: 20),
              isLoading
                  ? const Center(
                      child: CircularProgressIndicator(),
                    )
```

```dart
                  : RoundButton(
                      onPressed: makePost,
                      label: 'Post',
                    ),
            ],
          ),
        ),
      ),
    );
  }

  Future<void> makePost() async {
    setState(() => isLoading = true);
    await ref
        .read(postsProvider)
        .makePost(
          content: _postController.text,
          file: file!,
          postType: fileType,
        )
        .then((value) {
      Navigator.of(context).pop();
    }).catchError((_) {
      setState(() => isLoading = false);
    });
    setState(() => isLoading = false);
  }
}

class PickFileWidget extends StatelessWidget {
  const PickFileWidget({
    super.key,
    required this.pickImage,
    required this.pickVideo,
  });

  final VoidCallback pickImage;
  final VoidCallback pickVideo;

  @override
  Widget build(BuildContext context) {
    return Column(
      children: [
        TextButton(
          onPressed: pickImage,
          child: const Text('Pick Image'),
        ),
        const Divider(),
        TextButton(
          onPressed: pickVideo,
          child: const Text('Pick Video'),
        ),
      ],
    );
  }
}
```

Get all comments

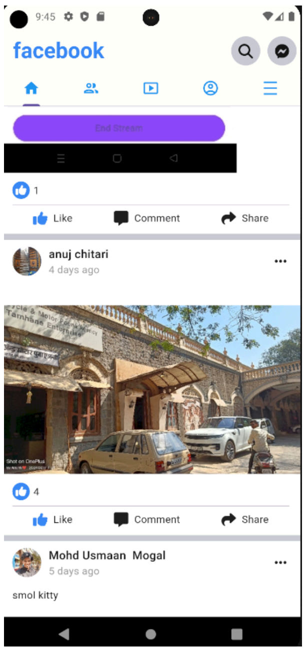
```dart
import 'package:facebook_clone/core/constants/firebase_field_names.dart';
import 'package:flutter/foundation.dart' show immutable;

@immutable
class Comment {
  final String commentId;
  final String authorId;
  final String postId;
  final String text;
  final DateTime createdAt;
  final List<String> likes;

  const Comment({
    required this.commentId,
    required this.authorId,
    required this.postId,
    required this.text,
    required this.createdAt,
    required this.likes,
  });
  Map<String, dynamic> toMap() {
    return <String, dynamic>{
      FirebaseFieldNames.commentId: commentId,
      FirebaseFieldNames.authorId: authorId,
      FirebaseFieldNames.postId: postId,
      FirebaseFieldNames.text: text,
      FirebaseFieldNames.createdAt: createdAt.millisecondsSinceEpoch,
      FirebaseFieldNames.likes: likes,
    };
  }

  factory Comment.fromMap(Map<String, dynamic> map) {
    return Comment(
      commentId: map[FirebaseFieldNames.commentId] ?? '',
      authorId: map[FirebaseFieldNames.authorId] ?? '',
      postId: map[FirebaseFieldNames.postId] ?? '',
      text: map[FirebaseFieldNames.text] ?? '',
      createdAt: DateTime.fromMillisecondsSinceEpoch(
        map[FirebaseFieldNames.createdAt] ?? 0,
      ),
      likes: List<String>.from(
        (map[FirebaseFieldNames.likes] ?? []),
      ),
    );
  }
}
```

Comments Screen

```
import
'package:facebook_clone/features/posts/presentation/widgets/comment_text_f
ield.dart';
import
'package:facebook_clone/features/posts/presentation/widgets/comments_list.
dart';
import 'package:flutter/material.dart';

class CommentsScreen extends StatelessWidget {
  const CommentsScreen({
```

```dart
    super.key,
    required this.postId,
  });

  final String postId;

  static const routeName = '/comments';

  // hello world

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Comments'),
      ),
      body: Column(
        children: [
          // Comments List
          CommentsList(postId: postId),

          // Comment Text field
          CommentTextField(
            postId: postId,
          ),
        ],
      ),
    );
  }
}
```

Output:



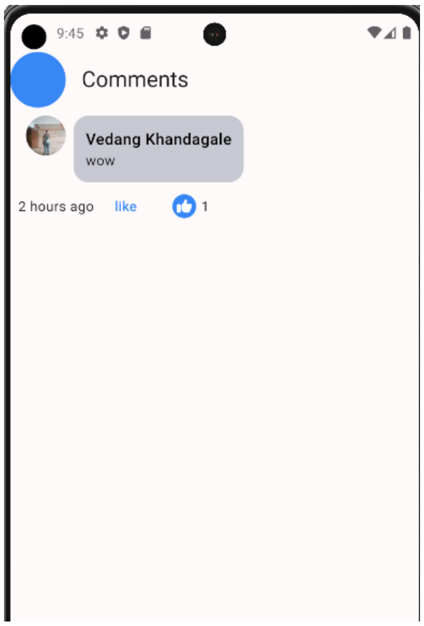
1

anuj chitari
4 days ago

4

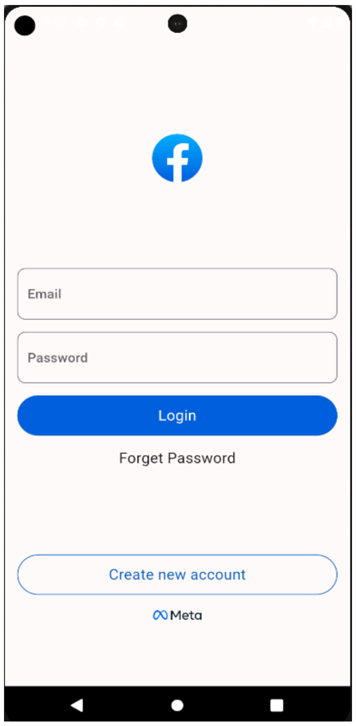
Mohd Usmaan Mogal
5 days ago

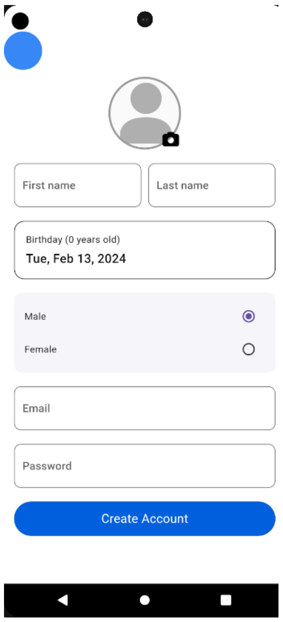
smol kitty



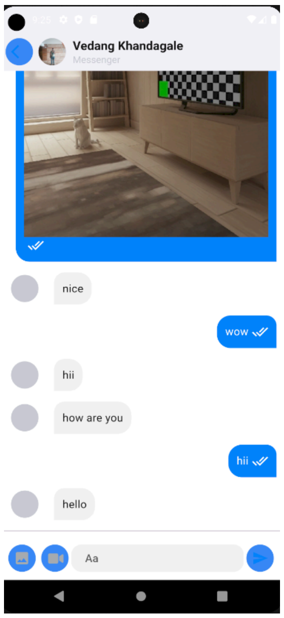
Vedang Khandagale
wow

2 hours ago 1





Birthday (0 years old)
Tue, Feb 13, 2024



Vedang Khandagale

nice

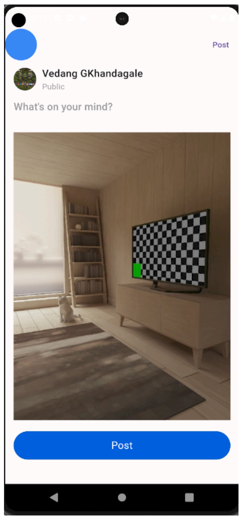
wow

hii

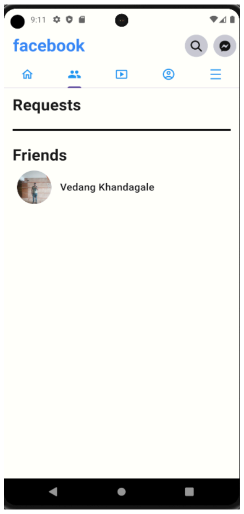
how are you

hii

hello



Vedang GKhandagale



Vedang Khandagale