

lib\carousel_slider.dart

```
import 'package:carousel_slider/carousel_slider.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/model/banner_model.dart';
import 'package:doc_fusion/screens/disease.dart';
import 'package:doc_fusion/screens/diseasedetail.dart';

class Carouselslider extends StatelessWidget {
  const Carouselslider({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return SizedBox(
      height: 150,
      width: MediaQuery.of(context).size.width,
      child: CarouselSlider.builder(
        itemCount: bannerCards.length,
        itemBuilder: (context, index, realIndex) {
          return Container(
            //alignment: Alignment.centerLeft,
            //width: MediaQuery.of(context).size.width,
            height: 140,
            margin: const EdgeInsets.only(left: 0, right: 0, bottom: 20),
            padding: const EdgeInsets.only(left: 0),
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(20),
              gradient: LinearGradient(
                stops: const [0.3, 0.7],
                begin: Alignment.centerLeft,
                end: Alignment.centerRight,
                colors: bannerCards[index].cardBackground,
              ),
            ),
            child: GestureDetector(
              onTap: () {
                index == 0
                ? Navigator.push(context,
                  MaterialPageRoute(builder: (BuildContext context) {
                    return const Disease();
                  })))
                : Navigator.push(context,
                  MaterialPageRoute(builder: (BuildContext context) {
                    return const DiseaseDetail(disease: 'Covid-19');
                  })));
              },
              child: Stack(
                children: [
                  Image.asset(
                    bannerCards[index].image,
                    //'assets/414.jpg',
                    fit: BoxFit.fitHeight,
                  ),
```

```
Container(  
padding: const EdgeInsets.only(top: 7, right: 5),  
alignment: Alignment.topRight,  
child: Row(  
crossAxisAlignment: CrossAxisAlignment.start,  
mainAxisAlignment: MainAxisAlignment.end,  
children: [  
Text(  
bannerCards[index].text,  
// 'Check Disease',  
style: GoogleFonts.lato(  
color: Colors.lightBlue[900],  
fontWeight: FontWeight.bold,  
fontSize: 15,  
),  
),  
Icon(  
Icons.chevron_right_rounded,  
color: Colors.lightBlue[900],  
size: 20,  
)  
],  
),  
),  
),  
),  
),  
),  
),  
);  
},  
options: CarouselOptions(  
autoPlay: true,  
enlargeCenterPage: true,  
enableInfiniteScroll: false,  
scrollPhysics: const ClampingScrollPhysics(),  
),  
),  
);  
}  
}
```

lib\firebase_options.dart

```
// File generated by FlutterFire CLI.
// ignore_for_file: type=lint
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:
        return macos;
      case TargetPlatform.windows:
        return windows;
      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: 'AIzaSyB8CZVb49xBKNDspp5v1kQ1DD8zQVu9Wgw',
    appId: '1:914888057780:web:dd3348b3aalf7e33f6c8e1',
    messagingSenderId: '914888057780',
    projectId: 'docfusion-30182',
    authDomain: 'docfusion-30182.firebaseio.com',
    storageBucket: 'docfusion-30182.firebaseio.com',
    measurementId: 'G-F6Y6YL7DC0',
  );
```

```
static const FirebaseOptions android = FirebaseOptions(  
  apiKey: 'AIzaSyBIFmZ966E4e7CN-Cu2_CDVArQJC24DlNg',  
  appId: '1:914888057780:android:d3947d31dcb8919af6c8e1',  
  messagingSenderId: '914888057780',  
  projectId: 'docfusion-30182',  
  storageBucket: 'docfusion-30182.firebaseiostorage.app',  
);
```

```
static const FirebaseOptions ios = FirebaseOptions(  
  apiKey: 'AIzaSyBWfS8_q1-jqyirCADu0ZtkdxRuAHFsL80',  
  appId: '1:914888057780:ios:2034c3722e06d5fcf6c8e1',  
  messagingSenderId: '914888057780',  
  projectId: 'docfusion-30182',  
  storageBucket: 'docfusion-30182.firebaseiostorage.app',  
  iosBundleId: 'com.example.docFusion',  
);
```

```
static const FirebaseOptions macos = FirebaseOptions(  
  apiKey: 'AIzaSyBWfS8_q1-jqyirCADu0ZtkdxRuAHFsL80',  
  appId: '1:914888057780:ios:2034c3722e06d5fcf6c8e1',  
  messagingSenderId: '914888057780',  
  projectId: 'docfusion-30182',  
  storageBucket: 'docfusion-30182.firebaseiostorage.app',  
  iosBundleId: 'com.example.docFusion',  
);
```

```
static const FirebaseOptions windows = FirebaseOptions(  
  apiKey: 'AIzaSyB8CZVb49xBKNDspp5v1kQ1DD8zQVu9Wgw',  
  appId: '1:914888057780:web:b3cff334bc07ba7bf6c8e1',  
  messagingSenderId: '914888057780',  
  projectId: 'docfusion-30182',  
  authDomain: 'docfusion-30182.firebaseio.com',  
  storageBucket: 'docfusion-30182.firebaseiostorage.app',  
  measurementId: 'G-SCJ1H51MVX',  
);  
}
```

lib\globals.dart

```
library globals;  
bool isDoctor = false;
```

lib/main.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:doc_fusion/globals.dart';
import 'package:doc_fusion/screens/doctor/main_page_doctor.dart';
import 'package:doc_fusion/screens/doctor_or_patient.dart';
import 'package:doc_fusion/screens/firebase_auth.dart';
import 'package:doc_fusion/screens/my_profile.dart';
import 'package:doc_fusion/screens/patient/appointments.dart';
import 'package:doc_fusion/screens/patient/doctor_profile.dart';
import 'package:doc_fusion/screens/patient/main_page_patient.dart';
import 'package:doc_fusion/screens/skip.dart';

import 'firebase_options.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );

  runApp(const MyApp());
}

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

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  User? user;

  Future<void> _getUser() async {
    user = _auth.currentUser!;
  }

  @override
  Widget build(BuildContext context) {
    _getUser();
    return MaterialApp(
      initialRoute: '/',
      routes: {
        // When navigating to the "/" route, build the FirstScreen widget.
        '/': (context) => user == null
          ? const Skip()
          : const DoctorOrPatient(),
        '/login': (context) => const FireBaseAuth(),
        '/home': (context) =>

```

```
isDoctor ? const MainPageDoctor() : const MainPagePatient(),
'/profile': (context) => const MyProfile(),
'/MyAppointments': (context) => const Appointments(),
'/DoctorProfile': (context) => DoctorProfile(),
},
theme: ThemeData(brightness: Brightness.light),
debugShowCheckedModeBanner: false,
// home: MainPageDoctor(),
// home: ChatRoom(
//   userId: '1234',
// ),
);
}
}
```

lib\splash_screen.dart

```
import 'dart:async';

import 'package:flutter/material.dart';

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

  @override
  State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    super.initState();
    Timer(const Duration(seconds: 3), () {
      Navigator.of(context).pushReplacementNamed('/home');
    });
  }

  @override
  Widget build(BuildContext context) {
    return const Scaffold(body: Text('hi'));
  }
}
```


lib\firestore_data\appointment_history_list.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/globals.dart';
import 'package:intl/intl.dart';

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

  @override
  State<AppointmentHistoryList> createState() => _AppointmentHistoryListState();
}

class _AppointmentHistoryListState extends State<AppointmentHistoryList> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  late User user;
  late String _documentID;

  Future<void> _getUser() async {
    user = _auth.currentUser!;
  }

  String _dateFormatter(String timestamp) {
    String formattedDate =
      DateFormat('dd-MM-yyyy').format(DateTime.parse(timestamp));
    return formattedDate;
  }

  Future<void> deleteAppointment(String docID) {
    return FirebaseFirestore.instance
      .collection('appointments')
      .doc(user.uid)
      .collection('all')
      .doc(docID)
      .delete();
  }

  @override
  void initState() {
    super.initState();
    _getUser();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: StreamBuilder(
          stream: FirebaseFirestore.instance
            .collection('appointments')
            .doc(user.uid)
```

```

.collection('all')
.orderBy('date', descending: true)
.snapshots(),
builder:
(BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
if (!snapshot.hasData) {
return const Center(
child: CircularProgressIndicator(),
);
}
return snapshot.data!.size == 0
? Text(
'History Appears here...',
style: GoogleFonts.lato(
color: Colors.grey,
fontSize: 18,
),
)
: ListView.builder(
scrollDirection: Axis.vertical,
physics: const ClampingScrollPhysics(),
shrinkWrap: true,
itemCount: snapshot.data!.size,
itemBuilder: (context, index) {
DocumentSnapshot document = snapshot.data!.docs[index];
return Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Container(
margin: const EdgeInsets.fromLTRB(5, 5, 5, 0),
padding: const EdgeInsets.only(left: 10, top: 5),
height: 70,
width: MediaQuery.of(context).size.width,
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(5),
color: Colors.blueGrey[50],
),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
// doctor name
Text(
'${index + 1}. ${isDoctor ? '${document['patientName']}' :
 '${document['doctorName']}'}',
style: GoogleFonts.lato(
fontSize: 15,
),
),
// date
Text(
_dateFormatter(
document['date'].toDate().toString()),
style: GoogleFonts.lato(
fontSize: 12,

```

```
fontWeight: FontWeight.bold),
),
// reason
Text(
document['description'] ?? 'No description')
],
),
),
const SizedBox(
height: 10,
),
],
);
},
);
},
),
),
);
}
}
```

lib\firestore_data\appointment_list.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/globals.dart';
import 'package:intl/intl.dart';

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

  @override
  State<AppointmentList> createState() => _AppointmentListState();
}

class _AppointmentListState extends State<AppointmentList> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  late User user;
  late String _documentID;

  Future<void> _getUser() async {
    user = _auth.currentUser!;
  }

  // delete appointment from both patient and doctor
  Future<void> deleteAppointment(
    String docID, String doctorId, String patientId) async {
    FirebaseFirestore.instance
      .collection('appointments')
      .doc(doctorId)
      .collection('pending')
      .doc(docID)
      .delete();
    return FirebaseFirestore.instance
      .collection('appointments')
      .doc(patientId)
      .collection('pending')
      .doc(docID)
      .delete();
  }

  String _dateFormatter(String timestamp) {
    String formattedDate =
      DateFormat('dd-MM-yyyy').format(DateTime.parse(timestamp));
    return formattedDate;
  }

  String _timeFormatter(String timestamp) {
    String formattedTime =
      DateFormat('kk:mm').format(DateTime.parse(timestamp));
    return formattedTime;
  }
}
```

```

// alert box for confirmation of deleting appointment
showAlertDialog(BuildContext context, String doctorId, String patientId) {
// No
Widget cancelButton = TextButton(
  child: const Text("No"),
  onPressed: () {
    Navigator.of(context).pop();
  },
);

// YES
Widget continueButton = TextButton(
  child: const Text("Yes"),
  onPressed: () {
    deleteAppointment(_documentID, doctorId, patientId);
    Navigator.of(context).pop();
  },
);

// set up the AlertDialog
AlertDialog alert = AlertDialog(
  title: const Text("Confirm Delete"),
  content: const Text("Are you sure you want to delete this Appointment?"),
  actions: [
    cancelButton,
    continueButton,
  ],
);

// show the dialog
showDialog(
  context: context,
  builder: (BuildContext context) {
    return alert;
  },
);

// helping in removing pending appointment
_checkDiff(DateTime date) {
  print(date);
  var diff = DateTime.now().difference(date).inSeconds;
  print('date difference : $diff');
  if (diff > 0) {
    return true;
  } else {
    return false;
  }
}

// for comparing date
_compareDate(String date) {
  if (_dateFormatter(DateTime.now().toString())
    .compareTo(_dateFormatter(date)) ==

```

```

0) {
return true;
} else {
return false;
}
}

@override
void initState() {
super.initState();
getUser();
}

@override
Widget build(BuildContext context) {
return SafeArea(
child: StreamBuilder(
stream: FirebaseFirestore.instance
.collection('appointments')
.doc(user.uid)
.collection('pending')
.orderBy('date')
.snapshots(),
builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
if (!snapshot.hasData) {
return const Center(
child: CircularProgressIndicator(),
);
}
return snapshot.data!.size == 0
? Center(
child: Text(
'No Appointment Scheduled',
style: GoogleFonts.lato(
color: Colors.grey,
fontSize: 18,
),
),
)
: ListView.builder(
scrollDirection: Axis.vertical,
physics: const ClampingScrollPhysics(),
shrinkWrap: true,
itemCount: snapshot.data!.size,
itemBuilder: (context, index) {
DocumentSnapshot document = snapshot.data!.docs[index];

// delete past appointments from pending appointment list
if (_checkDiff(document['date'].toDate())) {
deleteAppointment(document.id, document['doctorId'],
document['patientId']);
}

// each appointment

```

```

return Card(
  elevation: 2,
  child: InkWell(
    onTap: () {},
    child: ExpansionTile(
      initiallyExpanded: true,

      // main info of appointment
      title: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          // doctor name
          Padding(
            padding: const EdgeInsets.only(left: 5),
            child: Text(
              isDoctor
                ? document['patientName']
                : document['doctorName'],
              style: GoogleFonts.lato(
                fontSize: 16,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
          // Today label
          Text(
            _compareDate(
              document['date'].toDate().toString()
            )
            ? "TODAY"
            : "",
            style: GoogleFonts.lato(
              color: Colors.green,
              fontSize: 18,
              fontWeight: FontWeight.bold),
          ),

          const SizedBox(
            width: 0,
          ),
        ],
      ),

      // appointment date
      subtitle: Padding(
        padding: const EdgeInsets.only(left: 5),
        child: Text(
          _dateFormatter(
            document['date'].toDate().toString(),
            style: GoogleFonts.lato(),
          ),
        ),

        // patient info

```

```

children: [
  Padding(
    padding: const EdgeInsets.only(
      bottom: 20, right: 10, left: 16),
    child: Row(
      mainAxisAlignment:
        MainAxisAlignment.spaceBetween,
      children: [
        // patient info
        Column(
          crossAxisAlignment:
            CrossAxisAlignment.start,
          children: [
            // patient name
            Text(
              isDoctor
                ? ''
                : "Patient name: ${document['patientName']}",
              style: GoogleFonts.lato(
                fontSize: 16,
              ),
            ),
            const SizedBox(
              height: 10,
            ),

            // Appointment time
            Text(
              'Time: ${_timeFormatter(document['date'].toDate().toString())}',
              style: GoogleFonts.lato(fontSize: 16),
            ),

            const SizedBox(
              height: 10,
            ),

            Text(
              'Description : ${document['description']}',
              style: GoogleFonts.lato(fontSize: 16),
            )
          ],
        ),

        // delete button
        IconButton(
          tooltip: 'Delete Appointment',
          icon: const Icon(
            Icons.delete,
            color: Colors.red,
          ),
          onPressed: () {
            _documentID = document.id;
            showAlertDialog(
              context,

```



```
document['doctorId'],
document['patientId']);
},
),
],
),
),
],
),
),
);
},
);
},
),
);
}
}
```

lib\firestore_data\notification_list.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

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

  @override
  State<NotificationList> createState() => _NotificationListState();
}

class _NotificationListState extends State<NotificationList> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,
        leading: IconButton(
          splashRadius: 20,
          icon: const Icon(
            Icons.arrow_back_ios,
            color: Colors.indigo,
          ),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
        title: Text(
          'Notifications',
          style: GoogleFonts.lato(
            color: Colors.indigo,
            fontSize: 18,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
    );
  }
}
```

lib\firestore_data\search_list.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/patient/doctor_profile.dart';
import 'package:typicons_flutter/typicons_flutter.dart';

class SearchList extends StatefulWidget {
  final String searchKey;

  const SearchList({Key? key, required this.searchKey}) : super(key: key);

  @override
  State<SearchList> createState() => _SearchListState();
}

class _SearchListState extends State<SearchList> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: StreamBuilder(
          stream: FirebaseFirestore.instance
            .collection('doctor')
            .orderBy('name')
            .startAt(['Dr. ${widget.searchKey}']).endAt(
              ['Dr. ${widget.searchKey}\uf8ff']),
          snapshots(),
          builder:
            (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
              if (!snapshot.hasData) {
                return const Center(
                  child: CircularProgressIndicator(),
                );
              }
              return snapshot.data!.size == 0
                ? Center(
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.center,
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      Text(
                        'No Doctor found!',
                        style: GoogleFonts.lato(
                          color: Colors.blue[800],
                          fontSize: 25,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
                      const Image(
                        image: AssetImage('assets/error-404.jpg'),
                        height: 250,
                        width: 250,
```

```

),
],
),
)
: Scrollbar(
child: ListView.builder(
scrollDirection: Axis.vertical,
physics: const ClampingScrollPhysics(),
shrinkWrap: true,
itemCount: snapshot.data!.size,
itemBuilder: (context, index) {
DocumentSnapshot doctor = snapshot.data!.docs[index];

String name = doctor['name'] ?? 'Unknown';
String specialization = doctor['specialization'] ?? 'Not specified';
String profilePhoto = doctor['profilePhoto'] ?? 'https://via.placeholder.com/150';
double rating = (doctor['rating'] ?? 0).toDouble();

return Padding(
padding: const EdgeInsets.only(top: 0.0),
child: Card(
color: Colors.blue[50],
elevation: 0,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(10),
),
child: Container(
padding: const EdgeInsets.only(left: 10, right: 10, top: 0),
width: MediaQuery.of(context).size.width,
height: MediaQuery.of(context).size.height / 9,
child: TextButton(
onPressed: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => DoctorProfile(doctor: name),
),
);
},
child: Row(
crossAxisAlignment: CrossAxisAlignment.center,
children: [
CircleAvatar(
backgroundImage: NetworkImage(profilePhoto),
backgroundColor: Colors.blue,
radius: 25,
),
const SizedBox(width: 20),
Column(
crossAxisAlignment: CrossAxisAlignment.start,
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(
name,

```

```

style: GoogleFonts.lato(
fontWeight: FontWeight.bold,
fontSize: 17,
color: Colors.black87,
),
),
Text(
specialization,
style: GoogleFonts.lato(
fontSize: 16,
color: Colors.black54,
),
),
],
),
const SizedBox(width: 10),
Expanded(
child: Container(
alignment: Alignment.centerRight,
child: Row(
crossAxisAlignment: CrossAxisAlignment.end,
mainAxisAlignment: MainAxisAlignment.end,
children: [
Icon(
Typicons.star_full_outline,
size: 20,
color: Colors.indigo[400],
),
const SizedBox(width: 3),
Text(
rating.toStringAsFixed(1),
style: GoogleFonts.lato(
fontWeight: FontWeight.bold,
fontSize: 15,
color: Colors.indigo,
),
),
],
),
),
),
),
),
),
),
),
),
);
}
),
);
},
),
),
);

```


lib\firestore_data\topRatedList.dart

```
import 'dart:math';

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/patient/doctor_profile.dart';
import 'package:typicons_flutter/typicons_flutter.dart';

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

  @override
  State<TopRatedList> createState() => _TopRatedListState();
}

class _TopRatedListState extends State<TopRatedList> {
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: StreamBuilder(
        stream: FirebaseFirestore.instance
          .collection('doctor')
          .orderBy('rating', descending: true)
          .snapshots(),
        builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
          if (!snapshot.hasData) {
            return const Center(
              child: CircularProgressIndicator(),
            );
          }
          return ListView.builder(
            scrollDirection: Axis.vertical,
            physics: const ClampingScrollPhysics(),
            shrinkWrap: true,
            itemCount: min(5, snapshot.data!.docs.length),
            itemBuilder: (context, index) {
              DocumentSnapshot doctor = snapshot.data!.docs[index];
              return Padding(
                padding: const EdgeInsets.only(top: 3.0),
                child: Card(
                  color: Colors.blue[50],
                  elevation: 0,
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10),
                  ),
                  child: Container(
                    padding: const EdgeInsets.only(left: 10, right: 10, top: 0),
                    width: MediaQuery.of(context).size.width,
                    height: MediaQuery.of(context).size.height / 9,
                    child: TextButton(
                      onPressed: () {
                        Navigator.push(
```

```

context,
MaterialPageRoute(
builder: (context) => DoctorProfile(
doctor: doctor['name'],
),
),
);
},
child: Row(
crossAxisAlignment: CrossAxisAlignment.center,
//mainAxisAlignment: MainAxisAlignment.spaceAround,
children: [
CircleAvatar(
backgroundImage:
NetworkImage(doctor['profilePhoto'] ?? ''),
backgroundColor: Colors.blue,
radius: 25,
),
const SizedBox(
width: 20,
),
Column(
crossAxisAlignment: CrossAxisAlignment.start,
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(
doctor['name'],
style: GoogleFonts.lato(
fontWeight: FontWeight.bold,
fontSize: 17,
color: Colors.black87,
),
),
Text(
doctor['specialization'],
style: GoogleFonts.lato(
fontSize: 16, color: Colors.black54),
),
],
),
const SizedBox(
width: 10,
),
Expanded(
child: Container(
alignment: Alignment.centerRight,
child: Row(
crossAxisAlignment: CrossAxisAlignment.end,
mainAxisAlignment: MainAxisAlignment.end,
children: [
Icon(
Typicons.star_full_outline,
size: 20,
color: Colors.indigo[400],

```



```
),  
const SizedBox(  
width: 3,  
),  
Text(  
doctor['rating'].toString(),  
style: GoogleFonts.lato(  
fontWeight: FontWeight.bold,  
fontSize: 15,  
color: Colors.indigo,  
),  
),  
],  
),  
),  
),  
],  
),  
),  
),  
),  
),  
);  
},  
);  
},  
),  
);  
}  
}
```

lib\firestore_data\user_details.dart

```
import 'dart:math';

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/globals.dart';
import 'package:doc_fusion/model/update_user_details.dart';

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

  @override
  State<UserDetails> createState() => _UserDetailsState();
}

class _UserDetailsState extends State<UserDetails> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  late User user;

  // map of all the details
  Map<String, dynamic> details = {};

  Future<void> _getUser() async {
    user = _auth.currentUser!;

    DocumentSnapshot snap = await FirebaseFirestore.instance
      .collection(isDoctor ? 'doctor' : 'patient')
      .doc(user.uid)
      .get();

    setState(() {
      details = snap.data() as Map<String, dynamic>;
    });
    print(snap.data());
  }

  @override
  void initState() {
    super.initState();
    _getUser();
  }

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.only(top: 8.0),
      child: ListView.builder(
        controller: ScrollController(),
        shrinkWrap: true,
        itemCount: details.length,
        itemBuilder: (context, index) {
```

```

String key = details.keys.elementAt(index);
String value =
details[key] == null ? 'Not Added' : details[key].toString();
String label = key[0].toUpperCase() + key.substring(1);

return Container(
margin: const EdgeInsets.symmetric(horizontal: 10, vertical: 5),
child: InkWell(
splashColor: Colors.grey.withOpacity(0.5),
borderRadius: BorderRadius.circular(10),
onTap: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => UpdateUserDetails(
label: label,
field: key,
value: value,
),
),
).then((value) {
// reload page
_getUser();
setState(() {});
});
},
child: Ink(
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
color: Colors.grey[200],
),
child: Container(
padding: const EdgeInsets.symmetric(horizontal: 14),
height: MediaQuery.of(context).size.height / 14,
width: MediaQuery.of(context).size.width,
child: Row(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
children: [
Text(
label,
style: GoogleFonts.lato(
color: Colors.black,
fontSize: 16,
fontWeight: FontWeight.bold,
),
),
Text(
value.substring(0, min(20, value.length)),
style: GoogleFonts.lato(
color: Colors.black54,
fontSize: 15,
fontWeight: FontWeight.bold,
),
),

```

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

lib\helperFunction\sharedpref_helper.dart

```
import 'package:shared_preferences/shared_preferences.dart';

class SharedPreferenceHelper {

  final String _userIdKey = "USERIDKEY";
  final String _userNameKey = "USERNAMEKEY";
  final String _accountTypeKey = "ACCOUNTTYPEKEY";
  final String _profileUrlKey = "PROFILEURLKEY";

  // save data
  Future<bool> saveUserName(String userName) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.setString(_userNameKey, userName);
  }

  Future<bool> saveUserId(String userId) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.setString(_userIdKey, userId);
  }

  Future<bool> saveAccountType(bool accountType) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.setBool(_accountTypeKey, accountType);
  }

  Future<bool> saveProfileUrl(String profileUrl) async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.setString(_profileUrlKey, profileUrl);
  }

  // get data
  Future<String?> getUserName() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.getString(_userNameKey);
  }

  Future<String?> getUserId() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.getString(_userIdKey);
  }

  Future<bool?> getAccountType() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.getBool(_accountTypeKey);
  }

  Future<String?> getProfileUrl() async {
    SharedPreferences prefs = await SharedPreferences.getInstance();
    return prefs.getString(_profileUrlKey);
  }
}
```


lib\model\banner_model.dart

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

class BannerModel {
  String text;
  List<Color> cardBackground;
  String image;

  BannerModel(this.text, this.cardBackground, this.image);
}

List<BannerModel> bannerCards = [
  BannerModel(
    "Check Disease",
    [
      const Color(0xffald4ed),
      const Color(0xffc0eaff),
    ],
    "assets/414-bg.png"),
  BannerModel(
    "Covid-19",
    [
      const Color(0xffb6d4fa),
      const Color(0xffcfe3fc),
    ],
    "assets/covid-bg.png"),
];
```

lib\model\card_model.dart

```
import 'package:flutter/material.dart';
import 'package:tabler_icons/tabler_icons.dart';

class CardModel {
  String doctor;
  int cardBackground;
  var cardIcon;

  CardModel(this.doctor, this.cardBackground, this.cardIcon);
}

List<CardModel> cards = [
  CardModel("Cardiologist", 0xFFec407a, Icons.heart_broken),
  CardModel("Dentist", 0xFF5c6bc0, const IconData(0x1f9b7)),
  CardModel("Eye Special", 0xFFfbc02d, TablerIcons.eye),
  CardModel("Orthopaedic", 0xFF1565C0, Icons.wheelchair_pickup_sharp),
  CardModel("Paediatrician", 0xFF2E7D32, const IconData(0x1f476)),
];
```


lib\model\update_user_details.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:flutter/material.dart';
import 'package:doc_fusion/globals.dart';

class UpdateUserDetails extends StatefulWidget {
  final String label;
  final String field;
  final String value;
  const UpdateUserDetails(
    {Key? key, required this.label, required this.field, required this.value})
    : super(key: key);

  @override
  State<UpdateUserDetails> createState() => _UpdateUserDetailsState();
}

class _UpdateUserDetailsState extends State<UpdateUserDetails> {
  final TextEditingController _textcontroller = TextEditingController();
  late FocusNode f1;
  final FirebaseAuth _auth = FirebaseAuth.instance;
  User? user;
  String? userID;

  Future<void> _getUser() async {
    user = _auth.currentUser!;
    userID = user!.uid;
  }

  @override
  void initState() {
    super.initState();
    _getUser();
  }

  @override
  Widget build(BuildContext context) {
    _textcontroller.text = widget.value;
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 2,
        // back button
        leading: GestureDetector(
          onTap: () {
            Navigator.pop(context);
          },
          child: const Icon(
            Icons.arrow_back_ios,
            color: Colors.indigo,
          ),
        ),
      ),
    );
  }
}
```

```

),
title: Container(
alignment: Alignment.centerLeft,
child: Text(
widget.label,
style: GoogleFonts.lato(
color: Colors.indigo,
fontSize: 21,
fontWeight: FontWeight.bold,
),
),
),
),
),
body: Padding(
padding: const EdgeInsets.fromLTRB(5, 20, 5, 0),
child: Column(
children: [
Container(
margin: const EdgeInsets.symmetric(horizontal: 15),
child: TextFormField(
controller: _textcontroller,
style: GoogleFonts.lato(
fontSize: 20,
fontWeight: FontWeight.bold,
),
onFieldSubmitted: (String data) {
_textcontroller.text = data;
},
textInputAction: TextInputAction.done,
validator: (value) {
if (value == null) {
return 'Please Enter the ${widget.label}';
}
return null;
},
),
),
),
const SizedBox(
height: 50,
),
Container(
padding: const EdgeInsets.symmetric(horizontal: 30),
height: 50,
width: MediaQuery.of(context).size.width,
child: ElevatedButton(
onPressed: () {
FocusScope.of(context).unfocus();
updateData();
Navigator.of(context).pop(context);
},
style: ElevatedButton.styleFrom(
foregroundColor: Colors.black, backgroundColor: Colors.indigo.withOpacity(0.9),
elevation: 2,
shape: RoundedRectangleBorder(

```

```

borderRadius: BorderRadius.circular(32.0),
),
),
child: Text(
  'Update',
  style: GoogleFonts.lato(
    color: Colors.white,
    fontSize: 18,
    fontWeight: FontWeight.bold),
),
),
)
],
),
),
);
}

```

```

Future<void> updateData() async {
  FirebaseFirestore.instance
    .collection(isDoctor ? 'doctor' : 'patient')
    .doc(userID)
    .set({
      widget.field: _textcontroller.text,
    }, SetOptions(merge: true));
  if (widget.field.compareTo('name') == 0) {
    await user!.updateDisplayName(_textcontroller.text);
  }
}
}
}

```

lib\screens\disease.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/diseasedetail.dart';

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

  @override
  State<Disease> createState() => _DiseaseState();
}

class _DiseaseState extends State<Disease> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,
        title: Text(
          'Disease',
          style: GoogleFonts.lato(
            color: Colors.black,
            fontSize: 20,
            fontWeight: FontWeight.bold,
          ),
        ),
        iconTheme: const IconThemeData(
          color: Colors.black,
        ),
        body: StreamBuilder(
          stream: FirebaseFirestore.instance
            .collection('disease')
            .orderBy('Name')
            .startAt(['']).endAt(['' '\uf8ff']).snapshots(),
          builder:
            (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
              if (!snapshot.hasData) {
                return const Center(
                  child: CircularProgressIndicator(),
                );
              }
              return ListView(
                physics: const BouncingScrollPhysics(),
                children: snapshot.data!.docs.map((document) {
                  return Container(
                    padding:
                      const EdgeInsets.only(left: 10, right: 10, top: 0),
                    width: MediaQuery.of(context).size.width,
                    height: MediaQuery.of(context).size.height / 10,
                    decoration: const BoxDecoration(
                      border: Border(
```

```

bottom: BorderSide(
  color: Colors.black87,
  width: 0.2,
),
),
),
child: TextButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => DiseaseDetail(
          disease: document['Name'],
        ),
      ),
    );
  },
  child: Row(
    children: [
      const SizedBox(
        width: 20,
      ),
      Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            document['Name'],
            style: GoogleFonts.lato(
              fontWeight: FontWeight.bold,
              fontSize: 16,
              color: Colors.black87,
            ),
          ),
          Text(
            document['Symtoms'],
            style: GoogleFonts.lato(
              fontSize: 14, color: Colors.black54),
          ),
        ],
      ),
    ],
  )),
).toList(),
);
})))
}
}

```

lib\screens\diseasedetail.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

class DiseaseDetail extends StatefulWidget {
  final String disease;

  const DiseaseDetail({Key? key, required this.disease}) : super(key: key);
  @override
  State<DiseaseDetail> createState() => _DiseaseDetailState();
}

class _DiseaseDetailState extends State<DiseaseDetail> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        iconTheme: const IconThemeData(color: Colors.black),
        backgroundColor: Colors.white,
        elevation: 0,
        title: Text(
          widget.disease,
          style: GoogleFonts.lato(color: Colors.black),
        ),
      ),
      body: StreamBuilder(
        stream: FirebaseFirestore.instance
          .collection('disease')
          .orderBy('Name')
          .startAt([widget.disease]).endAt(
            ['${widget.disease}\uf8ff']).snapshots(),
        builder:
          (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
            if (!snapshot.hasData) {
              return const Center(
                child: CircularProgressIndicator(),
              );
            }
            return ListView(
              physics: const ClampingScrollPhysics(),
              children: snapshot.data!.docs.map((document) {
                return Container(
                  margin: const EdgeInsets.only(top: 10),
                  child: Column(
                    children: <Widget>[
                      const SizedBox(
                        height: 20,
                      ),
                      Container(
                        margin: const EdgeInsets.only(left: 15, right: 15),
                        padding: const EdgeInsets.only(left: 20, right: 20),
```

```

alignment: Alignment.centerLeft,
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
color: Colors.blueGrey[50],
),
child: Text(
document['Description'],
style: GoogleFonts.lato(
color: Colors.black54, fontSize: 18),
)),
const SizedBox(
height: 20,
),
Container(
margin: const EdgeInsets.only(left: 15, right: 15),
padding: const EdgeInsets.only(left: 20, right: 20),
alignment: Alignment.centerLeft,
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
color: Colors.blueGrey[50],
),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
'How does it spread?',
style: GoogleFonts.lato(
color: Colors.black87,
fontSize: 20,
fontWeight: FontWeight.bold),
textAlign: TextAlign.left,
),
const SizedBox(
height: 15,
),
Text(
document['Spread'],
style: GoogleFonts.lato(
color: Colors.black54,
fontSize: 18,
),
),
],
)),
const SizedBox(
height: 20,
),
Container(
margin: const EdgeInsets.only(left: 15, right: 15),
padding: const EdgeInsets.only(left: 20, right: 20),
alignment: Alignment.centerLeft,
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
color: Colors.blueGrey[50],

```

```

),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Text(
      'Symtoms',
      style: GoogleFonts.lato(
        color: Colors.black87,
        fontSize: 20,
        fontWeight: FontWeight.bold),
      textAlign: TextAlign.left,
    ),
    const SizedBox(
      height: 15,
    ),
    Text(
      document['Symtoms'],
      style: GoogleFonts.lato(
        color: Colors.black54,
        fontSize: 18,
      ),
    ),
  ],
)),
const SizedBox(
  height: 20,
),
Container(
  margin: const EdgeInsets.only(left: 15, right: 15),
  padding: const EdgeInsets.only(left: 20, right: 20),
  alignment: Alignment.centerLeft,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    color: Colors.blueGrey[50],
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Warning Signs - Seek medical attention',
        style: GoogleFonts.lato(
          color: Colors.black87,
          fontSize: 20,
          fontWeight: FontWeight.bold),
        textAlign: TextAlign.left,
      ),
      const SizedBox(
        height: 15,
      ),
      Text(
        document['Warning'],
        style: GoogleFonts.lato(
          color: Colors.black54,
          fontSize: 18,

```



```
),  
)  
],  
)),  
],  
,  
,  
);  
}).toList());  
}),  
);  
}  
}
```

lib\screens\doctor_or_patient.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:doc_fusion/globals.dart';
import 'package:doc_fusion/screens/doctor/main_page_doctor.dart';
import 'package:doc_fusion/screens/patient/main_page_patient.dart';

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

  @override
  State<DoctorOrPatient> createState() => _DoctorOrPatientState();
}

class _DoctorOrPatientState extends State<DoctorOrPatient> {
  bool _isLoading = true;
  void _setUser() async {
    final User? user = FirebaseAuth.instance.currentUser;
    DocumentSnapshot snap = await FirebaseFirestore.instance
      .collection('users')
      .doc(user!.uid)
      .get();

    var basicInfo = snap.data() as Map<String, dynamic>;

    isDoctor = basicInfo['type'] == 'doctor' ? true : false;
    print('isdoctor : $isDoctor');
    setState(() {
      _isLoading = false;
    });
  }

  @override
  void initState() {
    super.initState();
    _setUser();
  }

  @override
  Widget build(BuildContext context) {
    return _isLoading
      ? const Scaffold(body: Center(child: CircularProgressIndicator()))
      : isDoctor
      ? const MainPageDoctor()
      : const MainPagePatient();
  }
}
```

lib\screens\explore_list.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/patient/doctor_profile.dart';
import 'package:typicons_flutter/typicons_flutter.dart';

class ExploreList extends StatefulWidget {
  final String type;
  const ExploreList({Key? key, required this.type}) : super(key: key);

  @override
  State<ExploreList> createState() => _ExploreListState();
}

class _ExploreListState extends State<ExploreList> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,
        title: Text(
          '${widget.type}s',
          style: GoogleFonts.lato(
            color: Colors.black,
            fontSize: 20,
            fontWeight: FontWeight.bold,
          ),
        ),
        iconTheme: const IconThemeData(
          color: Colors.black,
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.only(top: 10),
        child: StreamBuilder(
          stream: FirebaseFirestore.instance
            .collection('doctor')
            .orderBy('specialization')
            .startAt([widget.type]).endAt(
              ['${widget.type}\uf8ff']).snapshots(),
          builder:
            (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
              if (!snapshot.hasData) {
                return const Center(
                  child: CircularProgressIndicator(),
                );
              }
              return ListView.builder(
                scrollDirection: Axis.vertical,
                physics: const ClampingScrollPhysics(),
                shrinkWrap: true,
                itemCount: snapshot.data!.size,
```

```

itemBuilder: (context, index) {
  DocumentSnapshot doctor = snapshot.data!.docs[index];
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 10),
    child: Card(
      color: Colors.blue[50],
      elevation: 0,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
      child: Container(
        padding: const EdgeInsets.only(left: 10, right: 10),
        width: MediaQuery.of(context).size.width,
        height: MediaQuery.of(context).size.height / 9,
        child: TextButton(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => DoctorProfile(
                  doctor: doctor['name'],
                ),
              ),
            );
          },
          child: Row(
            crossAxisAlignment: CrossAxisAlignment.center,
            //mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: [
              CircleAvatar(
                backgroundImage:
                  NetworkImage(doctor['profilePhoto']),
                backgroundColor: Colors.blue,
                radius: 25,
              ),
              const SizedBox(
                width: 20,
              ),
              Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Text(
                    doctor['name'],
                    style: GoogleFonts.lato(
                      fontWeight: FontWeight.bold,
                      fontSize: 17,
                      color: Colors.black87,
                    ),
                  ),
                  Text(
                    doctor['specialization'],
                    style: GoogleFonts.lato(
                      fontSize: 16, color: Colors.black54),

```

```
,
],
),
const SizedBox(
width: 10,
),
Expanded(
child: Container(
alignment: Alignment.centerRight,
child: Row(
crossAxisAlignment: CrossAxisAlignment.end,
mainAxisAlignment: MainAxisAlignment.end,
children: [
Icon(
Typicons.star_full_outline,
size: 20,
color: Colors.indigo[400],
),
const SizedBox(
width: 3,
),
Text(
doctor['rating'].toString(),
style: GoogleFonts.lato(
fontWeight: FontWeight.bold,
fontSize: 15,
color: Colors.indigo,
),
),
),
],
),
),
),
],
),
),
),
);
},
);
},
);
```

lib\screens\firebase_auth.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

import 'register.dart';
import 'sign_in.dart';

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

  @override
  State<FireBaseAuth> createState() => _FireBaseAuthState();

  // void out2(BuildContext context) {
  //   Navigator.pop(context);
  // }

  class _FireBaseAuthState extends State<FireBaseAuth> {
    @override
    Widget build(BuildContext context) {
      return Scaffold(
        body: Stack(
          children: [
            Container(
              constraints: const BoxConstraints.expand(),
              decoration: const BoxDecoration(
                image: DecorationImage(
                  image: AssetImage(
                    "assets/image-medical.jpg",
                  ),
                  fit: BoxFit.cover,
                ),
              ),
            SafeArea(
              child: Align(
                alignment: Alignment.topCenter,
                child: Container(
                  alignment: Alignment.topLeft,
                  padding: const EdgeInsets.only(top: 80.0, left: 25),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Text(
                        'HELLO',
                        style: GoogleFonts.b612(
                          color: Colors.black,
                          fontSize: 50,
                          fontWeight: FontWeight.w700),
                      ),
                      Text(
                        'Welcome to Medico!',

```

```

style: GoogleFonts.b612(
color: Colors.indigo[800],
fontSize: 17,
fontWeight: FontWeight.w400),
),
],
),
),
),
),
),
),
SafeArea(
child: Align(
alignment: Alignment.bottomCenter,
child: Column(
mainAxisAlignment: MainAxisAlignment.end,
children: [
Container(
height: 220,
decoration: BoxDecoration(
color: Colors.black26.withOpacity(0.25),
borderRadius: BorderRadius.circular(20),
),
child: Column(
crossAxisAlignment: CrossAxisAlignment.center,
mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
Container(
width: MediaQuery.of(context).size.width / 1.1,
padding: const EdgeInsets.all(16),
alignment: Alignment.center,
child: SizedBox(
width: double.infinity,
height: 50.0,
child: ElevatedButton(
onPressed: () => _pushPage(context, const SignIn()),
style: ElevatedButton.styleFrom(
foregroundColor: Colors.indigo[800], backgroundColor: Colors.indigo[800], elevation: 2,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(32.0),
),
),
child: Text(
"Sign in",
style: GoogleFonts.lato(
color: Colors.white,
fontSize: 18.0,
fontWeight: FontWeight.bold,
),
),
),
),
),
),
),
Container(

```

```
width: MediaQuery.of(context).size.width / 1.1,
padding: const EdgeInsets.all(16),
alignment: Alignment.center,
child: SizedBox(
width: double.infinity,
height: 50.0,
child: ElevatedButton(
onPressed: () => _pushPage(context, const Register()),
style: ElevatedButton.styleFrom(
foregroundColor: Colors.white, backgroundColor: Colors.white, elevation: 2,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(32.0),
),
),
child: Text(
"Create an Account",
style: GoogleFonts.lato(
color: Colors.black,
fontSize: 18.0,
fontWeight: FontWeight.bold,
),
),
),
),
),
),
),
),
],
),
),
const SizedBox(
height: 80,
),
],
),
),
),
),
],
), //<--
);
}

void _pushPage(BuildContext context, Widget page) {
Navigator.of(context).push(
MaterialPageRoute<void>(builder: (_) => page),
);
}
}
```


lib\screens\my_profile.dart

```
import 'dart:convert';
import 'dart:typed_data';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/firestore_data/appointment_history_list.dart';
import 'package:doc_fusion/globals.dart';
import 'package:image_picker/image_picker.dart';
import 'package:http/http.dart' as http;

import 'setting.dart';

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

  @override
  State<MyProfile> createState() => _MyProfileState();
}

class _MyProfileState extends State<MyProfile> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  late User user;
  final FirebaseStorage storage = FirebaseStorage.instance;

  // details
  String? email;
  String? name;
  String? phone;
  String? bio;
  String? specialization;

  // default dp
  String image =
    'https://cdn.icon-icons.com/icons2/1378/PNG/512/avatardefault_92824.png';

  Future<void> _getUser() async {
    user = _auth.currentUser!;

    DocumentSnapshot snap = await FirebaseFirestore.instance
      .collection(isDoctor ? 'doctor' : 'patient')
      .doc(user.uid)
      .get();

    setState(() {
      var snapshot = snap.data() as Map<String, dynamic>;
      email = snapshot['email'];
      name = snapshot['name'];
      phone = snapshot['phone'];
      bio = snapshot['bio'];
      image = snapshot['profilePhoto'] ?? image;
    });
  }
}
```

```

specialization = snapshot['specialization'];
});
print(snap.data());
}

@override
void initState() {
  super.initState();
  _getUser();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SafeArea(
      child: NotificationListener<OverscrollIndicatorNotification>(
        onNotification: (OverscrollIndicatorNotification overscroll) {
          overscroll.disallowIndicator();
        },
        return true;
      ),
      child: ListView(
        physics: const ClampingScrollPhysics(),
        shrinkWrap: true,
        children: <Widget>[
          Stack(
            alignment: Alignment.center,
            children: <Widget>[
              Column(
                children: [
                  Container(
                    decoration: const BoxDecoration(
                      gradient: LinearGradient(
                        begin: Alignment.topCenter,
                        end: Alignment.bottomCenter,
                        stops: [0.1, 0.5],
                        colors: [
                          Colors.indigo,
                          Colors.indigoAccent,
                        ],
                      ),
                    height: MediaQuery
                      .of(context)
                      .size
                      .height / 5,
                    child: Container(
                      padding: const EdgeInsets.only(top: 10, right: 7),
                      alignment: Alignment.topRight,
                      // edit user info button
                      child: IconButton(
                        icon: const Icon(
                          Icons.settings,
                          color: Colors.white,
                          size: 20,

```

```

),
onPressed: () {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => const UserSettings(),
    ),
  ).then((value) {
    // reload page
    _getUser();
    setState(() {});
  });
},
),
),
),
// user name
Container(
  alignment: Alignment.center,
  height: MediaQuery
    .of(context)
    .size
    .height / 6,
  padding: const EdgeInsets.only(top: 75),
  child: Text(
    name ?? 'Name Not Added',
    style: GoogleFonts.lato(
      fontSize: 25,
      fontWeight: FontWeight.bold,
    ),
  ),
),
),

Text(specialization == null ? '' : '($specialization)'),
],
),

// user image
Container(
  decoration: BoxDecoration(
    border: Border.all(
      color: Colors.teal.shade50,
      width: 5,
    ),
    shape: BoxShape.circle),
  child: InkWell(
    onTap: () {
      _showSelectionDialog(context);
    },
    child: CircleAvatar(
      radius: 80,
      backgroundColor: Colors.white,
      backgroundImage: NetworkImage(image),
    ),
  ),
),

```

```

),
),
],
),

const SizedBox(
height: 20,
),

// user basic info
Container(
margin: const EdgeInsets.only(left: 15, right: 15),
padding: const EdgeInsets.only(left: 20),
height: MediaQuery
.of(context)
.size
.height / 7,
width: MediaQuery
.of(context)
.size
.width,
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
color: Colors.blueGrey[50],
),
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
// user email
Row(
crossAxisAlignment: CrossAxisAlignment.center,
children: [
ClipRRect(
borderRadius: BorderRadius.circular(30),
child: Container(
height: 27,
width: 27,
color: Colors.red[900],
child: const Icon(
Icons.mail_rounded,
color: Colors.white,
size: 16,
),
),
),
const SizedBox(
width: 10,
),
Text(
user.email ?? 'Email Not Added',
style: GoogleFonts.lato(
fontSize: 16,
fontWeight: FontWeight.w600,
color: Colors.black54,

```

```

),
),
],
),
const SizedBox(
height: 15,
),
// user phone number
Row(
crossAxisAlignment: CrossAxisAlignment.center,
children: [
ClipRRect(
borderRadius: BorderRadius.circular(30),
child: Container(
height: 27,
width: 27,
color: Colors.blue[800],
child: const Icon(
Icons.phone,
color: Colors.white,
size: 16,
),
),
),
const SizedBox(
width: 10,
),
Text(
phone ?? 'Not Added',
style: GoogleFonts.lato(
fontSize: 16,
fontWeight: FontWeight.w600,
color: Colors.black54,
),
),
],
),
],
),
),

// user bio
Container(
margin: const EdgeInsets.only(left: 15, right: 15, top: 20),
padding: const EdgeInsets.only(left: 20, top: 20),
height: MediaQuery
.of(context)
.size
.height / 7,
width: MediaQuery
.of(context)
.size
.width,
decoration: BoxDecoration(

```

```

borderRadius: BorderRadius.circular(10),
color: Colors.blueGrey[50],
),
child: Column(
  children: [
    Row(
      children: [
        ClipRRect(
          borderRadius: BorderRadius.circular(30),
          child: Container(
            height: 27,
            width: 27,
            color: Colors.indigo[600],
            child: const Icon(
              Icons.edit,
              color: Colors.white,
              size: 16,
            ),
          ),
        const SizedBox(
          width: 10,
        ),
        Text(
          'Bio',
          style: GoogleFonts.lato(
            fontSize: 16,
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
      ],
    ),
    // bio
    Container(
      alignment: Alignment.centerLeft,
      padding: const EdgeInsets.only(top: 10, left: 40),
      child: Text(
        bio ?? 'Not Added',
        style: GoogleFonts.lato(
          fontSize: 16,
          fontWeight: FontWeight.w500,
          color: Colors.black38,
        ),
      ),
    ),
  ],
),
),
// Appointment history
Container(
  margin: const EdgeInsets.only(left: 15, right: 15, top: 20),
  padding: const EdgeInsets.only(left: 20, top: 20),

```

```

height: MediaQuery
.of(context)
.size
.height / 2,
width: MediaQuery
.of(context)
.size
.width,
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
color: Colors.blueGrey[50],
),
child: Column(
children: [
Row(
children: [
ClipRRect(
borderRadius: BorderRadius.circular(30),
child: Container(
height: 27,
width: 27,
color: Colors.green[900],
child: const Icon(
Icons.history,
color: Colors.white,
size: 16,
),
),
),
const SizedBox(
width: 10,
),
Text(
"Appointment History",
style: GoogleFonts.lato(
fontSize: 16,
fontWeight: FontWeight.bold,
color: Colors.black,
),
),
Expanded(
child: Container(
padding: const EdgeInsets.only(right: 10),
alignment: Alignment.centerRight,
child: SizedBox(
height: 30,
child: TextButton(
onPressed: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) =>
const AppointmentHistoryList()));
},

```

```

child: const Text('View all'),
),
),
),
)
],
),
const SizedBox(
height: 10,
),
Expanded(
child: Scrollbar(
thumbVisibility: true,
child: Container(
padding: const EdgeInsets.only(right: 15),
child: const AppointmentHistoryList(),
),
),
),
),
),
const SizedBox(
height: 30,
),
),
),
),
),
);
}

// for picking image from device
Future selectOrTakePhoto(ImageSource imageSource) async {
XFile? file =
await ImagePicker().pickImage(source: imageSource, imageQuality: 12);

if (file != null) {
var im = await file.readAsBytes();
// upload image to cloud
await uploadFile(im, file.name);
return;
}

print('No photo was selected or taken');
}

// dialog for option of take photo from
Future _showSelectionDialog(BuildContext context) async {
showDialog(
context: context,
builder: (BuildContext context) {
return SimpleDialog(
title: const Text('Select photo'),

```



```

children: <Widget>[
  SimpleDialogOption(
    child: const Text('From gallery'),
    onPressed: () {
      selectOrTakePhoto(ImageSource.gallery);
      Navigator.pop(context);
    },
  ),
  SimpleDialogOption(
    child: const Text('Take a photo'),
    onPressed: () {
      selectOrTakePhoto(ImageSource.camera);
      Navigator.pop(context);
    },
  ),
],
);
}
}

```

```

Future uploadFile(Uint8List imgBytes, String fileName) async {
  const cloudName = 'dik3t7jpt';
  const uploadPreset = 'docfusion';

```

```

  final url = Uri.parse(
    "https://api.cloudinary.com/v1_1/$cloudName/image/upload");

```

```

  try {
    final request = http.MultipartRequest('POST', url);

```

```

    request.fields['upload_preset'] =
      uploadPreset;

```

```

    request.files.add(http.MultipartFile.fromBytes(
      'file',
      imgBytes,
      filename: fileName,
    ));

```

```

    final response = await request.send();

```

```

    if (response.statusCode == 200) {
      final res = await http.Response.fromStream(response);
      final data = json.decode(res.body);
      final imageUrl = data['secure_url'];

```

```

    print("Uploaded to Cloudinary: $imageUrl");

```

```

    setState(() {

```

```

image = imageUrl;
});

// Save to Firestore
await FirebaseFirestore.instance
.collection(isDoctor ? 'doctor' : 'patient')
.doc(user.uid)
.set({'profilePhoto': imageUrl}, SetOptions(merge: true));

await FirebaseFirestore.instance
.collection('users')
.doc(user.uid)
.set({'profilePhoto': imageUrl}, SetOptions(merge: true));
} else {
print("Failed to upload to Cloudinary: ${response.statusCode}");
}
} catch (e) {
print("Cloudinary Upload Error: $e");
}
}

// upload image
// Future uploadFile(Uint8List img, String fileName) async {
//   final destination = 'dp/${user.displayName}-${fileName}';
//   try {
//     final ref = storage.ref(destination);
//
//     UploadTask uploadTask = ref.putData(img);
//     TaskSnapshot snapshot = await uploadTask;
//
//     String downloadUrl = await snapshot.ref.getDownloadURL();
//     print('image url : $downloadUrl');
//
//     setState(() {
//       image = Uri.decodeFull(downloadUrl.toString());
//     });
//     FirebaseFirestore.instance
//       .collection(isDoctor ? 'doctor' : 'patient')
//       .doc(user.uid)
//       .set({
//         'profilePhoto': downloadUrl,
//       }, SetOptions(merge: true));
//
//     // main user data
//     FirebaseFirestore.instance.collection('users').doc(user.uid).set({
//       'profilePhoto': downloadUrl,
//     }, SetOptions(merge: true));
//
//     print("uploaded !!!");
//   } catch (e) {
//     print(e.toString());
//     print('error occured');

```

```
// }  
// }  
// }
```

lib\screens\register.dart

```
import 'dart:math';
import 'package:doc_fusion/globals.dart' as globals;
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/sign_in.dart';

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

  @override
  State<Register> createState() => _RegisterState();
}

class _RegisterState extends State<Register> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController _displayName = TextEditingController();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final TextEditingController _passwordConfirmController =
    TextEditingController();

  int type = -1;

  FocusNode f1 = FocusNode();
  FocusNode f2 = FocusNode();
  FocusNode f3 = FocusNode();
  FocusNode f4 = FocusNode();

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: Center(
          child: NotificationListener<OverscrollIndicatorNotification>(
            onNotification: (OverscrollIndicatorNotification overscroll) {
              overscroll.disallowIndicator();
            },
            return true;
          ),
          child: SingleChildScrollView(
            child: Column(
              children: <Widget>[
```

```

Container(
padding: const EdgeInsets.fromLTRB(10, 40, 10, 10),
child: _signUp(),
),
],
),
),
),
),
),
);
}

```

```

Widget _signUp() {
return Form(
key: _formKey,
child: Padding(
padding: const EdgeInsets.only(right: 16, left: 16),
child: Column(
crossAxisAlignment: CrossAxisAlignment.center,
children: [
const SizedBox(
height: 20,
),
Container(
padding: const EdgeInsets.only(bottom: 50),
child: Text(
'Sign up',
style: GoogleFonts.lato(
fontSize: 30,
fontWeight: FontWeight.bold,
),
),
),
// display name
TextFormField(
focusNode: f1,
style: GoogleFonts.lato(
fontSize: 18,
fontWeight: FontWeight.w800,
),
keyboardType: TextInputType.emailAddress,
controller: _displayName,
decoration: InputDecoration(
contentPadding:
const EdgeInsets.only(left: 20, top: 10, bottom: 10),
border: const OutlineInputBorder(
borderRadius: BorderRadius.all(Radius.circular(90.0)),
borderSide: BorderSide.none,
),
filled: true,
fillColor: Colors.grey[350],
hintText: 'Name',
hintStyle: GoogleFonts.lato(

```

```

color: Colors.black26,
fontSize: 18,
fontWeight: FontWeight.w800,
),
),
onFieldSubmitted: (value) {
  f1.unfocus();
  FocusScope.of(context).requestFocus(f2);
},
textInputAction: TextInputAction.next,
validator: (value) {
  if (value!.isEmpty) return 'Please enter the Name';
  return null;
},
),
const SizedBox(
  height: 25.0,
),

// email
TextFormField(
  focusNode: f2,
  style: GoogleFonts.lato(
    fontSize: 18,
    fontWeight: FontWeight.w800,
  ),
  keyboardType: TextInputType.emailAddress,
  controller: _emailController,
  decoration: InputDecoration(
    contentPadding:
      const EdgeInsets.only(left: 20, top: 10, bottom: 10),
    border: const OutlineInputBorder(
      borderRadius: BorderRadius.all(Radius.circular(90.0)),
      borderSide: BorderSide.none,
    ),
    filled: true,
    fillColor: Colors.grey[350],
    hintText: 'Email',
    hintStyle: GoogleFonts.lato(
      color: Colors.black26,
      fontSize: 18,
      fontWeight: FontWeight.w800,
    ),
  ),
  onFieldSubmitted: (value) {
    f2.unfocus();
    if (_passwordController.text.isEmpty) {
      FocusScope.of(context).requestFocus(f3);
    }
  },
  textInputAction: TextInputAction.next,
  validator: (value) {
    if (value!.isEmpty) {
      return 'Please enter the Email';
    }
  },
)

```

```

} else if (!emailValidate(value)) {
return 'Please enter correct Email';
}
return null;
},
),
const SizedBox(
height: 25.0,
),

// password
TextFormField(
focusNode: f3,
style: GoogleFonts.lato(
fontSize: 18,
fontWeight: FontWeight.w800,
),
//keyboardType: TextInputType.visiblePassword,
controller: _passwordController,
decoration: InputDecoration(
contentPadding:
const EdgeInsets.only(left: 20, top: 10, bottom: 10),
border: const OutlineInputBorder(
borderRadius: BorderRadius.all(Radius.circular(90.0)),
borderSide: BorderSide.none,
),
filled: true,
fillColor: Colors.grey[350],
hintText: 'Password',
hintStyle: GoogleFonts.lato(
color: Colors.black26,
fontSize: 18,
fontWeight: FontWeight.w800,
),
),
onFieldSubmitted: (value) {
f3.unfocus();
if (_passwordConfirmController.text.isEmpty) {
FocusScope.of(context).requestFocus(f4);
}
},
textInputAction: TextInputAction.next,
validator: (value) {
if (value!.isEmpty) {
return 'Please enter the Password';
} else if (value.length < 8) {
return 'Password must be at least 8 characters long';
} else {
return null;
}
},
obscureText: true,
),
const SizedBox(

```

```

height: 25.0,
),

// confirm password
TextFormField(
  focusNode: f4,
  style: GoogleFonts.lato(
    fontSize: 18,
    fontWeight: FontWeight.w800,
  ),
  controller: _passwordConfirmController,
  decoration: InputDecoration(
    contentPadding:
      const EdgeInsets.only(left: 20, top: 10, bottom: 10),
    border: const OutlineInputBorder(
      borderRadius: BorderRadius.all(Radius.circular(90.0)),
      borderSide: BorderSide.none,
    ),
    filled: true,
    fillColor: Colors.grey[350],
    hintText: 'Confirm Password',
    hintStyle: GoogleFonts.lato(
      color: Colors.black26,
      fontSize: 18,
      fontWeight: FontWeight.w800,
    ),
  ),
  onFieldSubmitted: (value) {
    f4.unfocus();
  },
  textInputAction: TextInputAction.done,
  validator: (value) {
    if (value!.isEmpty) {
      return 'Please enter the Password';
    } else if (value.compareTo(_passwordController.text) != 0) {
      return 'Password not Matching';
    } else {
      return null;
    }
  },
  obscureText: true,
),

const SizedBox(
  height: 20,
),
// type of account
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    SizedBox(
      width: MediaQuery.of(context).size.width / 2.5,
      height: 50,
      child: ElevatedButton(

```



```

onPressed: () {
  setState(() {
    type = 0;
  });
},
style: ElevatedButton.styleFrom(
  elevation: 2, backgroundColor: Colors.grey[350],
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(32.0),
  ),
  side: BorderSide(
    width: 5.0,
    color: Colors.black38,
    style:
      type == 0 ? BorderStyle.solid : BorderStyle.none,
  )),
child: Text(
  "Doctor",
  style: GoogleFonts.lato(
    color: type == 0 ? Colors.black38 : Colors.white,
    fontSize: 18.0,
    fontWeight: FontWeight.bold,
  ),
),
),
),
),
const SizedBox(
  height: 50,
  child: Center(
    child: Text(
      'or',
      textAlign: TextAlign.center,
    ),
  ),
),
SizedBox(
  width: MediaQuery.of(context).size.width / 2.5,
  height: 50,
  child: ElevatedButton(
    onPressed: () {
      setState(() {
        type = 1;
      });
    },
    style: ElevatedButton.styleFrom(
      elevation: 2, backgroundColor: Colors.grey[350],
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(32.0),
      ),
      side: BorderSide(
        style: type == 1 ? BorderStyle.solid : BorderStyle.none,
        width: 5,
        color: Colors.black38,
      ),

```

```

),
child: Text(
  "Patient",
  style: GoogleFonts.lato(
    color: type == 1 ? Colors.black38 : Colors.white,
    fontSize: 18.0,
    fontWeight: FontWeight.bold,
  ),
),
),
),
],
),

// signin button
Container(
  padding: const EdgeInsets.only(top: 25.0),
  child: SizedBox(
    width: double.infinity,
    height: 50,
    child: ElevatedButton(
      onPressed: () async {
        if (_formKey.currentState!.validate() && type != -1) {
          showLoaderDialog(context);
          _registerAccount();
        }
      },
      style: ElevatedButton.styleFrom(
        foregroundColor: Colors.black, backgroundColor: Colors.indigo[900], elevation: 2,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(32.0),
        ),
      ),
      child: Text(
        "Sign In",
        style: GoogleFonts.lato(
          color: Colors.white,
          fontSize: 18.0,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ),

// divider
Container(
  padding: const EdgeInsets.only(top: 25, left: 10, right: 10),
  width: MediaQuery.of(context).size.width,
  child: const Divider(
    thickness: 1.5,
  ),
),

```

```

// signin page option
Padding(
padding: const EdgeInsets.only(top: 5.0),
child: Row(
mainAxisAlignment: MainAxisAlignment.center,
crossAxisAlignment: CrossAxisAlignment.center,
children: [
Text(
"Already have an account?",
style: GoogleFonts.lato(
fontSize: 15.0,
fontWeight: FontWeight.w700,
),
),
TextButton(
style: ButtonStyle(
overlayColor:
MaterialStateProperty.all(Colors.transparent)),
onPressed: () => _pushPage(context, const SignIn()),
child: Text(
'Sign in',
style: GoogleFonts.lato(
fontSize: 15,
color: Colors.indigo[700],
fontWeight: FontWeight.w600,
),
),
),
],
),
),
),
),
),
),
),
),
),
);
}

showAlertDialog(BuildContext context) {
Navigator.pop(context);
// set up the button
Widget okButton = TextButton(
child: Text(
"OK",
style: GoogleFonts.lato(fontWeight: FontWeight.bold),
),
onPressed: () {
Navigator.pop(context);
FocusScope.of(context).requestFocus(f2);
},
);

// set up the AlertDialog
AlertDialog alert = AlertDialog(
title: Text(

```

```

"Error!",
style: GoogleFonts.lato(
fontWeight: FontWeight.bold,
),
),
content: Text(
"Email already Exists",
style: GoogleFonts.lato(),
),
actions: [
okButton,
],
);

// show the dialog
showDialog(
context: context,
builder: (BuildContext context) {
return alert;
},
);
}

showLoaderDialog(BuildContext context) {
AlertDialog alert = AlertDialog(
content: Row(
children: [
const CircularProgressIndicator(),
Container(
margin: const EdgeInsets.only(left: 15),
child: const Text("Loading..."),
],
),
);
showDialog(
barrierDismissible: false,
context: context,
builder: (BuildContext context) {
return alert;
},
);
}

bool emailValidate(String email) {
if (RegExp(
r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*-+/?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+"
).hasMatch(email)) {
return true;
} else {
return false;
}
}

void _registerAccount() async {

```

```

User? user;
UserCredential? credential;

try {
credential = await _auth.createUserWithEmailAndPassword(
email: _emailController.text,
password: _passwordController.text,
);
} catch (error) {
if (error.toString().compareTo(
'[firebase_auth/email-already-in-use] The email address is already in use by another
account.') ==
0) {
showAlertDialog(context);
}
print(error.toString());
}
user = credential!.user;

if (user != null) {
if (!user.emailVerified) {
await user.sendEmailVerification();
}
await user.updateDisplayName(_displayName.text);

String name =
(type == 0) ? 'Dr. ${_displayName.text}' : _displayName.text;
String accountType = (type == 0) ? 'doctor' : 'patient';
FirebaseFirestore.instance.collection('users').doc(user.uid).set({
'name': name,
'type': accountType,
'email': user.email,
}, SetOptions(merge: true));

// set data according to type (doctor or patient)
Map<String, dynamic> mp = {
'id': user.uid,
'type': accountType,
'name': name,
'birthDate': null,
'email': user.email,
'phone': null,
'bio': null,
'address': null,
'profilePhoto': null,
};
// if doctor
if (type == 0) {
mp.addAll({
'openHour': "09:00",
'closeHour': "21:00",
'rating': double.parse(
(3 + Random().nextDouble() * 1.9).toStringAsPrecision(2)),
'specification': null,

```

```
'specialization': 'general',
});
globals.isDoctor = true;
}

// sep
FirebaseFirestore.instance.collection(accountType).doc(user.uid).set(mp);

Navigator.of(context)
.pushNamedAndRemoveUntil('/home', (Route<dynamic> route) => false);
} else {}
}

void _pushPage(BuildContext context, Widget page) {
Navigator.of(context).push(
MaterialPageRoute<void>(builder: (_) => page),
);
}
}
```

lib\screens\setting.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/firestore_data/user_details.dart';

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

  @override
  State<UserSettings> createState() => _UserSettingsState();
}

class _UserSettingsState extends State<UserSettings> {
  UserDetails detail = const UserDetails();

  final FirebaseAuth _auth = FirebaseAuth.instance;
  User? user;

  Future<void> _getUser() async {
    user = _auth.currentUser;
  }

  Future _signOut() async {
    await _auth.signOut();
  }

  @override
  void initState() {
    super.initState();
    _getUser();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        elevation: 2,
        backgroundColor: Colors.white,
        leading: Padding(
          padding: const EdgeInsets.only(left: 10),
          child: IconButton(
            splashRadius: 25,
            icon: const Icon(
              Icons.arrow_back_ios,
              color: Colors.indigo,
            ),
            onPressed: () => Navigator.of(context).pop(),
          ),
        ),
        title: Text(
          'User Settings',
          style: GoogleFonts.lato(
```

```

color: Colors.indigo, fontSize: 20, fontWeight: FontWeight.bold),
),
),
body: SingleChildScrollView(
  child: Column(
    children: [
      // info to edit
      const UserDetails(),

      // log out button
      Container(
        margin: const EdgeInsets.only(
          left: 15, right: 15, bottom: 30, top: 5),
        padding: const EdgeInsets.symmetric(horizontal: 14),
        height: MediaQuery.of(context).size.height / 14,
        width: MediaQuery.of(context).size.width,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          color: Colors.red,
        ),
        child: TextButton(
          onPressed: () {
            Navigator.of(context).pushNamedAndRemoveUntil(
              '/login', (Route<dynamic> route) => false);
            _signOut();
          },
          style: TextButton.styleFrom(backgroundColor: Colors.grey),
          child: Text(
            'Sign out',
            style: GoogleFonts.lato(
              color: Colors.black,
              fontSize: 16,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ],
    ),
  );
}
}

```


lib\screens\sign_in.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/globals.dart';
import 'package:doc_fusion/helperFunction/sharedpref_helper.dart';
import 'package:doc_fusion/screens/register.dart';

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

  @override
  State<SignIn> createState() => _SignInState();
}

class _SignInState extends State<SignIn> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  final TextEditingController _passwordController = TextEditingController();

  FocusNode f1 = FocusNode();
  FocusNode f2 = FocusNode();
  FocusNode f3 = FocusNode();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      key: _scaffoldKey,
      body: Builder(builder: (BuildContext context) {
        return SafeArea(
          child: NotificationListener<OverscrollIndicatorNotification>(
            onNotification: (OverscrollIndicatorNotification overscroll) {
              overscroll.disallowIndicator();
            },
            child: SingleChildScrollView(
              child: Column(
                children: <Widget>[
                  Container(
                    padding: const EdgeInsets.fromLTRB(10, 30, 10, 10),
                    child: withEmailPassword(),
                  ),
                ],
              ),
            ),
          ),
        );
      },
    );
  }
}
```

```
}
```

```
Widget withEmailPassword() {  
  return Form(  
    key: _formKey,  
    child: Padding(  
      padding: const EdgeInsets.only(right: 16, left: 16),  
      child: Column(  
        crossAxisAlignment: CrossAxisAlignment.center,  
        children: [  
          SizedBox(  
            width: double.infinity,  
            child: Image.asset(  
              'assets/vector-doc2.jpg',  
              scale: 3.5,  
            ),  
            ),  
            ),  
            const SizedBox(  
              height: 10,  
            ),  
            Container(  
              padding: const EdgeInsets.only(bottom: 25),  
              child: Text(  
                'Login',  
                style: GoogleFonts.lato(  
                  fontSize: 30,  
                  fontWeight: FontWeight.bold,  
                ),  
                ),  
                ),  
                ),  
                // email  
                TextFormField(  
                  focusNode: f1,  
                  style: GoogleFonts.lato(  
                    fontSize: 18,  
                    fontWeight: FontWeight.w800,  
                  ),  
                  keyboardType: TextInputType.emailAddress,  
                  controller: _emailController,  
                  decoration: InputDecoration(  
                    contentPadding:  
                      const EdgeInsets.only(left: 20, top: 10, bottom: 10),  
                    border: const OutlineInputBorder(  
                      borderRadius: BorderRadius.all(Radius.circular(90.0)),  
                      borderSide: BorderSide.none,  
                    ),  
                    filled: true,  
                    fillColor: Colors.grey[350],  
                    hintText: 'Email',  
                    hintStyle: GoogleFonts.lato(  
                      color: Colors.black26,  
                      fontSize: 18,  
                      fontWeight: FontWeight.w800,  
                    ),  
                  ),  
                ),  
              ),  
            ),  
          ],  
        ),  
      ),  
    ),  
  ),  
);
```

```

),
onFieldSubmitted: (value) {
  f1.unfocus();
  FocusScope.of(context).requestFocus(f2);
},
textInputAction: TextInputAction.next,
validator: (value) {
  if (value!.isEmpty) {
    return 'Please enter the Email';
  } else if (!emailValidate(value)) {
    return 'Please enter correct Email';
  } else {
    return null;
  }
},
),
const SizedBox(
  height: 25.0,
),

// password
TextFormField(
  focusNode: f2,
  style: GoogleFonts.lato(
    fontSize: 18,
    fontWeight: FontWeight.w800,
  ),
  //keyboardType: TextInputType.visiblePassword,
  controller: _passwordController,
  decoration: InputDecoration(
    contentPadding:
    const EdgeInsets.only(left: 20, top: 10, bottom: 10),
    border: const OutlineInputBorder(
      borderRadius: BorderRadius.all(Radius.circular(90.0)),
      borderSide: BorderSide.none,
    ),
    filled: true,
    fillColor: Colors.grey[350],
    hintText: 'Password',
    hintStyle: GoogleFonts.lato(
      color: Colors.black26,
      fontSize: 18,
      fontWeight: FontWeight.w800,
    ),
  ),
  onFieldSubmitted: (value) {
    f2.unfocus();
    FocusScope.of(context).requestFocus(f3);
  },
  textInputAction: TextInputAction.done,
  validator: (value) {
    if (value!.isEmpty) return 'Please enter the Passord';
    return null;
  },

```

```

obscureText: true,
),
// sign in button
Container(
padding: const EdgeInsets.only(top: 25.0),
child: SizedBox(
width: double.infinity,
height: 50,
child: ElevatedButton(
focusNode: f3,
onPressed: () async {
if (_formKey.currentState!.validate()) {
showLoaderDialog(context);
_signInWithEmailAndPassword();
}
},
style: ElevatedButton.styleFrom(
foregroundColor: Colors.black, backgroundColor: Colors.indigo[900], elevation: 2,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(32.0),
),
),
child: Text(
"Sign In",
style: GoogleFonts.lato(
color: Colors.white,
fontSize: 18.0,
fontWeight: FontWeight.bold,
),
),
),
// forget password
Container(
padding: const EdgeInsets.only(top: 15),
child: TextButton(
style: ButtonStyle(
overlayColor:
MaterialStateProperty.all(Colors.transparent)),
onPressed: () {},
child: Text(
'Forgot Password?',
style: GoogleFonts.lato(
fontSize: 16,
color: Colors.black54,
fontWeight: FontWeight.w600,
),
),
),
// dont have account
Padding(

```

```
padding: const EdgeInsets.only(top: 18.0),
child: Row(
  mainAxisAlignment: MainAxisAlignment.center,
  crossAxisAlignment: CrossAxisAlignment.center,
  children: [
    Text(
      "Don't have an account?",
      style: GoogleFonts.lato(
        fontSize: 15.0,
        fontWeight: FontWeight.w700,
      ),
    ),
    TextButton(
      style: ButtonStyle(
        overlayColor:
          MaterialStateProperty.all(Colors.transparent)),
      onPressed: () => _pushPage(context, const Register()),
      child: Text(
        'Signup here',
        style: GoogleFonts.lato(
          fontSize: 15,
          color: Colors.indigo[700],
          fontWeight: FontWeight.w600,
        ),
      ),
    ],
  ),
);
}
```

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

```
showLoaderDialog(BuildContext context) {
  AlertDialog alert = AlertDialog(
    content: Row(
      children: [
        const CircularProgressIndicator(),
        Container(
          margin: const EdgeInsets.only(left: 15),
          child: const Text("Loading..."),
        ),
      ],
    ),
  );
  showDialog(
```

```

barrierDismissible: false,
context: context,
builder: (BuildContext context) {
return alert;
},
);
}

bool emailValidate(String email) {
if (RegExp(
r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*-+/-=?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+"
).hasMatch(email)) {
return true;
} else {
return false;
}
}

void _signInWithEmailAndPassword() async {
try {
final User? user = (await _auth.signInWithEmailAndPassword(
email: _emailController.text,
password: _passwordController.text,
))
.user;
if (!user!.emailVerified) {
await user.sendEmailVerification();
}

DocumentSnapshot snap = await FirebaseFirestore.instance
.collection('users')
.doc(user.uid)
.get();
var basicInfo = snap.data() as Map<String, dynamic>;

isDoctor = basicInfo['type'] == 'doctor' ? true : false;

// save data to local storage
SharedPreferencesHelper().saveUserId(user.uid);
// SharedPreferencesHelper().saveProfileUrl(user);
SharedPreferencesHelper().saveUserName(basicInfo['name']);
SharedPreferencesHelper().saveAccountType(basicInfo['type']== 'doctor'? true :false);

Navigator.of(context)
.pushNamedAndRemoveUntil('/home', (Route<dynamic> route) => false);
} catch (e) {
final snackBar = SnackBar(
content: Row(
children: const [
Icon(
Icons.info_outline,
color: Colors.white,
),
Text(" There was a problem signing you in"),

```

```
],
),
);
Navigator.pop(context);
ScaffoldMessenger.of(context).showSnackBar(snackBar);
}
}

void _pushPage(BuildContext context, Widget page) {
Navigator.of(context).push(
MaterialPageRoute<void>(builder: (_) => page),
);
}
}
```

lib\screens\skip.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:introduction_screen/introduction_screen.dart';

import 'firebase_auth.dart';

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

  @override
  State<Skip> createState() => _SkipState();
}

class _SkipState extends State<Skip> {
  List<PageViewModel> getpages() {
    return [
      PageViewModel(
        title: '',
        image: Image.asset(
          'assets/doc.png',
          //fit: BoxFit.cover,
        ),
        //body: "Search Doctors",
        bodyWidget: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'Search Doctors',
              style:
                GoogleFonts.lato(fontSize: 30, fontWeight: FontWeight.w900),
            ),
            Text(
              'Find popular doctors nearby you',
              style: GoogleFonts.lato(
                fontSize: 15,
                color: Colors.grey[500],
                fontWeight: FontWeight.w800),
            ),
          ],
        ),
      ),
      PageViewModel(
        title: '',
        image: Image.asset(
          'assets/disease.png',
          //fit: BoxFit.cover,
        ),
        //body: "Search Doctors",
        bodyWidget: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
```



```

'Search Disease',
style:
GoogleFonts.lato(fontSize: 30, fontWeight: FontWeight.w900),
),
Text(
'Find information about disease',
style: GoogleFonts.lato(
fontSize: 15,
color: Colors.grey[500],
fontWeight: FontWeight.w800),
),
],
),
),
];
}

@override
Widget build(BuildContext context) {
return Scaffold(
body: IntroductionScreen(
globalBackgroundColor: Colors.lightBlue[100],
pages: getpages(),
showNextButton: false,
showSkipButton: true,
skip: SizedBox(
width: 80,
height: 48,
child: Card(
shape:
RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),
color: Colors.blue[300],
shadowColor: Colors.blueGrey[100],
elevation: 5,
child: Center(
child: Text(
'Skip',
textAlign: TextAlign.center,
style:
GoogleFonts.lato(fontSize: 25, fontWeight: FontWeight.w900),
),
),
),
),
done: SizedBox(
height: 48,
child: Card(
shape:
RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),
color: Colors.blue[300],
shadowColor: Colors.blueGrey[200],
elevation: 5,
child: Center(
child: Text(

```

```
'Continue',
textAlign: TextAlign.center,
style:
  GoogleFonts.lato(fontSize: 15, fontWeight: FontWeight.w900),
),
),
),
),
onDone: () => _pushPage(context, const FirebaseAuth()),
),
);
}
```

```
void _pushPage(BuildContext context, Widget page) {
  Navigator.of(context).push(
    MaterialPageRoute<void>(builder: (_) => page),
  );
}
}
```

lib\screens\chat\chats.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_database/ui/firebase_animated_list.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/chat/chat_dao.dart';
import 'package:doc_fusion/screens/chat/chat_room.dart';

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

  @override
  State<Chats> createState() => _ChatsState();
}

class _ChatsState extends State<Chats> {
  final ScrollController _scrollController = ScrollController();
  final FirebaseAuth _auth = FirebaseAuth.instance;
  late User user;
  late var chatDao;

  Future<void> _getUser() async {
    user = _auth.currentUser!;
  }

  Future<void> _set() async {
    chatDao = ChatDao(user.uid);
  }

  @override
  void initState() {
    super.initState();
    _getUser();
    _set();
    setState(() {});
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: Column(
          children: [_getChatList()],
        ),
      ),
    );
  }

  Widget _getChatList() {
    return Expanded(
      child: FirebaseAnimatedList(
        controller: _scrollController,
        query: chatDao.getChatQuery(),
      ),
    );
  }
}
```

```

itemBuilder: (context, snapshot, animation, index) {
final json = snapshot.value as Map<dynamic, dynamic>;
print('ok:' + json.toString());
return ChatCard(
  userId: json['uid'] ?? 'No id',
  profileUrl: json['photo'] ?? " ",
  userName: json['name'] ?? 'Not Set');
},
),
);
}
}

```

```

class ChatCard extends StatelessWidget {
final String userId;
final String profileUrl;
final String userName;
const ChatCard(
  {Key? key,
  required this.userId,
  required this.profileUrl,
  required this.userName})
: super(key: key);

```

```

@override
Widget build(BuildContext context) {
return Container(
padding: const EdgeInsets.symmetric(vertical: 1, horizontal: 14),
child: Card(
color: Colors.blue[50],
elevation: 1,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(10),
),
child: Container(
padding: const EdgeInsets.only(left: 10, right: 10),
width: MediaQuery.of(context).size.width,
height: MediaQuery.of(context).size.height / 9,
child: TextButton(
onPressed: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => ChatRoom(
user2Id: userId,
user2Name: userName,
profileUrl: profileUrl),
),
);
},
child: Row(
crossAxisAlignment: CrossAxisAlignment.center,
//mainAxisAlignment: MainAxisAlignment.spaceAround,
children: [

```

```
CircleAvatar(  
  backgroundImage: NetworkImage(profileUrl),  
  backgroundColor: Colors.grey[300],  
  radius: 30,  
)  
const SizedBox(  
  width: 20,  
)  
Column(  
  crossAxisAlignment: CrossAxisAlignment.start,  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
    Text(  
      userName,  
      style: GoogleFonts.lato(  
        fontWeight: FontWeight.bold,  
        fontSize: 19,  
        color: Colors.black87,  
      ),  
    ),  
  ],  
)  
];  
};  
}
```

lib\screens\chat\chat_dao.dart

```
import 'package:firebase_database/firebase_database.dart';

class ChatDao {
  late DatabaseReference _chatRef;
  final String yourId;
  ChatDao(this.yourId) {
    _chatRef =
    FirebaseDatabase.instance.ref('users').child(yourId);
  }

  Query getChatQuery() {
    return _chatRef;
  }
}
```

lib\screens\chat\chat_room.dart

```
import 'dart:async';
import 'dart:math';

import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_database/ui/firebase_animated_list.dart';
import 'package:flutter/material.dart';

import 'message.dart';
import 'message_dao.dart';

class ChatRoom extends StatefulWidget {
  final String user2Id;
  final String user2Name;
  String defaultProfileUrl =
    'https://cdn.icon-icons.com/icons2/1378/PNG/512/avatardefault_92824.png';
  final String profileUrl;

  ChatRoom(
    {Key? key,
    required this.user2Id,
    required this.user2Name,
    required this.profileUrl})
    : super(key: key);

  ////////////////////////////////////////////////////
  // final chatRoomId = "";

  @override
  State<ChatRoom> createState() => _ChatRoomState();
}

class _ChatRoomState extends State<ChatRoom> {
  final TextEditingController _messageController = TextEditingController();
  final ScrollController _scrollController = ScrollController();

  final FirebaseAuth _auth = FirebaseAuth.instance;
  late User user;
  late var messageDao;

  Future<void> _getUser() async {
    user = _auth.currentUser!;
  }

  Future<void> _setChatRoomId() async {
    messageDao = MessageDao(user1: user.uid, user2: widget.user2Id);
  }

  bool _canSendMessage() {
    return _messageController.text.isNotEmpty;
  }

  void _sendMessage() {
```

```

if (_canSendMessage()) {
var currTime = DateTime.now().toUtc().toString();
final message = Message(
message: _messageController.text, senderId: user.uid, time: currTime);
messageDao.saveMessage(message);
_messageController.clear();
setState(() {});
}
}

@override
void initState() {
super.initState();
_getUser();
_setChatRoomId();
setState(() {});
}

@override
Widget build(BuildContext context) {
return Scaffold(
// app bar with back button, profile picture, and name, and call button
appBar: AppBar(
leading: IconButton(
icon: const Icon(Icons.arrow_back),
onPressed: () => Navigator.pop(context),
),
title: Row(
children: [
CircleAvatar(
backgroundImage: NetworkImage(widget.profileUrl),
),
const SizedBox(width: 10),
// prevent over flow text if name is too long
Text(
widget.user2Name.substring(0, min(widget.user2Name.length, 12)),
),
],
),
actions: <Widget>[
IconButton(
icon: const Icon(Icons.call),
onPressed: () {},
),
IconButton(
icon: const Icon(Icons.videocam),
onPressed: () {},
),
IconButton(
icon: const Icon(Icons.more_vert),
onPressed: () {},
),
],
),

```



```

// body with messages
body: Column(children: [_getMessageList()]),

// type message and send button
bottomSheet: BottomAppBar(
  child: SizedBox(
    height: 50,
    child: Row(
      children: <Widget>[
        IconButton(
          icon: const Icon(Icons.attach_file),
          onPressed: () {},
        ),
        Expanded(
          child: Container(
            margin: const EdgeInsets.symmetric(horizontal: 10),
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(30),
              color: Colors.grey[200],
            ),
            child: Row(
              children: <Widget>[
                const SizedBox(width: 10),
                const Icon(Icons.insert_emoticon),
                const SizedBox(width: 10),
                Expanded(
                  child: TextField(
                    controller: _messageController,
                    keyboardType: TextInputType.text,
                    decoration: const InputDecoration(
                      border: InputBorder.none,
                      hintText: 'Type a message',
                    ),
                  ),
                ),
              ],
            ),
          ),
        IconButton(
          icon: const Icon(Icons.send),
          onPressed: () {
            _sendMessage();
          },
        ),
      ],
    ),
  ),
);
}

```

```

Widget _getMessageList() {
  return Expanded(

```

```

child: FirebaseAnimatedList(
  controller: _scrollController,
  query: messageDao.getMessageQuery(),
  itemBuilder: (context, snapshot, animation, index) {
    final json = snapshot.value as Map<dynamic, dynamic>;
    final message = Message.fromJson(json);
    return MessageWidget(
      message: message.message,
      time: message.time,
      isMe: message.senderId == user.uid);
  },
),
);
}
}

```

```

class MessageWidget extends StatelessWidget {
  final String message;
  final String time;
  final bool isMe;
  const MessageWidget(
    {Key? key, required this.message, required this.time, required this.isMe})
    : super(key: key);

```

```

@override
Widget build(BuildContext context) {
  // final size = msg.length;

  return Container(
    alignment: isMe ? Alignment.centerRight : Alignment.centerLeft,
    // margin: const EdgeInsets.symmetric(vertical: 10, horizontal: 10),

```

```

    child: Container(
      padding: const EdgeInsets.symmetric(vertical: 10, horizontal: 14),
      // margin: const EdgeInsets.symmetric(vertical: 5, horizontal: 8),
      margin: EdgeInsets.fromLTRB(isMe ? 60 : 8, 5, isMe ? 8 : 60, 5),
      decoration: BoxDecoration(
        // borderRadius: BorderRadius.circular(15),
        borderRadius: BorderRadius.only(
          topLeft: isMe ? const Radius.circular(15) : Radius.zero,
          topRight: isMe ? Radius.zero : const Radius.circular(15),
          bottomLeft: const Radius.circular(15),
          bottomRight: const Radius.circular(15),
        ),
        color: isMe ? Colors.blue[300] : Colors.grey[300],
      ),
      child: Text(
        message,
        style: const TextStyle(
          fontSize: 16,
        ),
      ),
    ),
  );
}

```


lib\screens\chat\message.dart

```
class Message {  
  final String senderId;  
  final String message;  
  final String time;  
  
  Message({required this.senderId, required this.message, required this.time});  
  
  Message.fromJson(Map<dynamic, dynamic> json)  
    : senderId = json['senderId'],  
      message = json['message'],  
      time = json['time'];  
  
  Map<dynamic, dynamic> toJson() => {  
    'senderId': senderId,  
    'message': message,  
    'time': time,  
  };  
}
```

lib\screens\chat\message_dao.dart

```
import 'package:cloud_firestore/cloud_firestore.dart' as store;
import 'package:firebase_database/firebase_database.dart';
import 'package:doc_fusion/screens/chat/message.dart';

class MessageDao {
  late DatabaseReference _messageRef;
  final String user1;
  final String user2;

  MessageDao({required this.user1, required this.user2}) {
    var users = [user1, user2];
    users.sort();

    String chatRoomId = users.join('-');

    // create gateway of two chats
    createGateWay();

    _messageRef =
    FirebaseDatabase.instance.ref('chats').child(chatRoomId).child('chat');
  }

  Future<void> createGateWay() async {
    // make two way terminal if not present
    store.DocumentSnapshot snap1 = await store.FirebaseFirestore.instance
    .collection('users')
    .doc(user1)
    .get();
    store.DocumentSnapshot snap2 = await store.FirebaseFirestore.instance
    .collection('users')
    .doc(user2)
    .get();

    var user1Info = snap1.data() as Map<String, dynamic>;
    var user2Info = snap2.data() as Map<String, dynamic>;

    var user1Data = {
      'uid': user1,
      'name': user1Info['name'],
      'photo': user1Info['profilePhoto']
    };
    var user2Data = {
      'uid': user2,
      'name': user2Info['name'],
      'photo': user2Info['profilePhoto']
    };
    print(user1Data);
    print(user2Data);
    FirebaseDatabase.instance
    .ref('users')
    .child(user1)
    .child(user2)
```

```
.set(user2Data);
FirebaseDatabase.instance
.ref('users')
.child(user2)
.child(user1)
.set(user1Data);
}
```

```
void saveMessage(Message message) {
print(message.toJson());
_messageRef.push().set(message.toJson());
}
```

```
Query getMessageQuery() {
return _messageRef;
}
}
```

lib\screens\doctor\main_page_doctor.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:google_nav_bar/google_nav_bar.dart';
import 'package:doc_fusion/screens/chat/chats.dart';
import 'package:doc_fusion/screens/my_profile.dart';
import 'package:doc_fusion/screens/patient/appointments.dart';
import 'package:typicons_flutter/typicons_flutter.dart';

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

  @override
  State<MainPageDoctor> createState() => _MainPageDoctorState();
}

class _MainPageDoctorState extends State<MainPageDoctor> {
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  int _selectedIndex = 2;
  final List<Widget> _pages = [
    const Chats(),
    const Appointments(),
    const MyProfile(),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.white,
      child: Scaffold(
        backgroundColor: Colors.transparent,
        key: _scaffoldKey,
        body: _pages[_selectedIndex],
        bottomNavigationBar: Container(
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: const BorderRadius.only(
              topLeft: Radius.circular(20),
              topRight: Radius.circular(20),
            ),
          ),
          boxShadow: [
            BoxShadow(
              blurRadius: 20,
              color: Colors.black.withOpacity(.2),
            ),
          ],
        ),
      ),
    );
  }
}
```

```

child: SafeArea(
  child: Padding(
    padding:
      const EdgeInsets.symmetric(horizontal: 10.0, vertical: 8),
    child: GNav(
      curve: Curves.easeOutExpo,
      rippleColor: Colors.grey.shade300,
      hoverColor: Colors.grey.shade100,
      haptic: true,
      tabBorderRadius: 20,
      gap: 5,
      activeColor: Colors.white,
      padding:
        const EdgeInsets.symmetric(horizontal: 20, vertical: 12),
      duration: const Duration(milliseconds: 200),
      tabBackgroundColor: Colors.blue.withOpacity(0.7),
      textStyle: GoogleFonts.lato(
        color: Colors.white,
      ),
      iconSize: 30,
      tabs: const [
        GButton(
          icon: Icons.chat_outlined /* Typicons.group_outline */,
          text: 'Chats',
        ),
        GButton(
          icon: Typicons.calendar,
          text: 'All Appointments',
        ),
        GButton(
          icon: Typicons.user,
          text: 'Profile',
        ),
      ],
      selectedIndex: _selectedIndex,
      onTabChange: _onItemTapped,
    ),
  ),
),
);
}

```


lib\screens\patient\appointments.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/firestore_data/appointment_list.dart';

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

  @override
  State<Appointments> createState() => _AppointmentsState();
}

class _AppointmentsState extends State<Appointments> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
        title: Container(
          alignment: Alignment.center,
          padding: const EdgeInsets.symmetric(vertical: 10),
          child: Text(
            'My Appointments',
            style: GoogleFonts.lato(
              color: Colors.black,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
        iconTheme: const IconThemeData(color: Colors.black),
      ),
      body: Container(
        padding: const EdgeInsets.only(right: 10, left: 10, top: 10),
        child: const AppointmentList(),
      ),
    );
  }
}
```

lib\screens\patient\booking_screen.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/patient/appointments.dart';
import 'package:intl/intl.dart';

class BookingScreen extends StatefulWidget {
  final String doctor;
  final String doctorUid;

  const BookingScreen({Key? key, required this.doctor, required this.doctorUid})
    : super(key: key);
  @override
  State<BookingScreen> createState() => _BookingScreenState();
}

class _BookingScreenState extends State<BookingScreen> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController _nameController = TextEditingController();
  final TextEditingController _phoneController = TextEditingController();
  final TextEditingController _descriptionController = TextEditingController();
  final TextEditingController _doctorController = TextEditingController();
  final TextEditingController _dateController = TextEditingController();
  final TextEditingController _timeController = TextEditingController();

  FocusNode f1 = FocusNode();
  FocusNode f2 = FocusNode();
  FocusNode f3 = FocusNode();
  FocusNode f4 = FocusNode();
  FocusNode f5 = FocusNode();

  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  DateTime selectedDate = DateTime.now();
  TimeOfDay currentTime = TimeOfDay.now();
  String timeText = 'Select Time';
  late String dateUTC;
  late String dateTime;

  final FirebaseAuth _auth = FirebaseAuth.instance;
  late User user;

  Future<void> _getUser() async {
    user = _auth.currentUser!;
  }

  Future<void> selectDate(BuildContext context) async {
    final DateTime? date = await showDatePicker(
      context: context,
      initialDate: DateTime.now(),
      firstDate: DateTime.now(),
```

```

lastDate: DateTime.now().add(const Duration(days: 365)),
);

if (date != null) {
  setState(() {
    selectedDate = date;
    String formattedDate = DateFormat('dd-MM-yyyy').format(selectedDate);
    _dateController.text = formattedDate;
    dateUTC = DateFormat('yyyy-MM-dd').format(selectedDate);
  });
}

// function for selecting appointment time
Future<void> selectTime(BuildContext context) async {
  TimeOfDay? selectedTime = await showTimePicker(
    context: context,
    initialTime: currentTime,
  );

  MaterialLocalizations localizations = MaterialLocalizations.of(context);
  String formattedTime = localizations.formatTimeOfDay(selectedTime!,
    alwaysUse24HourFormat: false);

  setState(() {
    timeText = formattedTime;
    _timeController.text = timeText;
  });
  dateTime = selectedTime.toString().substring(10, 15);
}

// for showing appointment booked
showAlertDialog(BuildContext context) {
  // OK button
  Widget okButton = TextButton(
    child: Text(
      "OK",
      style: GoogleFonts.lato(fontWeight: FontWeight.bold),
    ),
    onPressed: () {
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(
          // view all pending appointments
          builder: (context) => const Appointments(),
        ),
      );
    },
  );

  // set up the AlertDialog
  AlertDialog alert = AlertDialog(
    title: Text(
      "Done!",

```

```

style: GoogleFonts.lato(
fontWeight: FontWeight.bold,
),
),
content: Text(
"Appointment is registered.",
style: GoogleFonts.lato(),
),
actions: [
okButton,
],
);

// show the dialog
showDialog(
context: context,
builder: (BuildContext context) {
return alert;
},
);
}

@override
void initState() {
super.initState();
getUser();
selectTime(context);
_doctorController.text = widget.doctor;
}

@override
Widget build(BuildContext context) {
return Scaffold(
backgroundColor: Colors.white,
key: _scaffoldKey,
appBar: AppBar(
backgroundColor: Colors.white,
elevation: 0,
title: Text(
'Appointment booking',
style: GoogleFonts.lato(
color: Colors.black,
fontSize: 20,
fontWeight: FontWeight.bold,
),
),
iconTheme: const IconThemeData(
color: Colors.black,
),
),
body: SafeArea(
child: NotificationListener<OverscrollIndicatorNotification>(
onNotification: (OverscrollIndicatorNotification overscroll) {
overscroll.dismissIndicator();

```

```

return true;
},
child: ListView(
  shrinkWrap: true,
  children: [
    const Image(
      image: AssetImage('assets/appointment.jpg'),
      height: 250,
    ),
    const SizedBox(
      height: 10,
    ),

    // form
    Form(
      key: _formKey,
      child: Container(
        margin: const EdgeInsets.symmetric(horizontal: 20),
        padding: const EdgeInsets.only(top: 0),
        child: Column(
          children: [
            // enter patient details
            Container(
              alignment: Alignment.centerLeft,
              padding: const EdgeInsets.only(left: 16),
              child: Text(
                'Enter Patient Details',
                style: GoogleFonts.lato(
                  fontSize: 18,
                  fontWeight: FontWeight.bold,
                  color: Colors.black54,
                ),
              ),
            ),
            const SizedBox(
              height: 30,
            ),

            // patient Name
            TextFormField(
              controller: _nameController,
              focusNode: f1,
              validator: (value) {
                if (value!.isEmpty) {
                  return 'Please Enter Patient Name';
                }
                return null;
              },
              style: GoogleFonts.lato(
                fontSize: 18, fontWeight: FontWeight.bold),
              decoration: InputDecoration(
                contentPadding: const EdgeInsets.only(
                  left: 20, top: 10, bottom: 10),
                border: const OutlineInputBorder(

```

```

borderRadius:
BorderRadius.all(Radius.circular(90.0)),
borderSide: BorderSide.none,
),
filled: true,
fillColor: Colors.grey[350],
hintText: 'Patient Name*',
hintStyle: GoogleFonts.lato(
color: Colors.black26,
fontSize: 18,
fontWeight: FontWeight.w800,
),
),
onFieldSubmitted: (String value) {
f1.unfocus();
FocusScope.of(context).requestFocus(f2);
},
textInputAction: TextInputAction.next,
),
const SizedBox(
height: 20,
),

// patient phone number
TextFormField(
keyboardType: TextInputType.phone,
focusNode: f2,
controller: _phoneController,
style: GoogleFonts.lato(
fontSize: 18, fontWeight: FontWeight.bold),
decoration: InputDecoration(
contentPadding: const EdgeInsets.only(
left: 20, top: 10, bottom: 10),
border: const OutlineInputBorder(
borderRadius:
BorderRadius.all(Radius.circular(90.0)),
borderSide: BorderSide.none,
),
filled: true,
fillColor: Colors.grey[350],
hintText: 'Mobile*',
hintStyle: GoogleFonts.lato(
color: Colors.black26,
fontSize: 18,
fontWeight: FontWeight.w800,
),
),
validator: (value) {
if (value!.isEmpty) {
return 'Please Enter Phone number';
} else if (value.length < 10) {
return 'Please Enter correct Phone number';
}
}
return null;

```

```

},
onFieldSubmitted: (String value) {
f2.unfocus();
FocusScope.of(context).requestFocus(f3);
},
textInputAction: TextInputAction.next,
),
const SizedBox(
height: 20,
),

// disease description
TextFormField(
focusNode: f3,
controller: _descriptionController,
keyboardType: TextInputType.multiline,
maxLines: null,
style: GoogleFonts.lato(
fontSize: 18, fontWeight: FontWeight.bold),
decoration: InputDecoration(
contentPadding: const EdgeInsets.only(
left: 20, top: 10, bottom: 10),
border: const OutlineInputBorder(
borderRadius:
BorderRadius.all(Radius.circular(90.0)),
borderSide: BorderSide.none,
),
filled: true,
fillColor: Colors.grey[350],
hintText: 'Description',
hintStyle: GoogleFonts.lato(
color: Colors.black26,
fontSize: 18,
fontWeight: FontWeight.w800,
),
),
onFieldSubmitted: (String value) {
f3.unfocus();
FocusScope.of(context).requestFocus(f4);
},
textInputAction: TextInputAction.next,
),
const SizedBox(
height: 20,
),

// Doctor name
TextFormField(
readOnly: true,
controller: _doctorController,
validator: (value) {
if (value!.isEmpty) return 'Please enter Doctor name';
return null;
},

```

```

style: GoogleFonts.lato(
  fontSize: 18, fontWeight: FontWeight.bold),
decoration: InputDecoration(
  contentPadding: const EdgeInsets.only(
    left: 20, top: 10, bottom: 10),
  border: const OutlineInputBorder(
    borderRadius:
      BorderRadius.all(Radius.circular(90.0)),
    borderSide: BorderSide.none,
  ),
  filled: true,
  fillColor: Colors.grey[350],
  hintText: 'Doctor Name*',
  hintStyle: GoogleFonts.lato(
    color: Colors.black26,
    fontSize: 18,
    fontWeight: FontWeight.w800,
  ),
),
const SizedBox(
  height: 20,
),

// appointment date
Container(
  alignment: Alignment.center,
  height: 60,
  width: MediaQuery.of(context).size.width,
  child: Stack(
    alignment: Alignment.centerRight,
    children: [
      TextFormField(
        onTap: () {
          selectDate(context);
        },
        focusNode: f4,
        decoration: InputDecoration(
          contentPadding: const EdgeInsets.only(
            left: 20,
            top: 10,
            bottom: 10,
          ),
          border: const OutlineInputBorder(
            borderRadius:
              BorderRadius.all(Radius.circular(90.0)),
            borderSide: BorderSide.none,
          ),
          filled: true,
          fillColor: Colors.grey[350],
          hintText: 'Select Date*',
          hintStyle: GoogleFonts.lato(
            color: Colors.black26,
            fontSize: 18,

```



```

fontWeight: FontWeight.w800,
),
),
controller: _dateController,
validator: (value) {
  if (value!.isEmpty) {
    return 'Please Enter the Date';
  }
  return null;
},
onFieldSubmitted: (String value) {
  f4.unfocus();
  FocusScope.of(context).requestFocus(f5);
},
textInputAction: TextInputAction.next,
style: GoogleFonts.lato(
  fontSize: 18, fontWeight: FontWeight.bold),
),
Padding(
  padding: const EdgeInsets.only(right: 5.0),
  child: ClipOval(
    child: Material(
      color: Colors.indigo, // button color
      child: InkWell(
        // inkwell color
        child: const SizedBox(
          width: 40,
          height: 40,
          child: Icon(
            Icons.date_range_outlined,
            color: Colors.white,
          ),
        ),
      ),
      onTap: () {
        selectDate(context);
      },
    ),
  ),
),
),
),
),
),
),
const SizedBox(
  height: 20,
),

// appointment time
Container(
  alignment: Alignment.center,
  height: 60,
  width: MediaQuery.of(context).size.width,
  child: Stack(
    alignment: Alignment.centerRight,

```

```
children: [
  TextFormField(
    onTap: () {
      selectTime(context);
    },
    focusNode: f5,
    decoration: InputDecoration(
      contentPadding: const EdgeInsets.only(
        left: 20,
        top: 10,
        bottom: 10,
      ),
      border: const OutlineInputBorder(
        borderRadius:
          BorderRadius.all(Radius.circular(90.0)),
        borderSide: BorderSide.none,
      ),
      filled: true,
      fillColor: Colors.grey[350],
      hintText: 'Select Time*',
      hintStyle: GoogleFonts.lato(
        color: Colors.black26,
        fontSize: 18,
        fontWeight: FontWeight.w800,
      ),
    ),
    controller: _timeController,
    validator: (value) {
      if (value!.isEmpty) {
        return 'Please Enter the Time';
      }
      return null;
    },
    onFieldSubmitted: (String value) {
      f5.unfocus();
    },
    textInputAction: TextInputAction.next,
    style: GoogleFonts.lato(
      fontSize: 18, fontWeight: FontWeight.bold),
  ),
  Padding(
    padding: const EdgeInsets.only(right: 5.0),
    child: ClipOval(
      child: Material(
        color: Colors.indigo, // button color
        child: InkWell(
          // inkwell color
          child: const SizedBox(
            width: 40,
            height: 40,
            child: Icon(
              Icons.timer_outlined,
              color: Colors.white,
            ),
          ),
        ),
      ),
    ),
  ),
),
```

```

),
onTap: () {
selectTime(context);
},
),
),
),
)
],
),
),
const SizedBox(
height: 40,
),

// book appointment button
SizedBox(
height: 50,
width: MediaQuery.of(context).size.width,
child: ElevatedButton(
style: ElevatedButton.styleFrom(
foregroundColor: Colors.black, backgroundColor: Colors.indigo, elevation: 2,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(32.0),
),
),
onPressed: () {
if (_formKey.currentState!.validate()) {
print(_nameController.text);
print(_dateController.text);
print(widget.doctor);
showAlertDialog(context);
_createAppointment();
}
},
child: Text(
"Book Appointment",
style: GoogleFonts.lato(
color: Colors.white,
fontSize: 18,
fontWeight: FontWeight.bold,
),
),
),
const SizedBox(
height: 40,
),
],
),
),
],
),

```

```

),
),
);
}

Future<void> _createAppointment() async {
// print(dateUTC + ' ' + date_Time + ':00');
String appointId = '${user.uid}${widget.doctorUid}$dateUTC $dateTime';
print('${widget.doctorUid}.');
print('${user.uid}.');
print('${appointId}.');

var details = {
'patientName': _nameController.text,
'phone': _phoneController.text,
'description': _descriptionController.text,
'doctorName': _doctorController.text,
'date': DateTime.parse('$dateUTC $dateTime:00'),
'patientId': user.uid,
'doctorId': widget.doctorUid,
//help in cancelling appointment
'appointmentID': appointId,
};

FirebaseFirestore.instance
.collection('appointments')
.doc(user.uid)
.collection('pending')
.doc(appointId)
.set(details, SetOptions(merge: true));

FirebaseFirestore.instance
.collection('appointments')
.doc(user.uid)
.collection('all')
.doc(appointId)
.set(details, SetOptions(merge: true));

// add to doctor data
FirebaseFirestore.instance
.collection('appointments')
.doc(widget.doctorUid)
.collection('pending')
.doc(appointId)
.set(details, SetOptions(merge: true));

FirebaseFirestore.instance
.collection('appointments')
.doc(widget.doctorUid)
.collection('all')
.doc(appointId)
.set(details, SetOptions(merge: true));
}
}

```

lib\screens\patient\doctor_list.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/firestore_data/search_list.dart';

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

  @override
  State<DoctorsList> createState() => _DoctorsListState();
}

class _DoctorsListState extends State<DoctorsList> {
  final TextEditingController _textController = TextEditingController();
  late String search;
  var _length = 0;

  @override
  void initState() {
    super.initState();
    search = _textController.text;
    _length = search.length;
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,
        title: const Text('Find Doctors'),
        actions: <Widget>[
          SafeArea(
            child: Container(
              padding: const EdgeInsets.symmetric(vertical: 7, horizontal: 15),
              width: MediaQuery.of(context).size.width,
              child: TextFormField(
                textCapitalization: TextCapitalization.words,
                controller: _textController,
                decoration: InputDecoration(
                  contentPadding:
                    const EdgeInsets.only(left: 20, top: 10, bottom: 10),
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(10),
                    borderSide: BorderSide.none,
                  ),
                  filled: true,
                  fillColor: Colors.grey[300],
                ),
              ),
            ),
          ],
        ),
    );
  }
}
```

```

hintText: 'Search Doctor',
hintStyle: GoogleFonts.lato(
color: Colors.black26,
fontSize: 18,
fontWeight: FontWeight.w800,
),
prefixIcon: const Icon(
Icons.search,
size: 20,
),
prefixStyle: TextStyle(
color: Colors.grey[300],
),
suffixIcon: _textController.text.isNotEmpty
? IconButton(
onPressed: () {
setState(() {
_textController.clear();
_length = search.length;
});
},
child: const Icon(
Icons.close_rounded,
size: 20,
),
: const SizedBox(),
),
// onFieldSubmitted: (String _searchKey) {
//   setState(
//     () {
//       print('>>>' + _searchKey);
//       search = _searchKey;
//     },
//   );
// },
onChanged: (String searchKey) {
setState(
() {
print('>>>$searchKey');
search = searchKey;
_length = search.length;
},
);
},
style: GoogleFonts.lato(
fontSize: 18,
fontWeight: FontWeight.bold,
),
textInputAction: TextInputAction.search,
autofocus: false,
),
),
)

```

```
],
),
body: Container(
padding: const EdgeInsets.all(10),
child: _length == 0
? Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
TextButton(
onPressed: () {
setState(() {
_length = 1;
});
},
child: Text(
'Show All',
style: GoogleFonts.lato(
fontSize: 18,
fontWeight: FontWeight.bold,
),
),
),
const Image(image: AssetImage('assets/search-bg.png')),
],
),
)
: SearchList(
searchKey: search,
),
),
);
}
```

lib\screens\patient\doctor_profile.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/screens/chat/chat_room.dart';
import 'package:doc_fusion/screens/patient/booking_screen.dart';
import 'package:url_launcher/url_launcher.dart';

class DoctorProfile extends StatefulWidget {
  String? doctor = "P";

  DoctorProfile({Key? key, this.doctor}) : super(key: key);
  @override
  State<DoctorProfile> createState() => _DoctorProfileState();
}

class _DoctorProfileState extends State<DoctorProfile> {
  // for making phone call
  _launchCaller(String phoneNumber) async {
    String url = "tel:$phoneNumber";
    launch(url);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      body: SafeArea(
        child: StreamBuilder(
          stream: FirebaseFirestore.instance
            .collection('doctor')
            .orderBy('name')
            .startAt([widget.doctor]).endAt(
              ['${widget.doctor!}\uf8ff']).snapshots(),
          builder:
            (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
              if (!snapshot.hasData) {
                return const Center(
                  child: CircularProgressIndicator(),
                );
              }
              return NotificationListener<OverscrollIndicatorNotification>(
                onNotification: (OverscrollIndicatorNotification overscroll) {
                  overscroll.disallowIndicator();
                  return true;
                },
              ),
              child: ListView.builder(
                itemCount: snapshot.data!.size,
                itemBuilder: (context, index) {
                  DocumentSnapshot document = snapshot.data!.docs[index];
                  return Container(
                    margin: const EdgeInsets.only(top: 5),
                    child: Column(
```



```

children: <Widget>[
  Container(
    alignment: Alignment.centerLeft,
    height: 50,
    width: MediaQuery.of(context).size.width,
    padding: const EdgeInsets.only(left: 5),
    child: IconButton(
      icon: const Icon(
        Icons.chevron_left_sharp,
        color: Colors.indigo,
        size: 30,
      ),
      onPressed: () {
        Navigator.pop(context);
      },
    ),
  ),

  // doctor profile pic
  CircleAvatar(
    backgroundImage:
      NetworkImage(document['profilePhoto'] ?? ''),
    backgroundColor: Colors.lightBlue[100],
    radius: 80,
  ),
  const SizedBox(
    height: 20,
  ),

  // doctor name
  Text(
    document['name'] ?? '-',
    style: GoogleFonts.lato(
      fontWeight: FontWeight.bold,
      fontSize: 24,
    ),
  ),
  const SizedBox(
    height: 10,
  ),

  // doctor specialization
  Text(
    document['specialization'],
    style: GoogleFonts.lato(
      fontWeight: FontWeight.bold,
      fontSize: 18,
      color: Colors.black54),
  ),
  const SizedBox(
    height: 16,
  ),

  // rating

```

```

Rating(
  rating:
    double.parse(document['rating'].toString()),
  const SizedBox(
    height: 14,
  ),

  // description
  Container(
    padding: const EdgeInsets.only(left: 22, right: 22),
    alignment: Alignment.center,
    child: Text(
      document['specification'] ?? '-',
      textAlign: TextAlign.center,
      style: GoogleFonts.lato(
        fontSize: 14,
        color: Colors.black54,
      ),
    ),
    const SizedBox(
      height: 20,
    ),

    // address
    Container(
      width: MediaQuery.of(context).size.width,
      margin: const EdgeInsets.symmetric(horizontal: 10),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.start,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const SizedBox(
            width: 15,
          ),
          const Icon(Icons.place_outlined),
          const SizedBox(
            width: 20,
          ),
          SizedBox(
            width: MediaQuery.of(context).size.width / 1.4,
            child: Text(
              document['address'] ?? '-',
              style: GoogleFonts.lato(
                fontSize: 16,
              ),
            ),
          const SizedBox(
            width: 10,
          ),
        ],
      ),
    ),
  ),

```

```

// phone number
Container(
  height: MediaQuery.of(context).size.height / 12,
  margin: const EdgeInsets.symmetric(horizontal: 10),
  child: Row(
    children: [
      const SizedBox(
        width: 15,
      ),
      const Icon(Icons.phone_in_talk),
      const SizedBox(
        width: 11,
      ),
      TextButton(
        onPressed: () =>
          _launchCaller("${document['phone']}"),
        child: Text(
          document['phone'] ?? '-',
          style: GoogleFonts.lato(
            fontSize: 16, color: Colors.blue),
        ),
      ),
      const SizedBox(
        width: 10,
      ),
    ],
  ),
  const SizedBox(
    height: 0,
  ),
)

```

```

// working hour
Container(
  margin: const EdgeInsets.symmetric(horizontal: 10),
  child: Row(
    children: [
      const SizedBox(
        width: 15,
      ),
      const Icon(Icons.access_time_rounded),
      const SizedBox(width: 20),
      Text(
        'Working Hours',
        style: GoogleFonts.lato(
          fontSize: 16,
        ),
      ),
      const SizedBox(
        height: 20,
      ),
    ],
  ),
)

```

```

),

// timing
Container(
margin: const EdgeInsets.symmetric(horizontal: 10),
padding: const EdgeInsets.only(left: 60),
child: Row(
children: [
Text(
'Today: ',
style: GoogleFonts.lato(
fontSize: 16,
fontWeight: FontWeight.bold,
),
),
const SizedBox(
width: 10,
),
Text(
document['openHour'] +
" - " +
document['closeHour'],
style: GoogleFonts.lato(
fontSize: 17,
),
),
],
),
const SizedBox(
height: 50,
),

// book appointment button
Container(
padding: const EdgeInsets.symmetric(horizontal: 30),
height: 50,
width: MediaQuery.of(context).size.width,
child: ElevatedButton(
style: ElevatedButton.styleFrom(
foregroundColor: Colors.black, backgroundColor: Colors.indigo.withOpacity(0.9),
elevation: 2,
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(32.0),
),
),
onPressed: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => BookingScreen(
doctorUid: document['id'],
doctor: document['name'],
),
),

```

```

),
);
},
child: Text(
  'Book an Appointment',
  style: GoogleFonts.lato(
    color: Colors.white,
    fontSize: 16,
    fontWeight: FontWeight.bold,
  ),
),
),
),
),

const SizedBox(
  height: 20,
  child: Text(
    'or',
    textAlign: TextAlign.center,
  ),
),

// direct call
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Container(
      padding:
        const EdgeInsets.symmetric(horizontal: 30),
      height: 50,
      // width: MediaQuery.of(context).size.width,
      child: ElevatedButton(
        style: ElevatedButton.styleFrom(
          foregroundColor: Colors.black,    backgroundColor: Colors.indigo.withOpacity(0.9),
          elevation: 2,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(32.0),
          ),
        ),
        onPressed: () {
          _launchCaller(document['phone']);
        },
        child: const Icon(Icons.call),
      ),
    ),
    Container(
      padding:
        const EdgeInsets.symmetric(horizontal: 30),
      height: 50,
      // width: MediaQuery.of(context).size.width,
      child: ElevatedButton(
        style: ElevatedButton.styleFrom(
          foregroundColor: Colors.black,    backgroundColor: Colors.indigo.withOpacity(0.9),
          elevation: 2,

```

```

shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(32.0),
),
),
onPressed: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => ChatRoom(
user2Id: document['id'] ?? ' ',
user2Name: document['name'] ?? ' ',
profileUrl:
document['profilePhoto'] ?? ' ',
),
));
},
child: const Icon(Icons
.message_outlined)), /* Text(
'Message',
style: GoogleFonts.lato(
color: Colors.white,
fontSize: 16,
fontWeight: FontWeight.bold,
), */
),
],
),
],
),
);
},
),
);
},
),
);
}
}

```

```

class Rating extends StatelessWidget {
const Rating({
Key? key,
required this.rating,
}) : super(key: key);

```

```

final double rating;

```

```

@override
Widget build(BuildContext context) {
return Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
for (var i = 0; i < rating.toInt(); i++)

```

```
const Icon(  
  Icons.star_rounded,  
  color: Colors.indigoAccent,  
  size: 30,  
)  
if (rating - rating.toInt() > 0)  
const Icon(  
  Icons.star_half_rounded,  
  color: Colors.indigoAccent,  
  size: 30,  
)  
if (5 - rating.ceil() > 0)  
for (var i = 0; i < 5 - rating.ceil(); i++)  
const Icon(  
  Icons.star_rounded,  
  color: Colors.black12,  
  size: 30,  
)  
],  
);  
}  
}
```

lib\screens\patient\home_page.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:doc_fusion/carousel_slider.dart';
import 'package:doc_fusion/firestore_data/notification_list.dart';
import 'package:doc_fusion/firestore_data/search_list.dart';
import 'package:doc_fusion/firestore_data/top_rated_list.dart';
import 'package:doc_fusion/model/card_model.dart';
import 'package:doc_fusion/screens/explore_list.dart';
import 'package:intl/intl.dart';

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

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

class _HomePageState extends State<HomePage> {
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  TextEditingController _doctorName = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;
  User? user;

  Future<void> _getUser() async {
    user = _auth.currentUser;
  }

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

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

  @override
  Widget build(BuildContext context) {
    String message = "Good";
    DateTime now = DateTime.now();
    String currentHour = DateFormat('kk').format(now);
    int hour = int.parse(currentHour);

    setState(
      () {
        if (hour >= 5 && hour < 12) {
          message = 'Good Morning';
        }
      }
    );
  }
}
```



```

} else if (hour >= 12 && hour <= 17) {
message = 'Good Afternoon';
} else {
message = 'Good Evening';
}
},
);
return Scaffold(
  backgroundColor: Colors.white,
  key: _scaffoldKey,
  appBar: AppBar(
    automaticallyImplyLeading: false,
    actions: <Widget>[Container()],
    backgroundColor: Colors.white,
    elevation: 0,
    title: Container(
      padding: const EdgeInsets.only(top: 5),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          Container(
            //width: MediaQuery.of(context).size.width/1.3,
            alignment: Alignment.center,
            child: Text(
              message,
              style: GoogleFonts.lato(
                color: Colors.black54,
                fontSize: 20,
                fontWeight: FontWeight.w400,
              ),
            ),
          ),
          const SizedBox(
            width: 55,
          ),
          IconButton(
            splashRadius: 20,
            icon: const Icon(Icons.notifications_active),
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => const NotificationList());
            },
          ),
        ],
      ),
    ),
    iconTheme: const IconThemeData(
      color: Colors.black,
    ),
  ),
  body: SafeArea(
    child: NotificationListener<OverscrollIndicatorNotification>(

```

```

onNotification: (OverscrollIndicatorNotification overscroll) {
  overscroll.disallowIndicator();
  return true;
},
child: ListView(
  physics: const ClampingScrollPhysics(),
  shrinkWrap: true,
  children: <Widget>[
    Column(
      children: [
        const SizedBox(
          height: 30,
        ),
        Container(
          alignment: Alignment.centerLeft,
          padding: const EdgeInsets.only(left: 20, bottom: 10),
          child: Text(
            "Hello ${user?.displayName}",
            style: GoogleFonts.lato(
              fontSize: 18,
              fontWeight: FontWeight.w500,
            ),
          ),
        ),
        Container(
          alignment: Alignment.centerLeft,
          padding: const EdgeInsets.only(left: 20, bottom: 25),
          child: Text(
            "Let's Find Your\nDoctor",
            style: GoogleFonts.lato(
              fontSize: 35,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
        Container(
          padding: const EdgeInsets.fromLTRB(20, 0, 20, 25),
          child: TextFormField(
            textCapitalization: TextCapitalization.words,
            textInputAction: TextInputAction.search,
            controller: _doctorName,
            decoration: InputDecoration(
              contentPadding: const EdgeInsets.only(
                left: 20, top: 10, bottom: 10),
              border: const OutlineInputBorder(
                borderRadius: BorderRadius.all(Radius.circular(15.0)),
                borderSide: BorderSide.none,
              ),
              filled: true,
              fillColor: Colors.grey[200],
              hintText: 'Search doctor',
              hintStyle: GoogleFonts.lato(
                color: Colors.black26,
                fontSize: 18,

```

```
fontWeight: FontWeight.w800,
),
suffixIcon: Container(
  decoration: BoxDecoration(
    color: Colors.blue.shade900.withOpacity(0.9),
    borderRadius: BorderRadius.circular(20),
  ),
  child: IconButton(
    iconSize: 20,
    splashRadius: 20,
    color: Colors.white,
    icon: const Icon(Icons.search),
    onPressed: () {},
  ),
),
),
style: GoogleFonts.lato(
  fontSize: 18,
  fontWeight: FontWeight.w800,
),
onFieldSubmitted: (String value) {
  setState(
    () {
      value.isEmpty
        ? Container()
        : Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => SearchList(
                searchKey: value,
              ),
            ),
          );
    },
  );
},
),
),
Container(
  padding: const EdgeInsets.only(left: 23, bottom: 10),
  alignment: Alignment.centerLeft,
  child: Text(
    "We care for you",
    textAlign: TextAlign.center,
    style: GoogleFonts.lato(
      color: Colors.blue[800],
      fontWeight: FontWeight.bold,
      fontSize: 18),
  ),
),
SizedBox(
  width: MediaQuery.of(context).size.width,
  child: const CarouselSlider(),
),
),
```

```

Container(
padding: const EdgeInsets.only(left: 20),
alignment: Alignment.centerLeft,
child: Text(
"Specialists",
textAlign: TextAlign.center,
style: GoogleFonts.lato(
color: Colors.blue[800],
fontWeight: FontWeight.bold,
fontSize: 18),
),
),
Container(
height: 150,
padding: const EdgeInsets.only(top: 14),
child: ListView.builder(
physics: const ClampingScrollPhysics(),
scrollDirection: Axis.horizontal,
padding: const EdgeInsets.symmetric(horizontal: 20.0),
itemCount: cards.length,
itemBuilder: (context, index) {
//print("images path: ${cards[index].cardImage.toString()}");
return Container(
margin: const EdgeInsets.only(right: 14),
height: 150,
width: 140,
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(20),
color: Color(cards[index].cardBackground),
boxShadow: [
BoxShadow(
color: Colors.grey.shade400,
blurRadius: 4.0,
spreadRadius: 0.0,
offset: const Offset(3, 3),
),
]
// image: DecorationImage(
//   image: AssetImage(cards[index].cardImage),
//   fit: BoxFit.fill,
// ),
),
child: TextButton(
onPressed: () {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => ExploreList(
type: cards[index].doctor,
)),
);
},
style: ButtonStyle(
shape: MaterialStateProperty.all<

```

```

RoundedRectangleBorder>(
RoundedRectangleBorder(
borderRadius: BorderRadius.circular(20),
),
),
),
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
const SizedBox(
height: 16,
),
CircleAvatar(
backgroundColor: Colors.white,
radius: 29,
child: Icon(
cards[index].cardIcon,
size: 26,
color: Color(cards[index].cardBackground),
)),
const SizedBox(
height: 10,
),
Container(
alignment: Alignment.bottomCenter,
child: Text(
cards[index].doctor,
style: GoogleFonts.lato(
color: Colors.white,
fontSize: 16,
fontWeight: FontWeight.w600),
),
),
],
),
);
},
),
const SizedBox(
height: 30,
),
Container(
padding: const EdgeInsets.only(left: 20),
alignment: Alignment.centerLeft,
child: Text(
"Top Rated",
textAlign: TextAlign.center,
style: GoogleFonts.lato(
color: Colors.blue[800],
fontWeight: FontWeight.bold,
fontSize: 18),
),

```

```
),
const SizedBox(
height: 10,
),
Container(
padding: const EdgeInsets.only(left: 15, right: 15),
child: const TopRatedList(),
),
const SizedBox(
height: 20,
),
],
),
],
),
),
),
),
);
}
```

lib\screens\patient\main_page_patient.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:google_nav_bar/google_nav_bar.dart';
import 'package:doc_fusion/screens/chat/chats.dart';
import 'package:doc_fusion/screens/my_profile.dart';
import 'package:doc_fusion/screens/patient/doctor_list.dart';
import 'package:doc_fusion/screens/patient/home_page.dart';
import 'package:doc_fusion/screens/patient/appointments.dart';
import 'package:typicons_flutter/typicons_flutter.dart';

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

  @override
  State<MainPagePatient> createState() => _MainPagePatientState();
}

class _MainPagePatientState extends State<MainPagePatient> {
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey<ScaffoldState>();
  int _selectedIndex = 0;
  final List<Widget> _pages = [
    const HomePage(),
    const DoctorsList(),
    const Appointments(),
    const Chats(),
    const MyProfile(),
  ];

  final FirebaseAuth _auth = FirebaseAuth.instance;
  User? user;

  Future<void> _getUser() async {
    user = _auth.currentUser;
  }

  @override
  void initState() {
    super.initState();
    _getUser();
  }

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.white,
```

```

child: Scaffold(
  backgroundColor: Colors.transparent,
  key: _scaffoldKey,
  body: _pages[_selectedIndex],
  bottomNavigationBar: Container(
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: const BorderRadius.only(
        topLeft: Radius.circular(20),
        topRight: Radius.circular(20),
      ),
    ),
    boxShadow: [
      BoxShadow(
        blurRadius: 20,
        color: Colors.black.withOpacity(.2),
      ),
    ],
  ),
  child: SafeArea(
    child: Padding(
      padding:
        const EdgeInsets.symmetric(horizontal: 10.0, vertical: 8),
      child: GNav(
        curve: Curves.easeOutExpo,
        rippleColor: Colors.grey.shade300,
        hoverColor: Colors.grey.shade100,
        haptic: true,
        tabBorderRadius: 20,
        gap: 5,
        activeColor: Colors.white,
        padding:
          const EdgeInsets.symmetric(horizontal: 20, vertical: 12),
        duration: const Duration(milliseconds: 200),
        tabBackgroundColor: Colors.blue.withOpacity(0.7),
        textStyle: GoogleFonts.lato(
          color: Colors.white,
        ),
        tabs: const [
          GButton(
            iconSize: 28,
            icon: Icons.home,
            // text: 'Home',
          ),
          GButton(
            icon: Icons.search,
            // text: 'Search',
          ),
          GButton(
            iconSize: 28,
            icon: Typicons.calendar,
            // text: 'Appointments',
          ),
          GButton(
            iconSize: 28,

```



```
icon: Icons.chat,  
// text: 'Chat',  
,  
GButton(  
  iconSize: 28,  
  icon: Typicons.user,  
  // text: 'Profile',  
,  
,  
  selectedIndex: _selectedIndex,  
  onTabChange: _onItemTapped,  
,  
,  
,  
,  
,  
,  
);  
}  
}
```

lib\services\chat_methods.dart

```
import 'package:firebase_database/firebase_database.dart';
import 'package:doc_fusion/helperFunction/sharedpref_helper.dart';

class ChatMethods {
  Future addMessage({required String chatRoomId, required String message}) {

    // add message to real time database
    return
    FirebaseDatabase.instance.ref('chats').child(chatRoomId).child('chat').push().set({
      'message': message,
      'timestamp': DateTime.now().toUtc(),
      'sender': SharedPreferencesHelper().getUserId(),
    });
  }
}
```

pubspec.yaml

```
name: doc_fusion
description: "A new Flutter project."
# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number is used as
CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build suffix.
version: 1.0.0+1

environment:
  sdk: ^3.7.0

# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  carousel_slider:
  cloud_firestore:
  firebase_auth:
  firebase_core:
  firebase_storage:
  firebase_database:
  flutter:
  sdk: flutter
  fluttericon:
  google_fonts:
  google_nav_bar:
  icon:
  image_picker:
  intl:
  introduction_screen:
  shared_preferences:
  tabler_icons:
  typicons_flutter:
```

url_launcher:

The following adds the Cupertino Icons font to your application.
Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.8

dev_dependencies:
 flutter_launcher_icons:
 flutter_test:
 sdk: flutter

flutter_icons:
 android: true
 ios: true
 image_path: "assets/icon/appIcon.png"
 remove_alpha_ios: true

The "flutter_lints" package below contains a set of recommended lints to
encourage good coding practices. The lint set provided by the package is
activated in the `analysis_options.yaml` file located at the root of your
package. See that file for information about deactivating specific lint
rules and activating additional ones.
flutter_lints: ^5.0.0

For information on the generic Dart part of this file, see the
following page: <https://dart.dev/tools/pub/pubspec>

The following section is specific to Flutter packages.
flutter:

The following line ensures that the Material Icons font is
included with your application, so that you can use the icons in
the material Icons class.
uses-material-design: true

To add assets to your application, add an assets section, like this:
assets:
 - assets/
- images/a_dot_burr.jpeg
- images/a_dot_ham.jpeg

An image asset can refer to one or more resolution-specific "variants", see
<https://flutter.dev/to/resolution-aware-images>

For details regarding adding assets from package dependencies, see
<https://flutter.dev/to/asset-from-package>

To add custom fonts to your application, add a fonts section here,
in this "flutter" section. Each entry in this list should have a
"family" key with the font family name, and a "fonts" key with a
list giving the asset and other descriptors for the font. For
example:

```
# fonts:
#   - family: Schyler
#     fonts:
#       - asset: fonts/Schyler-Regular.ttf
#       - asset: fonts/Schyler-Italic.ttf
#         style: italic
#   - family: Trajan Pro
#     fonts:
#       - asset: fonts/TrajanPro.ttf
#       - asset: fonts/TrajanPro_Bold.ttf
#         weight: 700
#
# For details regarding fonts from package dependencies,
# see https://flutter.dev/to/font-from-package
```