Ahmed Bagha

Mobile Application

1.1 App Overview

The app is made with Flutter (Flutter is a Google-developed open-source toolkit for building cross-platform applications using a single codebase). It links up with Firebase to save patient data and handle different tasks.

The App consists of a lot of screens:

- Welcome Screen.
- Sign Up Screen.
- Sign In Screen.
- Survey Screen.
- Home Screen.
- Results screen.
- Health Blog Screen.
- Appointments Screen.
- Notifications Screen.
- Contact Us Screen.
- Reach Us Screen.
- Profile Screen.

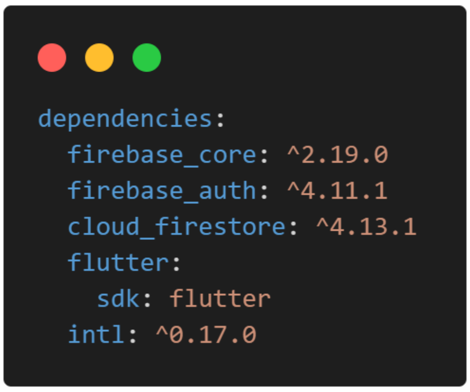
This app helps patients understand their AI-powered health results through a user-friendly interface. It also makes scheduling appointments and contacting doctors quick and easy.

1.2 Firebase Connection

(1) I have created firebase project in firebase and register my app.

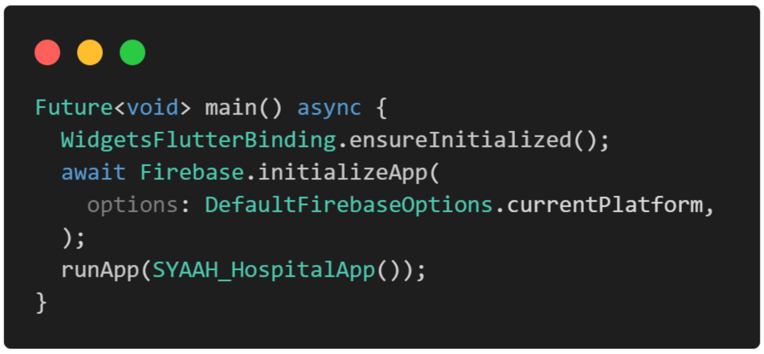
(2) Add configuration file **(google-services.json)** to my App in **(android – app)** directory.

(3) Add firebase dependencies in my project at **(pubspec.yaml)**

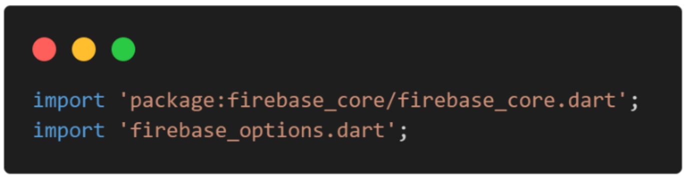


```yaml
dependencies:
  firebase_core: ^2.19.0
  firebase_auth: ^4.11.1
  cloud_firestore: ^4.13.1
  flutter:
    sdk: flutter
  intl: ^0.17.0
```

(4) Initialize Firebase in the **main** function



```dart
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(SYAAH_HospitalApp());
}
```



```dart
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
```

1.3 Application Screens

❑ Welcome Screen

Welcome to [Fusion Hospital Management System]. Sign up to get started or sign in if you're already part of our community. Need help? Tap the 'Help' button. Want to get in touch? Hit 'Contact.' Have feedback? Share it through 'Feedback.' It's that easy! Let's make your experience awesome together.

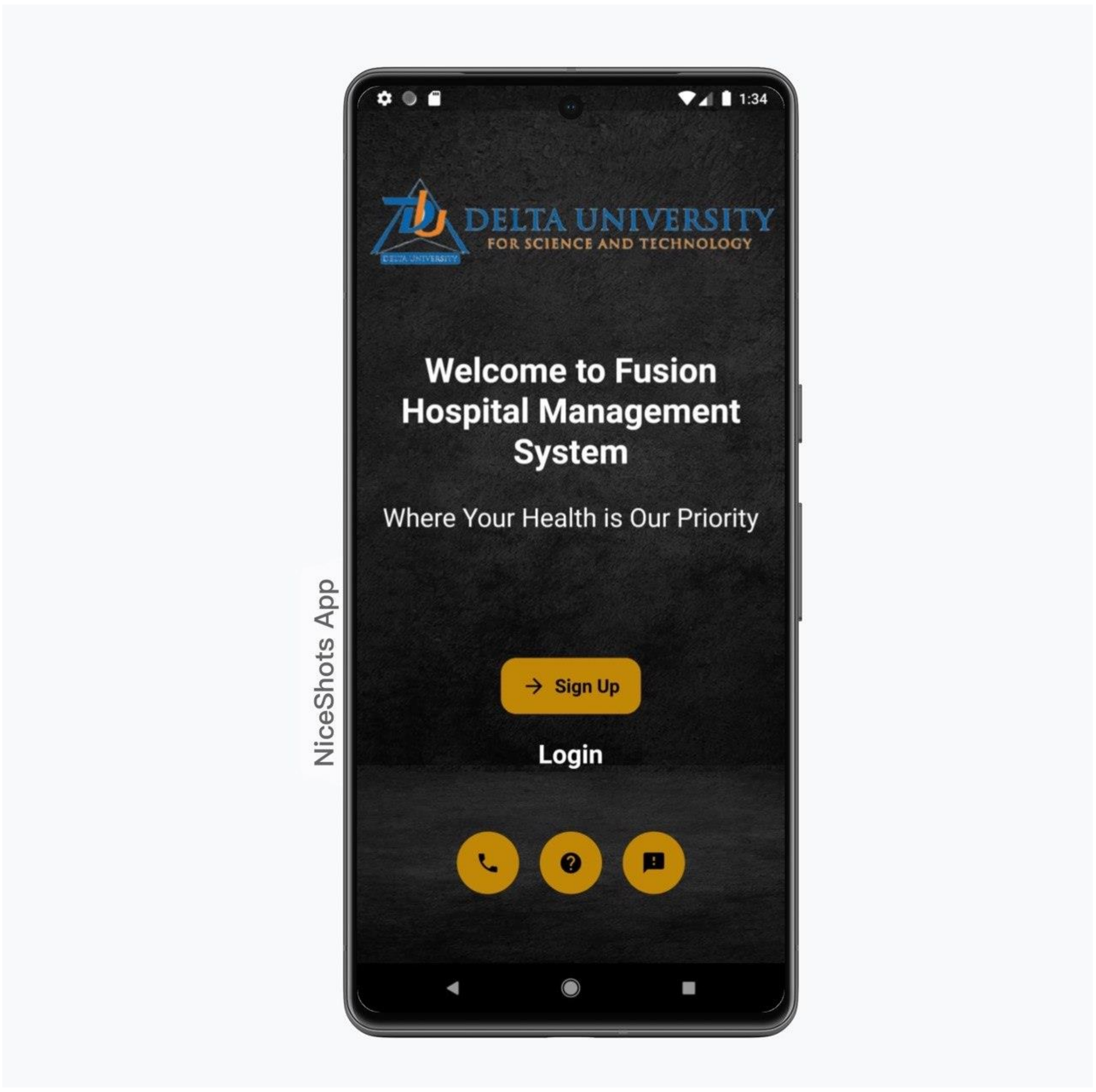

1:34
DELTA UNIVERSITY
FOR SCIENCE AND TECHNOLOGY
Welcome to Fusion Hospital Management System
Where Your Health is Our Priority
Sign Up
Login
NiceShots App

Figure 1: *Welcome Screen*

This screen is your starting point for a seamless healthcare experience. Check your results, book appointments, and reach out to your doctor hassle-free. Your journey to better health begins now.

❑ Sign In Screen

The sign-in screen facilitates a seamless experience for patients who have previously created an account, allowing them to easily access their account by entering their email and password.

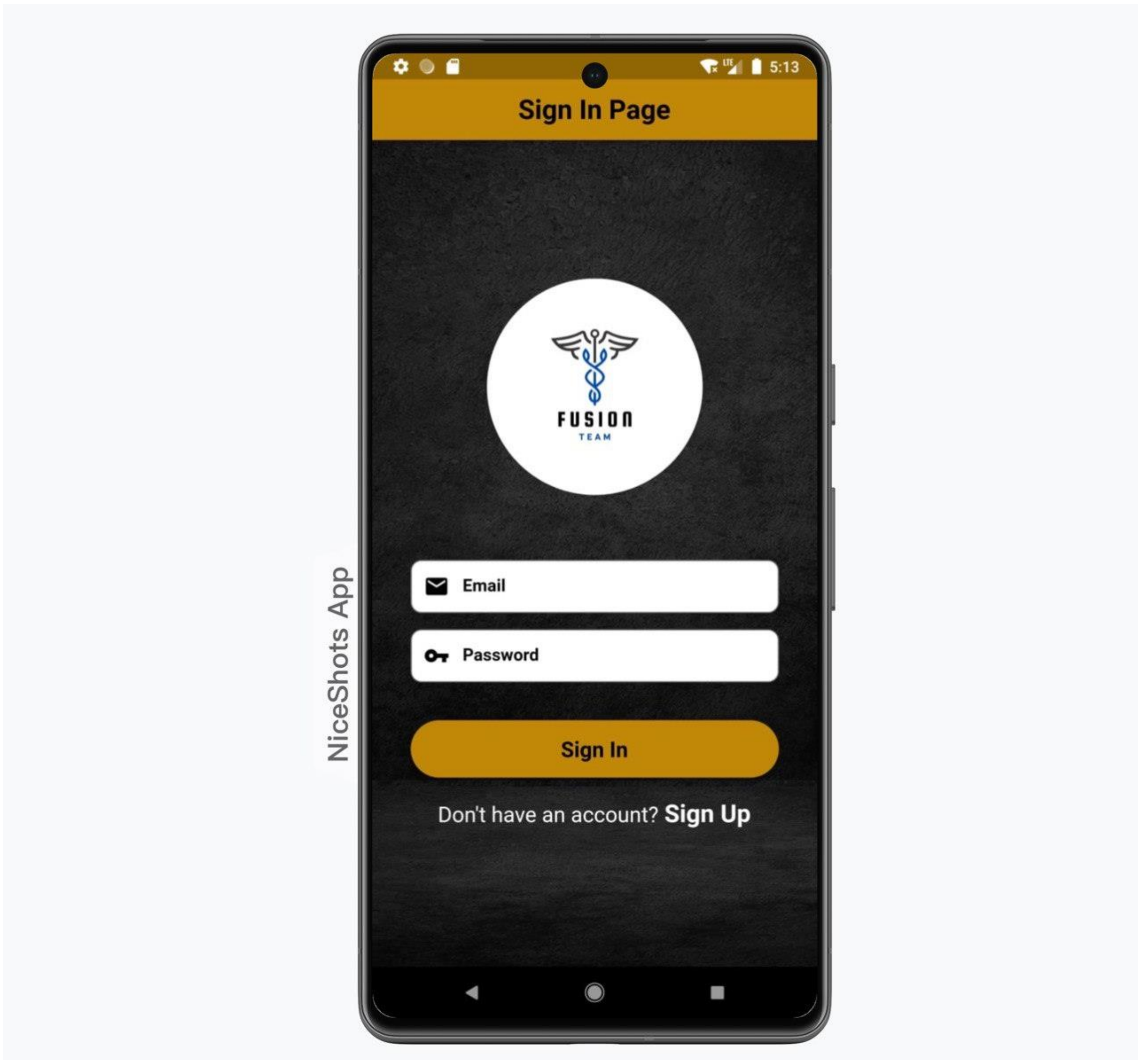
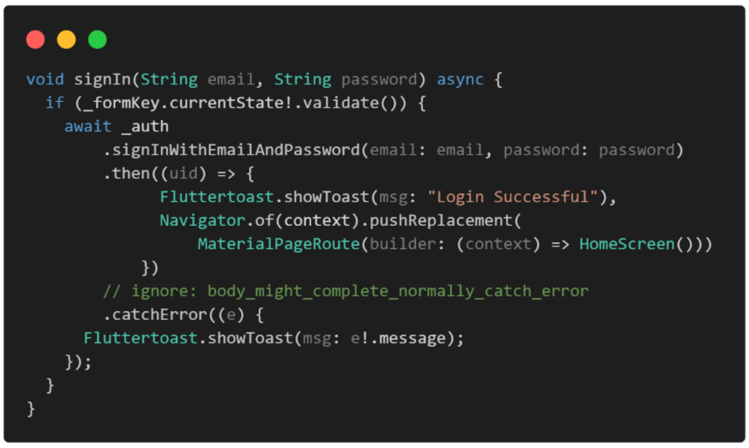


Figure 2: Sign Up Screen

I'll now demonstrate the sign-in process. It involves taking the patient's email and password, checking Firebase for the email's existence, and signing in if the email is found.



```
void signIn(String email, String password) async {
  if (_formKey.currentState!.validate()) {
    await _auth
        .signInWithEmailAndPassword(email: email, password: password)
        .then((uid) => {
              Fluttertoast.showToast(msg: "Login Successful"),
              Navigator.of(context).pushReplacement(
                  MaterialPageRoute(builder: (context) => HomeScreen()))
            })
        // ignore: body_might_complete_normally_catch_error
        .catchError((e) {
      Fluttertoast.showToast(msg: e!.message);
    });
  }
}
```

❑ Sign Up Screen

The signup screen makes it easy for patients to create an account. You just need to enter your first name, last name, phone number, birthdate, gender, address, email, and password.

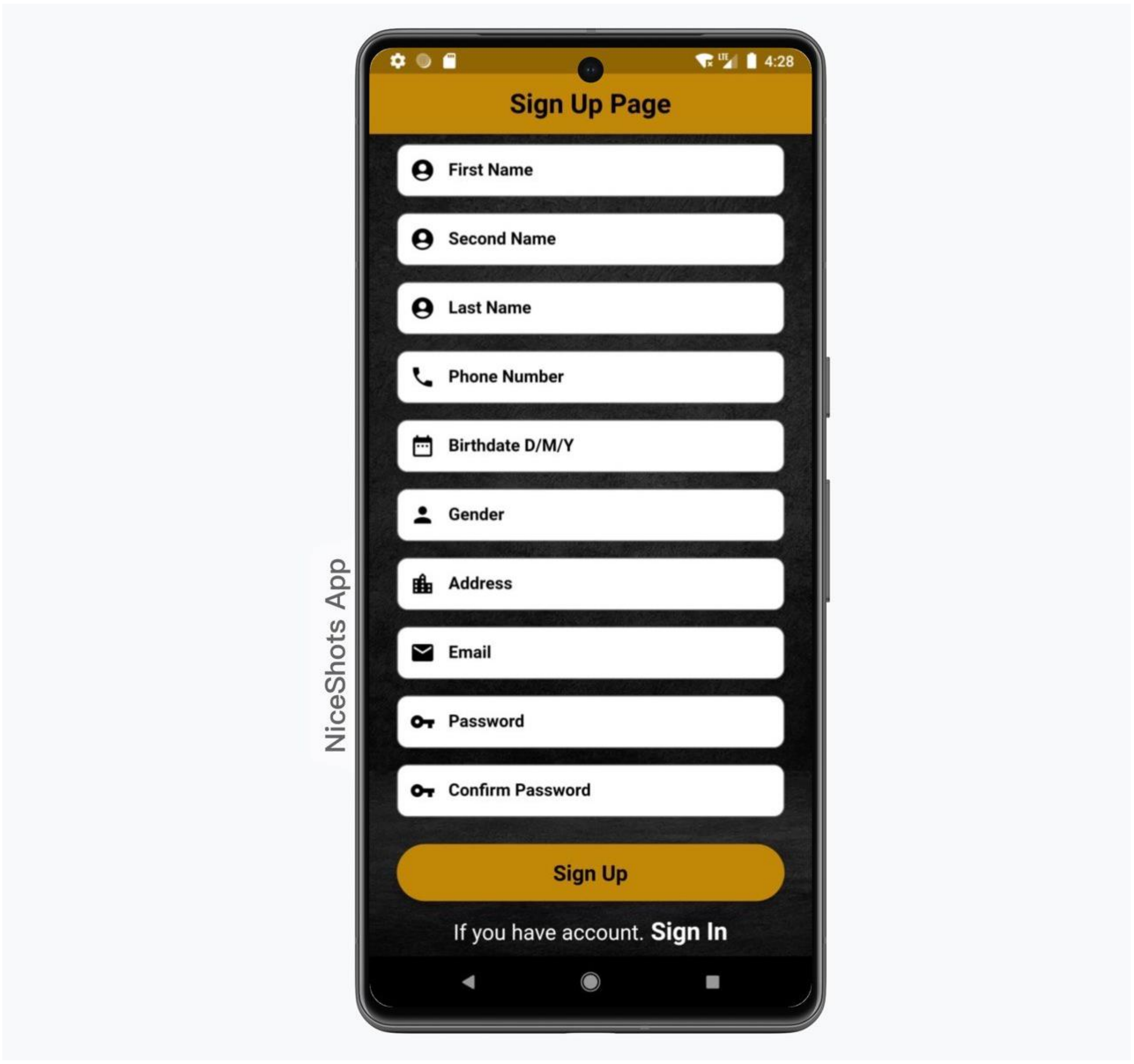
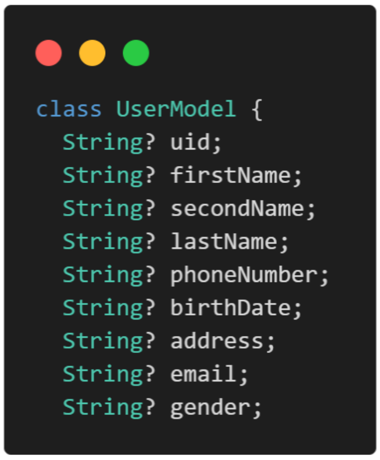


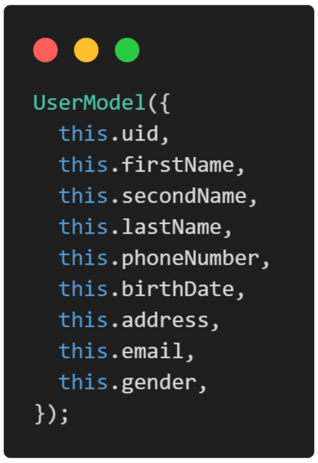
Figure 3: *Sign Up Screen*

I will now demonstrate how to securely store user input data in Firebase, ensuring that it can be conveniently retrieved for future use during the sign-in process.

Firstly, I will establish a model within the application to gather user data, ensuring a seamless process of storing it locally before transmitting it to Firebase.

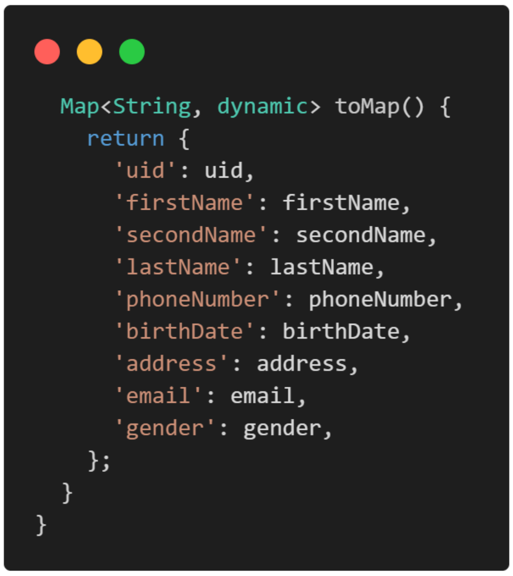


```
class UserModel {
  String? uid;
  String? firstName;
  String? secondName;
  String? lastName;
  String? phoneNumber;
  String? birthDate;
  String? address;
  String? email;
  String? gender;
```

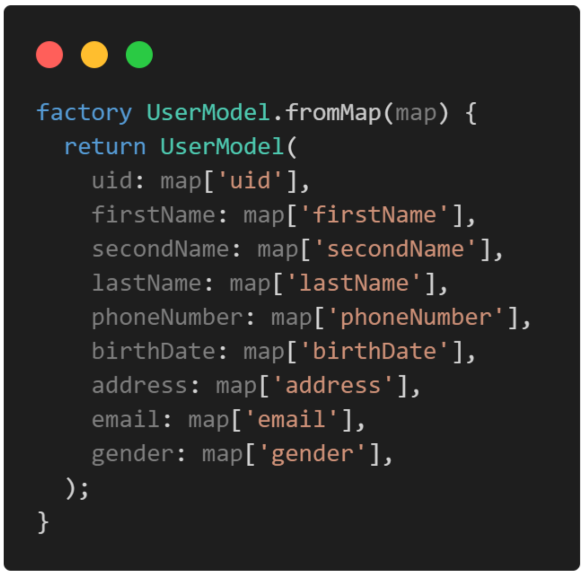


```
UserModel({
  this.uid,
  this.firstName,
  this.secondName,
  this.lastName,
  this.phoneNumber,
  this.birthDate,
  this.address,
  this.email,
  this.gender,
});
```

Secondly, I will illustrate the two models responsible for transmitting and retrieving data to and from Firebase.

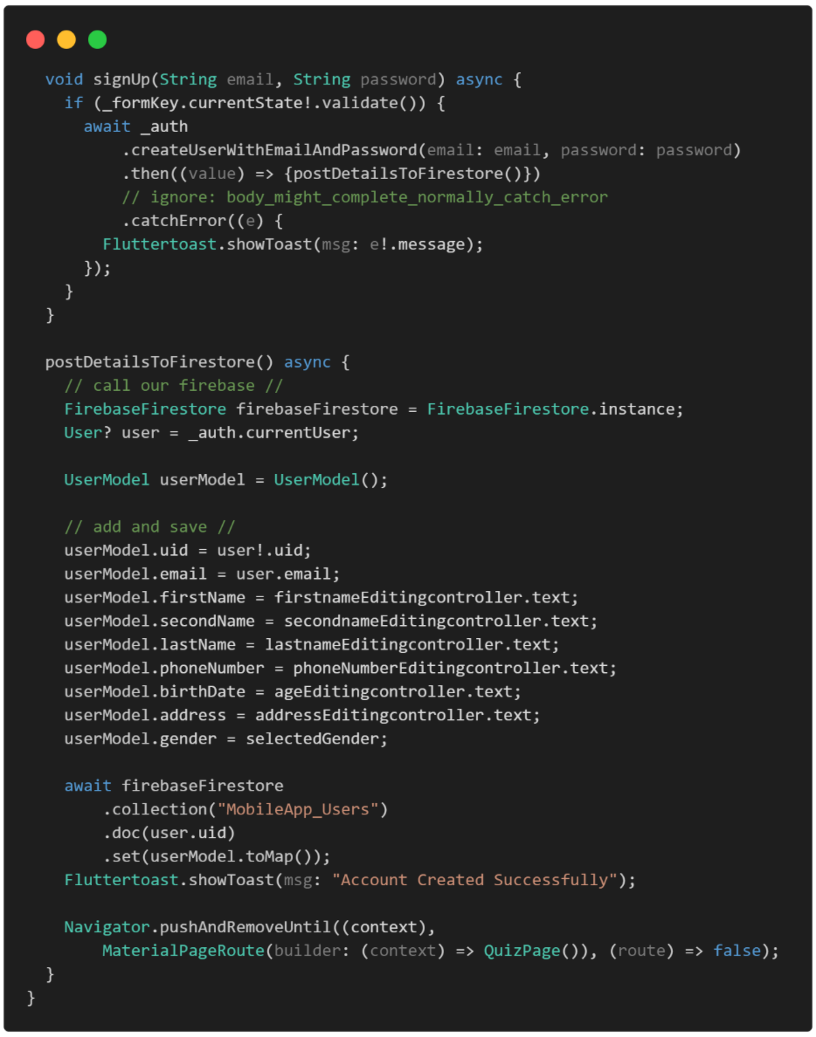


```
  Map<String, dynamic> toMap() {
    return {
      'uid': uid,
      'firstName': firstName,
      'secondName': secondName,
      'lastName': lastName,
      'phoneNumber': phoneNumber,
      'birthDate': birthDate,
      'address': address,
      'email': email,
      'gender': gender,
    };
  }
}
```



```
factory UserModel.fromMap(map) {
  return UserModel(
    uid: map['uid'],
    firstName: map['firstName'],
    secondName: map['secondName'],
    lastName: map['lastName'],
    phoneNumber: map['phoneNumber'],
    birthDate: map['birthDate'],
    address: map['address'],
    email: map['email'],
    gender: map['gender'],
  );
}
```

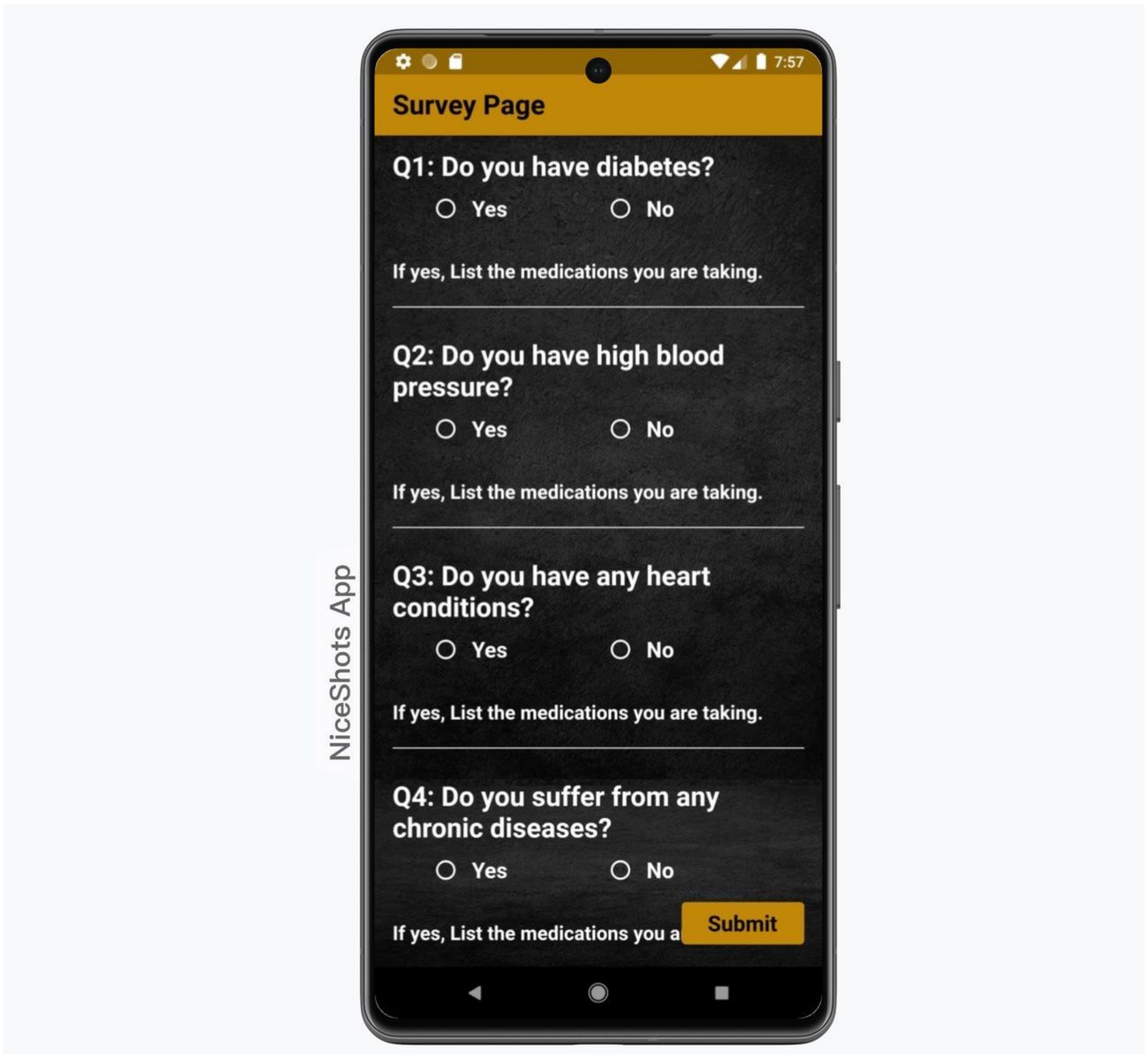
Thirdly, I will demonstrate the registration process method, utilizing the UserModel model to transmit all entered data and call firebase for saving this data in Firebase.



```
  void signUp(String email, String password) async {
    if (_formKey.currentState!.validate()) {
      await _auth
          .createUserWithEmailAndPassword(email: email, password: password)
          .then((value) => {postDetailsToFirestore()})
          // ignore: body_might_complete_normally_catch_error
          .catchError((e) {
        Fluttertoast.showToast(msg: e!.message);
      });
    }
  }

  postDetailsToFirestore() async {
    // call our firebase //
    FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;
    User? user = _auth.currentUser;

    UserModel userModel = UserModel();

    // add and save //
    userModel.uid = user!.uid;
    userModel.email = user.email;
    userModel.firstName = firstnameEditingcontroller.text;
    userModel.secondName = secondnameEditingcontroller.text;
    userModel.lastName = lastnameEditingcontroller.text;
    userModel.phoneNumber = phoneNumberEditingcontroller.text;
    userModel.birthDate = ageEditingcontroller.text;
    userModel.address = addressEditingcontroller.text;
    userModel.gender = selectedGender;

    await firebaseFirestore
        .collection("MobileApp_Users")
        .doc(user.uid)
        .set(userModel.toMap());
    Fluttertoast.showToast(msg: "Account Created Successfully");

    Navigator.pushAndRemoveUntil((context),
        MaterialPageRoute(builder: (context) => QuizPage()), (route) => false);
  }
}
```

❑ Survey Screen

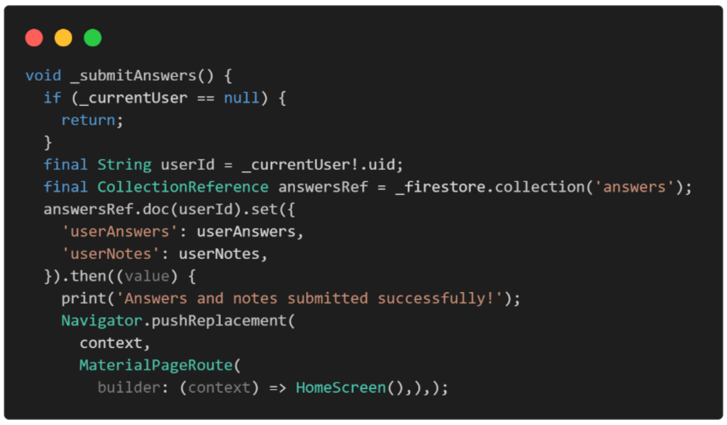
After **Sign up**, take a short health survey sharing info on diabetes, high blood pressure, heart conditions, chronic diseases, and current medications. This helps doctors create personalized treatment plans. Your answers are securely stored in Firebase after submission.



7:57
Survey Page
Q1: Do you have diabetes?
Yes
No
If yes, List the medications you are taking.
Q2: Do you have high blood pressure?
Yes
No
If yes, List the medications you are taking.
Q3: Do you have any heart conditions?
Yes
No
If yes, List the medications you are taking.
Q4: Do you suffer from any chronic diseases?
Yes
No
If yes, List the medications you a
Submit
NiceShots App

Figure 4: *Survey Screen*

This code shows how to gather survey responses and save them on Firebase for storage. It allows easy retrieval of the stored data.



```
void _submitAnswers() {
  if (_currentUser == null) {
    return;
  }
  final String userId = _currentUser!.uid;
  final CollectionReference answersRef = _firestore.collection('answers');
  answersRef.doc(userId).set({
    'userAnswers': userAnswers,
    'userNotes': userNotes,
  }).then((value) {
    print('Answers and notes submitted successfully!');
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(
        builder: (context) => HomeScreen(),),);
```

❑ Home Screen

The homepage includes six cards with simple and user-friendly features:

(1) "Results": Displays test results and predictions based on X-ray analysis.

(2) "Health Blog": Shows the patient's health status and survey responses.

(3) "Appointments": Allows patients to schedule their appointments.

(4) "Notifications": Provides updates from the hospital regarding appointments.

(5) "Contact Us": Enables patients to easily reach out to the hospital with any questions.

(6) "Reach Us": Provides the hospital's location for easy navigation.

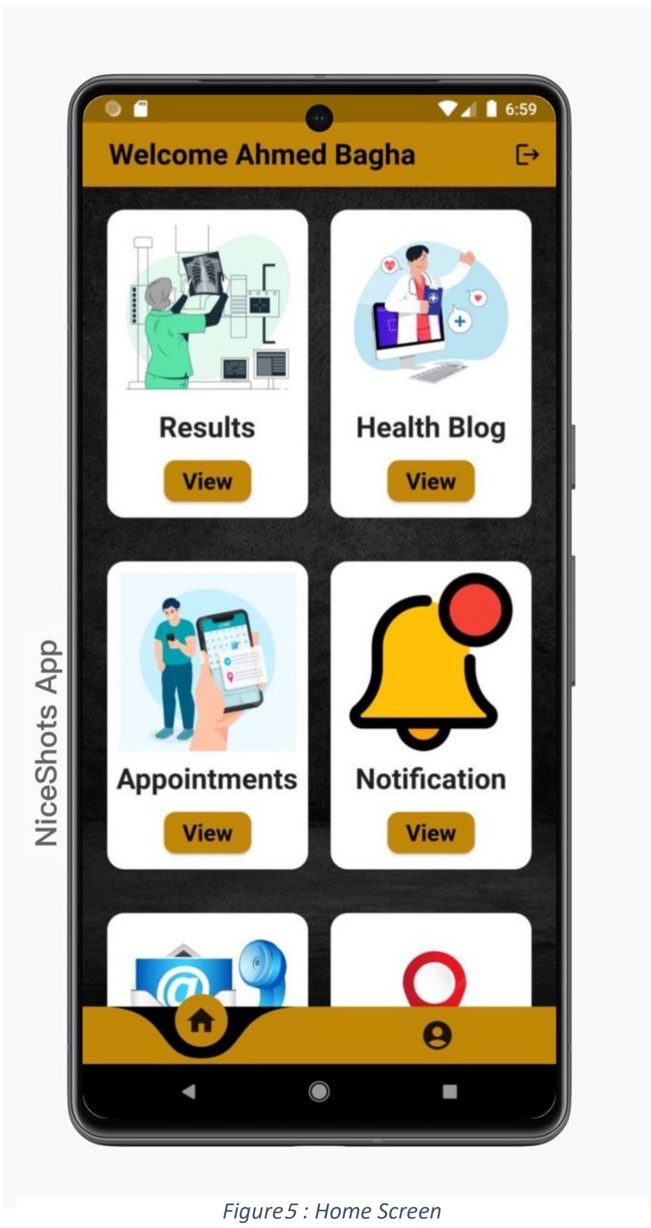


Figure5 : Home Screen

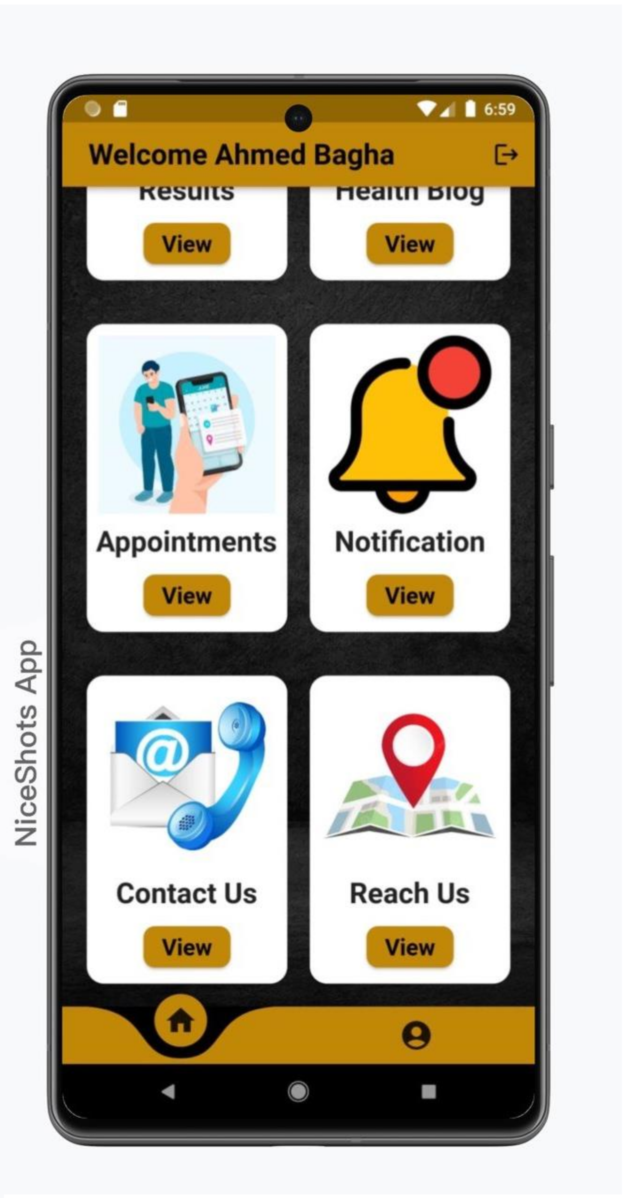
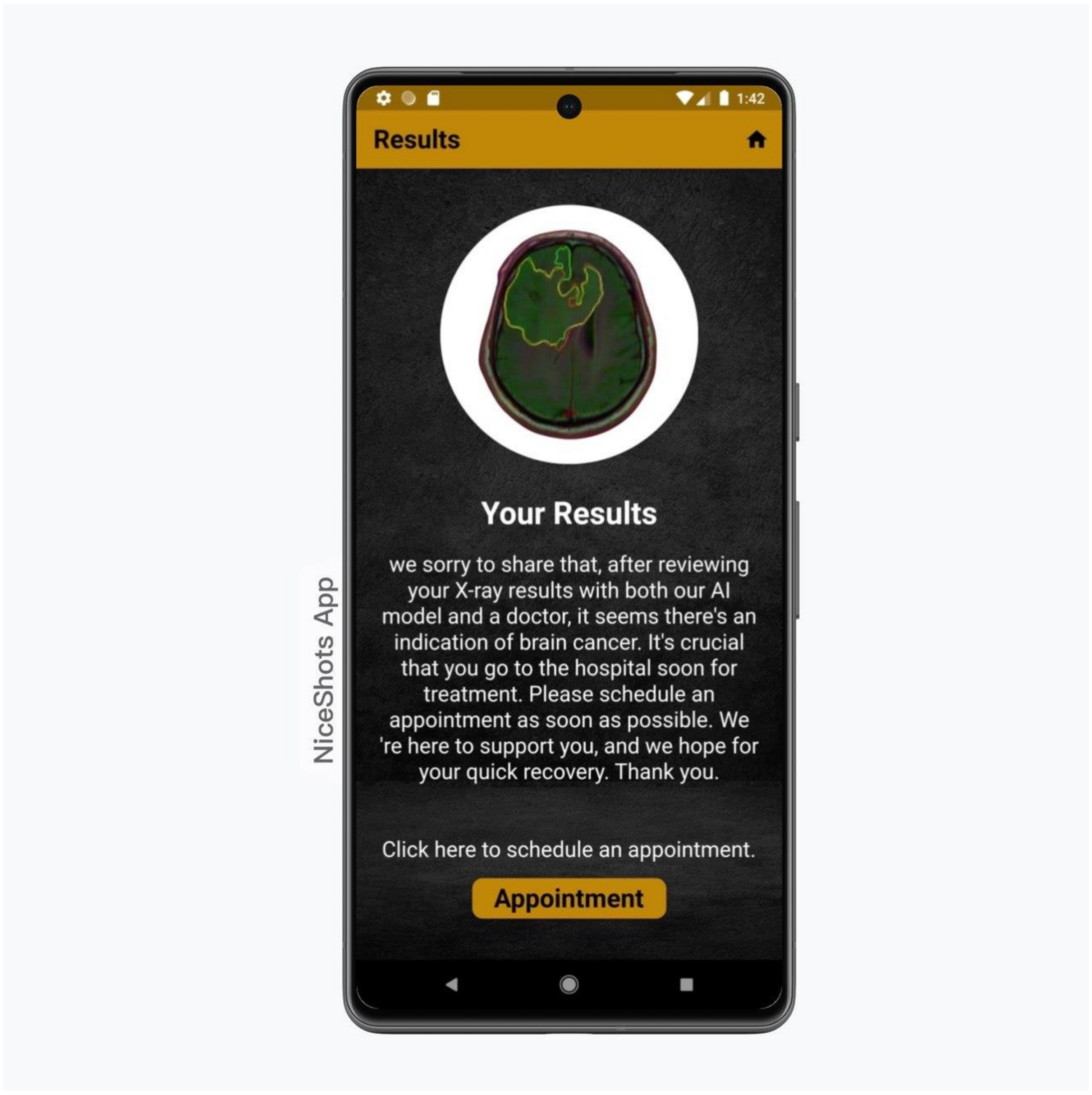


Figure5: Home Screen

❑ Results Screen

On the results screen, the patient's outcomes are displayed, indicating whether the X-ray, examined by the AI model, reveals the presence of cancer. Subsequently, the findings are transmitted to the doctor, who, in turn, sends them to the mobile application. This enables the patient to view both the X-ray and the results indicating whether cancer is present or not.



Your Results

we sorry to share that, after reviewing your X-ray results with both our AI model and a doctor, it seems there's an indication of brain cancer. It's crucial that you go to the hospital soon for treatment. Please schedule an appointment as soon as possible. We 're here to support you, and we hope for your quick recovery. Thank you.

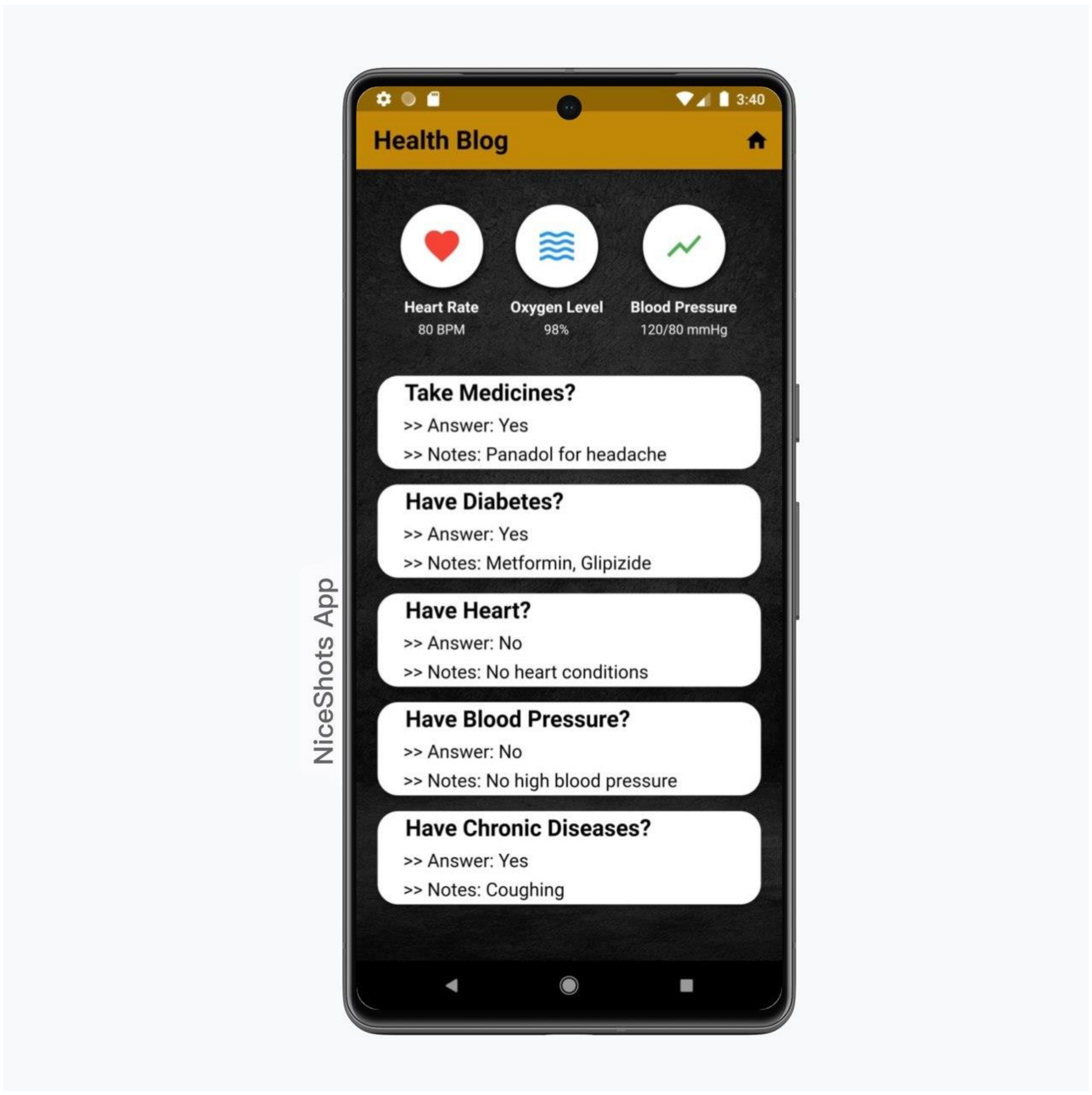
Click here to schedule an appointment.

Figure 6: Results Screen

In this image, the X-ray and the corresponding result are visible. If the patient has a response, there is a button at the end of the page to click, leading to the appointment page. This allows the patient to schedule an appointment, go to the hospital, and begin treatment.

❑ Health Blog Screen.

This screen shows everything about the patient's health, like heart rate, oxygen, and pressure. It also includes the answers to a survey the patient filled out after signing up.



Health Blog

Heart Rate
80 BPM

Oxygen Level
98%

Blood Pressure
120/80 mmHg

Take Medicines?
>> Answer: Yes
>> Notes: Panadol for headache

Have Diabetes?
>> Answer: Yes
>> Notes: Metformin, Glipizide

Have Heart?
>> Answer: No
>> Notes: No heart conditions

Have Blood Pressure?
>> Answer: No
>> Notes: No high blood pressure

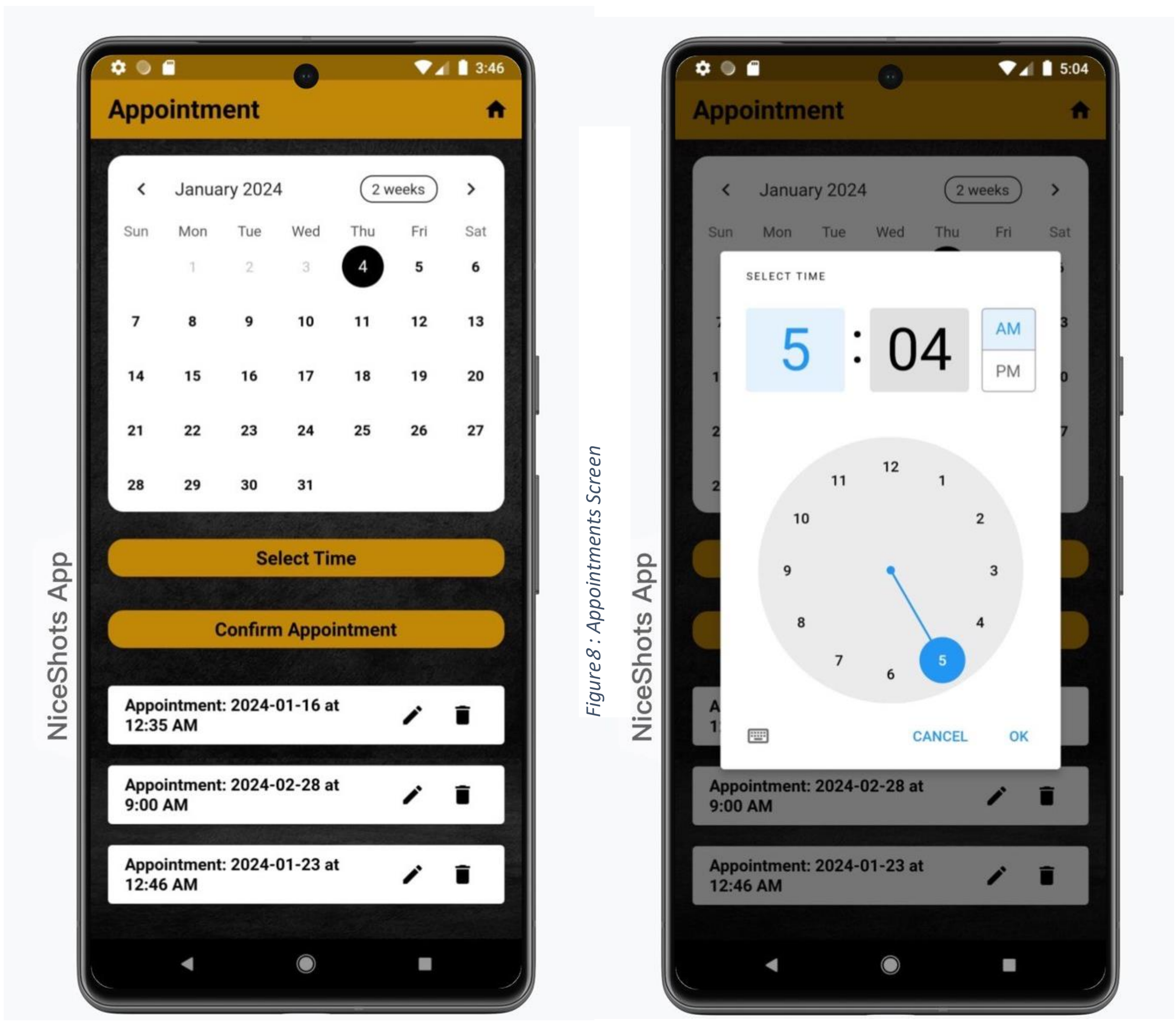
Have Chronic Diseases?
>> Answer: Yes
>> Notes: Coughing

Figure 7: Health Blog Screen

This page receives and stores all the saved data related to the patient's health, providing the patient with comprehensive information about their well-being.

❑ Appointments Screen.

This page assists the patient in scheduling appointments for the date and time of their preference, providing them with flexibility and convenience. You can also edit the appointment, and this page ensures minimal waiting time.



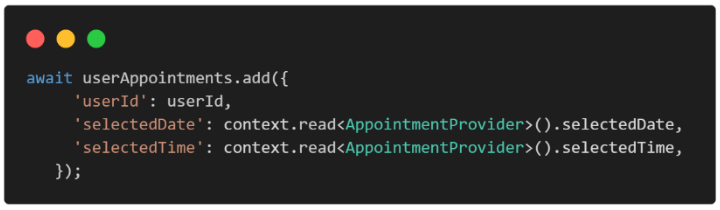
Appointment: 2024-01-16 at 12:35 AM

Appointment: 2024-02-28 at 9:00 AM

Appointment: 2024-01-23 at 12:46 AM

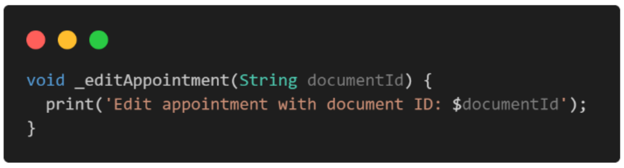
Figure 8 : Appointments Screen

Add Appointment to firebase



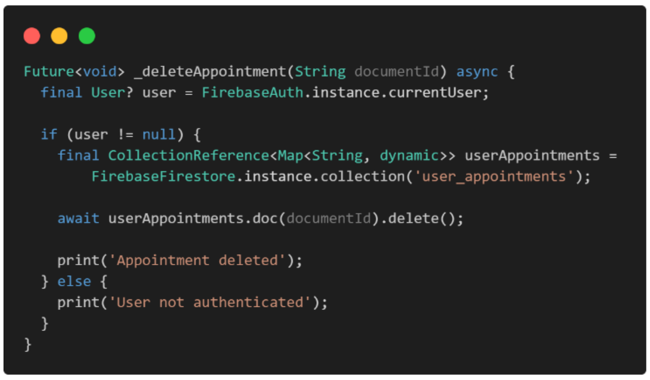
```
await userAppointments.add({
      'userId': userId,
      'selectedDate': context.read<AppointmentProvider>().selectedDate,
      'selectedTime': context.read<AppointmentProvider>().selectedTime,
    });
```

Edit Appointment



```
void _editAppointment(String documentId) {
  print('Edit appointment with document ID: $documentId');
}
```

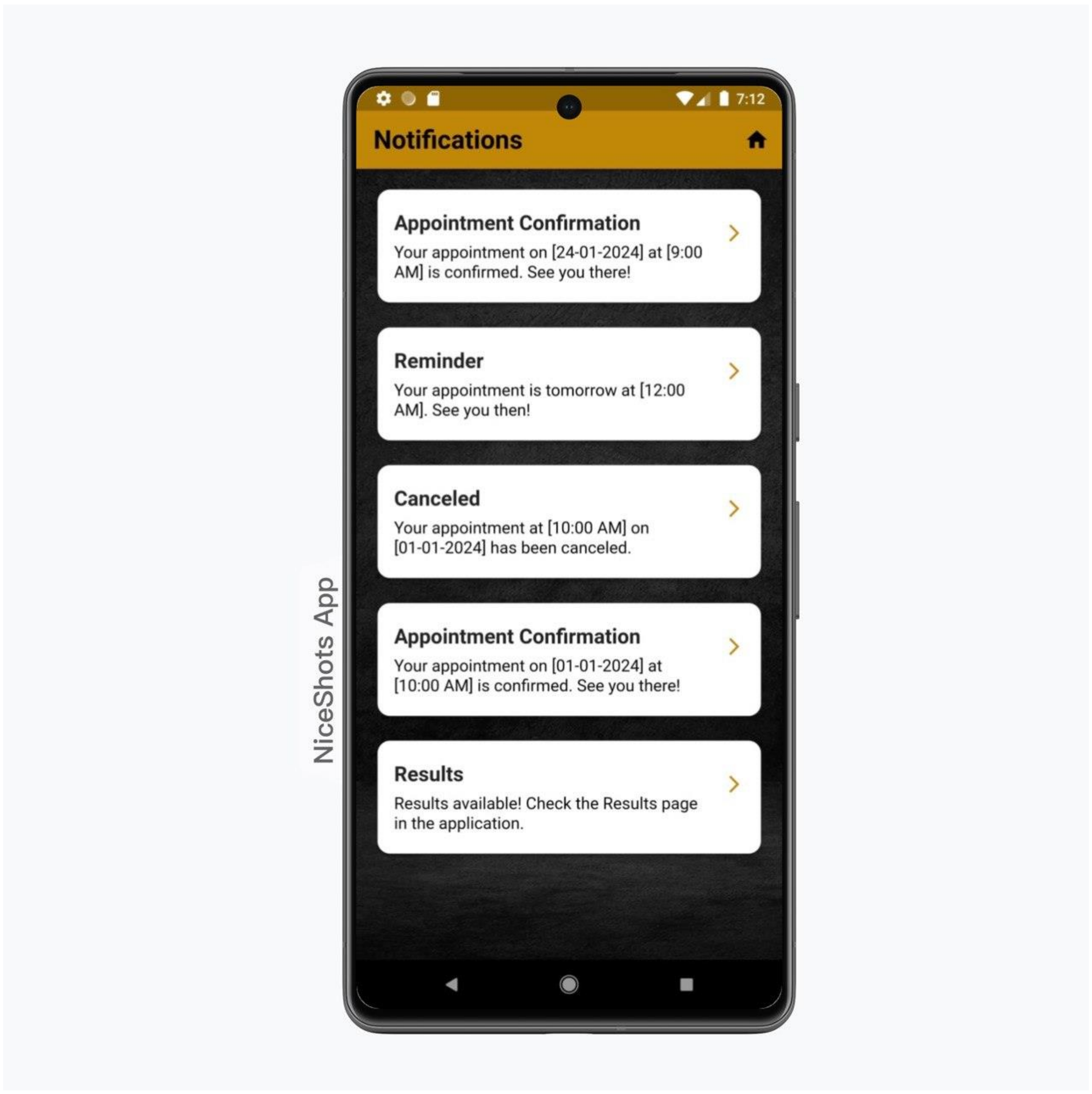
Delete Appointment



```
Future<void> _deleteAppointment(String documentId) async {
  final User? user = FirebaseAuth.instance.currentUser;

  if (user != null) {
    final CollectionReference<Map<String, dynamic>> userAppointments =
        FirebaseFirestore.instance.collection('user_appointments');

    await userAppointments.doc(documentId).delete();

    print('Appointment deleted');
  } else {
    print('User not authenticated');
  }
}
```

❑ Notifications Screen

This page displays notifications sent by the hospital to the patient, including appointment confirmations, reminders, and notifications for appointment cancellations or modifications.



Notifications

Appointment Confirmation
Your appointment on [24-01-2024] at [9:00 AM] is confirmed. See you there!

Reminder
Your appointment is tomorrow at [12:00 AM]. See you then!

Canceled
Your appointment at [10:00 AM] on [01-01-2024] has been canceled.

Appointment Confirmation
Your appointment on [01-01-2024] at [10:00 AM] is confirmed. See you there!

Results
Results available! Check the Results page in the application.

Figure 9: Notifications Screen

On the Notifications Screen, you can find important messages from the hospital. This includes appointment confirmations, friendly reminders, and notifications for any changes or cancellations. It's your go-to place for staying updated on your healthcare appointments.

❑ Contact Us Screen

On the Contact Us screen, you'll find the hospital's email and phone number. This page also lets you provide feedback, which is important for improving the hospital and our services.

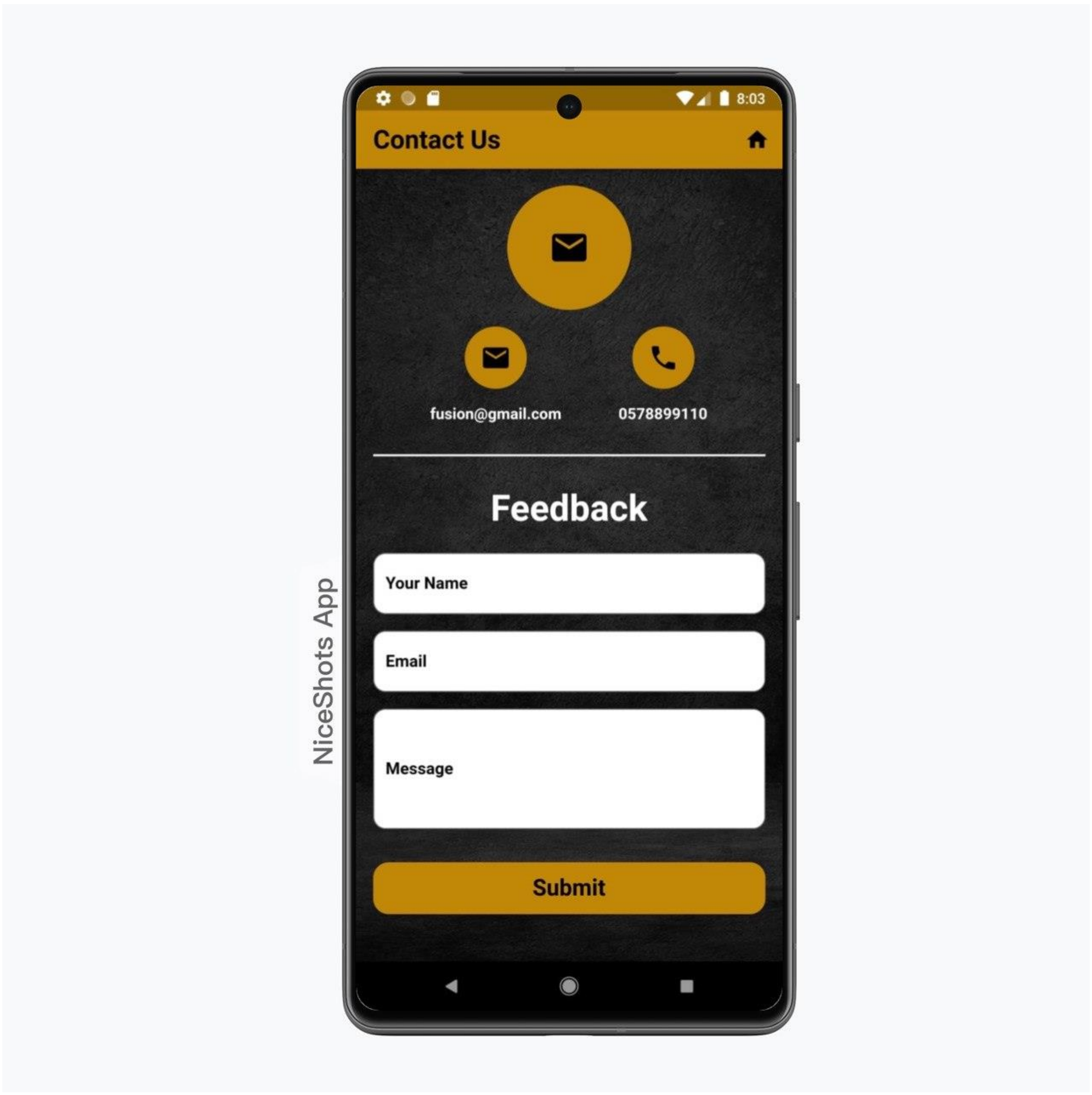


Figure 10: Contact Us Screen

The Contact Us screen is simple to use. You can find the hospital's email and phone number. Also, when you provide feedback, know that it's important for making the hospital better. We store this information securely in Firebase.

❑ Reach Us Screen

The "Find Us" screen makes it easy for patients to discover the hospital's location. Whether you're a new visitor or returning patient, our Reach Us feature ensures you can locate us with convenience.

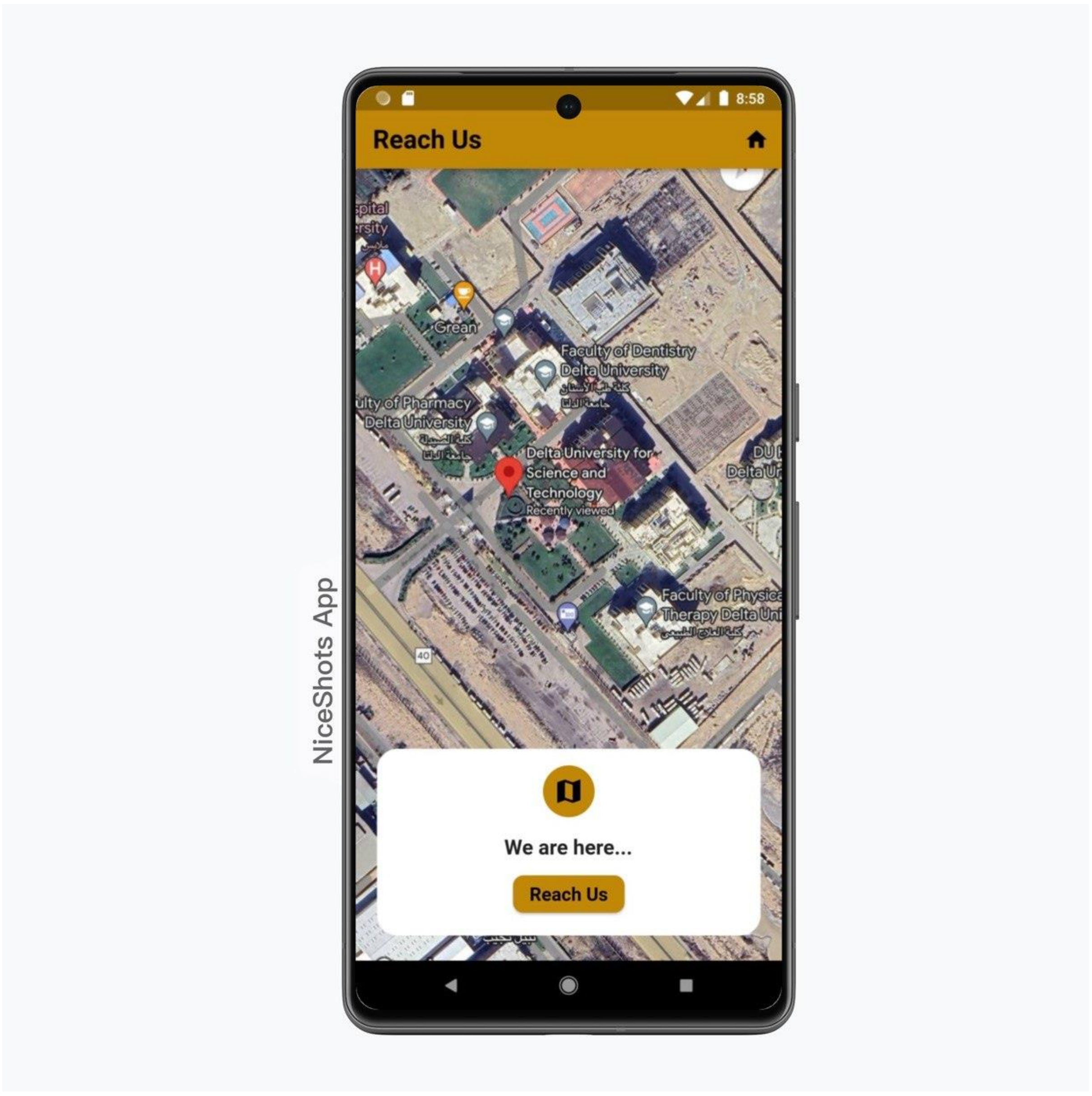
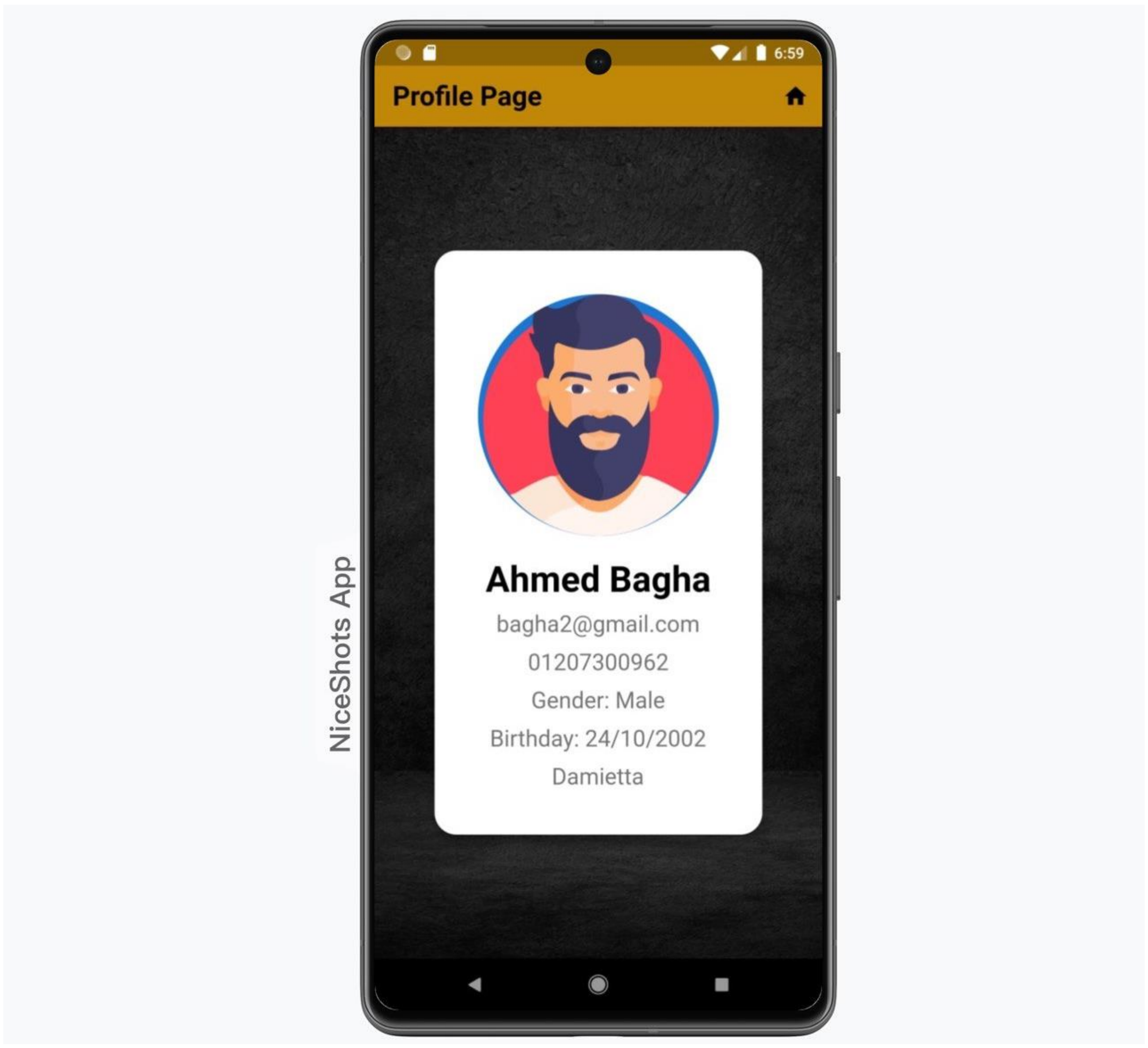

8:58
Reach Us
Grean
Faculty of Dentistry Delta University
ulty of Pharmacy Delta University
Delta University for Science and Technology
Recently viewed
Faculty of Physica Therapy Delta Uni
We are here...
Reach Us
NiceShots App

Figure 11: Reach Us Screen

Clicking the "Reach Us" button on the location screen effortlessly takes you to Google Maps, ensuring a seamless and user-friendly experience in finding the hospital's exact location. Whether you're a first-time visitor or a returning patient, this quick and efficient navigation feature simplifies your journey to the hospital.

❑ Profile Screen

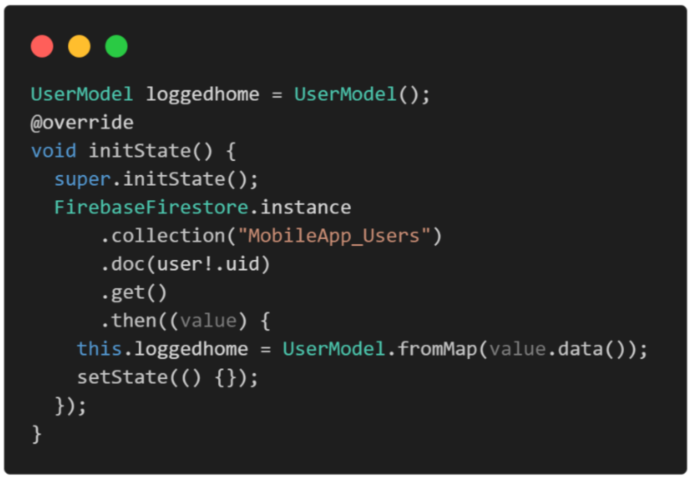
The profile screen displays all the information of the user who is signed in.



Ahmed Bagha
bagha2@gmail.com
01207300962
Gender: Male
Birthday: 24/10/2002
Damietta

Figure 12: Profile Screen

This code exemplifies how to interact with Firebase to retrieve the data of the current user and display it on a profile screen.

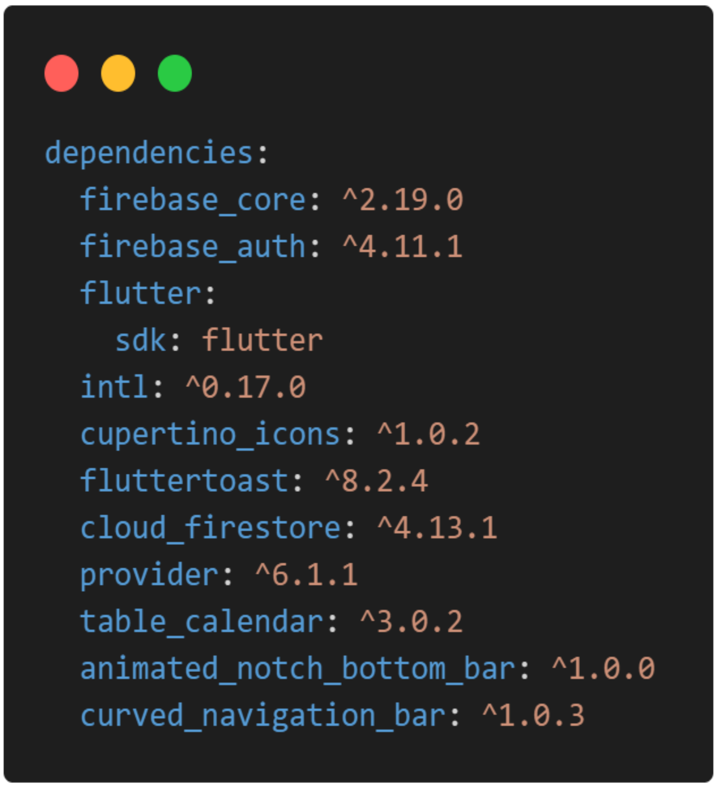


```
UserModel loggedhome = UserModel();
@override
void initState() {
  super.initState();
  FirebaseFirestore.instance
      .collection("MobileApp_Users")
      .doc(user!.uid)
      .get()
      .then((value) {
    this.loggedhome = UserModel.fromMap(value.data());
    setState(() {});
  });
}
```

❑ Importants Widgets Used In Building The App

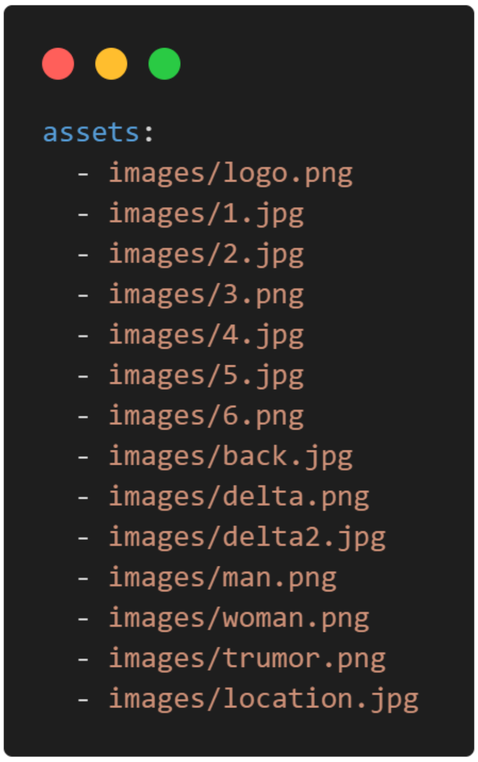
- **Scaffold:** basic app structure including app bar, body, and drawer.
- **AppBar:** top app bar for navigation and titles.
- **Container:** basic rectangular box that can contain other widgets.
- **Row:** arranges its children in a horizontal line.
- **Column:** arranges its children in a vertical line.
- **ListView:** scrollable list of widgets.
- **Stack:** overlapping widgets.
- **Card:** material design card with elevation and rounded corners.
- **TextField:** input field for text.
- **IconButton:** a button with an icon.
- **AssetsImage:** displaying images that added in pubspec.yaml.
- **BottomNavigationBar:** navigation bar at the bottom of the screen.
- **Radio:** for creating radio button that select a single option from a group of options.
- **DatePicker:** allows the user to pick a date from a calendar.
- **TimePicker:** allows the user to pick a time.
- **ListView.builder:** creates a scrollable list of widgets on demand, useful for large lists.
- **Form:** container for form fields and manages the form state.
- **SingleChildScrollView:** make page as box that can be scrolled vertically.
- **Divider:** thin horizontal line, often used to separate content.
- **Positioned:** widget that positions its child at a specified offset.
- **SizedBox:** to make space vertically or horizontally.
- **Navigator.push:** pushes a new route onto the navigator's stack.
- **MaterialPageRoute:** to add the new screen to take me to it when click.

Important Dependencies



```yaml
dependencies:
  firebase_core: ^2.19.0
  firebase_auth: ^4.11.1
  flutter:
    sdk: flutter
  intl: ^0.17.0
  cupertino_icons: ^1.0.2
  fluttertoast: ^8.2.4
  cloud_firestore: ^4.13.1
  provider: ^6.1.1
  table_calendar: ^3.0.2
  animated_notch_bottom_bar: ^1.0.0
  curved_navigation_bar: ^1.0.3
```

Adding Images



```yaml
assets:
  - images/logo.png
  - images/1.jpg
  - images/2.jpg
  - images/3.png
  - images/4.jpg
  - images/5.jpg
  - images/6.png
  - images/back.jpg
  - images/delta.png
  - images/delta2.jpg
  - images/man.png
  - images/woman.png
  - images/trumor.png
  - images/location.jpg
```