

# **Chapter 1: INTRODUCTION**

## **1.1. Motivation**

The Appointment Booking System can be motivated by a number of factors, including:

1. Efficiency: By implementing an appointment booking system, businesses can streamline the booking process and reduce the time spent manually managing appointments. This can help to increase the efficiency of the organization and free up staff time to focus on other tasks.
2. Customer satisfaction: With an appointment booking system, customers can easily schedule appointments at their own convenience, without having to wait on hold or go through a complicated booking process. This can improve customer satisfaction and lead to increased repeat business.
3. Cost savings: By automating the appointment booking process, businesses can save money on administrative costs, such as staff time and resources needed to manage appointments manually.
4. Improved record-keeping: An appointment booking system can help to improve record-keeping and provide a centralized platform for managing appointments. This can help businesses to better track appointments, manage resources, and generate reports and analytics to inform decision-making.
5. Flexibility: An appointment booking system can offer businesses the flexibility to customize appointment scheduling based on their unique needs, such as setting availability times, appointment types, and booking rules.

Overall, an appointment booking system can offer a number of benefits to businesses, including improved efficiency, customer satisfaction, cost savings, record-keeping, and flexibility.

## **1.2. Problem Definition**

The problem that an appointment booking system seeks to address is the inefficient and time-consuming process of manually managing appointments. In a traditional setting, customers may have to wait on hold, navigate complex phone menus or wait in long queues to book an appointment. This can lead to frustration for customers and create additional workload for staff members who manage the booking process manually.

Additionally, manual appointment booking can often result in errors such as double bookings, missed appointments, or incorrect appointment details. This can lead to wasted time and resources, as well as negative experiences for customers.

An appointment booking system seeks to address these issues by automating the appointment booking process, making it more convenient and efficient for both customers and staff. By providing a centralized platform for managing appointments, businesses can streamline the booking process, reduce the risk of errors, and free up staff time to focus on other tasks.

Furthermore, an appointment booking system can provide valuable analytics and data insights that can be used to inform business decisions, such as identifying peak booking times or determining which services are most popular.

Overall, the problem that an appointment booking system seeks to address is the inefficiency and errors associated with manual appointment booking, and the potential negative impact this can have on customer experience and business operations.

## **1.3. Objectives of the Project**

The objectives of a doctor appointment booking system are as follows:

1. Streamline appointment scheduling: One of the main objectives of a doctor appointment booking system is to make the appointment scheduling process more efficient and streamlined. By automating the process, patients can easily schedule appointments, and doctors can manage their schedules more effectively.
2. Improve patient experience: Another objective of the appointment booking system is to improve the overall patient experience. By providing a user-friendly platform for scheduling appointments, patients can book appointments at their convenience, without having to wait on hold or navigate complicated phone menus.
3. Reduce no-shows: The appointment booking system can also help to reduce the number of missed appointments, which can be costly for doctors and frustrating for patients. By sending reminders and confirmations, patients are less likely to forget their appointments, and doctors can better manage their schedules.
4. Increase productivity: With an appointment booking system, doctors can focus on providing quality care to their patients rather than managing appointment scheduling manually. This can increase productivity and allow doctors to see more patients.
5. Enhance record-keeping: The appointment booking system can also provide a centralized platform for managing patient appointments, which can enhance record-keeping and provide valuable insights into patient behavior.

Overall, the objectives of a doctor appointment booking system are to improve the efficiency of appointment scheduling, enhance the patient experience, reduce no-shows, increase productivity, and improve record-keeping. By achieving these objectives, doctors can provide better care to their patients while also increasing the efficiency of their practices.

## **1.4. Limitations of the Project**

While a doctor appointment booking system can provide many benefits, there are also some limitations to consider, including:

1. Limited access for certain patients: Not all patients may have access to the internet or be comfortable with online booking, which can limit the reach of the system. In such cases, additional support may be needed to assist patients with scheduling appointments.
2. Dependence on technology: The appointment booking system relies on technology, which can be prone to technical issues, such as system crashes, internet connectivity issues, or other unforeseen

technical problems. This can potentially lead to appointment scheduling disruptions, and alternate arrangements may need to be made.

3. Increased workload for staff: While an appointment booking system can streamline the scheduling process, it can also increase the workload for staff members responsible for managing the system. These staff members may need to manage technical issues, support patients with booking appointments, and keep the system up-to-date with current information.

4. Privacy and security concerns: Online booking systems may be vulnerable to cyber attacks, which can compromise patient data privacy and security. This requires careful planning and implementation of security measures to ensure patient data is kept secure and confidential.

5. Resource constraints: Implementing an appointment booking system requires an initial investment in technology, staff training, and ongoing maintenance. Small practices or clinics may have limited resources, which can make it challenging to implement and maintain such systems.

Overall, while a doctor appointment booking system can provide many benefits, it is important to carefully consider and address the potential limitations to ensure that it is effectively implemented and meets the needs of patients and providers.

## **Chapter 2: LITERATURE SURVEY**

### **2.1) Introduction**

Here we present an interaction system for doctor and patient communication. It has an exceptional administration of several nodes through which doctors and patients interact with each other. The patients can easily access the clinic server nodes. Here the patients are allowed to interact with the doctors about their symptoms. The doctors can list and track their patients. Here the patients book their appointments online depending on the doctor's availability and their time feasibility. The doctors on the other hand can either extend or reduce their working hours depending on the number of patients arriving for that day.

### **2.2) Existing System**

#### **❖ KiviHealth**

##### **1. Owned by –**

VITALIC HEALTH PRIVATE LIMITED

No.250, Lloyds Road, Royapettah,

Chennai, Tamil Nadu, INDIA

PIN - 600014

##### **2. Author - Rajandeep Singh**

## **2.3) Disadvantages of Existing System**

<b>Disadvantages</b>	<b>Description</b>
Purchase Cost	The Purchase cost for KiviHealth is around 18,000 rs per year which is quite expensive.
Sms and Alert Cost	The App offers 2,000/- for 10,000 messages. So, It will cost around 6000/- to 20000/- per year for only Paying for SMS feature.
Patient Compliance	Young generation patients accept the technology but middle age-old age population still prefer paper receipt and calls
Difficult to use	Teaching staff regarding software and its use is very time consuming. Since, most of the staff at small hospital and small clinic are not much educated.
Inventory Management	Inventory Management is bit tricky and difficult. Not all Software provide customization options

## **2.4) Comparative Study**

The Product Cost of CareDent is Zero. The Application is free of cost. It is easily available at Play Store.

Also, there are no charges for any SMS Service. So, the total cost of using the application is zero. Which is a good thing in today's world.

The UI of the application is very very simple. Any age group people can use it with ease.

# **Chapter 3: ANALYSIS**

## **3.1. Introduction**

### **3.11 Technical Feasibility :**

A doctor appointment system is a software application that allows patients to schedule appointments with doctors online. The technical feasibility of a doctor appointment system depends on various factors, including the platform on which it is built, the scalability of the system, and the integration with other systems.

Here are some key technical considerations for a doctor appointment system:

1. Platform: The system should be built on a reliable and scalable platform, such as a cloud-based infrastructure, that can handle a large number of users and transactions.
2. Database: The system should have a robust database that can store patient information, doctor information, appointment schedules, and other relevant data.
3. User interface: The system should have a user-friendly interface that allows patients to easily schedule appointments, view available slots, and receive confirmation of their appointments.
4. Testing: The system should undergo thorough testing to ensure its functionality and reliability, including load testing to ensure it can handle a high volume of traffic.

Overall, a doctor appointment system is technically feasible, provided it is built on a reliable platform, has a robust database, a user-friendly interface, strong security measures, and can integrate with other healthcare systems.

### **3.12 Economical Feasibility :**

The economical feasibility of a doctor appointment system depends on various factors, including the development cost, maintenance cost, and potential revenue streams.

Here are some key economical considerations for a doctor appointment system:

1. Development Cost: The cost of developing a doctor appointment system can vary depending on the platform, the features, and the complexity of the system. A cost-benefit analysis should be performed to determine if the development cost is worth the potential benefits.
2. Maintenance Cost: The maintenance cost of a doctor appointment system includes ongoing software updates, bug fixes, and server maintenance. The cost of maintenance should be factored in when determining the overall cost of the system.
3. Revenue Streams: A doctor appointment system can generate revenue through charging patients a fee for using the system, or taking a commission on appointments booked through the system. A revenue model should be established to ensure the system is economically viable.

Overall, a doctor appointment system can be economically feasible if the development cost, maintenance cost, and potential revenue streams are carefully considered and analyzed. If there is a demand for such a system and it can generate enough revenue to cover the development and maintenance costs, then it can be a viable investment.

## **3.2. Software/Hardware Requirements**

### **1.Hardware Requirements:**

- **Device** - HP Pavilion is a line of consumer-oriented laptop and desktop computers produced by HP Inc. It was introduced in 1995 by Hewlett-Packard. The name is applied to both desktops and laptops for the Home and Home Office product range. The Pavilion mainly competes against computers such as Acer's Aspire, Dell's Inspiron and XPS, Lenovo's IdeaPad, Samsung's Sens and Toshiba's Satellite.
- **Processor** - Ryzen 5 4600H is a 64-bit hexa-core mid-range performance x86 mobile microprocessor introduced by AMD in early 2020. Fabricated on TSMC's 7-nanometer process and based on AMD's Zen 2 microarchitecture, the 4600H operates at a base frequency of 3.0 GHz with a TDP of 45 W and a boost frequency of up to 4.0 GHz.
- **Ram** – 8 GB
- **Hard Disk Capacity** – 1 TB
- **SSD Capacity** – 128 GB
- **GPU** - The GeForce 16 series is a series of graphics processing units developed by Nvidia, based on the Turing microarchitecture, announced in February 2019.<sup>[5]</sup> The 16 series, commercialized within the same timeframe as the 20 series, aims to cover the entry-level to mid-range market, not addressed by the latter. As a result, the media have mainly compared it to AMD's Radeon RX 500 series of GPUs.

### **2.Software Requirements:**

- **Android Studio** - Android Studio is the official<sup>[7]</sup> integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.<sup>[8]</sup> It is available for download on Windows, macOS and Linux based operating systems.<sup>[9]</sup> It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.
- **Google Firebase** - Firebase is a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication, and integration for a variety of applications, including Android, iOS, JavaScript, Node.js, Java, Unity, PHP, and C++.

### **3.4. Algorithms / API used**

#### **1.RazorPay :**

Razorpay is an India-based, fintech company that provides payment gateway services to vendors, merchants, and e-commerce platforms. It is an online system that is designed to handle end-to-end payments that can also be integrated with different websites and apps. The company was founded by IIT Roorkee alumni Shashank Kumar and Harshil Mathur in 2014. It is headquartered in Bangalore, India.

# **Chapter 4: DESIGN AND ARCHITECTURE**

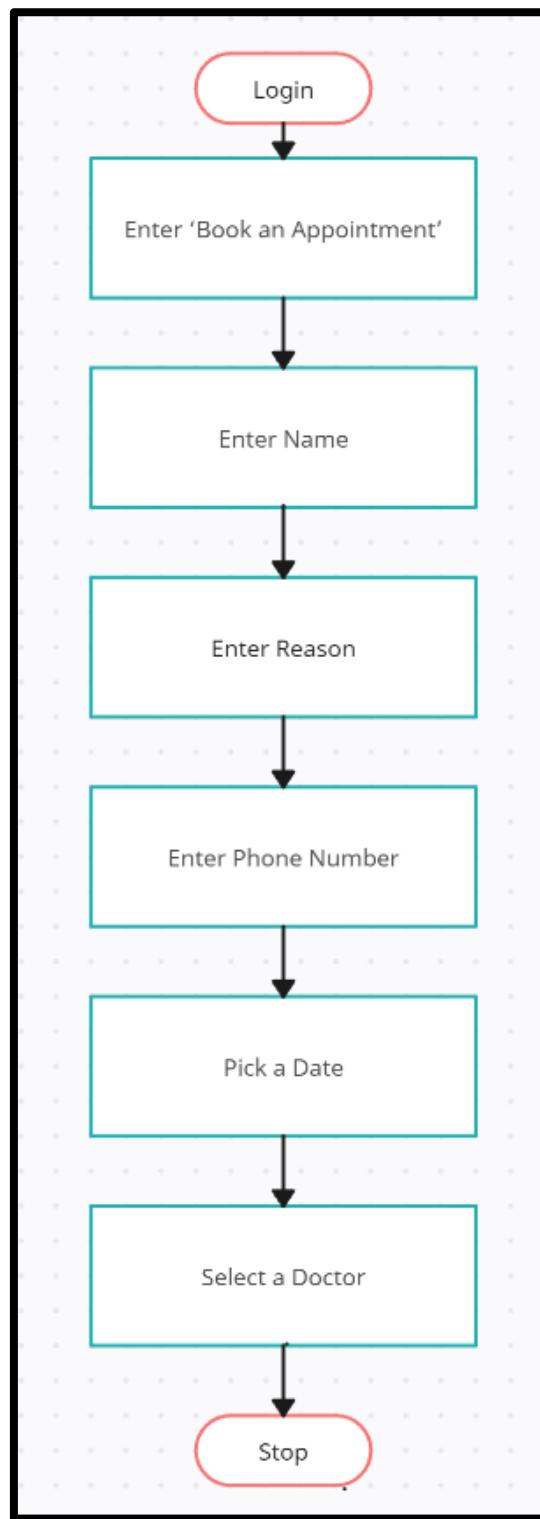
## **4.1. Introduction**

Designing an Android project requires careful planning and attention to detail. The first step in the process is to identify the problem that the app will solve and the target audience. This information will help to determine the app's features and functionality. The next step is to create wireframes and prototypes of the app, which will serve as the visual representation of the app's layout and user interface. The app's design should be user-friendly, intuitive, and aesthetically pleasing. Color schemes, fonts, and icons should be carefully chosen to create a cohesive and visually appealing app. It's also essential to consider accessibility and ensure that the app is usable for users with disabilities. Finally, testing and feedback from users will help to refine and improve the app's design before its release.

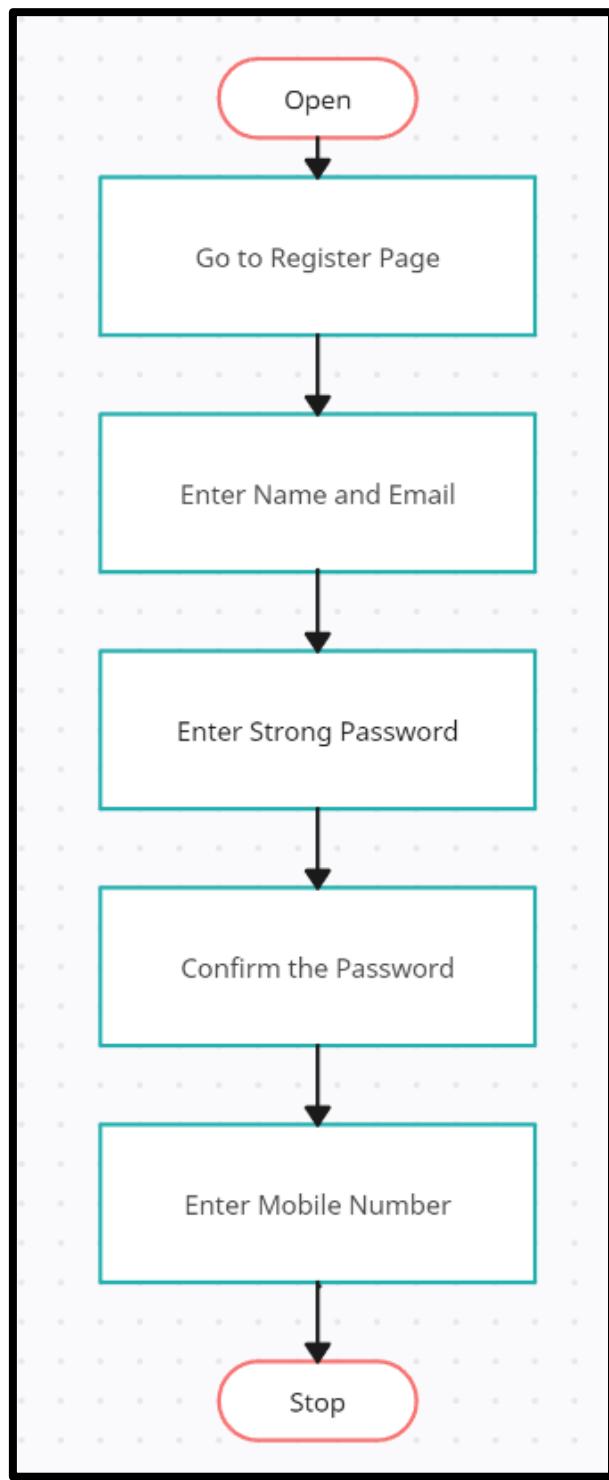
Design is the first step in development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. On the software requirement have been analyzed and specified the software design involves three technical activities- Design, Coding, Implementation, Testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activities decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decision has the final bearing upon reliability and maintainability of the a system. Design is only way to accurately transfer the customers requirements into finished software or system Design is the place where quality is fostered in development. Software design is the process through which requirements are translated into 2 representation of software. Software requirement is conducted in two steps Preliminary design is concerned with the transformation of requirements into data.

## 4.2. USE CASE DIAGRAM

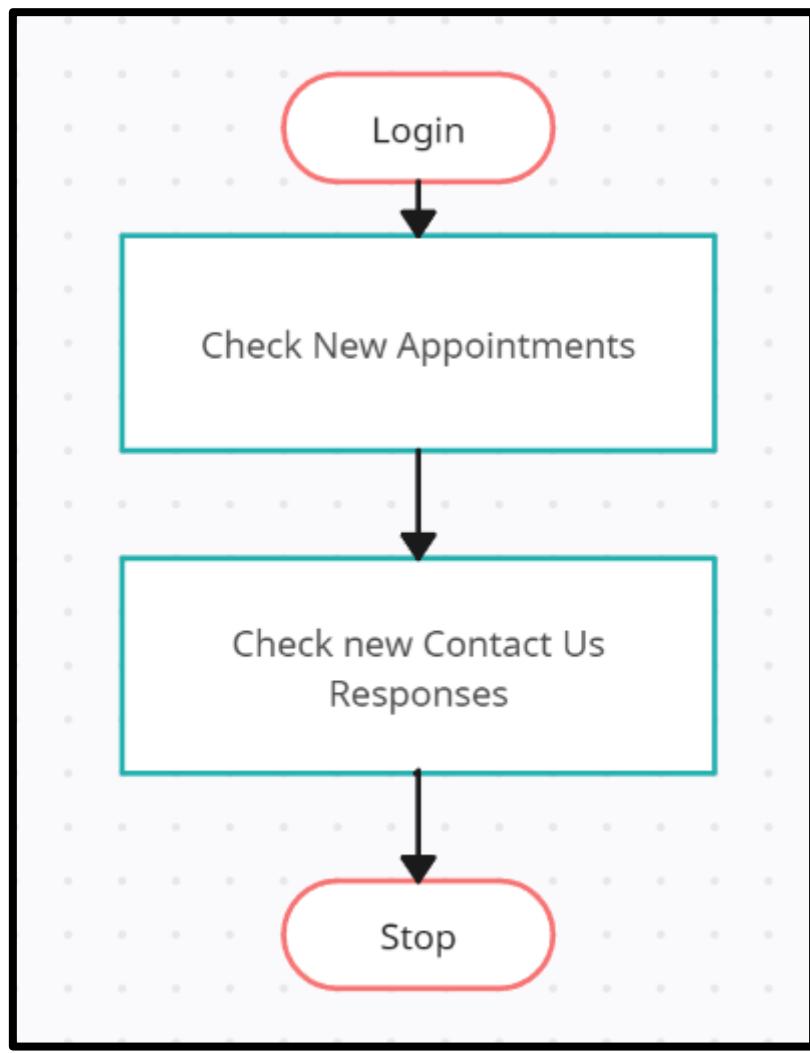
### 1. Login Page



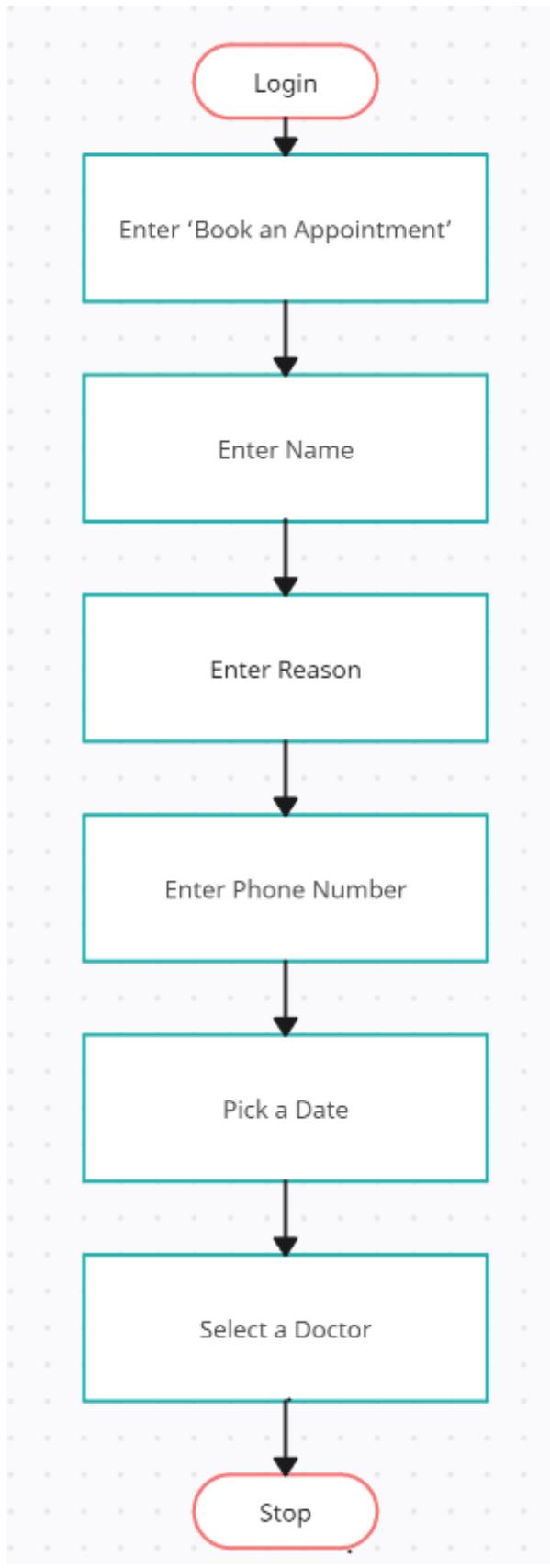
## 2. Register Page



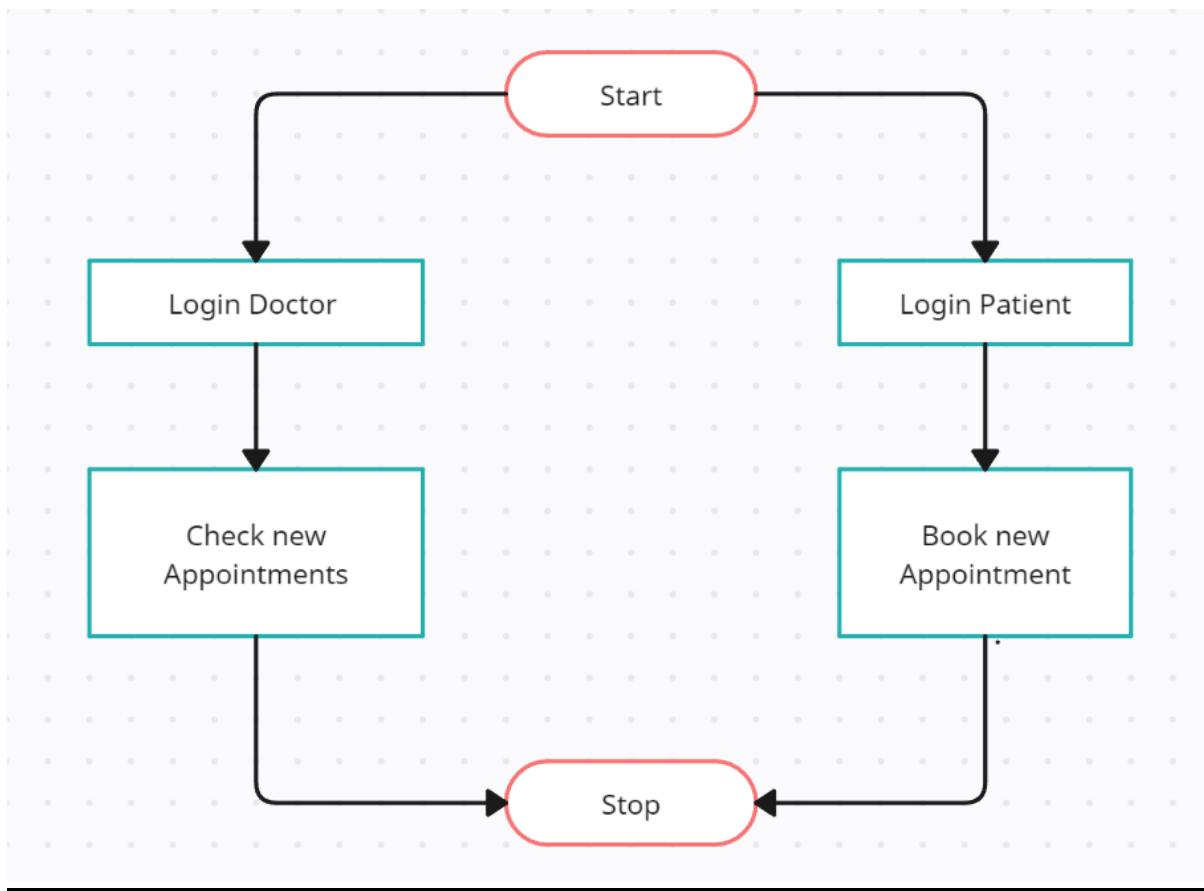
### 3. Doctor Login



#### **4. User Login**



## 5. Application Flow



# **Chapter 5.**

## **IMPLEMENTATION AND RESULT**

### **5.1 Introduction**

Code output is the result that a computer program generates when it is executed. When a programmer writes code, they use a specific programming language and syntax to give instructions to the computer on what actions to perform. The program can range from a few lines of code to a large, complex software system.

Once the programmer has written the code, they will compile or run the program. This process generates an output, which can be in various forms, such as text, numbers, graphics, or other types of data. The output serves as feedback on whether the program is functioning correctly or not.

Programmers use the output to debug their code, which involves identifying and fixing errors or bugs in the program. The output helps programmers understand what is happening within the program and allows them to make necessary changes to ensure it is working correctly.

Code output is an essential part of software development because it provides valuable information about the program's behavior and enables programmers to refine their code until it meets the desired outcome. The output can be used to generate reports, visualize data, or provide feedback to the user, among other purposes.

### **5.2 Explanation of Key functions**

#### **➤ Authentication**

Firebase Authentication provides backend services for easy use the SDKs and in-built UI libraries to authenticate the user in the

application. Most of the apps need the identity of the user, and after knowing their status, the app saves the user's data securely in the cloud. It aims to build a secure authentication system.

Using Firebase Authentication, we will authenticate the log-in of Google, Facebook, GitHub, Twitter and more.

#### **➤ Database**

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

#### **➤ RazorPay**

Razorpay Android App Payment Gateway supports over 100+ payment modes including Credit and Debit Cards (Visa, Mastercard, Rupay, AMEX, Diners), Net Banking from top 50+ banks, UPI (Web Collect & UPI Intent), Online Wallets, EMIs (Credit/Debit Cards & Cardless), PayLater and NEFT/RTGS payments.

### **5.3 Method of implementation (code and snapshots)**

#### **➤ Manifest File Code:**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/
    apk/res/android"
    xmlns:tools="http://schemas.android.com/to
    ols"
    package="com.example.caredentapp">
```

```

<uses-permission
    android:name="android.permission.INTERNET"
/>
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.Light"
    tools:targetApi="31">
<activity
    android:name=".MainActivity3"
    android:exported="false" />
<activity
    android:name=".MainActivity2"
    android:exported="false"
    android:label="@string/title_activity_main2"
    android:theme="@style/Theme.Design.NoActionBar" />
<activity
    android:name=".RetrieveData"
    android:exported="false" />
<activity
    android:name=".AboutClinic"
    android:exported="false" />
<activity
    android:name=".OTP_Verification_Page_Richa"
    android:exported="false" />
<activity
    android:name=".AllAppointmentRicha"
    android:exported="false" />
<activity
    android:name=".AllAppointmentAkshay"
    android:exported="false" />
<activity
    android:name=".ContactUsDataForDoctor"
    android:exported="false" />
<activity
    android:name=".ContactUs"
    android:exported="false" />
<activity
    android:name=".EditProfilePage"
    android:exported="false" />
<activity
    android:name=".ContactUsPage"
    android:exported="false" />
<activity
    android:name=".OTP_Verification_Registratio
n"
    android:exported="false" />
<activity
    android:name=".Doctor1Dash"
    android:exported="false" />
<activity
    android:name=".Doctor3Dash"
    android:exported="false" />
<activity
    android:name=".Doctor2Dash"
    android:exported="false" />
<activity
    android:name=".UserProfilePage"
    android:exported="false" />
<activity
    android:name=".OTP_Verification_Page"
    android:exported="false" />
<activity
    android:name=".DoctorLogin"
    android:exported="false" />
<activity
    android:name=".AllAppointmentForDoctor"
    android:exported="false" />
<activity
    android:name=".DoctorDash"
    android:exported="false" />
<activity
    android:name=".Doctor2"
    android:exported="false" />
<activity

```

```

        android:name=".Doctor3"
        android:exported="false" />
<activity
    android:name=".BookAppointment"
    android:exported="false" />
<activity
    android:name=".Doctor1"
    android:exported="false" />
<activity
    android:name=".AllAppointment"
    android:exported="false" />
<activity
    android:name=".DashboardPage"
    android:exported="false" />
<activity
    android:name=".RegisterPage"
    android:exported="false" />
<activity
    android:name=".LoginPage"
    android:exported="false" />
<activity
    android:name=".MainActivity"

    android:exported="true">
<intent-filter>
    <action
        android:name="android.intent.action.MAIN"
        />

        <category
        android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

</manifest>

```

## ➤ LoginPage.xml Code:

```

?      <?xml version="1.0" encoding="utf-
8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/
    apk/res/android"

    xmlns:app="http://schemas.android.com/apk
    /res-auto"

```

```

    xmlns:tools="http://schemas.android.com/to
    ols"

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".LoginPage"
        android:background="@color/white"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginTop="45sp"
            >
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="40dp"
                android:fontFamily="@font/poppins_semibol
                d"
                android:gravity="center"
                android:text="Login"
                android:textColor="@color/black"
                android:textSize="35dp" />
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_marginTop="80dp"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <ImageView
                android:layout_marginLeft="30dp"
                android:layout_width="35dp"
                android:layout_height="35dp"
                android:background="@drawable/emaillogo"
                android:layout_gravity="center"
                />
            <EditText
                android:id="@+id/etLoginEmail"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="00dp"
                android:inputType="textEmailAddress"
                android:layout_marginRight="50dp"
                android:hint="Email"

```

```

        android:fontFamily="@font/poppins"
        android:backgroundTint="#079BF5"
        android:textColor="@color/black"
        android:textColorHint="#D3D3D3"
        android:textSize="20dp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
    >

        <ImageView
            android:layout_width="35dp"
            android:layout_height="35dp"

            android:background="@drawable/locklogo"
            android:layout_gravity="center"
            android:layout_marginLeft="30dp"
        />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            android:textColor="@color/black"
            android:fontFamily="@font/poppins"
            android:id="@+id/etLoginPass"
            android:inputType="textPassword"
            android:backgroundTint="#079BF5"
            android:layout_marginRight="50dp"
            android:textColorHint="#D3D3D3"
            android:textSize="20dp"
        />
    </LinearLayout>

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Remember Me"
        android:id="@+id/remember_me"
        android:textColor="@color/black"
        android:buttonTint="@color/black"
        android:layout_marginLeft="50dp"
    />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginRight="100dp"
        android:id="@+id(btnLogin"
        android:background="@drawable/all_buttons"
        android:text="LOGIN"
        android:layout_marginTop="50dp"
    />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginRight="100dp"
        android:id="@+id/btnLoginforDoctor"
        android:background="@drawable/all_buttons"
        android:text="LOGIN FOR DOCTORS
ONLY"
        android:layout_marginTop="10dp"
    />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="00dp"
        android:orientation="vertical"
    >

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Don't have an account?"
            android:gravity="center"
            android:layout_gravity="center"
            android:layout_marginTop="120dp"
            android:textColor="@color/black"
            android:textSize="12dp"
        />

        <Button
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_marginLeft="100dp"
            android:layout_gravity="center"
            android:id="@+id/btnRegisterfrmLogin"
            android:background="@drawable/all_buttons"
        />
    
```

```

        android:layout_marginRight="100dp"
        android:text="REGISTER"
        />

    </LinearLayout>

</LinearLayout>

```

➤ **LoginPage.java Code:**

```

? package com.example.caredentapp;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import
com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import
com.google.firebaseio.auth.FirebaseAuth;

public class LoginPage extends
AppCompatActivity {

    EditText etLoginEmail,etLoginPass;
    Button
    btnLogin,btn_register,btndoctorlogin;
    FirebaseAuth mAuth;
    public static final String sharedprefs=
"sharedprefs";

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.activity_login_page);
        getSupportActionBar().hide();

        checkBox();

        btn_register =
findViewById(R.id.btnRegisterfrmLogin);
        etLoginEmail =
findViewById(R.id.etLoginEmail);
        etLoginPass =
findViewById(R.id.etLoginPass);
        btnLogin = findViewById(R.id.btnLogin);
        btndoctorlogin =
findViewById(R.id.btnLoginforDoctor);

        mAuth = FirebaseAuth.getInstance();

        btndoctorlogin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(LoginPage.this,DoctorLogin.class));
                finish();
            }
        });
        btn_register.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                movetoregisterpage();
            }
        });
        btnLogin.setOnClickListener(view ->{
            LoginUser();
        });
    }

    private void checkBox() {
        SharedPreferences sharedPreferences =
getSharedPreferences(sharedprefs,MODE_pri
vate);
        String check =
sharedPreferences.getString("name","");
        if (check.equals("true")){

```

```

        Toast.makeText(LoginPage.this, "User
Logged In successfully",
Toast.LENGTH_LONG).show();
        startActivity(new Intent(LoginPage.this,
DashboardPage.class));
        finish();
    }
}

private void LoginUser() {
    String email =
etLoginEmail.getText().toString();
    String password =
etLoginPass.getText().toString();

    if (TextUtils.isEmpty(email)) {
        etLoginEmail.setError("Email cannot be
Empty");
        etLoginEmail.requestFocus();
    } else if (TextUtils.isEmpty(password)) {
        etLoginPass.setError("Password cannot
be Empty");
        etLoginPass.requestFocus();
    } else {

mAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull
Task<AuthResult> task) {
        if (task.isSuccessful()) {
            SharedPreferences
sharedPreferences =
getSharedPreferences(sharedprefs, MODE_PRI
VATE);
            SharedPreferences.Editor editor
= sharedPreferences.edit();

            editor.putString("name", "true");
            editor.apply();

            Toast.makeText(LoginPage.this,
"User Logged In successfully",
Toast.LENGTH_LONG).show();
            startActivity(new
Intent(LoginPage.this, DashboardPage.class));
        } else {
            Toast.makeText(LoginPage.this,
"Login In Error - " +
task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }
});

private void movetoregisterpage(){
    Intent intent = new
Intent(LoginPage.this, RegisterPage.class);
    startActivity(intent);
}
}

```

## ➤ RegisterPage.xml Code:

```

?      <?xml version="1.0" encoding="utf-
8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/
apk/res/android"

xmlns:app="http://schemas.android.com/apk
/res-auto"

xmlns:tools="http://schemas.android.com/to
ols"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/white"
    tools:context=".RegisterPage">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="45dp"
    android:orientation="vertical"
    >

<TextView
    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:text="Register"
        android:gravity="center"

        android:fontFamily="@font/poppins_semiibold"
        android:textSize="35dp"
        android:textColor="@color/black"
        android:layout_marginTop="40dp"
    />

</LinearLayout>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Name"
    android:fontFamily="@font/poppins"
    android:backgroundTint="#079BF5"
    android:layout_marginLeft="50dp"
    android:id="@+id/etRegName"
    android:layout_marginRight="50dp"
    android:textColorHint="#D3D3D3"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:layout_marginTop="35dp"
/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:id="@+id/etRegEmail"
    android:fontFamily="@font/poppins"
    android:backgroundTint="#079BF5"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="13dp"
    android:textColor="@color/black"
    android:layout_marginRight="50dp"
    android:textColorHint="#D3D3D3"
    android:textSize="20dp"
/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="@font/poppins"
    android:backgroundTint="#079BF5"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="13dp"
    android:textColorHint="#D3D3D3"
    android:textColor="@color/black"
    android:layout_marginRight="100dp"
    android:layout_marginLeft="50dp"
    android:background="@drawable/all_buttons"
    android:layout_marginRight="100dp"
    android:text="REGISTER"
/>

```

```

        android:id="@+id(btnRegister"
        android:layout_marginTop="100dp"
        />

    </LinearLayout>

```

## ➤ RegisterPage.java Code:

```

package com.example.caredentapp;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import
com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import
com.google.firebase.FirebaseException;
import com.google.firebase.auth.AuthResult;
import
com.google.firebase.auth.FirebaseAuth;
import
com.google.firebase.auth.PhoneAuthCredential;
import
com.google.firebase.auth.PhoneAuthProvider;
import
com.google.firebaseio.database.FirebaseDatabase;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;

public class RegisterPage extends
AppCompatActivity {

    FirebaseAuth mAuth;

```

```

        EditText
etRegEmail,etRegPass,etRegName,etRegPhon
eNo,etRegConPass;
        Button btnRegister;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_register_pa
ge);
        getSupportActionBar().hide();

        etRegConPass =
findViewById(R.id.etRegConPass);
        etRegPhoneNo =
findViewById(R.id.etRegPhoneNo);
        etRegName =
findViewById(R.id.etRegName);
        btnRegister =
findViewById(R.id.btnRegister);
        etRegEmail =
findViewById(R.id.etRegEmail);
        etRegPass =
findViewById(R.id.etRegPass);
        mAuth = FirebaseAuth.getInstance();

        btnRegister.setOnClickListener(view ->{
            createUser();
        });
    }

    private void createUser(){
        String email =
etRegEmail.getText().toString();
        String password =
etRegPass.getText().toString();
        String conpassword =
etRegConPass.getText().toString();
        String name =
etRegName.getText().toString();
        String phoneno =
etRegPhoneNo.getText().toString();
    }

```

```
if(TextUtils.isEmpty(name)){
    etRegName.setError("Name cannot be
empty");
    etRegName.requestFocus();
}else if(TextUtils.isEmpty(email)){
    etRegEmail.setError("Email cannot be
Empty");
    etRegEmail.requestFocus();
}else if(TextUtils.isEmpty(password)){
    etRegPass.setError("Password cannot
be Empty");
    etRegPass.requestFocus();
}else
if(!TextUtils.equals(conpassword,password)){
    etRegConPass.setError("Confirm
Password doesn't match Password");
    etRegConPass.requestFocus();
}else if(etRegPhoneNo.length() != 10){
    etRegPhoneNo.setError("Enter Correct
Phone Number");
    etRegPhoneNo.requestFocus();
}else {
    //startActivity(intent1);
}else{
    Toast.makeText(RegisterPage.this,
    "Registration Error" +
    task.getException().getMessage(),
    Toast.LENGTH_SHORT).show();
}
}
}
}
}
}

private void registerUser(){
    String reg_nametosend =
etRegName.getText().toString();
    String reg_phonenumbertosend =
etRegPhoneNo.getText().toString();
    String reg_emailtosend =
etRegEmail.getText().toString();
}
```

```
mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()){
            registerUser();
        }
        Toast.makeText(RegisterPage.this,"Successfull
y reigstered",Toast.LENGTH_LONG).show();
        //startActivity(new Intent(RegisterPage.this,LoginPage.class));
        Intent intent = new Intent(RegisterPage.this, LoginPage.class);
        startActivity(intent);
        Intent intent1 = new Intent(RegisterPage.this, UserProfilePage.class
);
        //intent.putExtra("profile_name_intent",name);
    }
}
```

```
    HashMap<String, Object> map1 = new  
    HashMap<>();  
  
    map1.put("name", etRegName.getText().toString());  
  
    map1.put("phone", etRegPhoneNo.getText().toString());  
    map1.put("email", reg_emailtosend);  
  
    FirebaseDatabase.getInstance().getReference()  
        .child("Profile").setValue(map1).addOnCompleteListener(new OnCompleteListener<Void>()  
{  
    @Override  
    public void onComplete(@NonNull  
        Task<Void> task) {  
        if(task.isSuccessful()){  
            Intent intent = new  
            Intent(RegisterPage.this, LoginPage.class);  
  
            intent.putExtra("name1", reg_name);  
        }  
    }  
});
```

```

        intent.putExtra("reason1",reg_phonenumbert
osend);
        startActivity(intent);
        finish();

Toast.makeText(getApplicationContext(),
"Registered", Toast.LENGTH_SHORT).show();
}else{

Toast.makeText(getApplicationContext(),
"Failed to Reg",
Toast.LENGTH_SHORT).show();
}
});

}

}

```

## ➤ BookAppointment.xml

```

?> <?xml version="1.0" encoding="utf-
8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com
    /apk/res/android"

    xmlns:app="http://schemas.android.com/apk
    /res-auto"

    xmlns:tools="http://schemas.android.com/to
    ols"
    android:orientation="vertical"
    android:background="@color/white"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BookAppointment">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:orientation="horizontal"
        android:background="#82CAFF"
        >
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="BOOK APPOINTMENT"
            android:textSize="25dp"
            android:layout_marginTop="5dp"
            android:textColor="@color/black"
            android:layout_marginLeft="50dp"
            android:layout_marginRight="50dp"
            android:layout_gravity="center"
            android:fontFamily="@font/poppins_semibol
d"
            android:gravity="center_horizontal"
            />
    
```

</EditText

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your Name"
        android:layout_marginLeft="30dp"
        android:fontFamily="@font/poppins"
        android:backgroundTint="#079BF5"
        android:textColor="@color/black"
        android:textColorHint="#D3D3D3"
        android:id="@+id/et_name"
        android:layout_marginRight="30dp"
        android:textSize="25dp"
        android:layout_marginTop="50dp"
        />
```

<EditText

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="@font/poppins"
        android:backgroundTint="#079BF5"
        android:textColor="@color/black"
        android:textColorHint="#D3D3D3"
        android:hint="Appointment Reason"
        android:layout_marginLeft="30dp"
        android:id="@+id/et_reason"
        android:layout_marginRight="30dp"
        android:textSize="25dp"
        android:layout_marginTop="20dp"
        />
```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Select your Date"
    android:textSize="25dp"
    android:gravity="center"
    android:layout_marginTop="20dp"
    android:textColor="@color/black"
    />

<Button
    android:id="@+id/date_input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:layout_marginTop="15dp"
    android:textSize="17dp"
    android:layout_marginRight="100dp"

    android:background="@drawable/all_buttons"
    android:text="Pick Date" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Available Doctors"
    android:textSize="25dp"
    android:gravity="center"
    android:layout_marginTop="20dp"
    android:textColor="@color/black"
    />
<HorizontalScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp"
    >
    <RadioGroup
        android:layout_width="match_parent"

        android:layout_height="wrap_content"
        android:orientation="horizontal"
        >
        <RadioButton
            android:layout_margin="10dp"
            android:textAlignment="center"
            android:id="@+id/radio_doct1"
            android:button="@color/white"
            android:background="@drawable/custom_radio_selector"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:text="DR. Akshay"
            android:textColor="@drawable/custom_radio_text"/>
        <RadioButton
            android:layout_width="150dp"
            android:layout_margin="10dp"
            android:button="@color/white"
            android:id="@+id/radio_doct2"
            android:layout_height="150dp"
            android:text="DR. Richa"
            android:textColor="@drawable/custom_radio_text"/>
        <RadioButton
            android:layout_width="150dp"
            android:layout_margin="10dp"
            android:button="@color/white"
            android:id="@+id/radio_doct3"
            android:layout_height="150dp"
            android:text="DR. VaZe"
            android:textColor="@color/black"/>
    </RadioGroup>
</HorizontalScrollView>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Book And Pay ₹150"
    android:id="@+id/book_appointment"
    android:background="@drawable/custom_radio_selector"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:text="Book And Pay ₹150"
    android:id="@+id/book_appointment"
    android:textColor="@drawable/custom_radio_text"/>

```

```

        android:layout_marginTop="20dp"
        android:layout_marginLeft="100dp"
        android:layout_marginRight="100dp"

    android:background="@drawable/all_button
s"
    />
</LinearLayout>

```

## ➤ BookAponintment.java

```

package com.example.caredentapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity
;
import
androidx.core.content.res.ResourcesCompat;

import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.TimePickerDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import
android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.text.TextUtils;
import android.text.format.DateFormat;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.RadioButton;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import
com.google.android.gms.tasks.OnCompleteListener
stener;

```

```

import com.google.android.gms.tasks.Task;
import
com.google.firebaseio.database.DatabaseRefer
ence;
import
com.google.firebaseio.database.FirebaseDataB
ase;
import com.razorpay.Checkout;
import com.razorpay.PaymentResultListener;

import org.json.JSONObject;

import java.util.Calendar;
import java.util.HashMap;

public class BookAppointment extends
AppCompatActivity implements
PaymentResultListener {
private static final String CHANNEL_ID = "My
Channel";
private static final int NOTIFICATION_ID =
100;

Button date_in, book_appointment;
EditText et_name, et_reason;
DatabaseReference info;
String dateString = "", selected_doctor;
ImageButton dr_akshay, dr_richa;
RadioButton
radio_doct1, radio_doct2, radio_doct3;

@Override
protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_book_appo
intment);
getSupportActionBar().hide();

date_in = findViewById(R.id.date_input);
book_appointment =
findViewById(R.id.book_appointment);
et_name = findViewById(R.id.et_name);
et_reason =
findViewById(R.id.et_reason);

```

```

radio_doct1 =
findViewById(R.id.radio_doct1);
radio_doct2 =
findViewById(R.id.radio_doct2);
radio_doct3 =
findViewById(R.id.radio_doct3);

Checkout.reloadData(getApplicationContext());

        date_in.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        selectdate();
    }
});

book_appointment.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {

        String nametosend =
et_name.getText().toString();
        String reasontosend =
et_reason.getText().toString();

        if (TextUtils.isEmpty(nametosend)){
//Toast.makeText(getApplicationContext(),"E
nter your Name",
Toast.LENGTH_SHORT).show();
        et_name.setError("Please Enter
your Name!");
        et_name.requestFocus();
    }else if
(TextUtils.isEmpty(reasontosend)){
        et_reason.setError("Please Enter
your Reason");
        et_reason.requestFocus();
    }else if(dateString.isEmpty()){

        Toast.makeText(BookAppointment.this,
"Please Select a Date",
Toast.LENGTH_SHORT).show();
    }else{
        paymentnow("15000");
        //goback();
    }
/*
        info =
FirebaseDatabase.getInstance().getReference
().child("Name");

        String name =
et_name.getText().toString();
        String reason =
et_reason.getText().toString();

        Patients patients = new
Patients(name,reason);

        info.push().setValue(patients);

        Toast.makeText(BookAppointment.this,"App
ointment
Booked",Toast.LENGTH_SHORT).show();
*/
    }
});
}

private void goback() {
    startActivity(new
Intent(BookAppointment.this,
DashboardPage.class));
}

public void selectdate(){
    Calendar calendar =
Calendar.getInstance();

    int Year = calendar.get(Calendar.YEAR);
    int Month =
calendar.get(Calendar.MONTH);
    int Day = calendar.get(Calendar.DATE);

    DatePickerDialog datePickerDialog = new
DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {

```

```

    @Override
    public void onDateSet(DatePicker
datePicker, int year, int month, int date) {
        dateString = date + "/" + (month + 1)
+ "/" + year;
        date_in.setText(dateString);
        /*
        HashMap<String, Object> map = new
HashMap<>();
        map.put("date", dateString);

        FirebaseDatabase.getInstance().getReference
().child("Patients").push().setValue(map).add
OnCompleteListener<Void>() {
            @Override
            public void
onComplete(@NonNull Task<Void> task) {
                if(task.isSuccessful()){
                    Intent intent = new
Intent(BookAppointment.this,AllAppointmen
t.class);

                    intent.putExtra("date1",dateString);

                    Toast.makeText(getApplicationContext(),
"Appointment Booked",
Toast.LENGTH_SHORT).show();
                }else{

                    Toast.makeText(getApplicationContext(),
"Failed to Book",
Toast.LENGTH_SHORT).show();
                }
            }
        });
        */
    }

    },Year,Month,Day);
    datePickerDialog.show();
}

private void paymentnow(String amount){
}

```

```

    final Activity activity = this;
    Checkout checkout = new Checkout();

    checkout.setKeyID("rzp_test_PyqlhTycsKqnM
A");

    double newamount =
Float.parseFloat(amount) * 100;

    try {
        JSONObject options = new
JSONObject();

        options.put("name", "CareDent");
        options.put("description", "Reference
No. #123456");
        options.put("image",
"https://s3.amazonaws.com/rzp-
mobile/images/rzp.jpg");
        //options.put("order_id",
"order_DBJOWzybf0sJbb");//from response
of step 3.
        options.put("theme.color",
"#3399cc");
        options.put("currency", "INR");
        options.put("amount",
"15000");//pass amount in currency subunits
        options.put("prefill.email",
"ayushvaze2003@gmail.com");

        options.put("prefill.contact", "9988776655");
        JSONObject retryObj = new
JSONObject();
        retryObj.put("enabled", true);
        retryObj.put("max_count", 4);
        options.put("retry", retryObj);

        checkout.open(activity, options);

    } catch(Exception e) {
        Log.e("TAG", "Error in starting
Razorpay Checkout", e);
    }
}

@Override

```

```

public void onPaymentSuccess(String s) {
    Toast.makeText(getApplicationContext(),
    "Payment Successful",
    Toast.LENGTH_SHORT).show();

    sendNotification();

    if (radio_doct1.isChecked()){
        select_doct1();
    } else if (radio_doct2.isChecked()){
        select_doct2();
    } else if (radio_doct3.isChecked()){
        select_doct3();
    }
    // else if (!(radio_doct1.isChecked()) &&
    !(radio_doct2.isChecked()) &&
    !(radio_doct3.isChecked())){
    //     Toast.makeText(this, "Please Select a
    Doctor", Toast.LENGTH_SHORT).show();
    // }
}

private void select_doct3() {
    String nametosend2 =
    et_name.getText().toString();
    String reasontosend2 =
    et_reason.getText().toString();
    selected_doctor = "VaZe";

    HashMap<String, Object> map = new
    HashMap<>();

    map.put("name", et_name.getText().toString());
    map.put("reason", et_reason.getText().toString());
    map.put("date", dateString);
    map.put("doctor", selected_doctor);

    FirebaseDatabase.getInstance().getReference()
    .child("VaZe").push().setValue(map).addOn
   CompleteListener(new
    OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull
        Task<Void> task) {
            if(task.isSuccessful()){
                Intent intent = new
                Intent(BookAppointment.this,AllAppointmen
                t.class);

                intent.putExtra("name1",nametosend2);

                intent.putExtra("reason1",reasontosend2);

                intent.putExtra("date1",dateString);

                intent.putExtra("doctor1",selected_doctor);
                startActivity(intent);

                Toast.makeText(getApplicationContext(),
                "Appintment Booked",
                Toast.LENGTH_SHORT).show();
            }else{
                Toast.makeText(getApplicationContext(),
                "Failed to Book",
                Toast.LENGTH_SHORT).show();
            }
        }
    });
}

private void select_doct2() {
    String nametosend1 =
    et_name.getText().toString();
    String reasontosend1 =
    et_reason.getText().toString();
    selected_doctor = "Richa";

    HashMap<String, Object> map = new
    HashMap<>();

    map.put("name", et_name.getText().toString());
    map.put("reason", et_reason.getText().toString());
    map.put("date", dateString);
    map.put("doctor", selected_doctor);

    FirebaseDatabase.getInstance().getReference()

```

```

        () .child("Richa") .push() .setValue (map) .addOn
        CompleteListener (new
        OnCompleteListener <Void> () {
            @Override
            public void onComplete (@NonNull
            Task <Void> task) {
                if (task.isSuccessful ()) {
                    Intent intent = new
                    Intent (BookAppointment.this, AllAppointmen
                    t.class);

                    intent.putExtra ("name1", nameToSend1);

                    intent.putExtra ("reason1", reasonToSend1);

                    intent.putExtra ("date1", dateString);

                    intent.putExtra ("doctor1", selected_doctor);
                    startActivity (intent);

                    Toast.makeText (getApplicationContext (),
                    "Appintment Booked",
                    Toast.LENGTH_SHORT).show ();
                } else {
                    Toast.makeText (getApplicationContext (),
                    "Failed to Book",
                    Toast.LENGTH_SHORT).show ();
                }
            });
        }

        private void select_doct1 () {
            String nameToSend =
            et_name.getText ().toString ();
            String reasonToSend =
            et_reason.getText ().toString ();
            selected_doctor = "Akshay";

            HashMap <String, Object> map = new
            HashMap <> ();

            map.put ("name", et_name.getText ().toString ());
            map.put ("reason", et_reason.getText ().toString ());
        }
    }

    map.put ("date", dateString);
    map.put ("doctor", selected_doctor);

    FirebaseDatabase.getInstance ().getReference
    () .child ("Akshay") .push () .setValue (map) .addOn
    CompleteListener (new
    OnCompleteListener <Void> () {
        @Override
        public void onComplete (@NonNull
        Task <Void> task) {
            if (task.isSuccessful ()) {
                Intent intent = new
                Intent (BookAppointment.this, AllAppointmen
                t.class);

                intent.putExtra ("name1", nameToSend);

                intent.putExtra ("reason1", reasonToSend);

                intent.putExtra ("date1", dateString);

                intent.putExtra ("doctor1", selected_doctor);
                startActivity (intent);
                finish();

                Toast.makeText (getApplicationContext (),
                "Appintment Booked",
                Toast.LENGTH_SHORT).show ();
            } else {
                Toast.makeText (getApplicationContext (),
                "Failed to Book",
                Toast.LENGTH_SHORT).show ();
            }
        }
    });

    @Override
    public void onPaymentError (int i, String s) {
        Toast.makeText (getApplicationContext (),
        "Payment Unsuccessful",
        Toast.LENGTH_SHORT).show ();
        Intent intent = new
        Intent (BookAppointment.this, DashboardPage
        .class);
        startActivity (intent);
    }
}

```

```

    }

    public void select_doctor(){
        selected_doctor = null;

        if (radio_doc1.isChecked()){
            selected_doctor = "Akshay";
        } else if (radio_doc2.isChecked()){
            selected_doctor = "Richa";
        } else {
            selected_doctor = "Ayush";
        }
        Toast.makeText(getApplicationContext(),
        "Your Doctor : " + selected_doctor,
        Toast.LENGTH_SHORT).show();
    }

    public void sendNotification(){

        Notification notification;
        Drawable drawable =
        ResourcesCompat.getDrawable(getResources
        (), R.drawable.icon1 , null);
        BitmapDrawable bitmapDrawable =
        (BitmapDrawable) drawable;
        Bitmap largelIcon =
        bitmapDrawable.getBitmap();

        NotificationManager nm =
        (NotificationManager)
        getSystemService(NOTIFICATION_SERVICE);
        if (android.os.Build.VERSION.SDK_INT >=
        android.os.Build.VERSION_CODES.O) {
            notification = new
            Notification.Builder(this)
                .setLargeIcon(largelIcon)
                .setSmallIcon(R.drawable.icon1)
                .setContentText("Your
Appointment has been booked Successfully
on " + dateString)
                .setSubText("Appointment
Message")
                .setChannelId(CHANNEL_ID)
                .build();
            nm.createNotificationChannel(new
NotificationChannel(CHANNEL_ID,"New
Channel",
NotificationManager.IMPORTANCE_HIGH));
        } else{
            notification = new
            Notification.Builder(this)
                .setLargeIcon(largelIcon)
                .setSmallIcon(R.drawable.icon1)
                .setContentText("Your
Appointment has been booked Successfully")
                .setSubText("Appointment
Message")
                .build();
            nm.notify(NOTIFICATION_ID,
            notification);
        }
    }
}

➤ UserProfile.xml
    <?xml version="1.0" encoding="utf-
8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com
    /apk/res/android"
    xmlns:app="http://schemas.android.com/apk
    /res-auto"
    xmlns:tools="http://schemas.android.com/to
    ols"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:orientation="vertical"
    tools:context=".UserProfilePage">
    <!-- top bar -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:orientation="horizontal"
        android:background="#82CAFF"
        >
        <LinearLayout
            android:layout_width="wrap_content"
            android:gravity="center"
            android:layout_height="match_parent"
            android:orientation="vertical">
            <ImageButton
                android:layout_width="50dp"

```

```

        android:layout_gravity="center"
        android:layout_height="50dp"

    android:background="@drawable/arrow2"

    android:id="@+id/back_to_dashboard_from
    _profile"
    />
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:gravity="center"
    android:layout_marginLeft="300dp"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <ImageButton
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:background="@drawable/editlogo1"
        android:id="@+id/edit_profile"
        />
</LinearLayout>

</LinearLayout>

<!-- Profile Top bar -->

<!-- Profile Middle Bar -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="20dp"
    android:layout_marginRight="20dp"
    android:background="@drawable/profile_ba
    ckground"
    android:orientation="vertical">

<de.hdodenhof.circleimageview.CircleImage
View
    android:id="@+id/profile_picture"
    android:layout_width="120dp"
    android:layout_height="120dp"
    android:layout_gravity="center"
    android:src="@drawable/female_doc_logo"
    android:layout_marginTop="20dp"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/profile_ba
        ckground"
        android:orientation="vertical">

        <TextView
            android:id="@+id/profile_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:gravity="center"
            android:text=""
            android:textColor="@color/black"
            android:textSize="25dp" />

        <TextView
            android:id="@+id/profile_phone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="0dp"
            android:fontFamily="@font/poppins_semiol
            d"
            android:gravity="center"
            android:text=""
            android:textColor="@color/black"
            android:textSize="20dp" />

        <TextView
            android:id="@+id/profile_email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:fontFamily="@font/poppins_semiol
            d"
            android:gravity="center"
            android:text=""
            android:textColor="@color/black"
            android:textSize="20dp" />

```

```

        android:layout_height="wrap_content"
        android:layout_marginTop="0dp"

    android:layout_marginBottom="10dp"
        android:gravity="center"
        android:text="ayush1@gmail.com"
        android:textColor="@color/black"
        android:textSize="20dp" />
</LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"

    android:background="@drawable/profile_background"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"

        android:layout_height="wrap_content"
        android:text="LOG OUT"
        android:id="@+id/logout_btn"
        android:textColor="@color/black"
        android:gravity="center"
        android:fontFamily="@font/poppins"
        android:textSize="25dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="10dp"
    />
</LinearLayout>
</LinearLayout>

➤ UserProfilePage.java
❷ package com.example.caredentapp;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import
androidx.appcompat.app.AppCompatActivity
;
import
androidx.recyclerview.widget.LinearLayoutManager;
import
androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;

```

```

import android.provider.MediaStore;
import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import
com.google.firebaseio.database.DataSnapshot;
import
com.google.firebaseio.database.DatabaseError;
import
com.google.firebaseio.database.DatabaseRefer
ence;
import
com.google.firebaseio.database.FirebaseDatab
ase;
import
com.google.firebaseio.database.ValueEventList
ener;

import java.util.ArrayList;

public class UserProfilePage extends
AppCompatActivity {

    TextView
payment_history,shareapp,contactus,change
password,logout_btn;
    TextView
profile_name,profile_phone,profile_email;
    ImageButton
editprofile,backtodashfromprofile;
    ImageView profilepic;
    public static final String sharedprefs=
"sharedprefs";
    private final int value = 1000;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_user_profil
e_page);

        shareapp = findViewById(R.id.shareapp);
        profilepic =
findViewById(R.id.profile_picture);
        logout_btn =
findViewById(R.id.logout_btn);
        contactus =
findViewById(R.id.contactus);
        editprofile =
findViewById(R.id.edit_profile);
        backtodashfromprofile =
findViewById(R.id.back_to_dashboard_from_
profile);
        profile_name =
findViewById(R.id.profile_name);
        profile_phone =
findViewById(R.id.profile_phone);
        profile_email =
findViewById(R.id.profile_email);

        FirebaseDatabase database1 =
FirebaseDatabase.getInstance();
        DatabaseReference myRef1 =
database1.getReference("Profile/name");
        DatabaseReference myRef2 =
database1.getReference("Profile/email");
        DatabaseReference myRef3 =
database1.getReference("Profile/phone");

        myRef1.addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull
DataSnapshot snapshot) {
                String name =
snapshot.getValue(String.class);
                profile_name.setText(name);
            }
        });

        @Override
        public void onCancelled(@NonNull
DatabaseError error) {
        }
    });
    myRef2.addValueEventListener(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull
DataSnapshot snapshot) {

```

```

String email =
snapshot.getValue(String.class);
profile_email.setText(email);
}

@Override
public void onCancelled(@NonNull
DatabaseError error) {

};

myRef3.addValueEventListener(new
ValueEventListener() {
    @Override
    public void onDataChange(@NonNull
DataSnapshot snapshot) {
        String phone =
snapshot.getValue(String.class);
        profile_phone.setText(phone);
    }

    @Override
    public void onCancelled(@NonNull
DatabaseError error) {

};

backtodashfromprofile.setOnClickListener(ne
w View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new
Intent(UserProfilePage.this,DashboardPage.cl
ass));
    }
});

editprofile.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent iGallery = new
Intent(Intent.ACTION_PICK);

iGallery.setData(MediaStore.Images.Media.E
XTERNAL_CONTENT_URI);
startActivityForResult(iGallery,value);
    }
});

contactus.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new
Intent(UserProfilePage.this,ContactUs.class));
    }
});

shareapp.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent();

intent.setAction(Intent.ACTION_SEND);
        intent.setType("text/plain");

intent.putExtra(Intent.EXTRA_TEXT,"Downloa
d BC");

startActivity(Intent.createChooser(intent,"Ch
oose one"));
    }
});

logout_btn.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        SharedPreferences
sharedPreferences =
getSharedPreferences(sharedprefs,MODE_PR
IVATE);
        SharedPreferences.Editor editor =
sharedPreferences.edit();

editor.putString("name","");
        editor.apply();
    }
});

```

```
Intent intent = new Intent(UserProfilePage.this,LoginPage.class);
startActivity(intent);
finish();
}
});

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode,
resultCode, data);

    if (resultCode == RESULT_OK){
        if (requestCode == value){

profilepic.setImageURI(data.getData());
    }
}
}


```

android:background="#82CAFF">>

<TextView  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:text="Appointments Here"  
    android:textSize="25dp"  
    android:layout\_marginTop="5dp"  
    android:textColor="@color/black"  
    android:layout\_marginLeft="50dp"  
    android:layout\_marginRight="50dp"  
    android:layout\_gravity="center"

        android:fontFamily="@font/poppins\_semiBold"  
        "  
            android:gravity="center\_horizontal"  
            />>

</LinearLayout>

<androidx.recyclerview.widget.RecyclerView  
    android:layout\_width="match\_parent"  
    android:layout\_marginTop="20dp"  
    android:layout\_marginLeft="20dp"

## ➤ AllAppointmentForDoctor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
    android:orientation="vertical"

    android:background="@drawable/profile_background"
    android:layout_height="match_parent"
    tools:context=".AllAppointmentForDoctor">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:orientation="horizontal"
```

```
        android:background="#82CAFF"
    >
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Appointments Here"
    android:textSize="25dp"
    android:layout_marginTop="5dp"
    android:textColor="@color/black"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:layout_gravity="center"

    android:fontFamily="@font/poppins_semiBold"
    android:gravity="center_horizontal"
    />
</LinearLayout>

<androidx.recyclerview.widget.RecyclerView
    android:layout_width="match_parent"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="20dp"
    android:background="@color/white"
    android:id="@+id/appointment_recycler"
    android:layout_marginRight="20dp"
    android:layout_height="match_parent">

</androidx.recyclerview.widget.RecyclerView>
```

## ➤ AllAppointmentForDoctor.java

```
package com.example.caredentapp;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayoutManager
nager;
import
androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.os.Bundle;
```

```

import android.util.AttributeSet;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;

public class AllAppointmentAkshay extends AppCompatActivity {
    DatabaseReference database;
    RecyclerView recyclerView;
    AkshayAdapter akshayAdapter;
    ArrayList<UserAkshay> list;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_all_appointment_akshay);
        getSupportActionBar().hide();

        recyclerView =
                findViewById(R.id.akshay_recycler);
        database =
                FirebaseDatabase.getInstance().getReference("Akshay");
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
    }

    list = new ArrayList<>();
    akshayAdapter = new AkshayAdapter(this, list);
    recyclerView.setAdapter(akshayAdapter);
}

recyclerView.setAdapter(akshayAdapter);

database.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        for (DataSnapshot dataSnapshot:snapshot.getChildren()){
            UserAkshay userAkshay =
                    dataSnapshot.getValue(UserAkshay.class);
            list.add(userAkshay);
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});

akshayAdapter.notifyDataSetChanged();
}

```

## ➤ UserAkshay.java

```

package com.example.caredentapp;

public class UserAkshay {

    String name,date,reason,doctor;

    public String getName() {
        return name;
    }

    public String getDate() {
        return date;
    }

    public String getReason() {
        return reason;
    }
}

```

```

        public String getDoctor() {
            return doctor;
        }
    }

    ➤ OTP_Verification_Page.xml

    package com.example.caredentapp;

    import androidx.annotation.NonNull;
    import
    androidx.appcompat.app.AppCompatActivity;

    import android.content.Intent;
    import android.os.Bundle;
    import android.text.Editable;
    import android.text.TextWatcher;
    import android.view.View;
    import android.widget.Button;
    import android.widget.EditText;
    import android.widget.Toast;

    import
    com.google.android.gms.tasks.OnCompleteListener;
    import com.google.android.gms.tasks.Task;
    import com.google.firebase.auth.AuthResult;
    import
    com.google.firebaseio.auth.FirebaseAuth;
    import
    com.google.firebaseio.auth.PhoneAuthCredential;
    import
    com.google.firebaseio.auth.PhoneAuthProvider;

    public class OTP_Verification_Page extends
    AppCompatActivity {

        Button otp_submit_btn;
        String mobiledata,getbackendotp;
        EditText
        num1,num2,num3,num4,num5,num6;

        @Override
        protected void onCreate(Bundle
        savedInstanceState) {
            super.onCreate(savedInstanceState);
    }

    setContentType(R.layout.activity_otp_verificat
    ion_page);
    getSupportActionBar().hide();

    otp_submit_btn =
    findViewById(R.id.submit_button);
    mobiledata =
    getIntent().getStringExtra("mobilenumber");
    getbackendotp =
    getIntent().getStringExtra("backendotp");

    num1 = findViewById(R.id.firstnumber);
    num2 =
    findViewById(R.id.secondnumber);
    num3 = findViewById(R.id.thirdnumber);
    num4 = findViewById(R.id.fournumber);
    num5 = findViewById(R.id.fivenumber);
    num6 = findViewById(R.id.sixnumber);

    otp_submit_btn.setOnClickListener(new
    View.OnClickListener() {
        @Override
        public void onClick(View view) {

            //startActivity(new
            Intent(OTP_Verification_Page.this,Doctor1Das
            h.class));

            sendotp();
        }
    });

    numbertomove();
}

public void sendotp(){

if(!num1.getText().toString().trim().isEmpty()
&&
!num2.getText().toString().trim().isEmpty() &&
!num3.getText().toString().trim().isEmpty() &&
!num4.getText().toString().trim().isEmpty() &&
!num5.getText().toString().trim().isEmpty() &&
!num6.getText().toString().trim().isEmpty())
}

```



```

        num3.requestFocus();
    }

}

@Override
public void afterTextChanged(Editable
editable) {
}

});

num3.addTextChangedListener(new
TextWatcher() {
    @Override
    public void
beforeTextChanged(CharSequence
charSequence, int i, int i1, int i2) {
}

    @Override
    public void
onTextChanged(CharSequence charSequence,
int i, int i1, int i2) {

if(!charSequence.toString().trim().isEmpty()){
    num4.requestFocus();
}
}

    @Override
    public void afterTextChanged(Editable
editable) {
}

});

num4.addTextChangedListener(new
TextWatcher() {
    @Override
    public void
beforeTextChanged(CharSequence
charSequence, int i, int i1, int i2) {
}

}

@Override
public void afterTextChanged(Editable
editable) {
}

}

public void
onTextChanged(CharSequence charSequence,
int i, int i1, int i2) {
}

    if(!charSequence.toString().trim().isEmpty())){
        num5.requestFocus();
    }
}

@Override
public void afterTextChanged(Editable
editable) {
}

});

num5.addTextChangedListener(new
TextWatcher() {
    @Override
    public void
beforeTextChanged(CharSequence
charSequence, int i, int i1, int i2) {
}

    @Override
    public void
onTextChanged(CharSequence charSequence,
int i, int i1, int i2) {
}

}

    if(!charSequence.toString().trim().isEmpty())){
        num6.requestFocus();
    }
}

@Override
public void afterTextChanged(Editable
editable) {
}

}
});
```

## ➤ Logo anim.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<set
    xmlns:android="http://schemas.android.com/
    apk/res/android">
    <scale
        android:fromXScale="0"
        android:fromYScale="0"
        android:toXScale="1"
        android:toYScale="1"
        android:pivotY="50%"
        android:pivotX="50%"
        android:duration="1000"
    />
    <alpha
        android:fromAlpha="0.6"
        android:toAlpha="1.0"
        android:duration="1000"
    />
</set>

```

## ➤ SMS Code:

```

?     private void sendSMS() {
    if
(ContextCompat.checkSelfPermission(BookApp
ointment.this,
Manifest.permission.SEND_SMS) ==
PackageManager.PERMISSION_GRANTED){
    String phone
=et_phone.getText().toString();
    String name =
et_name.getText().toString();
    smsShow = "Your Appointment Has been
Successfully Booked on " + dateString;

    if (!name.isEmpty() &&
!smsShow.isEmpty()){
        SmsManager smsManager =
SmsManager.getDefault();

        smsManager.sendTextMessage(phone,null,sm
sShow,null,null);
        Toast.makeText(this, "SMS Sent",
Toast.LENGTH_SHORT).show();
    }else {
        Toast.makeText(this, "ENTER Number
and Message", Toast.LENGTH_SHORT).show();
    }
}else{

```

```

ActivityCompat.requestPermissions(BookAppoi
ntment.this, new
String[]{Manifest.permission.SEND_SMS},100);
    }
}

```

## ➤ DoctorProfile.xml Code:

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/
    res/android"

    xmlns:app="http://schemas.android.com/apk/res-
    auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Doctor1"
    android:orientation="vertical"
    android:background="@color/white"
    >

<ScrollView

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    >

<LinearLayout

```

```

    android:layout_width="match_parent"
    android:layout_margin="15dp"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    >
<LinearLayout
    android:layout_width="170dp"
    android:layout_height="170dp"
    android:layout_marginTop="50dp"
    android:gravity="center"
    android:layout_marginLeft="12dp"
    android:layout_marginBottom="50dp"
    android:orientation="vertical">
<de.hdodenhof.circleimageview.CircleImageView
    android:layout_width="170dp"
    android:layout_height="170dp"
    android:src="@drawable/akshayprof"
></de.hdodenhof.circleimageview.CircleImageVie
w>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_marginTop="50dp"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    >
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dr. Akshay"
        android:textColor="@color/black"
        android:textSize="20dp"
        android:textStyle="bold"
        android:layout_marginTop="58dp"
        />
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DENTIST"
        android:textSize="17dp"
        android:textColor="@color/black"
        />
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BDS (Bachelor of Dental
Surgery)"
        android:textColor="@color/black"
        />
</LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="0dp"

```

```
    android:background="#FAF9F6"
    android:orientation="vertical">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SPECIALITY"
    android:textSize="20dp"
    android:layout_marginLeft="20dp"
    android:textStyle="bold"
    android:textColor="#0089ED"
/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="EndoSopic"
    android:padding="5dp"
    android:layout_gravity="center"
    android:background="@drawable/custom_radio_s
elector"
    android:textSize="20dp"
    android:textStyle="bold"
    android:textColor="#0089ED"
/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="EXPIRIENCE"
    android:layout_marginTop="25dp"
    android:background="#FAF9F6"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="EXPIRIENCE"
        android:textSize="20dp"
        android:layout_marginLeft="20dp"
        android:textStyle="bold"
        android:textColor="#0089ED"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:padding="5dp"
        android:layout_gravity="center"
        android:background="@drawable/custom_radio_s
elector"
        android:textSize="20dp"
        android:textStyle="bold"
        android:textColor="#0089ED"
    />

```

```

        android:text="Hello, Myself Akshay and I am
        Dentist."
        android:textSize="20dp"
        android:textStyle="bold"
        android:textColor="#000000"
    />
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:text="akshay@gmail.com"
        android:textSize="20dp"
        android:textStyle="bold"
        android:textColor="#000000"
    />
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:text="Expirience - 5+ years"
        android:textSize="20dp"
        android:textStyle="bold"
        android:textColor="#000000"
    />
</LinearLayout>
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:background="#FAF9F6"
        android:orientation="vertical">
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CONTACTUS"
        android:textSize="20dp"
        android:layout_marginLeft="20dp"
        android:textStyle="bold"
    />
</LinearLayout>
</LinearLayout>
</ScrollView>
</LinearLayout>

```

➤ **DoctorProfile.xml Code:**

```
package com.example.caredentapp;
```

```

import
    androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class Doctor1 extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle
        savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_doctor1);

        getSupportActionBar().hide();

    }

}

```

➤ **ContactUs.xml Code:**

```

<?xml version="1.0" encoding="utf-
8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/
    apk/res/android"

    xmlns:app="http://schemas.android.com/apk
    /res-auto"

    xmlns:tools="http://schemas.android.com/to
    ols"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:background="@color/white"
        tools:context=".ContactUs">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"

```

```

        android:orientation="horizontal"
        android:background="#82CAFF"
    >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Contect Us"
        android:textSize="25dp"
        android:layout_marginTop="5dp"
        android:textColor="@color/black"
        android:layout_marginLeft="50dp"
        android:layout_marginRight="50dp"
        android:layout_gravity="center"

        android:fontFamily="@font/poppins_semibol
        d"
        android:gravity="center_horizontal"
    />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
    >

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Name"
            android:inputType="textPersonName"
            android:fontFamily="@font/poppins"
            android:backgroundTint="#079BF5"
            android:id="@+id/contactus_name"
            android:layout_marginLeft="50dp"
            android:layout_marginRight="50dp"
            android:textColorHint="#D3D3D3"
            android:textColor="@color/black"
            android:textSize="20dp"
            android:layout_marginTop="35dp"
        />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="phone"
            android:hint="Contact Number"
            android:fontFamily="@font/poppins"

```

```

        android:id="@+id/contactus_number"
        android:backgroundTint="#079BF5"
        android:layout_marginLeft="50dp"
        android:layout_marginRight="50dp"
        android:textColorHint="#D3D3D3"
        android:textColor="@color/black"
        android:textSize="20dp"
        android:layout_marginTop="35dp"
    />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Message/Query"
    android:inputType="textLongMessage"
    android:fontFamily="@font/poppins"
    android:backgroundTint="#079BF5"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:id="@+id/contactus_message"
    android:textColorHint="#D3D3D3"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:layout_marginTop="35dp"
/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:background="@drawable/all_buttons"
    android:fontFamily="@font/poppins_semibold"
    android:gravity="center_horizontal"
/>

    android:id="@+id/submit_contactus_btn"
    android:layout_marginRight="100dp"
    android:text="SUBMIT"
    android:layout_marginTop="100dp"
/>
</LinearLayout>

</LinearLayout>

```

## ➤ **ContactUs.java Code:**

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

```

```

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:background="@color/white"
        tools:context=".ContactUs">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:orientation="horizontal"
            android:background="#82CAFF"
        >
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Contect Us"
                android:textSize="25dp"
                android:layout_marginTop="5dp"
                android:textColor="@color/black"
                android:layout_marginLeft="50dp"
                android:layout_marginRight="50dp"
                android:layout_gravity="center"
                android:fontFamily="@font/poppins_semibold"
                android:gravity="center_horizontal"
            />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
        >
            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Name"
                android:inputType="textPersonName"
                android:fontFamily="@font/poppins"

```

```

        android:backgroundTint="#079BF5"
        android:id="@+id/contactus_name"
        android:layout_marginLeft="50dp"
        android:layout_marginRight="50dp"
        android:textColorHint="#D3D3D3"
        android:textColor="@color/black"
        android:textSize="20dp"
        android:layout_marginTop="35dp"
    />
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="phone"
    android:hint="Contact Number"
    android:fontFamily="@font/poppins"
    android:id="@+id/contactus_number"
    android:backgroundTint="#079BF5"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:textColorHint="#D3D3D3"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:layout_marginTop="35dp"
/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Message/Query"
    android:inputType="textLongMessage"
    android:fontFamily="@font/poppins"
    android:backgroundTint="#079BF5"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:id="@+id/contactus_message"
    android:textColorHint="#D3D3D3"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:layout_marginTop="35dp"
/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:background="@drawable/all_buttons"
    >

```

```

        android:id="@+id/submit_contactus_btn"
        android:layout_marginRight="100dp"
        android:text="SUBMIT"
        android:layout_marginTop="100dp"
    />

```

</LinearLayout>

</LinearLayout>

## ➤ Recycler View Code:

```

    <?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
    android:orientation="vertical"
    android:background="@color/white"
    android:layout_height="match_parent"
    tools:context=".ContactUsDataForDoctor">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/contactus_recycler"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="20dp"
        android:background="@color/white">

    </androidx.recyclerview.widget.RecyclerView
    >

```

```

</LinearLayout>
➤ DoctorDashboard.xml
    ?      <?xml version="1.0" encoding="utf-
8"?>
    <LinearLayout
        xmlns:android="http://schemas.android.com/
        apk/res/android"

        xmlns:app="http://schemas.android.com/apk
        /res-auto"
        android:orientation="vertical"

        xmlns:tools="http://schemas.android.com/to
        ols"
        android:background="@color/white"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".Doctor1Dash">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Welcome Akshay"
            android:textSize="50dp"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:layout_marginTop="50dp"
            android:gravity="center"
        />
        <ImageButton
            android:layout_width="250dp"
            android:layout_height="200dp"

            android:id="@+id/appointment_btn_for_
            doctor_akshay"

            android:background="@drawable/appoinmen
            t_logo"
            android:layout_gravity="center"
            android:layout_marginTop="50dp"
        />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Check new Appointments"
            android:textColor="@color/black"
            android:textSize="25dp"

```

```

            android:layout_marginTop="20dp"
        />
        <ImageButton
            android:layout_width="250dp"
            android:layout_height="200dp"
            android:id="@+id/contactuspage"

            android:background="@drawable/appoinmen
            t_logo"
            android:layout_gravity="center"
            android:layout_marginTop="50dp"
        />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Contact Us Page"
            android:textColor="@color/black"
            android:textSize="25dp"
            android:layout_marginTop="20dp"
        />
    </LinearLayout>

```

➤ **DoctorDashboard.java Code:**

```

package com.example.caredentapp;

import
    androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.CompoundButton;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.ToggleButton;

public class Doctor1Dash extends
    AppCompatActivity {

    ImageButton
        check_appointments,contactuspage;

    @Override
    protected void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_doctor1_da
sh);
getSupportActionBar().hide();

check_appointments =
findViewById(R.id.appointment_btn_for_doc
or_akshay);
contactuspage =
findViewById(R.id.contactuspage);

contactuspage.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new
Intent(Doctor1Dash.this,ContactUsDataForDo
ctor.class));
    }
});

check_appointments.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new
Intent(Doctor1Dash.this,AllAppointmentAksh
ay.class));
    }
});

```

## ➤ AdapterClass Code:

```

package com.example.caredentapp;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;

```

```

public class PatientAdapter extends
RecyclerView.Adapter<PatientAdapter.Patient
Holder> {

    Context context;
    ArrayList<Patients_name> list;

    public PatientAdapter(Context context,
    ArrayList<Patients_name> list) {
        this.context = context;
        this.list = list;
    }

    @NonNull
    @Override
    public PatientHolder
onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
    View v =
LayoutInflater.from(context).inflate(R.layout.a
ppointment_item,parent,false);
    return new PatientHolder(v);
}

@Override
public void onBindViewHolder(@NonNull
PatientHolder holder, int position) {

    Patients_name patients_name =
list.get(position);

    holder.patient_name.setText(patients_name.
getName());

    holder.patient_reason.setText(patients_name
.getReason());

    holder.patient_doctor.setText(patients_name
.getDoctor());

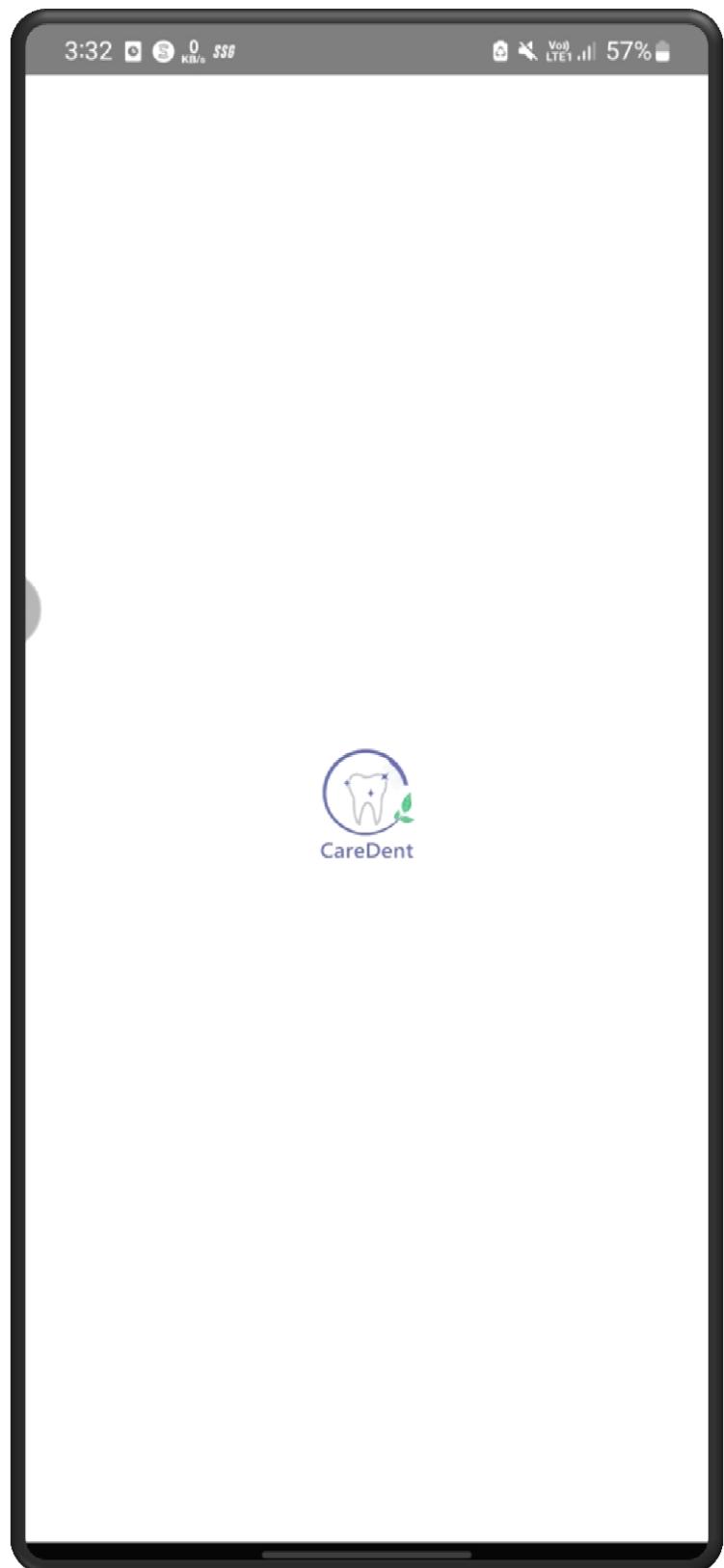
}

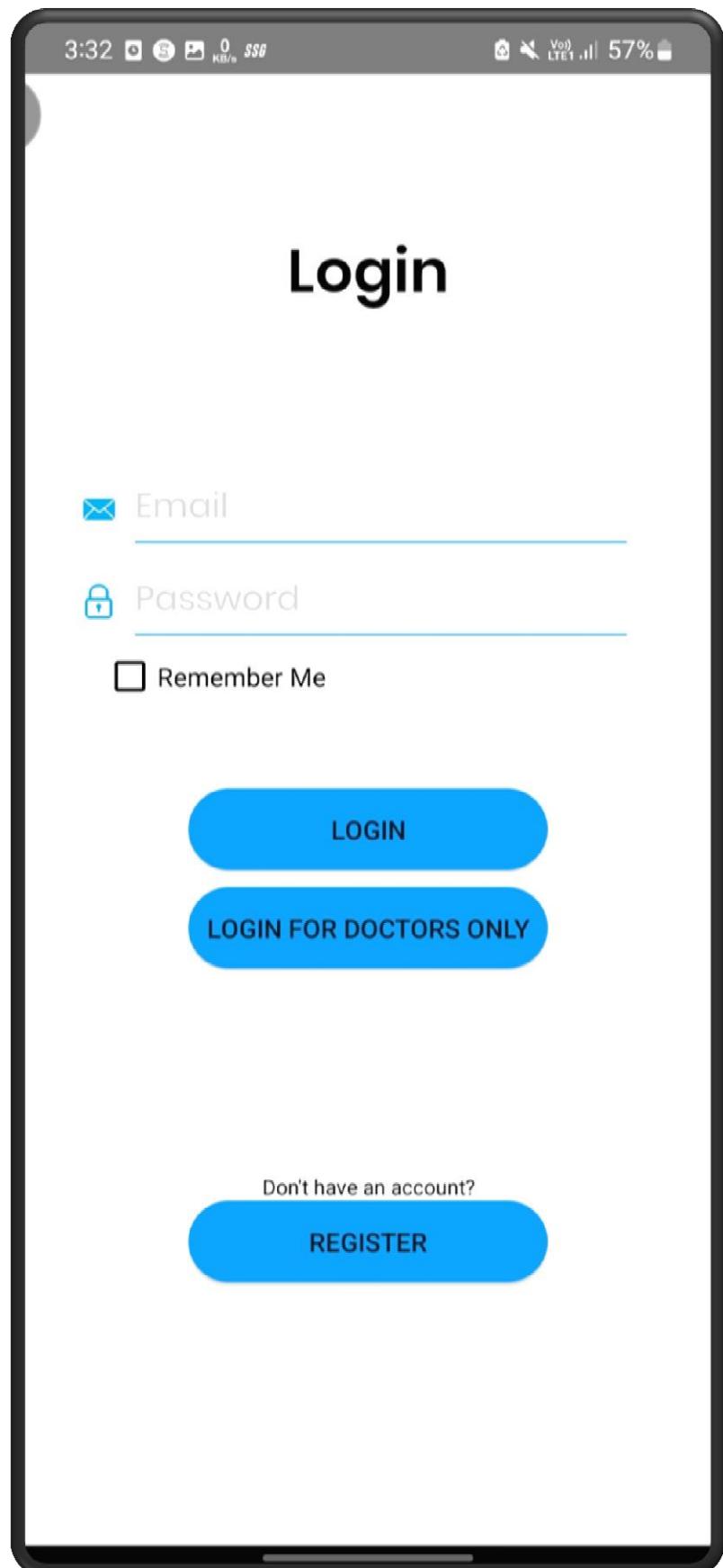
@Override
public int getItemCount() {
    return list.size();
}

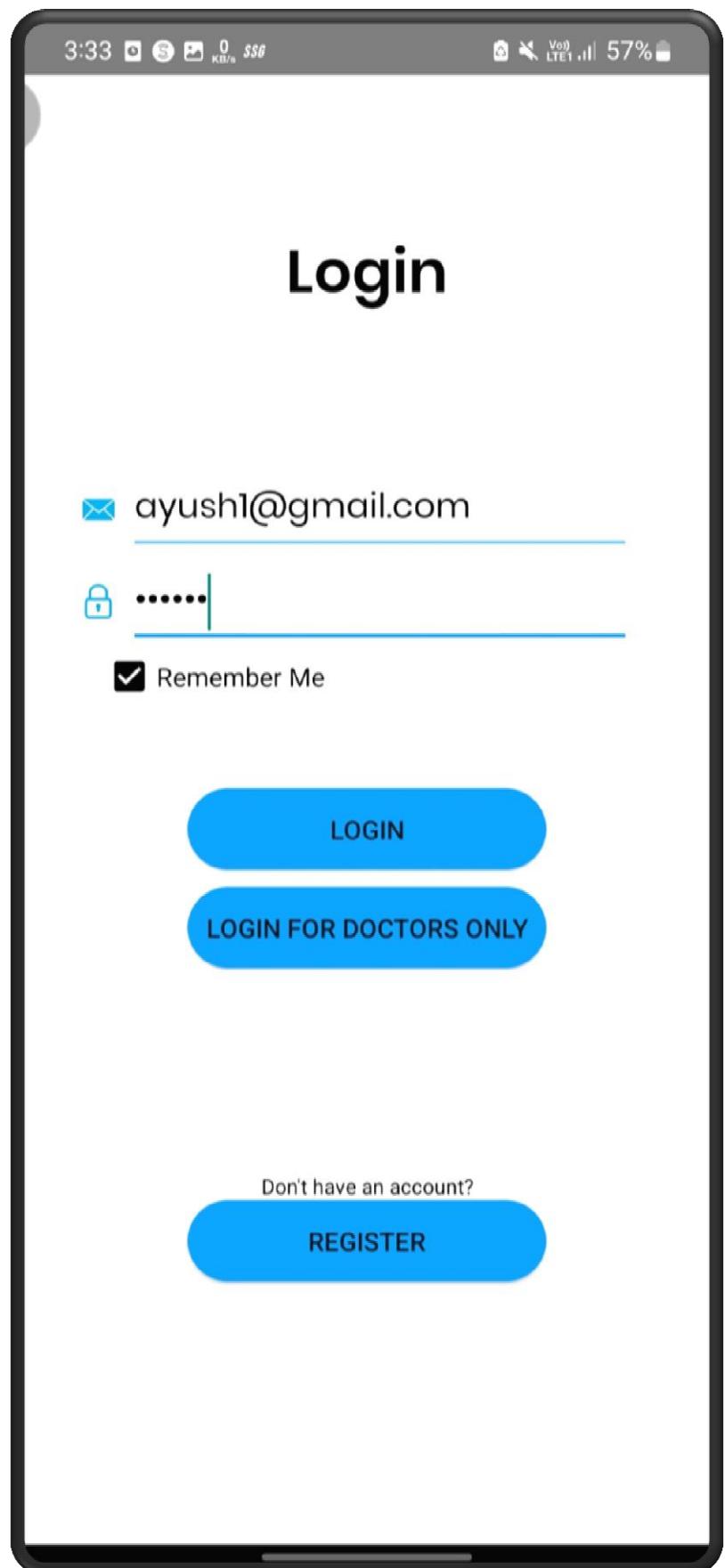
```

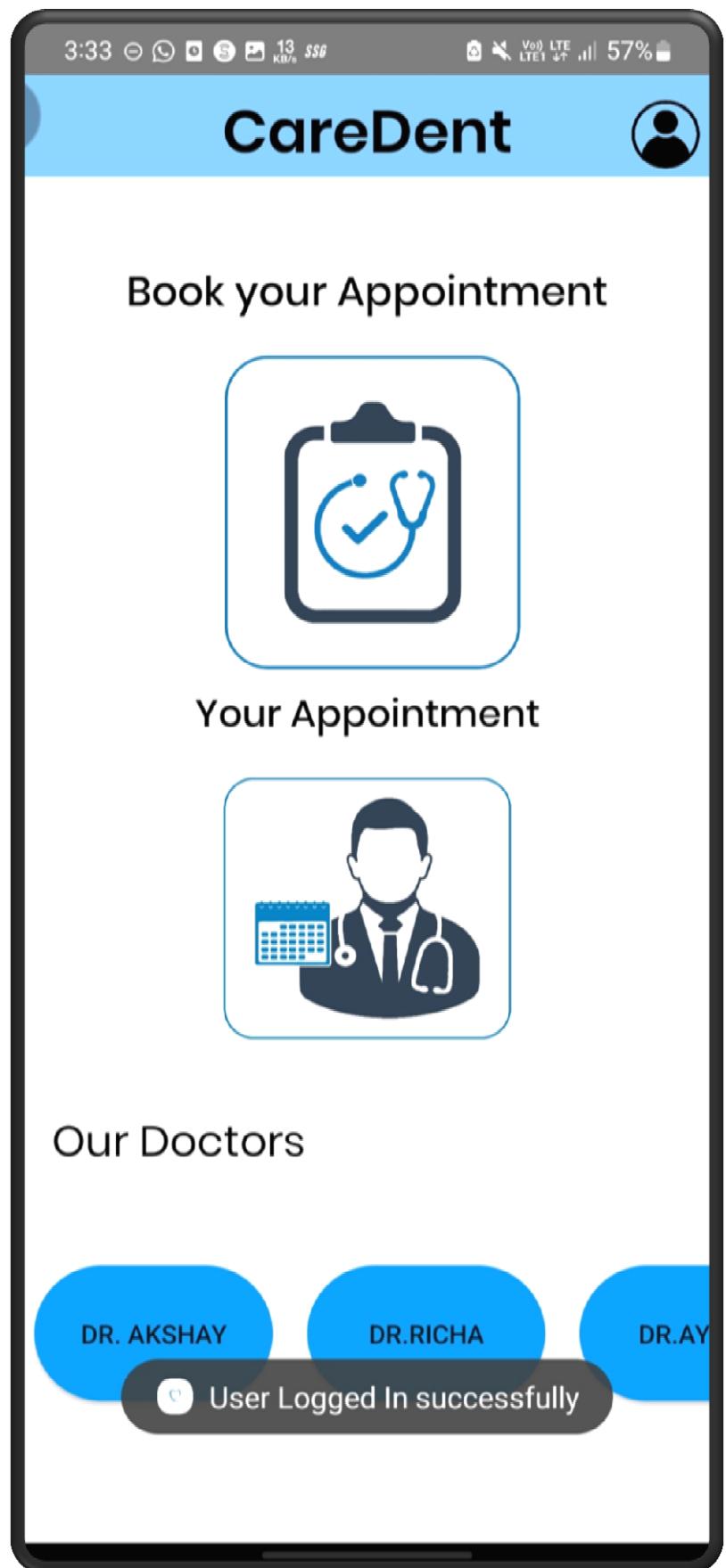
```
public static class PatientHolder extends  
RecyclerView.ViewHolder{  
  
    TextView  
    patient_name,patient_reason,patient_doctor;  
  
    public PatientHolder(@NonNull View  
itemView) {  
        super(itemView);  
  
        patient_name =  
itemView.findViewById(R.id.disp_name_to_d  
octor);  
        patient_reason =  
itemView.findViewById(R.id.disp_reason_to_  
doctor);  
        patient_doctor =  
itemView.findViewById(R.id.disp_doctor_to_d  
octor);  
  
    }  
}  
  
}
```

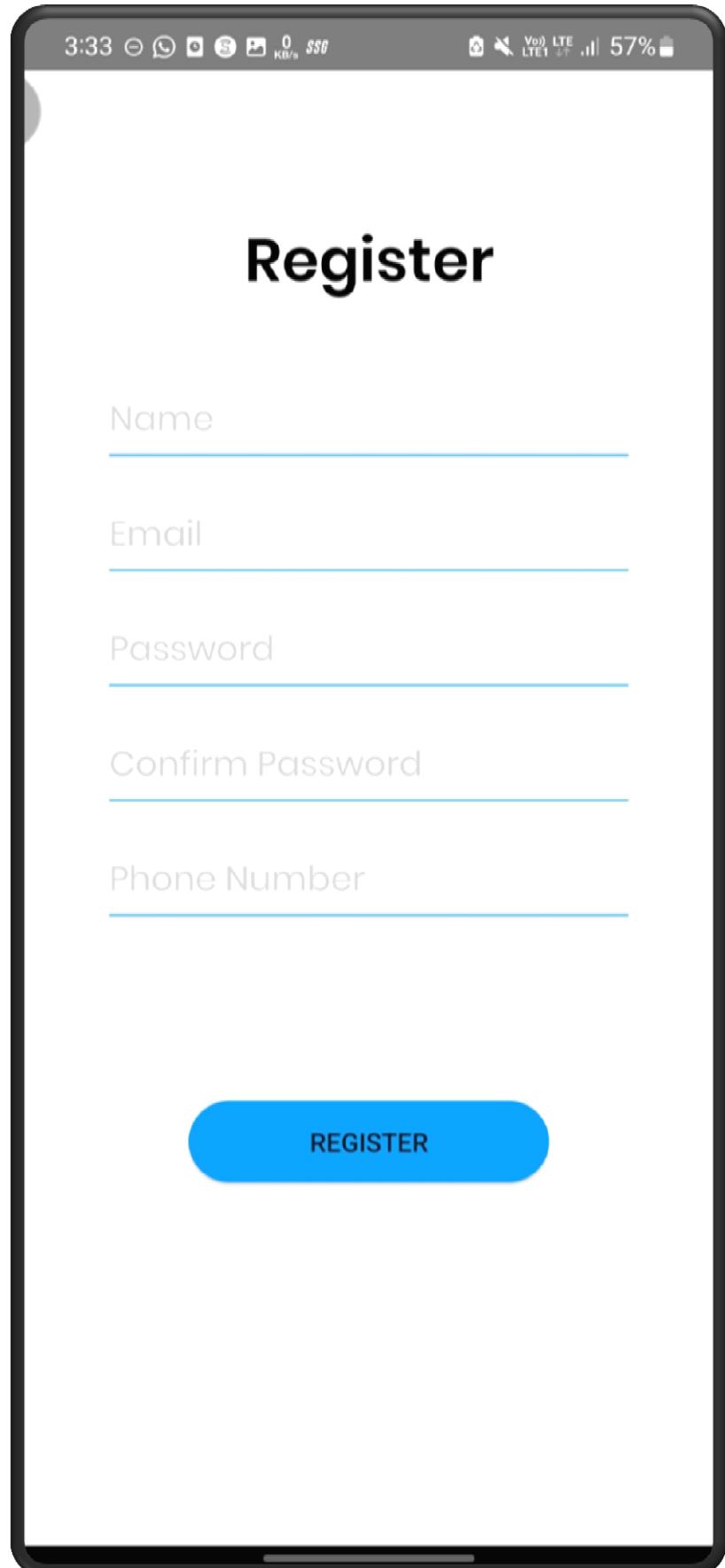
➤ **Output(snapshots) :**

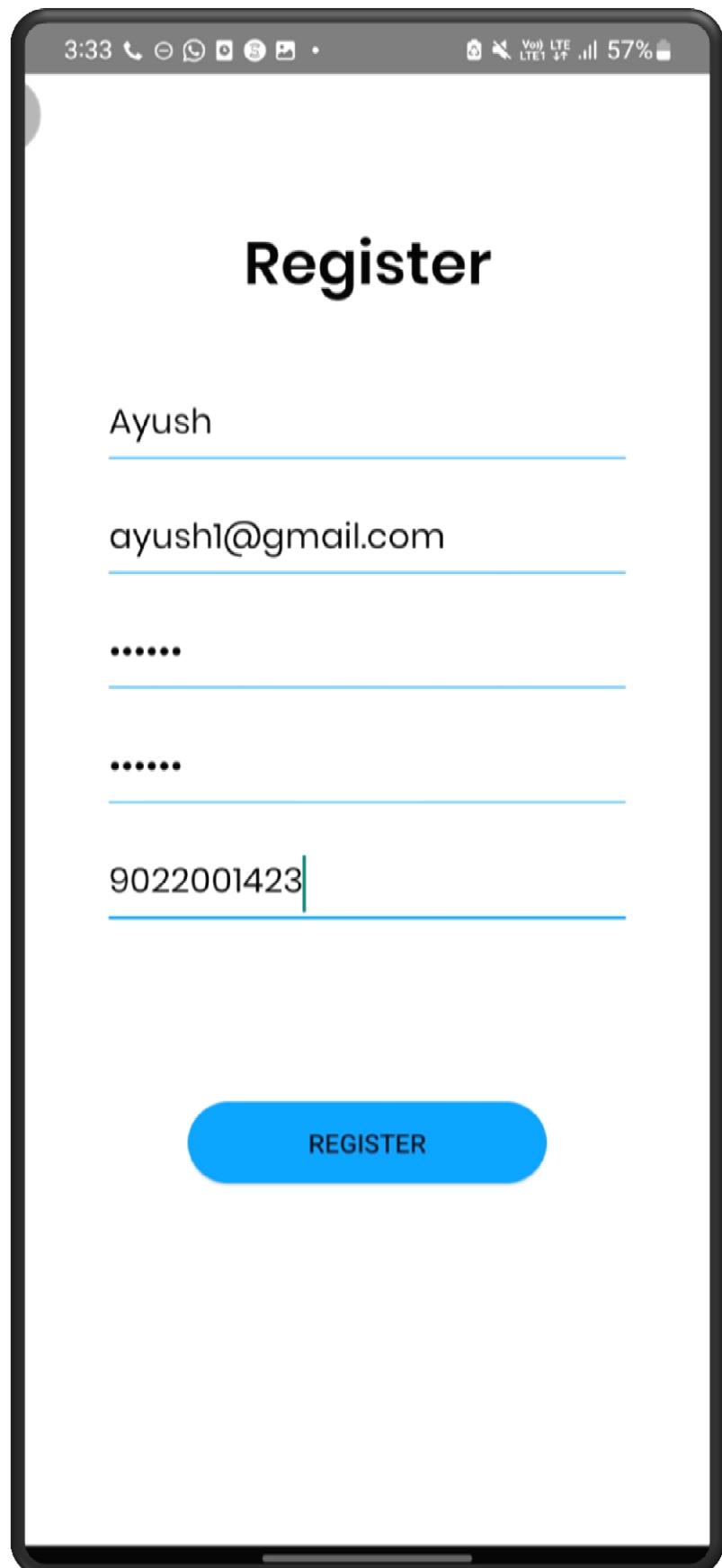












# Register

Ayush

ayush1@gma

.....

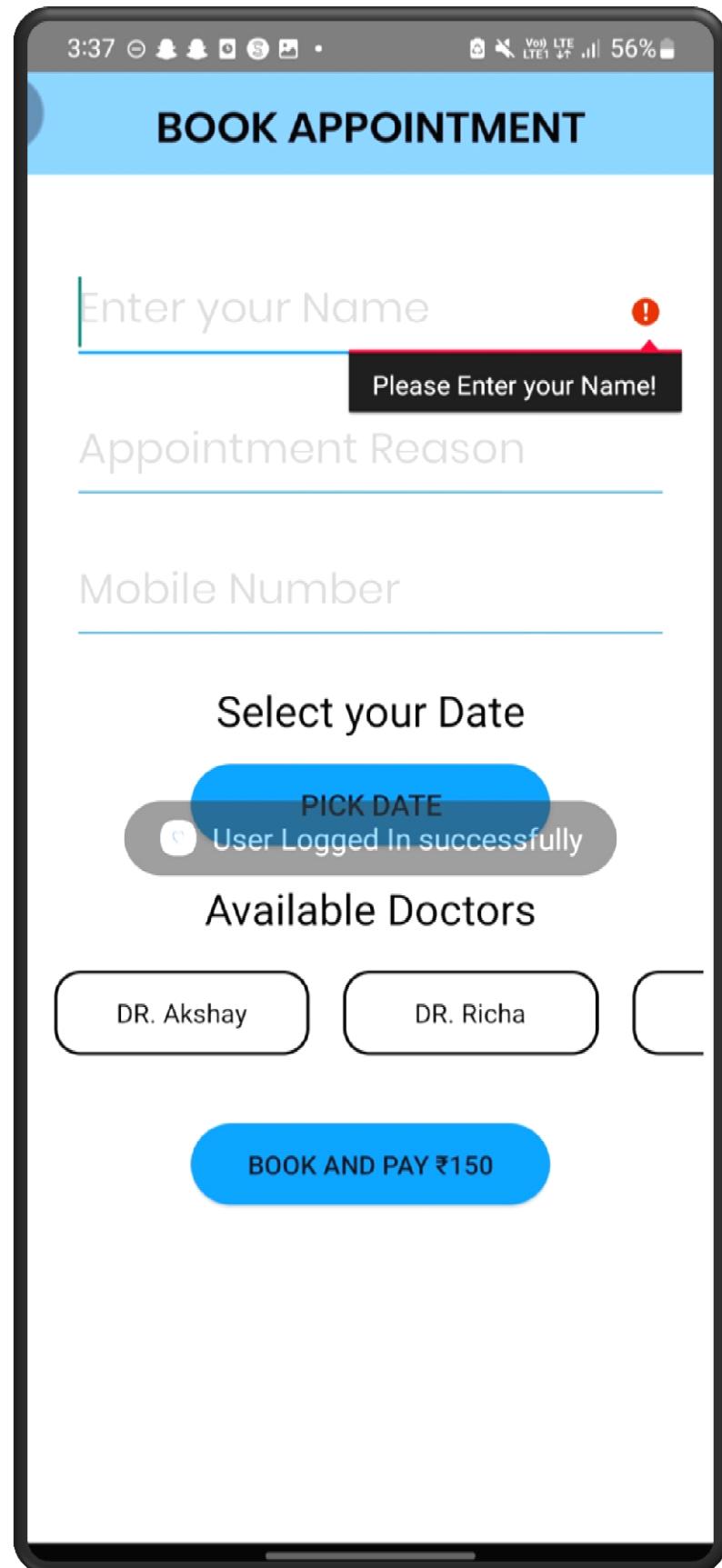
.....

9022001423

REGISTER



Registration Error  
The email address is badly formatted.



## BOOK APPOINTMENT

Ayush

Ch

2023

Sun, 30 Apr

90

< April 2023 >

M T W T F S S

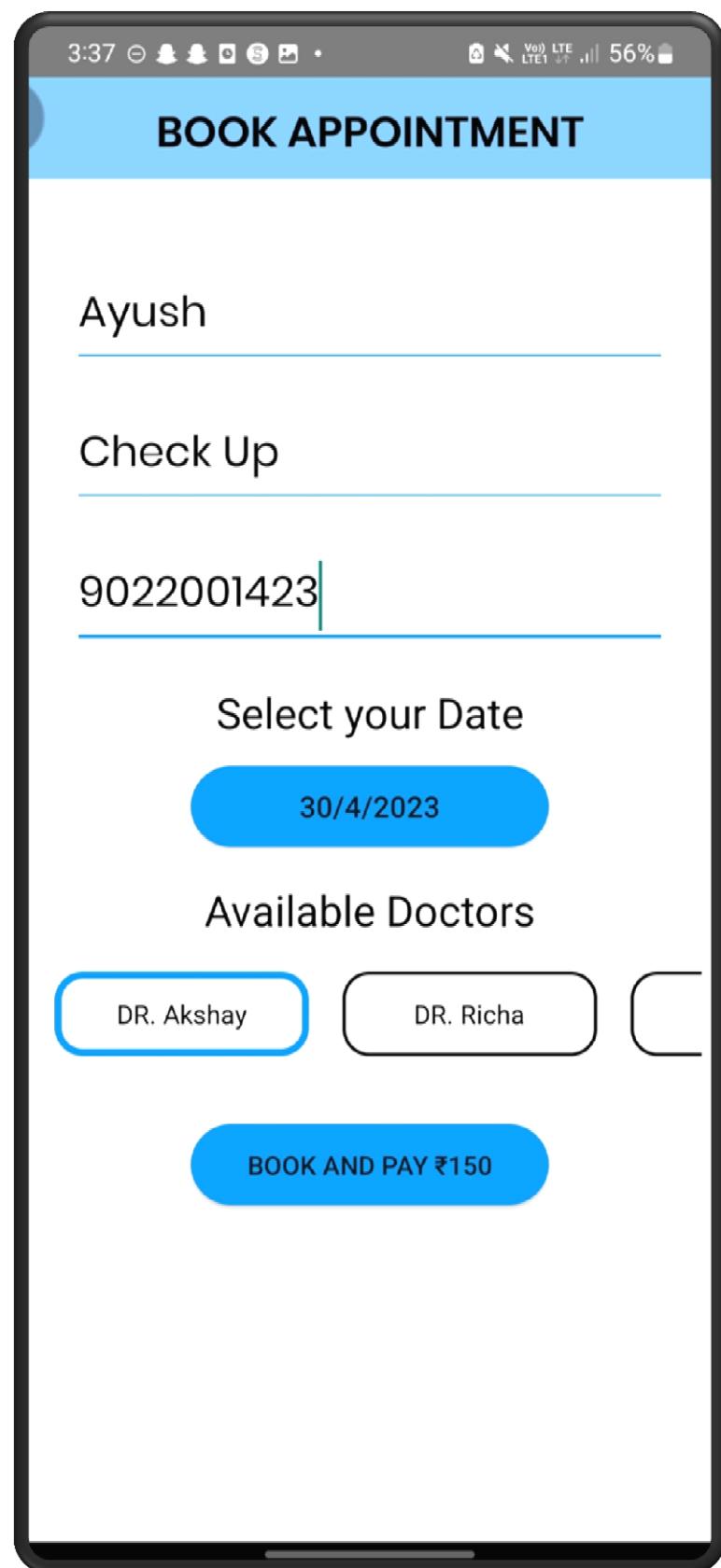
3 4 5 6 7 8 9

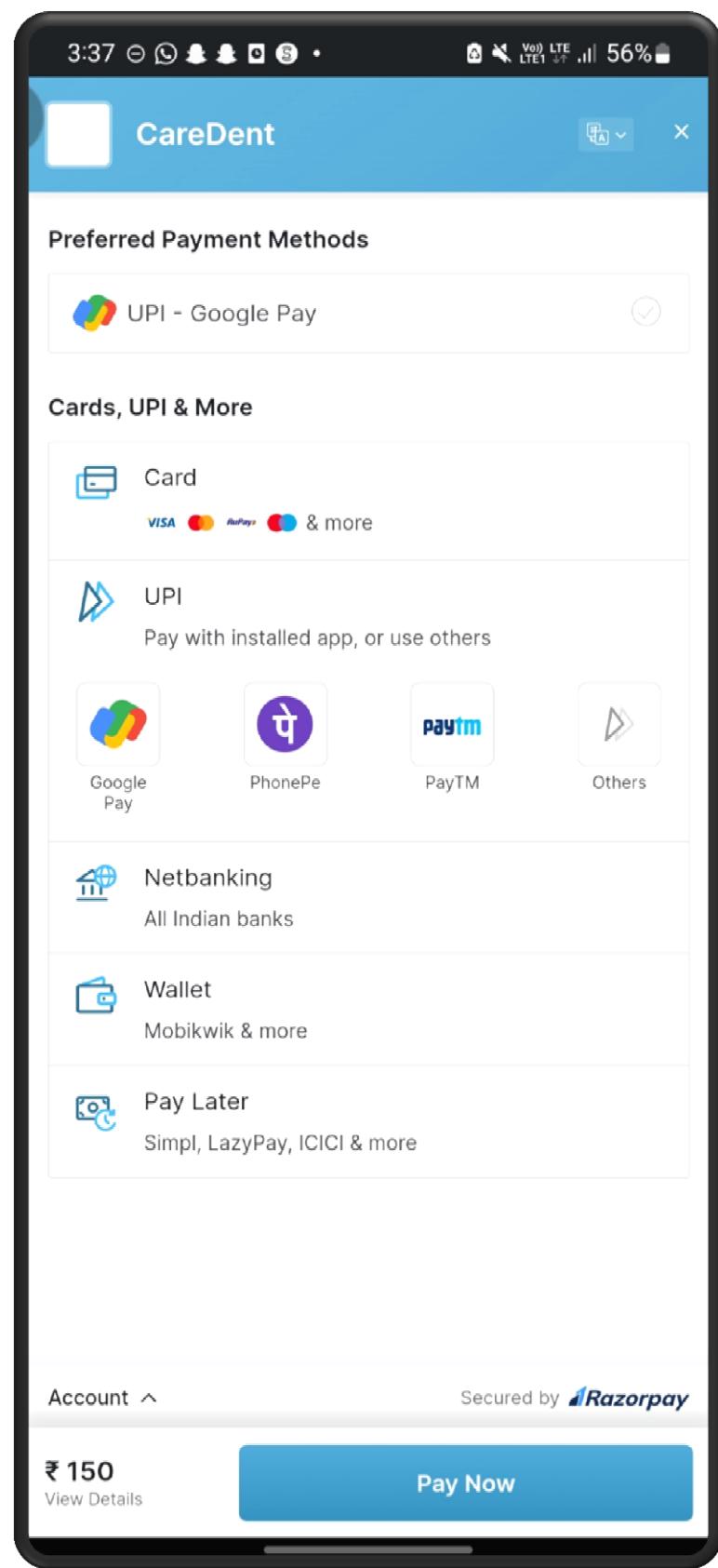
10 11 12 13 14 15 16

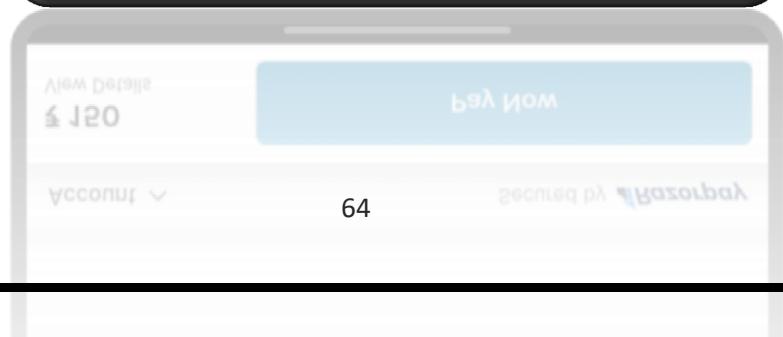
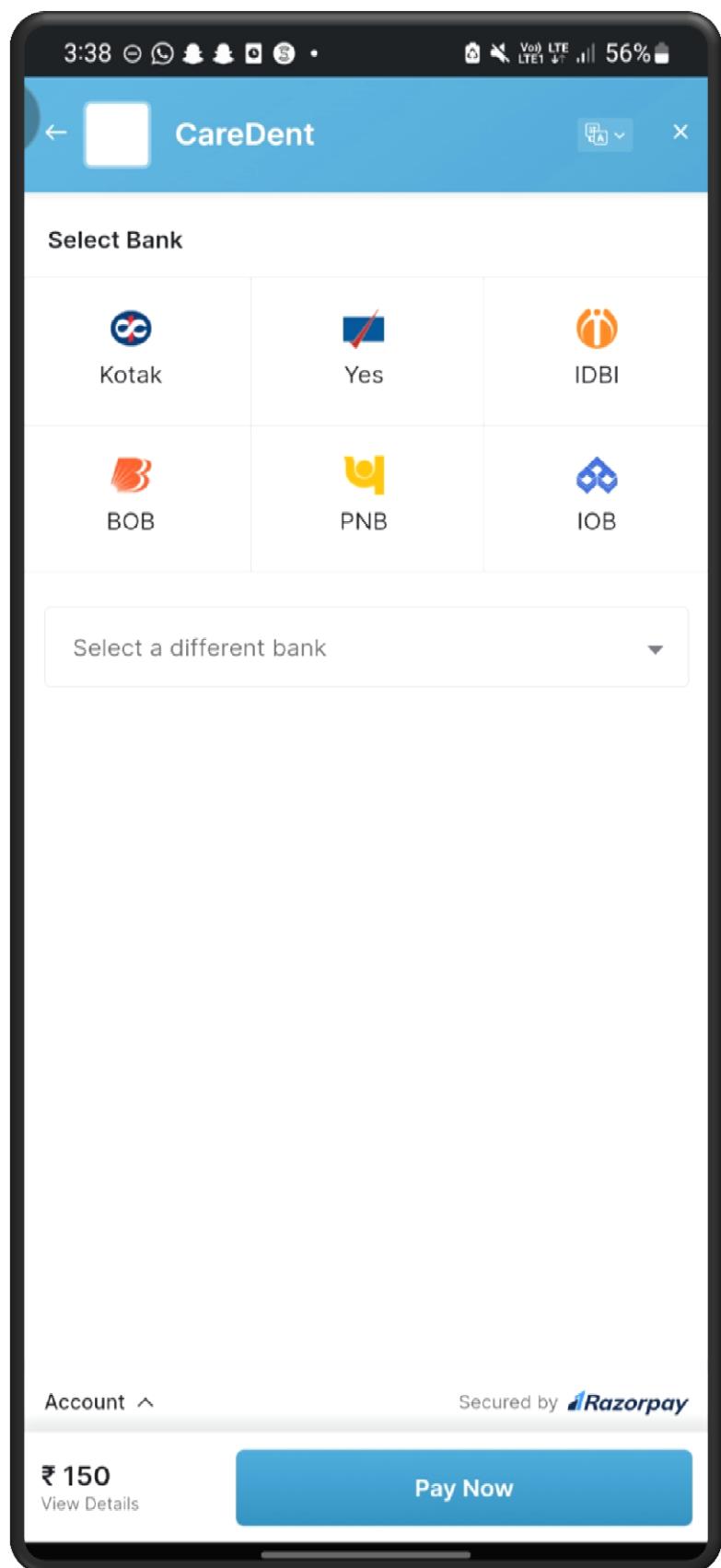
17 18 19 20 21 22 23

24 25 26 27 28 29 30

CANCEL OK







3:38 ☺ 🔍 📲 📩 📩

56% 🔋

# Appointments

Your Name : Ayush

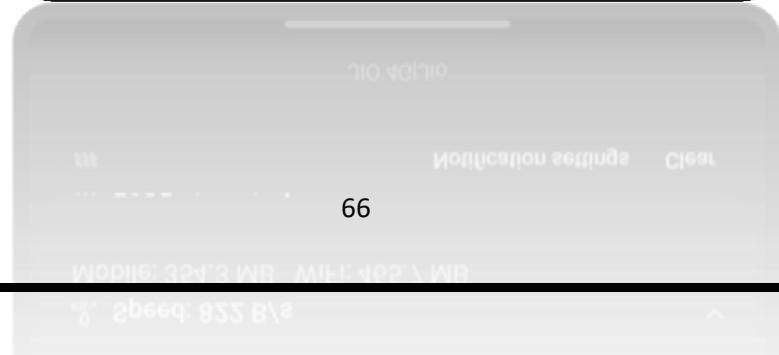
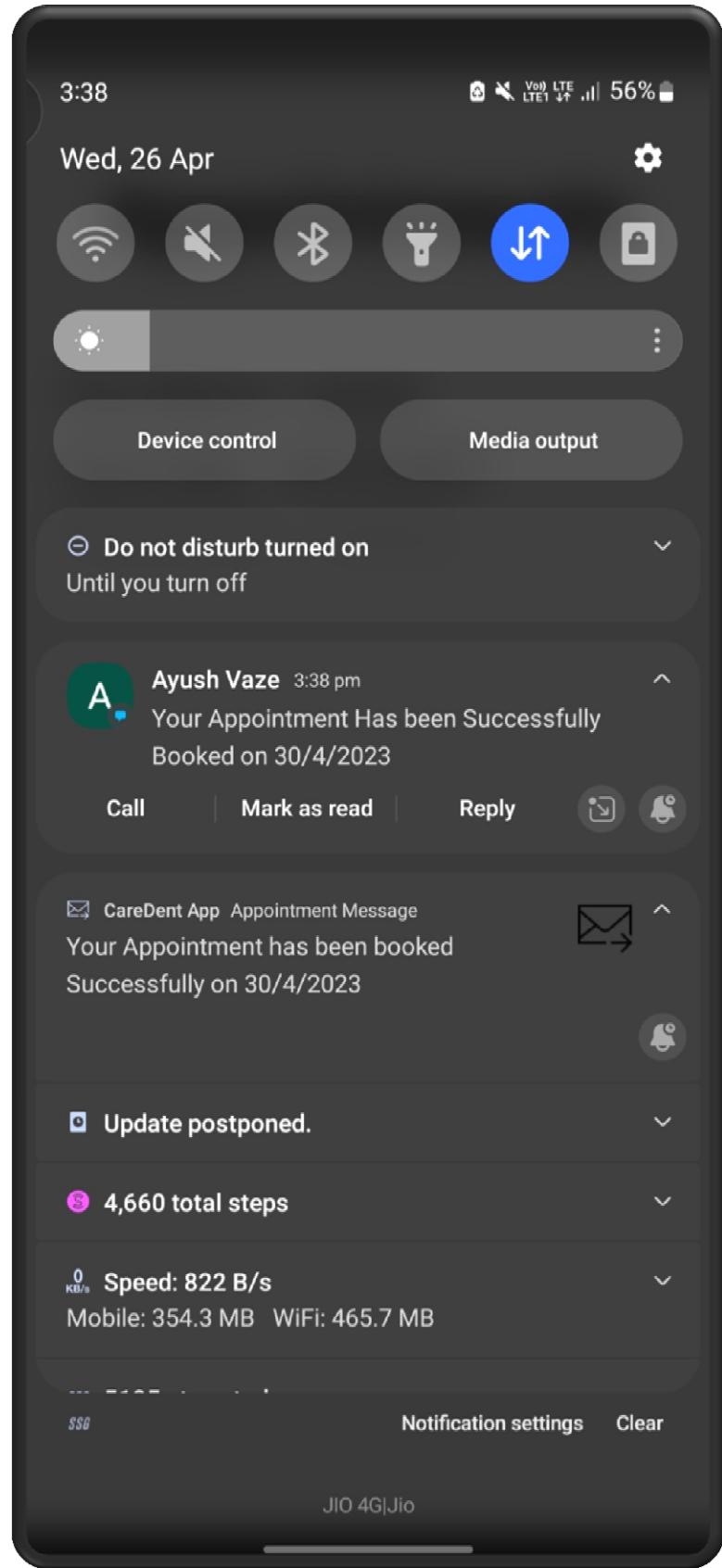
Your Date : 30/4/2023

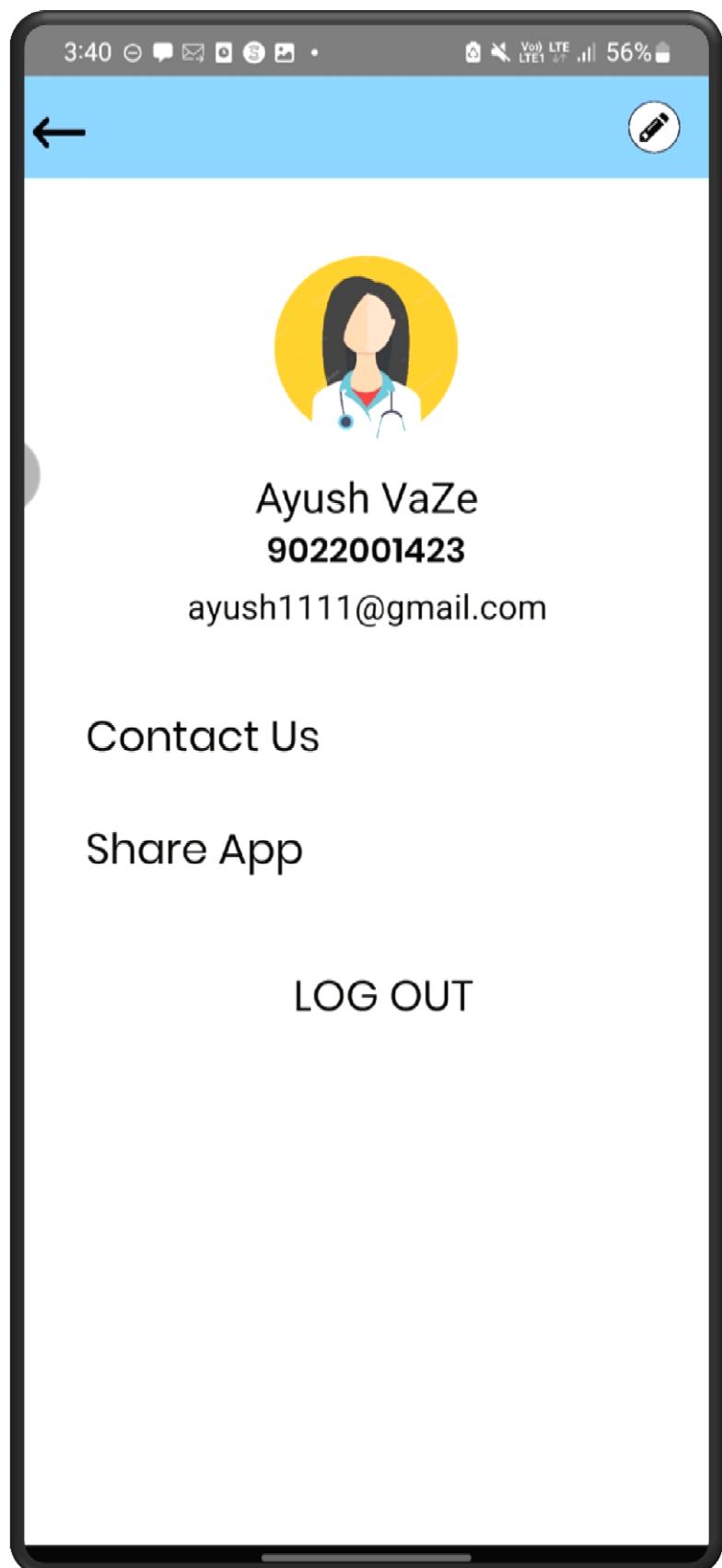
Your Doctor : Akshay

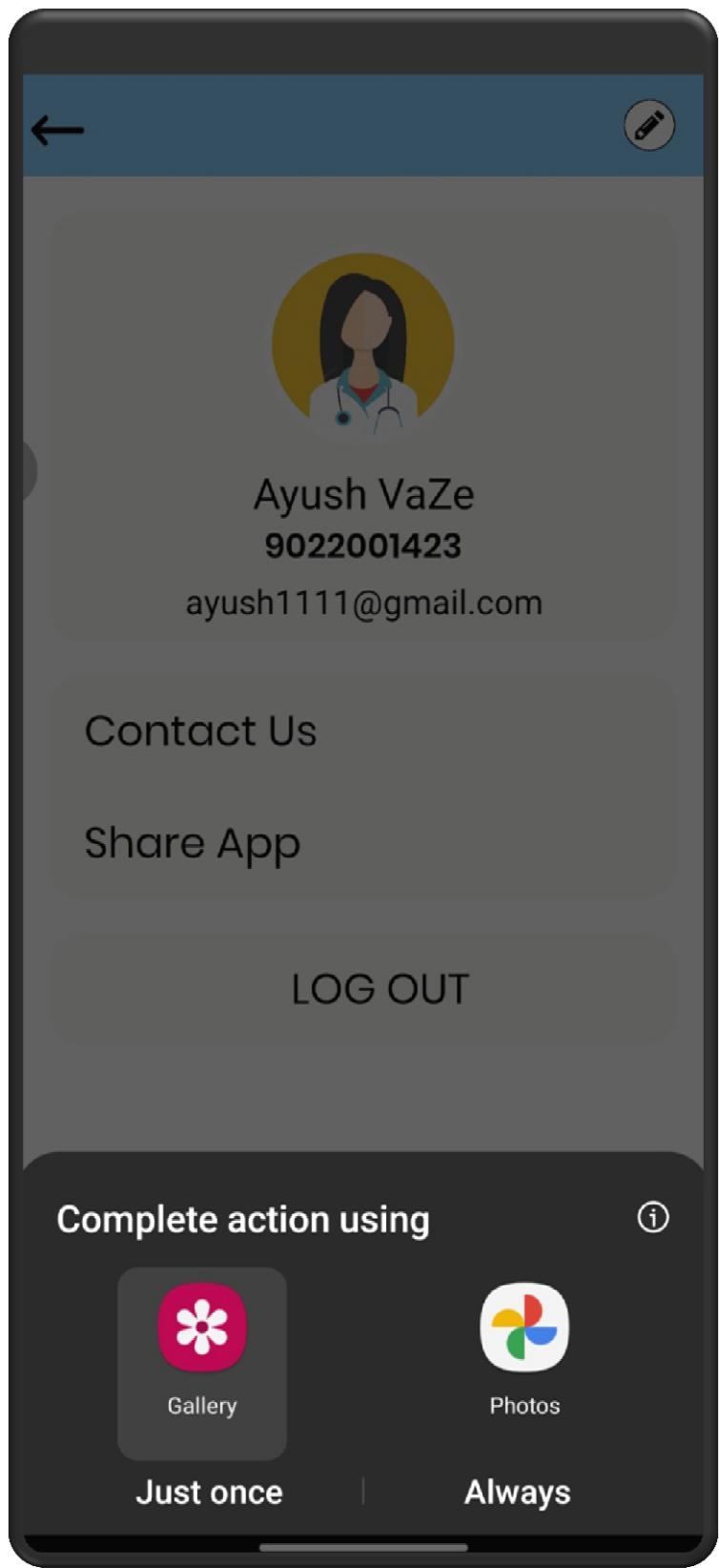
 SMS Sent successful

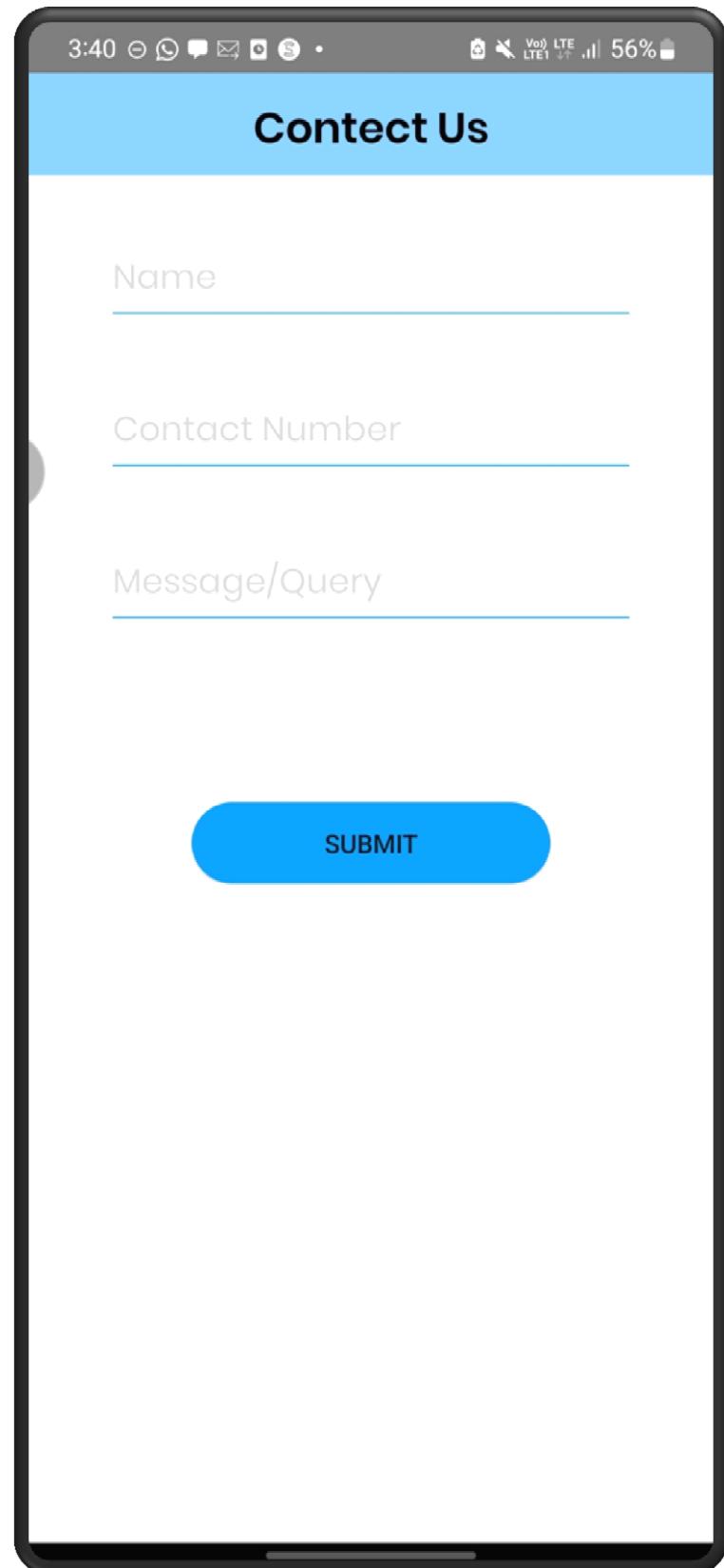
65

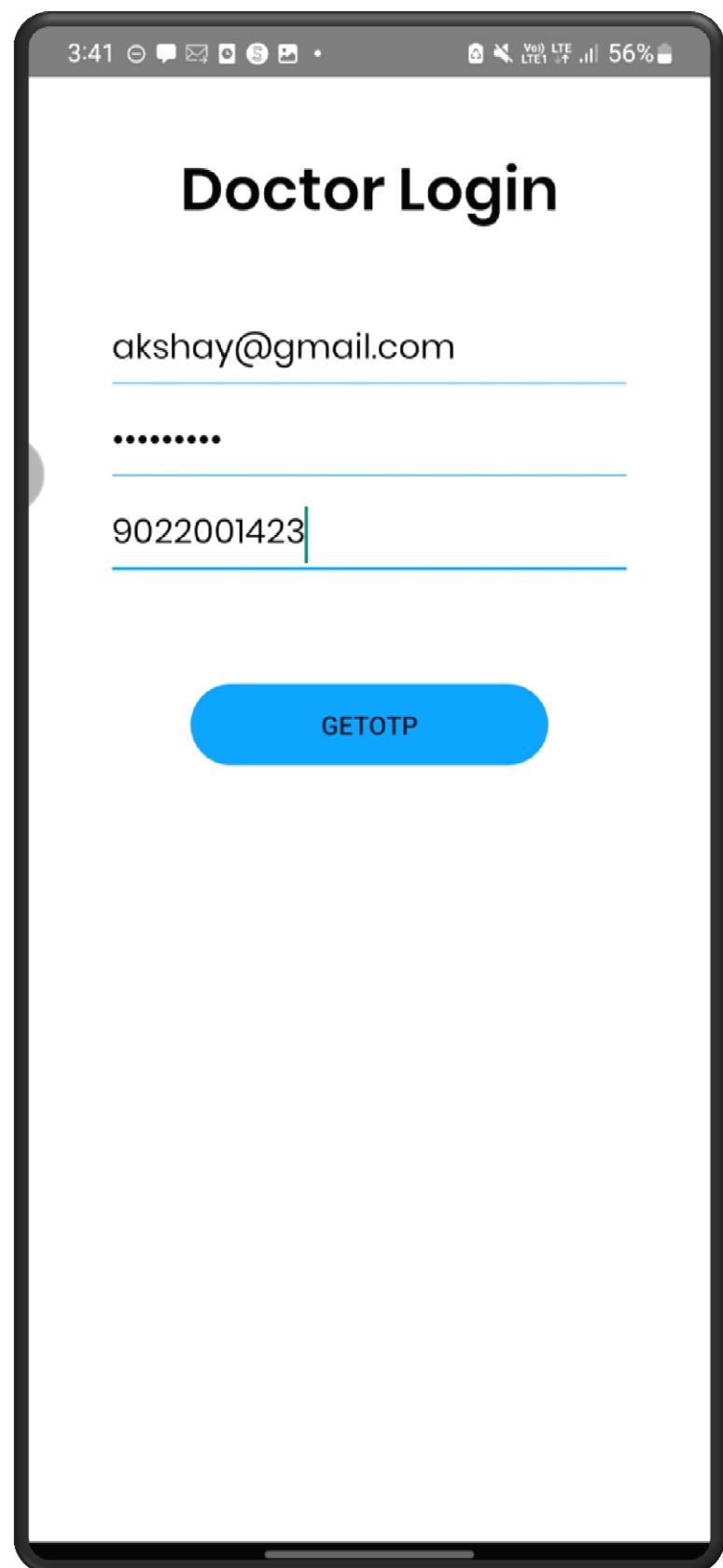
Unread SMS

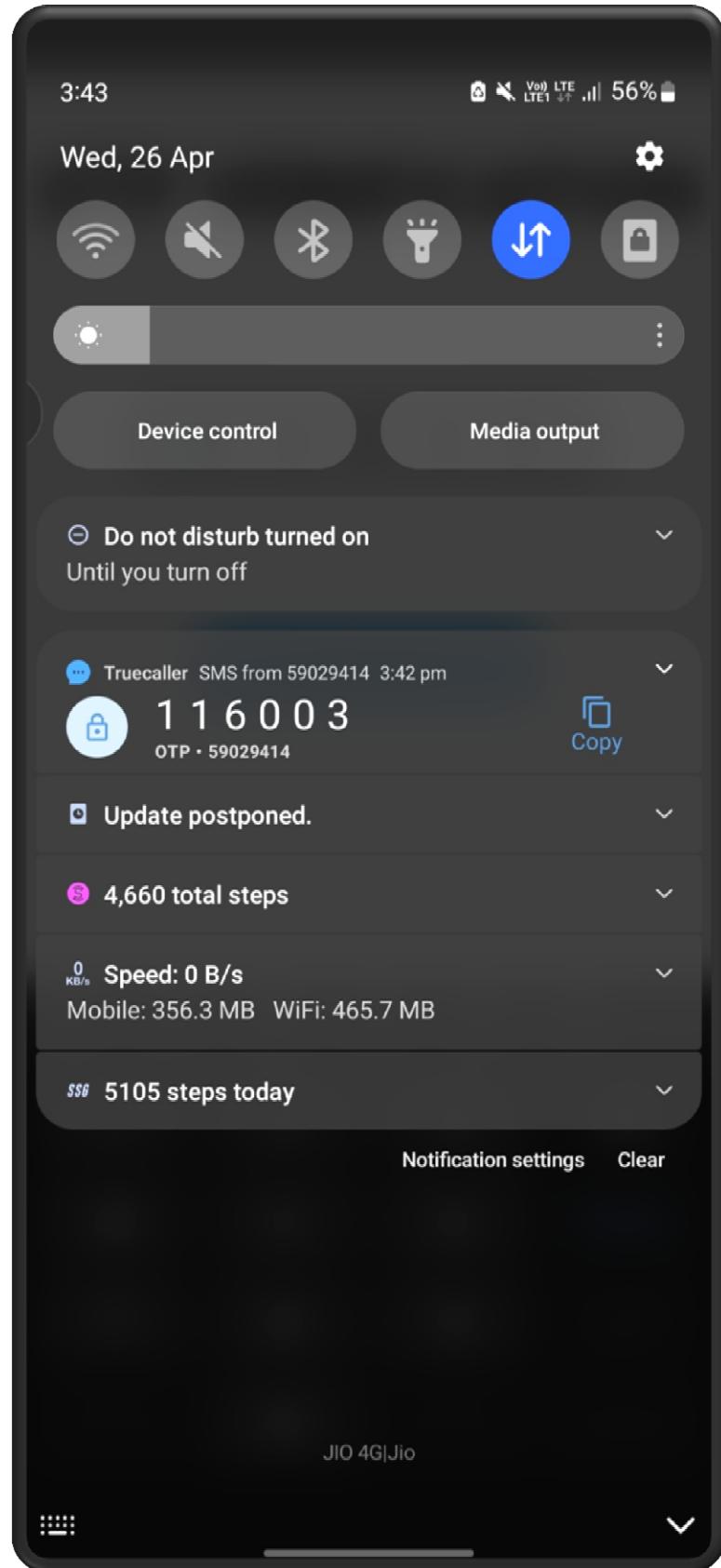












3:43 ⊜ 9 KB/s SSB

VoLTE 56%

## Appointments Akshay

**Doctor :** Akshay

**Name :** Ayush

**Reason :** New Check Up 2

**Time :** 29/4/2023

**Doctor :** Akshay

**Name :** Ayush

**Reason :** new check up

**Time :** 30/4/2023

**Doctor :** Akshay

**Name :** Ayush

**Reason :** Check up

**Time :** 28/4/2023

**Doctor :** Akshay

**Name :** Ayush

**Reason :** check kr in a

**Time :** 30/4/2023

**Doctor :** Akshay

**Name :** Ayush

**Reason :** Check Up

**Time :** 30/4/2023

**Doctor :** Akshay

Doctor : Akshay

Time : 30/4/2023

Reason : Check Up

Ayush : Name

3:43 ☺ 🔍 📲 📸 📹 📷

Yoo! LTE1 4G 56%

**Name :** ayush

**Number :** 6699663355

**Message :** new message

**Name :** ayush

**Number :** 3434649

**Message :** jsisje

**Name :** Ayush

**Number :** 9022001423

**Message :** Hello

**Name :** Ayush

**Number :** 9022001423

**Message :** Hello 2

**Name :** Ayush

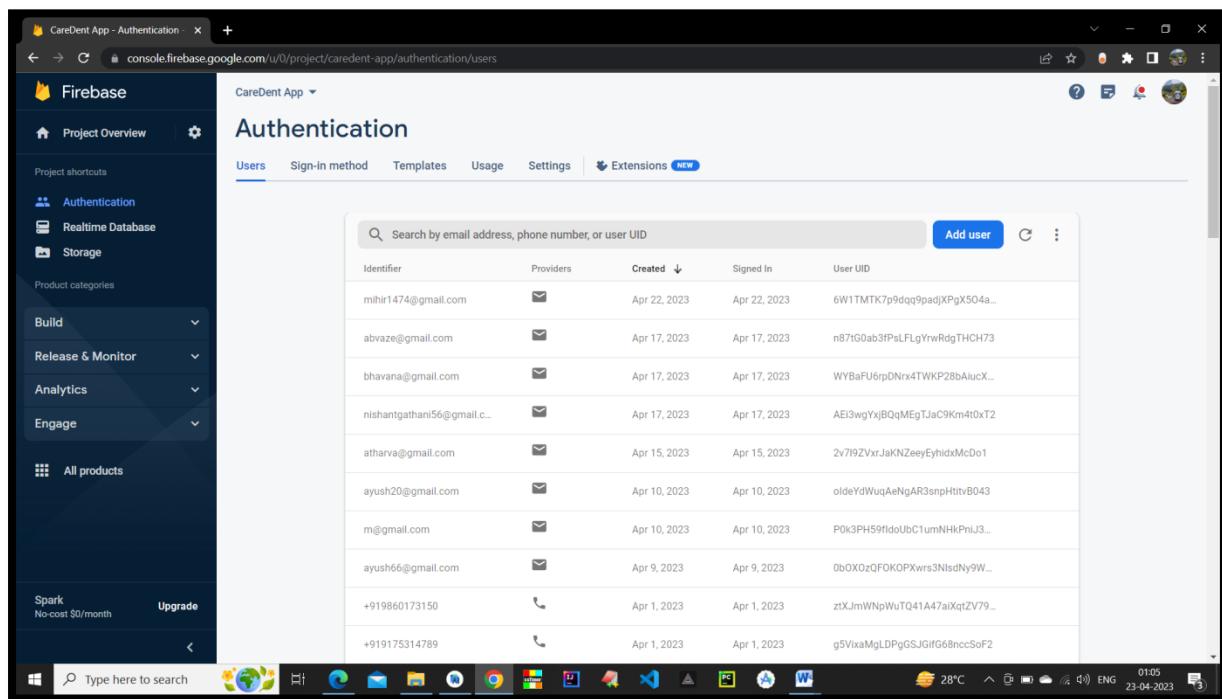
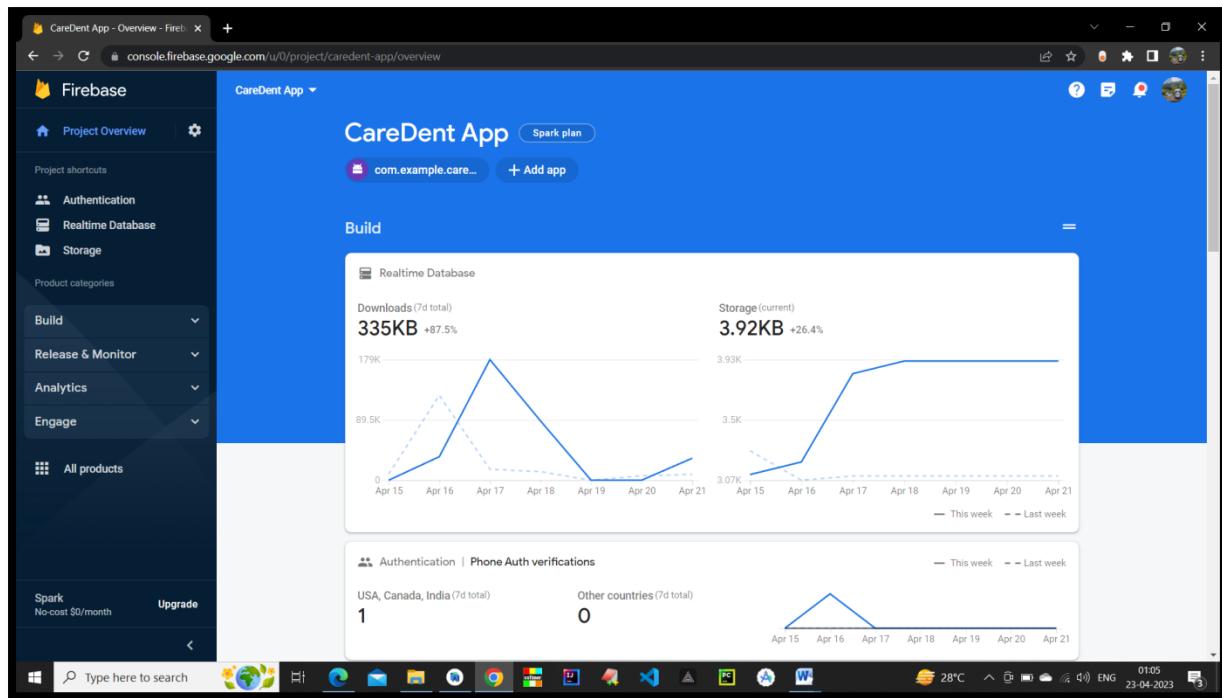
**Number :** 9022001423

**Message :** Hello 3

**Name :** xyz

**Number :** 1234567890

**Message :** hello 4



The screenshot shows the Firebase Realtime Database console for the project 'CareDent App'. The left sidebar includes 'Project Overview', 'Authentication', 'Realtime Database' (selected), and 'Storage'. The main area displays the database structure at the root level:

```
https://caredent-app-default.firebaseio.com/
- Akshay
  - ContactUs
  - New Appointment
  - Profile
  - Richa
  - Vaze
```

A red warning bar at the top states: "⚠ Your security rules are defined as public, so anyone can steal, modify, or delete data in your database". Below the tree view, it says "Database location: United States (us-central1)".

This screenshot shows the same Firebase Realtime Database console, but with more data added under the 'Akshay' node:

```
https://caredent-app-default.firebaseio.com/
- Akshay
  - -NRmzFU4raPq9QZA0a12
    - date: "31/3/2023"
    - doctor: "Akshay"
    - name: "akshay"
    - reason: "new data"
  - -NRn-D1ZSE1fRzEFwKpo
    - date: "31/3/2023"
    - doctor: "Akshay"
    - name: "Akshay"
```

The red warning bar and database location information remain the same.

The screenshot shows a web browser window with the KiviHealth website open. The search bar at the top has "smile design" entered. Below the search bar, there is a sidebar with dropdown menus for "Select Area" and "Select Speciality". The main content area displays a search result for "Smile Design in Mumbai" with one result listed: "Dr. Rahul Shetty". The doctor's profile includes a placeholder image, the name "Dr. Rahul Shetty", title "BDS", specialty "Dentist", location "Kalyan, Thane", and interest areas "Cosmetic/Aesthetic Surgery", "Smile Design", and "Invisible Braces". A "BOOK APPOINTMENT" button is visible. At the bottom of the page, there is a sidebar for "Doctor Login" and a footer with links for Case Studies, Testimonials, FAQs, Contact Us, Blog, and Privacy Policy, along with social media icons for Google+, Facebook, Twitter, and LinkedIn.

# **Chapter 6: TESTING AND VALIDATION**

## **6.1 Introduction**

Testing is a critical aspect of software development that ensures the quality and reliability of the software. It is a process of verifying that the software meets its requirements and specifications and performs as expected. Testing helps to identify defects, errors, and inconsistencies in the software and ensures that the software is free from bugs and meets the user's expectations.

This report aims to provide an introduction to testing and its importance in software development. It covers the basic concepts of testing, the different types of testing, and the benefits of testing.

## **6.2 Design of Test Cases and scenarios**

### **1.Login Page**

<b>Test Case ID</b>	<b>Test Case Objective</b>	<b>Input Data</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Status</b>
TC1	Checking the Email Format	abc.com	Email Address Badly Formatted	Email Address Badly Formatted	Pass
TC2	Checking the Password field	123	The Password is Invalid	The Password is Invlid	Pass
TC3	Putting Unregistered Email Address	abc@gamil.com	There is no User with this Email Address	There is no User with this Email Address	Pass
TC4	Username Field is Left Blank	-	Email connot be empty	Email connot be empty	Pass
TC5	Password Field is Left Blank	-	Password cannot be empty	Password cannot be empty	Pass

Table 1

## 2.Register Page

Test Case ID	Test Case Objective	Input Data	Expected Output	Actual Output	Status
TC1	Keeping Name field empty	-	Name connot be empty	Name connot be empty	Pass
TC2	Keeping Email field empty	-	Email connot be empty	Email connot be empty	Pass
TC3	Keeping Password field empty	-	Password cannot be empty	Password cannot be empty	Pass
TC4	Keeping Confirm Password field empty	-	Confirm Password doesn't match Password	Confirm Password doesn't match Password	Pass
TC5	Keeping Phone Number field empty	-	Phone Number cannot be empty	Phone Number cannot be empty	Pass
TC6	Checking Length of Phone Number	987654321	Enter Correct Phone Number	Enter Correct Phone Number	Pass
TC7	Giving Worng email Format	abc@gmail	Email Badly Foramtted	Email badly Formatted	Pass

Table 2

### 3.Book Appointment Page

Test Case ID	Test Case Objective	Input Data	Expected Output	Actual Output	Status
TC1	Keeping Name field empty	-	Name connot be empty	Name connot be empty	Pass
TC2	Keeping Reason field empty	-	Reason connot be empty	Reason connot be empty	Pass
TC3	Keeping Mobile No field empty	-	Mobile cannot be empty	Mobile cannot be empty	Pass
TC4	Keeping Date field empty	-	Select a Date	Select a Date	Pass
TC5	Not selecting a Doctor	-	Select a Doctor	Select a Doctor	Pass

Table 3

#### 4.Doctor Login Page

Test Case ID	Test Case Objective	Input Data	Expected Output	Actual Output	Status
TC1	Checking Email Field	abc@gmail.com	Worng Email	Wrong Email	Pass
TC2	Checking Email Field	Abc123	Wrong Password	Wrong Password	Pass
TC3	Checking Mobile Field	987654321	Enter Correct Number	Enter Correct Number	Pass
TC4	Chekking OTP Field	654321	Enter Correct OTP	Enter Correct OTP	Pass

Table 4

## **6.3 Validation**

Validation is an essential aspect of developing an appointment booking app to ensure that the app functions as intended and meets the needs of its users. Here are some examples of validation steps for an appointment booking app:

1. Usability testing: Conducting usability testing with a diverse group of users to ensure that the app is user-friendly and easy to navigate. This testing can include feedback on the layout, design, and overall usability of the app.
2. Performance testing: Testing the app's performance to ensure that it can handle the expected number of users and appointments. This testing can include load testing and stress testing to identify potential bottlenecks or issues with the app's performance.
3. Security testing: Testing the app's security features to ensure that user data is protected and secure. This testing can include vulnerability testing, penetration testing, and other security assessments.
4. Functional testing: Testing the app's functionality to ensure that all features work as intended. This testing can include testing individual features such as appointment scheduling, reminders, and calendar integration.
5. User acceptance testing: Conducting user acceptance testing with a group of users to ensure that the app meets their needs and expectations. This testing can include feedback on features, usability, and overall user experience.

By conducting these validation steps, developers can ensure that their appointment booking app meets the needs of its users, functions as intended, and provides a positive user experience.

## **6.4 Future Scope**

The future of doctor appointment systems is promising, as the healthcare industry continues to embrace technology and digital solutions. Here are some potential developments that could shape the future of doctor appointment systems:

1. Artificial Intelligence: The use of artificial intelligence (AI) could revolutionize the doctor appointment system, as it could help to automate various aspects of the appointment process, such as appointment scheduling, patient triage, and reminder notifications. AI-powered chatbots could also be used to handle patient inquiries and provide assistance.
2. Blockchain Technology: Blockchain technology could be used to enhance the security and privacy of doctor appointment systems, as it provides a decentralized and secure method of storing and sharing patient data.
3. Predictive Analytics: Predictive analytics could be used to anticipate patient needs and preferences, making it easier for doctors to schedule appointments and provide personalized care.

Overall, the future of doctor appointment systems looks bright, as technology continues to advance and transform the healthcare industry. With the right tools and strategies, doctor appointment systems can help to improve patient outcomes, reduce healthcare costs, and enhance the patient experience.

## **Chapter 7: CONCLUSION**

In conclusion, a doctor appointment system is an efficient and effective way of managing appointments between doctors and patients. It provides a convenient platform for patients to schedule appointments, receive reminders, while also allowing doctors to better organize their schedules and ensure that they provide timely and appropriate care to their patients. With the rise of technology and the increasing demand for healthcare services, a doctor appointment system is becoming increasingly necessary to improve the overall quality of healthcare delivery.

With the development of web based NHIS medical appointment booking system, patients are able to book and manage their own appointment with ease. They will be reminded of their appointments via SMS/email that will be promptly sent to them before their appointment date. The system itself also provides a quick view of their appointment at the Home page. These functions could indirectly help to reduce the number of missed medical appointments and patients no-show up for their appointment. Patients would be notified via SMS/email if their appointment were affected.

The system is mainly designed to facilitate appointment booking between the patient and the health personnel. Time will not be wasted on converting paper-based appointment record into electronic-based. The system further helps to reduces healthcare personnel workload by allowing them to generate medical reports easily. They could now maximize their competence and allocate more time to maximize service quality.

## Chapter 8 : REFERENCES

1. <https://developer.android.com/docs>
2. <https://developer.android.com/studio/write/firebase>
3. <https://firebase.google.com/docs/auth>
4. <https://razorpay.com/docs/payments/payment-gateway/android-integration/standard/build-integration/>
5. <https://kivihealth.com/for-doctors>
6. <https://developer.android.com/reference/android/telephony/SmsManager>