



**CA PROJECT REPORT**

On

**CARELINK**(Doctor Appointment App)

Submitted By:

**Yash Ubana**

Registration No: 12017732

Section: KO202

Couse Code: CSE227

Roll No: 61

Programme Name: B. Tech (CSE)

Under the Guidance Of

**Shubhita Ma'am**

School Of Computer Science and Engineering

Lovely Professional University

## DESCRIPTION

Introducing CareLink, your all-in-one solution for seamless doctor appointments, connecting patients, doctors, and administrators effortlessly. CareLink revolutionizes the way you manage your healthcare appointments, streamlining the process for patients, doctors, and administrators alike. With intuitive interfaces tailored for each role, CareLink ensures a smooth and efficient experience for everyone involved.

## FEATURES COVERED

1. **Firestore Authentication:** CareLink sets the gold standard in healthcare appointment management, offering a secure and efficient platform powered by Firestore Authentication. Built with the latest advancements in technology and security, CareLink ensures that every user interaction is safeguarded with the highest level of protection. Admin, doctor and patient have their own according to access level.
2. **Firestore Storage:** CareLink harnesses the power of Firestore and Realtime Database to provide a robust and real-time data management solution, revolutionizing the way healthcare appointments are managed and coordinated. In addition to Firestore, CareLink utilizes Firestore Realtime Database to facilitate real-time communication and collaboration between users. Realtime Database provides a JSON-based cloud database that synchronizes data in real-time across connected clients.
3. **Advanced Graphics:** Introducing the CareLink app with a splash screen that captures attention and sets the stage for a seamless healthcare experience. As the app loads, users are greeted with a dynamic animation that symbolizes the app's commitment to innovation, efficiency, and care.
4. **Speech To Text:** Incorporating speech-to-text functionality into the CareLink app enhances accessibility and efficiency, allowing users to interact with the app using natural language and voice commands.
5. **Web Search:** Integrating web search functionality into the CareLink app expands its capabilities beyond healthcare appointment management, providing users with access to valuable health-related information and resources directly within the app interface.
6. **Book Appointments:** Patients can log into the CareLink app and navigate to the appointment booking section. They can select their preferred date, and type of appointment. Patients can also specify any additional information or requirements for their appointment.

# **ABSTRACTION**

Introducing CareLink, a groundbreaking doctor dashboard application leveraging Firebase for seamless authentication and storage solutions, setting a new standard in healthcare management. CareLink redefines the healthcare landscape with its innovative features, including speech-to-text integration and web search capabilities, empowering medical professionals to work smarter and more efficiently. With distinct access levels for administrators, doctors, and patients, CareLink ensures secure and tailored experiences for every user. Its advanced graphics elevate data visualization, providing insights at a glance for informed decision-making. Whether accessing patient records, scheduling appointments, or collaborating with colleagues, CareLink offers a unified platform for streamlined workflow management.

From secure communication channels to comprehensive data storage, CareLink prioritizes privacy and compliance, adhering to stringent regulatory standards such as HIPAA. With a mobile-friendly interface and offline functionality, CareLink ensures accessibility and continuity of care, anytime and anywhere. In essence, CareLink is more than an application—it's a catalyst for transformative healthcare delivery, empowering providers, and patients alike to achieve optimal outcomes.

CareLink stands at the forefront of healthcare innovation, revolutionizing the way medical professionals interact with patient data and manage their daily responsibilities. By harnessing the power of Firebase for authentication and storage, CareLink ensures a secure and reliable platform for accessing and storing sensitive medical information. Its integration of speech-to-text technology not only streamlines data entry but also enhances accessibility for users with diverse needs, enabling hands-free interaction with the application. Additionally, the inclusion of web search capabilities expands the horizons of medical knowledge available to practitioners, fostering continuous learning and informed decision-making.

With CareLink's multi-tiered access system, administrators wield powerful oversight capabilities, doctors navigate patient care seamlessly, and patients engage with their health information confidently. This tailored approach not only enhances user experience but also strengthens data security and confidentiality, preserving patient trust in the healthcare system.

Moreover, CareLink's commitment to advanced graphics ensures that complex data sets are presented in an intuitive and digestible format, empowering users to derive meaningful insights efficiently. Whether tracking patient outcomes, monitoring clinic productivity, or analysing treatment efficacy, CareLink's visualizations provide clarity and context to inform clinical practice and drive quality improvement initiatives.

# **TOPICS COVERED**

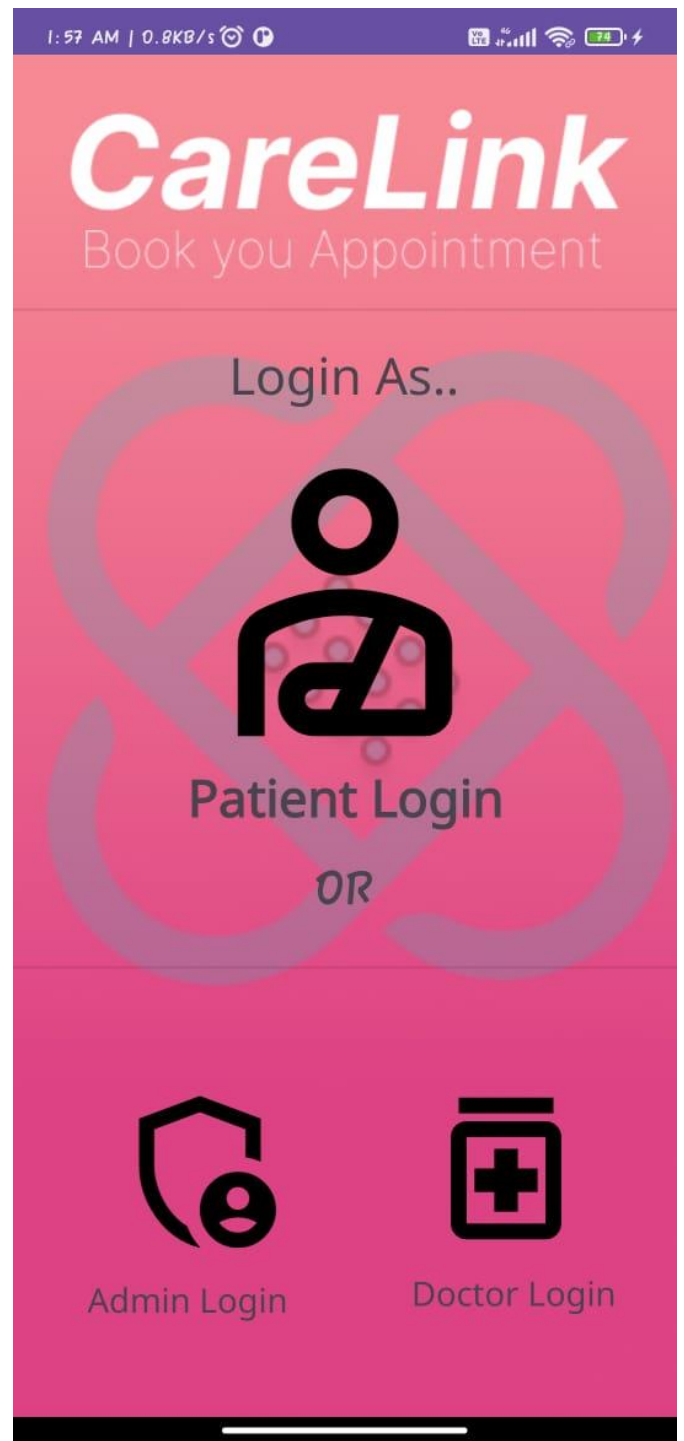
- **Splash Screen**
- **Advance Animation**
- **Fire Base Authentication**
- **Realtime Database**
- **Firestore Database**
- **Speech to Text**
- **Web Search**
- **Explicit Intent**
- **Implicit Intent**
- **Recycle View**
- **Scroll View**
- **Bottom Navigation**
- **Card View**
- **Progress Dialog Box**
- **Image View**
- **Drop Down Menu**
- **Switch Button**
- **Toast**
- **Information/Input Validation**

# Screen Shots of Application

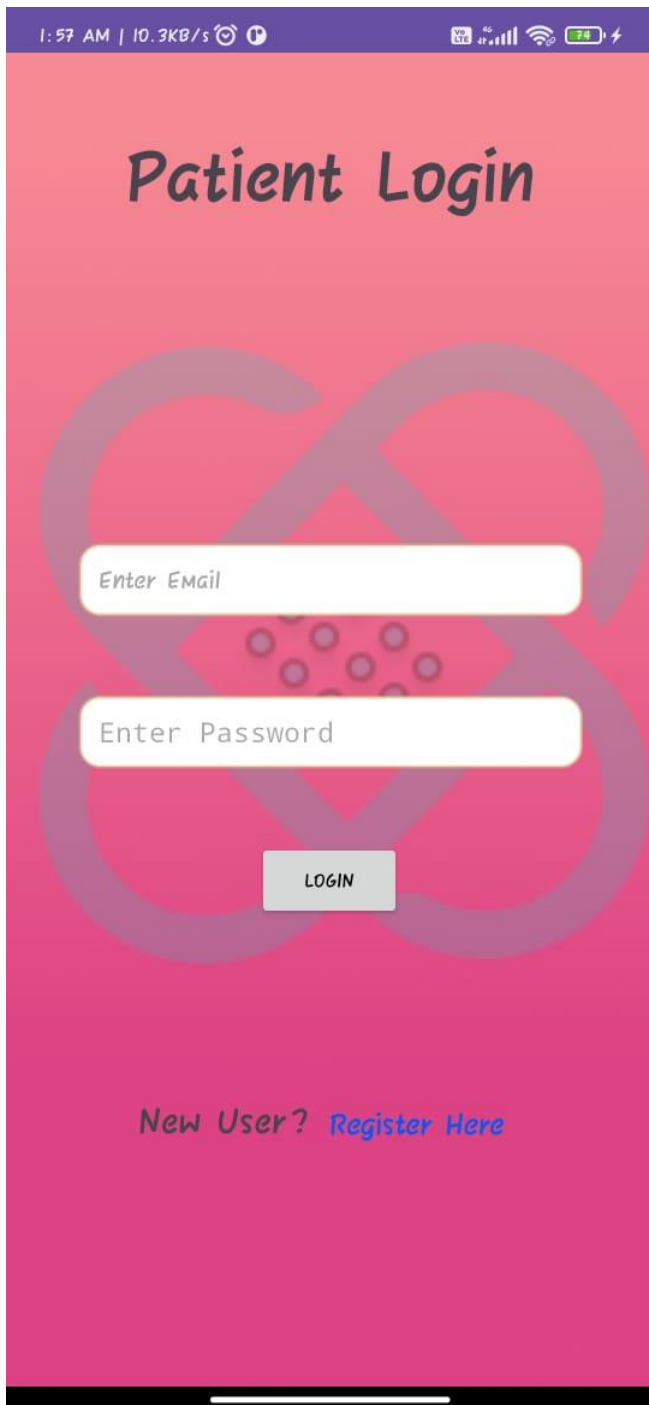
**Splash Screen Activity**



**Main Screen**



## Patient Login



1:57 AM | 10.3KB/s

# Patient Login

Enter Email

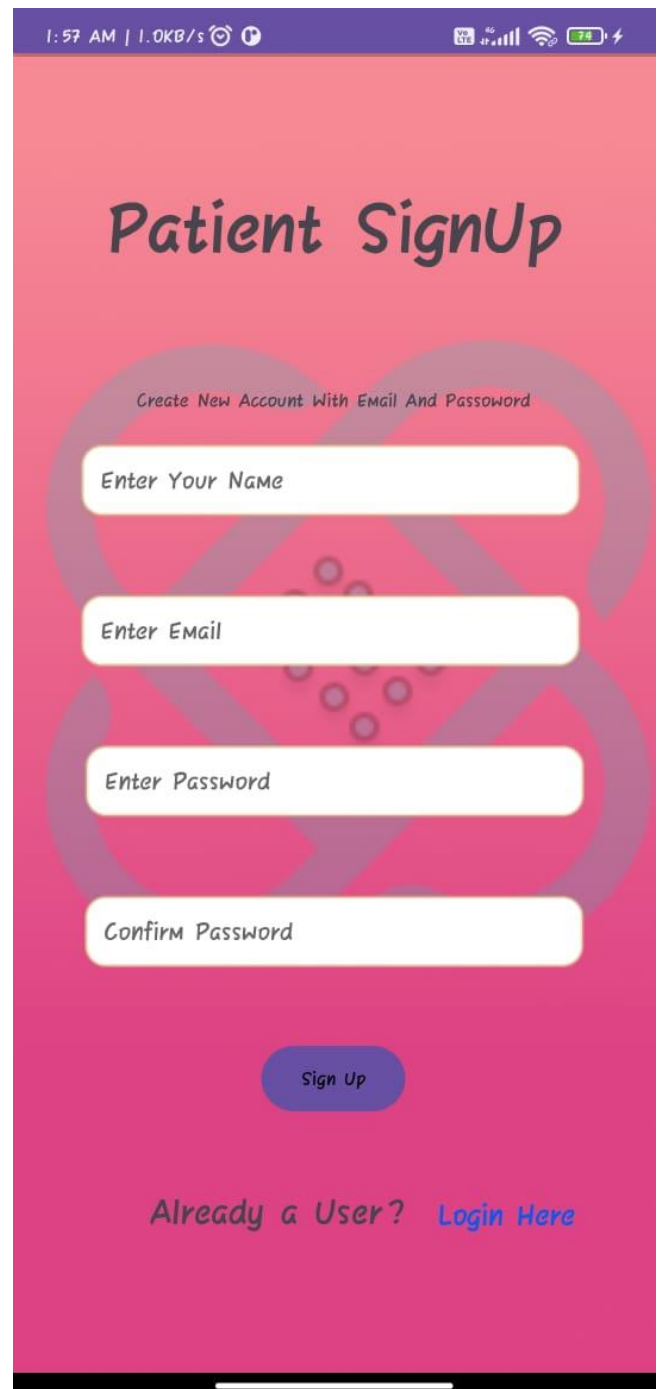
Enter Password

LOGIN

New User? [Register Here](#)

This is a mobile app screen for patient login. It features a pink background with a large, faint, stylized 'X' watermark. The screen has a status bar at the top showing the time as 1:57 AM and a data speed of 10.3KB/s. The main heading is 'Patient Login'. Below it are two white input fields with rounded corners, labeled 'Enter Email' and 'Enter Password'. A grey button with the text 'LOGIN' is positioned below the password field. At the bottom, there is a link that says 'New User? Register Here'.

## Patient Sign Up



1:57 AM | 1.0KB/s

# Patient SignUp

Create New Account With Email And Password

Enter Your Name

Enter Email

Enter Password

Confirm Password

Sign Up

Already a User? [Login Here](#)


This is a mobile app screen for patient sign up. It features a pink background with a large, faint, stylized 'X' watermark. The screen has a status bar at the top showing the time as 1:57 AM and a data speed of 1.0KB/s. The main heading is 'Patient SignUp'. Below it is a sub-heading 'Create New Account With Email And Password'. There are four white input fields with rounded corners, labeled 'Enter Your Name', 'Enter Email', 'Enter Password', and 'Confirm Password'. A blue button with the text 'Sign Up' is positioned below the 'Confirm Password' field. At the bottom, there is a link that says 'Already a User? Login Here'.

## Patient Dashboard

1:57 AM | 0.2KB/s

4G

78




Welcome, **YASH UBANA**

# CareLink

Book your Appointment


Ask Anything...

>



List of Doctors

mon-sun



300 Rs/Hour

Dr. Yash


eye

rps


Jaipur

12 Years


+911234567890



Dashboard



Book Appointment



History

## Appointment Booking Page

1:57 AM | 0.2KB/s

4G

78

## Book Appointment

Enter Patient Name

Date of Birth (DD-MM-YYYY)

Male

☐


Female

Mobile Number


Enter Problem / Reason

Select Doctor


Submit



Dashboard






Book Appointment



History

## Patient Booking History

1:58 AM | 7.3KB/s



MOHIT

▼


Date of Birth : 05-05-2000


Gender : Male


Mobile : 9509341181

Reason : i am suffering from high fever

Doctor Name : Yash ubana

Dashboard

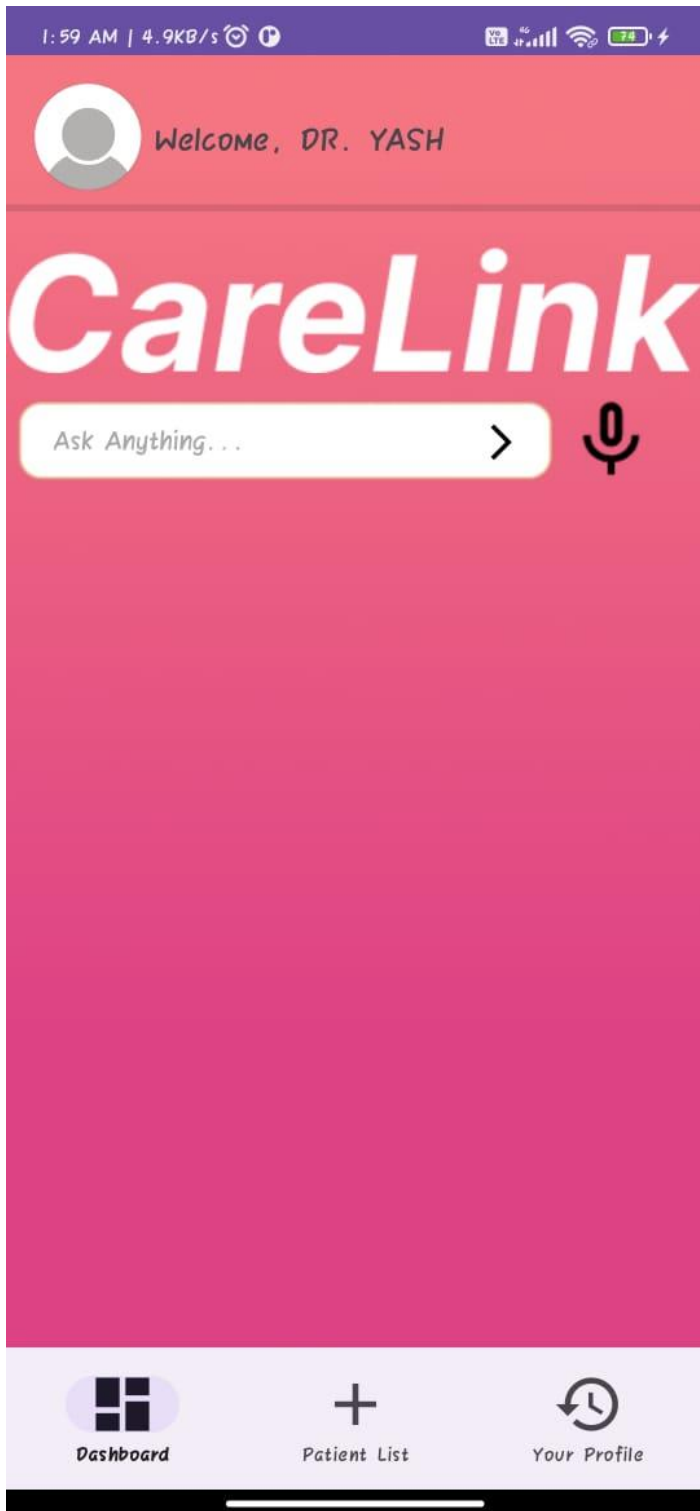
Book Appointment

History

## Doctor Login Screen



## Doctor Dashboard



## Patient List for Doctor



## Doctor Profile Detail Input

[illegible]

### Doctor Profile Info.

The image is a screenshot of a mobile application interface for a doctor's profile. At the top, there is a status bar with the time '1:59 AM', data usage '135KB/s', and various connectivity icons. The profile section has a light blue background with a white border. It features a circular placeholder for a profile picture. Below the picture, the text 'MON-SUN' and '300 Rs/Hour' are displayed. The doctor's name 'DR. YASH' is centered, followed by 'eye', 'rps', 'JAIPUR', '12 Years', and the phone number '+911234567890'. Two buttons, 'Edit Info' (purple) and 'Log Out' (red), are at the bottom of the profile section. The bottom navigation bar is purple and contains three icons: a dashboard icon, a plus sign for 'Patient List', and a clock icon for 'Your Profile'.

## Admin Login

1:57 AM | 0.6KB/s

# Admin Login

Enter Email

Enter Password

LOGIN

## Admin Dashboard

1:59 AM | 4.2MB/s

# Admin Dashboard

Create Doctor Account

Enter Doctor's Name

Enter doctor email

Password

Confirm Password

CREATE

Dashboard Doctor List Patient List

2:00 AM | 6.1KB/s

SHSHSJS

▼

Date of Birth : 05-05-8000

Gender : Female

Mobile : 4646

Reason : shshsj

Doctor Name : Yash ubana

MOHIT

▼

Date of Birth : 05-05-2000

Gender : Male

Mobile : 9509341181

Reason : i am suffering from high fever

Doctor Name : Yash ubana

YASHUBANA

▼

Date of Birth : 05082002

Dashboard

+

Doctor List

Patient List

## **Motive behind “CareLink” App**

The driving force behind the creation of CareLink stems from a deep-rooted commitment to revolutionize healthcare delivery and enhance patient outcomes. Recognizing the increasingly complex challenges faced by medical professionals in managing patient care, accessing vital information, and collaborating effectively within multidisciplinary teams, the creators of CareLink envisioned a comprehensive solution that seamlessly integrates advanced technology with intuitive design principles. At its core, CareLink seeks to bridge the gap between healthcare providers and patients, empowering both parties with the tools and resources needed to navigate the complexities of modern healthcare delivery. By leveraging cutting-edge features such as speech-to-text, web search capabilities, and advanced graphics, CareLink aims to streamline workflows, improve communication, and foster informed decision-making across the continuum of care. Ultimately, the motive behind CareLink is to empower healthcare professionals to deliver high-quality, patient-centred care while promoting efficiency, transparency, and innovation within the healthcare ecosystem.

At the heart of CareLink lies a profound desire to redefine the healthcare experience, transcending the boundaries of traditional patient care and administrative tasks. Motivated by a vision of a healthcare landscape characterized by seamless coordination, enhanced accessibility, and empowered patient engagement, the creators embarked on a journey to develop a transformative solution that addresses the multifaceted needs of healthcare providers and patients alike. With a relentless focus on innovation, CareLink aims to break down silos, facilitate interdisciplinary collaboration, and promote data-driven decision-making to drive continuous improvement in healthcare delivery.

## MainActivity.kt

```
package com.example.carelink

import android.annotation.SuppressLint
import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.carelink.Auth.Admin.AdminLoginPage
import com.example.carelink.Auth.Doctor.LoginDoc
import com.example.carelink.Auth.Patient.LoginPage
import com.example.carelink.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        binding = ActivityMainBinding.inflate(layoutInflater)
        super.onCreate(savedInstanceState)
        setContentView(binding.root)

        binding.imageView7.setOnClickListener{
            val intent = Intent(this,
AdminLoginPage::class.java )
            startActivity(intent)
        }
        binding.imageView8.setOnClickListener{
            val intent = Intent(this, LoginDoc::class.java )
            startActivity(intent)
        }
        binding.imageView9.setOnClickListener{
            val intent = Intent(this, LoginPage::class.java )
            startActivity(intent)
        }

    }
    @SuppressLint("MissingSuperCall")
    override fun onBackPressed() {

    }
}
```

## Patient/Doctor/Admin Login

```
package com.example.carelink.Auth.Patient

import android.app.ProgressDialog
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import
com.example.carelink.Auth.Patient.Fragments.BookAppointment
import com.example.carelink.MainActivity
import com.example.carelink.R
import
com.example.carelink.databinding.ActivityLoginPageBinding
import com.google.firebase.auth.FirebaseAuth

class LoginPage : AppCompatActivity() {
    private lateinit var binding: ActivityLoginPageBinding
    private lateinit var firebaseAuth: FirebaseAuth
    lateinit var progressDialog: ProgressDialog
    override fun onCreate(savedInstanceState: Bundle?) {
        binding =
ActivityLoginPageBinding.inflate(layoutInflater)
        super.onCreate(savedInstanceState)
        setContentView(binding.root)

        firebaseAuth = FirebaseAuth.getInstance()
        progressDialog = ProgressDialog(this)
        progressDialog.setTitle("Getting database")
        progressDialog.setMessage("Loading...")
        progressDialog.setCancelable(false)

        binding.textView3.setOnClickListener {
            val intent = Intent(this, SignupPage::class.java)
            startActivity(intent)
            finish()
        }
        binding.button.setOnClickListener {
            progressDialog.show()
            val email = binding.patientEmail.text.toString()
            val pass = binding.patientPass.text.toString()

            if (email.isNotEmpty() && pass.isNotEmpty()) {
                if (pass.length < 6) {
                    binding.patientPass.error = "Password must
```

```

be at least 6 characters"
        return@setOnClickListener
    } else {

firebaseAuth.signInWithEmailAndPassword(email,
pass).addOnCompleteListener {
        progressDialog.hide()
        if (it.isSuccessful) {
            val intent = Intent(this,
Dashboard::class.java)
            startActivity(intent)
            finish()
        } else {
            Toast.makeText(this,
it.exception.toString(), Toast.LENGTH_SHORT).show()
        }
    }
}

}

}

}

override fun onStart() {
    super.onStart()
    if (firebaseAuth.currentUser != null) {

        val intent = Intent(this, Dashboard::class.java)
        startActivity(intent)
        finish()
    }
}

override fun onBackPressed() {
    super.onBackPressed()
    val intent = Intent(this, MainActivity::class.java)
    startActivity(intent)
    finish()
}
}

```



# Patient Appointment Booking

```
package com.example.carelink.Auth.Patient.Fragments

import android.graphics.Color
import android.os.Bundle
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.AdapterView
import android.widget.AdapterView.Adapter
import android.widget.ArrayAdapter
import android.widget.AutoCompleteTextView
import android.widget.Toast
import androidx.fragment.app.Fragment
import com.example.carelink.R
import com.example.carelink.databinding.FragmentBookAppointmentBinding
import com.google.firebase.Firebase
import com.google.firebase.auth.FirebaseAuth
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 com.google.firebase.database.getValue
import com.google.firebase.firestore.DocumentReference
import com.google.firebase.firestore.SetOptions
import com.google.firebase.firestore.firestore

class BookAppointment : Fragment() {
    private var binding: FragmentBookAppointmentBinding? = null
    private lateinit var gen: String
    private lateinit var doc: String
    private lateinit var ids: String
    private val db = Firebase.firestore
    lateinit var ref: DocumentReference
    lateinit var ref2: DocumentReference
    lateinit var ref3: DocumentReference
    lateinit var docName: DatabaseReference
    var items = mutableMapOf<String, String>()
    var keysList = mutableListOf<String>()
    val uid = FirebaseAuth.getInstance().currentUser!!.uid

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        docName = FirebaseDatabase.getInstance().getReference("DocDetail")
        fetchDocName()
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        binding = FragmentBookAppointmentBinding.inflate(inflater, container,
false)
        gen = "Male"
        binding!!.textView9.setTextColor(Color.parseColor("#72FFEB3B"))

        ref = db.collection("Users").document(uid).collection(uid).document()
        ids = ref.id
        ref2 = db.collection("Admin").document(ids)

        doc = "Select Doctor"
```

```

binding!!.switch1.setOnCheckedChangeListener { _, isChecked ->
    if (isChecked) {
        gen = "Female"
        binding!!.textView7.setTextColor(Color.parseColor("#72FFEB3B"))
        binding!!.textView9.setTextColor(Color.BLACK)
    } else {
        gen = "Male"
        binding!!.textView9.setTextColor(Color.parseColor("#72FFEB3B"))
        binding!!.textView7.setTextColor(Color.BLACK)
    }
}

binding!!.button3.setOnClickListener {
    val name = binding!!.name.text.toString().trim()
    val dob = binding!!.date.text.toString().trim()
    val gender = gen
    val phone = binding!!.phone.text.toString().trim()
    val reason = binding!!.reason.text.toString().trim()
    val doctor = doc
    val ids = ids
    val uid = uid
    val doc_id = items[doc]
    ref3 =
db.collection("Doctor_Patient").document(doc_id.toString()).collection("Patients")
        .document();

    if (name.isEmpty()) {
        binding!!.name.error = "Please enter your name"
        return@setOnClickListener
    }
    if (dob.isEmpty()) {
        binding!!.date.error = "Please enter your date of birth"
        return@setOnClickListener
    }

    if (phone.isEmpty()) {
        binding!!.phone.error = "Please enter your phone number"
        return@setOnClickListener
    }
    else if (phone.length != 10) {
        binding!!.phone.error = "Please enter a valid phone number"
        return@setOnClickListener
    }

    if (reason.isEmpty()) {
        binding!!.reason.error = "Please enter your reason"
        return@setOnClickListener
    }
    else if (reason.length < 10){
        binding!!.reason.error = "Please enter detailed reason (min 10
characters)"
        return@setOnClickListener
    }

    if (doctor.isEmpty()) {
        binding!!.autoComplete.error = "Please select a doctor"
        return@setOnClickListener
    }
    if (doctor == "Select Doctor") {
        Toast.makeText(requireContext(), "Please select a doctor",

```

```

Toast.LENGTH_SHORT).show()
        return@setOnClickListener
    }

    if(name.isNotEmpty() && dob.isNotEmpty() && phone.isNotEmpty() &&
reason.isNotEmpty() && doctor.isNotEmpty() && ids.isNotEmpty() && uid.isNotEmpty()
&&
        phone.length == 10 && reason.length >= 10 && doctor != "Select
Doctor") {

        val userMap = hashMapOf(
            "name" to name,
            "dob" to dob,
            "gender" to gender,
            "phone" to phone,
            "reason" to reason,
            "doctor" to doctor,
            "id" to ids,
            "uid" to uid,
        )
        val docpatient = hashMapOf(
            "name" to name,
            "dob" to dob,
            "gender" to gender,
            "phone" to phone,
            "reason" to reason,
        )
        ref.set(userMap)
        .addOnSuccessListener {
            Toast.makeText(requireContext(), "Appointment Booked",
Toast.LENGTH_SHORT)
                .show()
                binding!!.name.text.clear()
                binding!!.date.text.clear()
                binding!!.phone.text.clear()
                binding!!.reason.text.clear()
            }.addOnFailureListener {
                Toast.makeText(requireContext(), it.message.toString(),
Toast.LENGTH_SHORT)
                    .show()
            }
        ref2.set(userMap)
        ref3.set(docpatient)
    }else{
        Toast.makeText(requireContext(), "Please enter every detail",
Toast.LENGTH_SHORT)
            .show()
    }
}
return binding!!.root
}

fun fetchDocName() {
    val docRef = docName
    docRef.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            snapshot.children.forEach { doc ->
                val name = doc.child("name").getValue(String::class.java)
                val doc_id = doc.child("id").getValue(String::class.java)

                items[name.toString()] = doc_id.toString()
                keysList = items.keys.toMutableList()

                spinner(keysList)
            }
        }
    })
}

```

```
        override fun onCancelled(error: DatabaseError) {
            Log.d("BookAppointment", "Error: ${error.message}")
        }
    })
}

private fun spinner(keysList: MutableList<String>) {
    val autoCompleteer: AutoCompleteTextView = binding!!.autoComplete

    println(keysList)
    val adapter = ArrayAdapter(requireContext(), R.layout.doctor_list_patient,
keysList)

    autoCompleteer.setAdapter(adapter)
    autoCompleteer.setOnItemClickListener =
        AdapterView.OnItemClickListener { adapterView, view, i, l ->

            doc = adapterView.getItemAtPosition(i).toString()
            Toast.makeText(requireContext(), doc, Toast.LENGTH_SHORT).show()
        }
    }
}
```