MOBILE PROGRAMMING
CA-4 REVIEW 1
MEDHUB

INTRODUCTION

Introducing "MedHub," an innovative Kotlin-based application designed to enhance healthcare accessibility worldwide. Utilizing advanced geolocation technology, MedHub pinpoints nearby labs and medical equipment stores, providing essential resources like wheelchairs, canes etc. It also integrates emergency services, offering quick access to hotlines. The app features a sophisticated appointment scheduling system for both in-person and virtual consultations, waiving charges for virtual appointments to ensure affordability. MedHub serves as a robust platform for blood, organ, and plasma donation initiatives, offering real-time updates on available resources across healthcare institutions. With a user-centric design, MedHub empowers individuals to manage their health and actively contribute to lifesaving efforts. Guided by Kotlin's versatility and accessibility, MedHub emerges as a transformative force, promoting healthcare equity and community well-being

TEAM MEMBERS

PAVITHRAA D [E0222041]
POORVAA SRI B [E0222033]

MAIN ACTIVITIES

1. Home
2. Signup & Login
3. Lab Page
4. Blood Donor Page
5. Organ Donor Page
6. Plasma Donor Page
7. Medical Shop Page

DATABASE REQUIREMENTS

1. Users Table

- userId (Primary Key)
- username
- password
- userType (e.g., patient, lab technician, shop owner)

2. Appointments Table

- appointmentId (Primary Key)
- userId (Foreign Key referencing Users table)
- appointmentDate
- appointmentTime
- location (for lab/shop appointments)
- type (e.g., lab appointment, blood donation, organ donation, plasma donation)

3. Blood Donors Table

- donorId (Primary Key)
- name
- address
- phone
- hospitalName
- hospitalAddress

4. Organ Donors Table

- donorId (Primary Key)
- name
- address
- phone
- hospitalName
- hospitalAddress

5. Plasma Donors Table

- donorId (Primary Key)
- name
- address
- phone
- hospitalName
- hospitalAddress

6. Medical Shops Table

- shopId (Primary Key)
- name
- address
- location
- phone

ANDROID COMPONENTS OF EACH ACTIVITY

For Home Activity (activity_home.xml)

- **ImageView**: To display a logo.
- **Button**: Login button.
- **ScrollView**: For scrolling our services section.
- **TextView**: For displaying text (about us, address).
- **WebView**: For displaying Google Maps or other web content.
- **ImageButton**: For clickable image buttons to navigate to other activities.

For Login Activity (activity_login.xml)

- **EditText**: For entering username and password.
- **Button**: Login button.
- **TextView**: For displaying instructions or messages.
- **ProgressBar**: Optional, for indicating loading or login progress.

For Lab Activity (activity_lab.xml)

- **MapView**: To display a map and get user's location.
- **Button**: For appointment booking.
- **TextView**: For displaying nearby labs and their details.

For Blood Donor Activity (activity_blood_donor.xml)

- **TextView**: For displaying a list of blood donors.

- **LinearLayout/RelativeLayout**: For each donor's information (name, address, phone number).

For Organ Donor Activity (activity_organ_donor.xml)

- **TextView**: For displaying a list of organ donors.
- **LinearLayout/RelativeLayout**: For each donor's information (name, hospital name, address, phone number).

For Plasma Donor Activity (activity_plasma_donor.xml)

- **TextView**: For displaying a list of plasma donors.
- **LinearLayout/RelativeLayout**: For each donor's information (name, hospital name, address, phone number).

For Medical Shop Activity (activity_medical_shop.xml)

- **EditText**: For entering search/filter criteria.
- **Button**: For applying filters.
- **TextView**: For displaying a list of medical shops.
- **LinearLayout/RelativeLayout**: For each shop's information (name, location, phone number).

General Components

- **TextView**: For displaying text information.
- **EditText**: For user input.
- **Button**: For user interaction.
- **ScrollView**: If content needs to be scrollable.
- **MapView**: For integrating Google Maps or displaying maps.
- **ImageButton**: For clickable images or icons.

INTENT

Manifest.xml:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.yourpackagename">
       <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
```

```
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

          <!-- Home Activity -->
          <activity android:name=".MainActivity">
           <intent-filter>
              <action android:name="android.intent.action.MAIN" />

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

          <!-- Blood Donor Activity -->
          <activity android:name=".BloodDonorActivity" />

          <!-- Organ Donor Activity -->
          <activity android:name=".OrganDonorActivity" />

          <!-- Plasma Donor Activity -->
          <activity android:name=".PlasmaDonorActivity" />

          <!-- Labs Activity -->
          <activity android:name=".LabActivity" />

          <!-- Medical Shop Activity -->
          <activity android:name=".MedicalShopActivity" />

    </application>

</manifest>
```

Home.kt:

```
 import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
```

```
  override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

          // Navigate to Blood Donor Activity
        buttonBloodDonor.setOnClickListener {
          val intent = Intent(this, BloodDonorActivity::class.java)
           startActivity(intent)
          }

          // Navigate to Organ Donor Activity
        buttonOrganDonor.setOnClickListener {
          val intent = Intent(this, OrganDonorActivity::class.java)
           startActivity(intent)
          }

          // Navigate to Plasma Donor Activity
        buttonPlasmaDonor.setOnClickListener {
          val intent = Intent(this, PlasmaDonorActivity::class.java)
           startActivity(intent)
          }
          // Navigate to Labs Activity
        buttonLabs.setOnClickListener {
          val intent = Intent(this, LabActivity::class.java)
           startActivity(intent)
          }

          // Navigate to Shops Activity
        buttonShops.setOnClickListener {
          val intent = Intent(this, MedicalShopActivity::class.java)
           startActivity(intent)
                }
          }
}
```