

Practical 1

- Develop a basic mobile application to display a message in center of screen

ActivityMain.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/user_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="80dp"
        android:layout_marginRight="20dp"
        android:hint="User ID"
        android:padding="20dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"

tools:ignore="Autofill,HardcodedText,TextContrastCheck,TextFields,VisualLintTextFie
ldSize" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="80dp"
        android:layout_marginRight="20dp"
        android:hint="Password"
        android:inputType="textPassword"
        android:padding="20dp"
```

```

        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/user_id"

tools:ignore="Autofill,HardcodedText,TextContrastCheck,VisualLintTextFieldSize" />

<Button
    android:id="@+id/submit_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="20dp"
    android:text="Submit"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/password"
    tools:ignore="HardcodedText" />

<TextView
    android:id="@+id/message"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="16dp"
    android:text=""
    android:textSize="18sp" />
</LinearLayout>

```

MainActivity.kt

```

package com.example.practical1

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

```

```

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

    val userIdEditText = findViewById<EditText>(R.id.user_id)
    val passwordEditText = findViewById<EditText>(R.id.password)
    val submitButton = findViewById<Button>(R.id.submit_button)
    val messageTextView = findViewById<TextView>(R.id.message)

    val predefinedUserId = "admin"
    val predefinedPassword = "1234"

    submitButton.setOnClickListener {
        val enteredUserId = userIdEditText.text.toString()
        val enteredPassword = passwordEditText.text.toString()

        if (enteredUserId == predefinedUserId && enteredPassword ==
predefinedPassword) {
            messageTextView.text = "Login successful!"

messageTextView.setTextColor(resources.getColor(android.R.color.holo_green_dark
))
        } else {
            messageTextView.text = "Invalid user ID or password."

messageTextView.setTextColor(resources.getColor(android.R.color.holo_red_dark))
        }
    }
}

```

practical 2

Develop an application that asks user to enter user id and password. Upon receiving user id and password, application should compare both of them with prescribed values. Application should display appropriate message to user.

ActivityMain.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:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp"
    tools:context=".MainActivity">
```

```
<EditText
    android:id="@+id/email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="12dp"
    android:hint="Enter your email"
    android:inputType="textEmailAddress"
    android:padding="20dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
    android:id="@+id/password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="12dp"
    android:hint="Enter your password"
    android:inputType="textPassword"
    android:padding="20dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/email" />
```

```
<Button
    android:id="@+id/loginButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```

        android:layout_marginLeft="20dp"
        android:layout_marginTop="60dp"
        android:layout_marginRight="20dp"
        android:enabled="false"
        android:padding="15dp"
        android:text="Log In"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/password" />

<TextView
    android:id="@+id/errorTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="20dp"
    android:textColor="@android:color/holo_red_dark"
    android:visibility="gone"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/password" />
</LinearLayout>

```

Welcome.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/welcomeTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="12dp"

```

```
        android:layout_marginRight="12dp"
        android:padding="20dp"
        android:text="Welcome!"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Welcome.kt

```
package com.example.practical3
```

```
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.example.practical2.R
```

```
class WelcomeActivity : AppCompatActivity() {
```

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

```
        val email = intent.getStringExtra("email")
```

```
        val welcomeTextView: TextView = findViewById(R.id.welcomeTextView)
        welcomeTextView.text = "Welcome, $email!"
```

```
    }
}
```

MainActivity.kt

```
package com.example.practical3
```

```
import android.content.Intent
import android.os.Bundle
import android.text.Editable
```

```
import android.text.TextWatcher
import android.util.Patterns
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.example.practical2.R
```

```
class MainActivity : AppCompatActivity() {
```

```
    private lateinit var emailEditText: EditText
    private lateinit var passwordEditText: EditText
    private lateinit var loginButton: Button
    private lateinit var errorTextView: TextView
```

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

```
        emailEditText = findViewById(R.id.email)
        passwordEditText = findViewById(R.id.password)
        loginButton = findViewById(R.id.loginButton)
        errorTextView = findViewById(R.id.errorTextView)
```

```
        // Listen for changes in the email EditText to validate email format
```

```
        emailEditText.addTextChangedListener(object : TextWatcher {
            override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int,
after: Int) {}
            override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count:
Int) {
                loginButton.isEnabled = isEmailValid(s.toString())
            }
            override fun afterTextChanged(s: Editable?) {}
        })
```

```
        loginButton.setOnClickListener {
            val email = emailEditText.text.toString()
            val password = passwordEditText.text.toString()
```

```
            if (authenticateUser(email, password)) {
```

```

        // Authentication successful
        val intent = Intent(this, WelcomeActivity::class.java)
        intent.putExtra("email", email)
        startActivity(intent)
        finish()
    } else {
        // Authentication failed
        errorTextView.text = "Authentication failed. Please check your credentials."
        errorTextView.visibility = TextView.VISIBLE
    }
}

// Function to validate email format
private fun isEmailValid(email: String): Boolean {
    return Patterns.EMAIL_ADDRESS.matcher(email).matches()
}

// Dummy authentication function (replace with actual authentication logic)
private fun authenticateUser(email: String, password: String): Boolean {
    return email == "user@example.com" && password == "password123"
}
}

```

practical 5

Develop an application that should increment the value of a text after specific interval of time. Also provide feature to stop incrementing value of the text.

ActivityMain.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"

```



```
android:padding="16dp">
```

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="0"  
    android:textSize="32sp" />
```

```
<Button  
    android:id="@+id/startButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="16dp"  
    android:text="Start Increment" />
```

```
<Button  
    android:id="@+id/stopButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="16dp"  
    android:enabled="true"  
    android:text="Stop Increment" />
```

```
</LinearLayout>
```

MainActivity.kt

```
package com.example.practical5  
  
import android.os.Bundle  
import android.os.Handler  
import android.os.Looper  
import androidx.appcompat.app.AppCompatActivity  
import android.widget.Button  
import android.widget.TextView  
  
class MainActivity : AppCompatActivity() {  
  
    private lateinit var tvValue: TextView  
    private lateinit var btnStart: Button  
    private lateinit var btnStop: Button
```

```

private var value = 0
private val handler = Handler(Looper.getMainLooper())
private lateinit var incrementRunnable: Runnable
private var isIncrementing = false

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

    tvValue = findViewById(R.id.textView)
    btnStart = findViewById(R.id.startButton)
    btnStop = findViewById(R.id.stopButton)

    incrementRunnable = object : Runnable {
        override fun run() {
            if (isIncrementing) {
                value++
                tvValue.text = value.toString()
                handler.postDelayed(this, 1000) // Increment every 1 second
            }
        }
    }

    btnStart.setOnClickListener {
        if (!isIncrementing) {
            isIncrementing = true
            handler.post(incrementRunnable)
        }
    }

    btnStop.setOnClickListener {
        isIncrementing = false
        handler.removeCallbacks(incrementRunnable)
    }
}

override fun onDestroy() {
    super.onDestroy()
    handler.removeCallbacks(incrementRunnable)
}

```

```
}  
}
```

practical 6

Develop an application that activates a progress bar on button click. The progress of progress bar (in form of %) should be shown accordingly. Appropriate message should be shown upon completion of increment in progress bar value.

ActivityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:padding="16dp">  
  
    <ProgressBar  
        android:id="@+id/progressBar"  
        style="@style/Widget.AppCompat.ProgressBar.Horizontal"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:max="100"  
        android:progress="0"  
        android:layout_centerInParent="true"  
        android:layout_marginTop="50dp" />  
  
    <TextView  
        android:id="@+id/tvProgress"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Progress: 0%"  
        android:layout_below="@id/progressBar"  
        android:layout_centerHorizontal="true"  
        android:layout_marginTop="20dp"  
        android:textSize="18sp"  
        android:textColor="@android:color/black"/>  
  
    <Button  
        android:id="@+id/btnStart"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"
```

```
        android:text="Start Progress"
        android:layout_below="@id/tvProgress"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp"/>
```

```
</RelativeLayout>
```

MainActivity.kt

```
package com.example.practical6
```

```
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.widget.Button
import android.widget.ProgressBar
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
```

```
class MainActivity : AppCompatActivity() {

    private lateinit var progressBar: ProgressBar
    private lateinit var tvProgress: TextView
    private lateinit var btnStart: Button

    private var progressStatus = 0
    private val handler = Handler(Looper.getMainLooper())

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

        progressBar = findViewById(R.id.progressBar)
        tvProgress = findViewById(R.id.tvProgress)
        btnStart = findViewById(R.id.btnStart)

        btnStart.setOnClickListener {
            progressStatus = 0
            progressBar.progress = progressStatus
            tvProgress.text = "Progress: 0%"

            Thread {
                while (progressStatus < 100) {
                    progressStatus += 1
                }
            }.start()
        }
    }
}
```

```

        // Update the progress bar and display the current percentage
        handler.post {
            progressBar.progress = progressStatus
            tvProgress.text = "Progress: $progressStatus%"
        }

        try {
            // Sleep for 50 milliseconds to simulate a time-consuming task
            Thread.sleep(50)
        } catch (e: InterruptedException) {
            e.printStackTrace()
        }
    }

    // Show the completion message
    handler.post {
        tvProgress.text = "Task Completed!"
    }
    }.start()
}
}
}
}
}

```

practical 7

Develop an android service that shows a message at specific interval of time. Also provide features to start and stop the service.

ActivityMain.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity">

    <Button

```

```
    android:id="@+id/startServiceButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Service" />
```

```
<Button
    android:id="@+id/stopServiceButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Stop Service"
    android:layout_marginTop="16dp"/>
</LinearLayout>
```

MyService.kt

```
package com.example.practical7
```

```
import android.app.Service
import android.content.Intent
import android.os.Handler
import android.os.IBinder
import android.util.Log
import android.widget.Toast
```

```
class MyService : Service() {

    private val handler = Handler()
    private val interval: Long = 5000 // 5 seconds
    private lateinit var runnable: Runnable

    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        Log.d("MyService", "Service started")

        runnable = object : Runnable {
            override fun run() {
                showMessage()
                handler.postDelayed(this, interval)
            }
        }
        handler.post(runnable)

        return START_STICKY
    }
}
```

```

private fun showMessage() {
    Toast.makeText(this, "Service is running", Toast.LENGTH_SHORT).show()
    Log.d("MyService", "Message displayed")
}

override fun onDestroy() {
    super.onDestroy()
    handler.removeCallbacks(runnable)
    Log.d("MyService", "Service stopped")
}

override fun onBind(intent: Intent?): IBinder? {
    return null
}
}

```

MainActivity.kt

```

package com.example.practical7

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var binding: activity_main

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = activity_main.inflate(layoutInflater)
        setContentView(binding.root)

        binding.startServiceButton.setOnClickListener {
            startService(Intent(this, MyService::class.java))
        }

        binding.stopServiceButton.setOnClickListener {
            stopService(Intent(this, MyService::class.java))
        }
    }
}

```

practical 8

Develop an android application that ask for upper bound and lower bound value of a range. The application should list out prime numbers existing between specified range on screen (utilize service running in background for searching prime numbers). Upon completion of operation, application should prompt the user through a notification.

ActivityMain.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:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/upperBoundEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="60dp"
        android:layout_marginRight="12dp"
        android:hint="Enter upper bound"
        android:inputType="number"
        android:textSize="20sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/lowerBoundEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="60dp"
        android:layout_marginRight="12dp"
        android:hint="Enter lower bound"
        android:inputType="number"
        android:textSize="20sp"
```



```
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toBottomOf="@id/upperBoundEditText" />
```

```
<Button
    android:id="@+id/findPrimesButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="12dp"
    android:text="Find Prime Numbers"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/lowerBoundEditText" />
```

```
<TextView
    android:id="@+id/resultTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="12dp"
    android:text="Prime numbers will be displayed here"
    android:textSize="16sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/findPrimesButton" />
```

```
</LinearLayout>
```

MainActivity.kt

```
package com.example.practical8
```

```
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.practical8.databinding.ActivityMainBinding
```

```
class MainActivity : AppCompatActivity() {
```

```

private lateinit var binding: ActivityMainBinding

private val primeResultReceiver = object : BroadcastReceiver() {
    override fun onReceive(context: Context?, intent: Intent?) {
        val primes = intent?.getIntegerArrayListExtra("PRIME_NUMBERS")
        binding.resultTextView.text = primes?.joinToString(", ") ?: "No primes found"
        Toast.makeText(this@MainActivity, "Prime search complete!",
Toast.LENGTH_SHORT).show()
    }
}

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

    registerReceiver(primeResultReceiver,
IntentFilter("com.example.practical7.PRIME_RESULT"))

    binding.findPrimesButton.setOnClickListener {
        val lowerBound = binding.lowerBoundEditText.text.toString().toIntOrNull() ?:
0
        val upperBound = binding.upperBoundEditText.text.toString().toIntOrNull() ?:
0

        if (lowerBound >= upperBound) {
            Toast.makeText(this, "Lower bound should be less than upper bound",
Toast.LENGTH_SHORT).show()
            return@setOnClickListener
        }

        val intent = Intent(this, PrimeNumberService::class.java).apply {
            putExtra("LOWER_BOUND", lowerBound)
            putExtra("UPPER_BOUND", upperBound)
        }
        startService(intent)
    }
}

override fun onDestroy() {
    super.onDestroy()
    unregisterReceiver(primeResultReceiver)
}

```

```
}  
}
```

PrimeNumberService.kt

```
package com.example.practical8
```

```
import android.Manifest  
import android.app.Service  
import android.content.Intent  
import android.content.pm.PackageManager  
import android.os.IBinder  
import androidx.core.app.ActivityCompat  
import androidx.core.app.NotificationCompat  
import androidx.core.app.NotificationManagerCompat
```

```
class PrimeNumberService : Service() {
```

```
    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {  
        val lowerBound = intent?.getIntExtra("LOWER_BOUND", 0) ?: 0  
        val upperBound = intent?.getIntExtra("UPPER_BOUND", 0) ?: 0
```

```
        val primes = findPrimes(lowerBound, upperBound)  
        val resultIntent = Intent("com.example.practical7.PRIME_RESULT")  
        resultIntent.putIntegerArrayListExtra("PRIME_NUMBERS", primes as  
ArrayList<Int>)  
        sendBroadcast(resultIntent)
```

```
        showCompletionNotification(primes.size)
```

```
        stopSelf()  
        return START_NOT_STICKY  
    }
```

```
    private fun findPrimes(lowerBound: Int, upperBound: Int): List<Int> {  
        val primes = mutableListOf<Int>()  
        for (i in lowerBound..upperBound) {  
            if (isPrime(i)) primes.add(i)  
        }  
        return primes  
    }
```

```
    private fun isPrime(n: Int): Boolean {
```

```

    if (n < 2) return false
    for (i in 2..Math.sqrt(n.toDouble()).toInt()) {
        if (n % i == 0) return false
    }
    return true
}

private fun showCompletionNotification(primeCount: Int) {
    val notification = NotificationCompat.Builder(this, "PRIME_CHANNEL")
        .setSmallIcon(android.R.drawable.ic_dialog_info)
        .setContentTitle("Prime Number Search")
        .setContentText("Found $primeCount prime numbers")
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .build()

    val notificationManager = NotificationManagerCompat.from(this)
    if (ActivityCompat.checkSelfPermission(
        this,
        Manifest.permission.POST_NOTIFICATIONS
    ) != PackageManager.PERMISSION_GRANTED
    ) {
        // TODO: Consider calling
        // ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        // public void onRequestPermissionsResult(int requestCode, String[] permissions,
        // int[] grantResults)
        // to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return
    }
    notificationManager.notify(1, notification)
}

override fun onBind(intent: Intent?): IBinder? {
    return null
}
}

```

practical 9

Develop an application that fetches all contacts phone book and displays on screen.

ActivityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <ListView
        android:id="@+id/contactsListView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

MainActivity.kt

```
package com.example.practical9

import android.Manifest
import android.R
import android.content.pm.PackageManager
import android.database.Cursor
import android.os.Bundle
import android.provider.ContactsContract
import android.widget.ArrayAdapter
import android.widget.ListView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding
    private lateinit var contactsListView: ListView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
```

```

setContentView(binding.root)

contactsListView = binding.contactsListView

// Check for permission to read contacts
if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_CONTACTS)
    != PackageManager.PERMISSION_GRANTED) {
    // Request permission if not granted
    ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.READ_CONTACTS), 1)
    } else {
        // If permission is already granted, fetch contacts
        fetchContacts()
    }
}

override fun onRequestPermissionsResult(
    requestCode: Int, permissions: Array<out String>, grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if (requestCode == 1) {
        if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            // If permission granted, fetch contacts
            fetchContacts()
        } else {
            // If permission denied, show a toast
            Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
        }
    }
}

private fun fetchContacts() {
    val contactsList = mutableListOf<String>()
    val cursor: Cursor? = contentResolver.query(
        ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
        null, null, null, null
    )

    cursor?.use {

```

```

        val nameIndex =
it.getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME)
        val numberIndex =
it.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER)

        while (it.moveToNext()) {
            val name = it.getString(nameIndex)
            val number = it.getString(numberIndex)
            contactsList.add("$name: $number")
        }
    }

    val adapter = ArrayAdapter(this, R.layout.simple_list_item_1, contactsList)
    contactsListView.adapter = adapter
}
}

```

in gradle ->

```

android {
    viewBinding {
        var enabled = true
    }
}

```

practical 10

Develop an application that accepts a number from user. The application should dynamically generate accepted number of list items in another activity

ActivityMain.xml

```

<!-- activity_main.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/numberInput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```
    android:layout_marginLeft="12dp"
    android:layout_marginTop="270dp"
    android:layout_marginRight="12dp"
    android:hint="Enter a number"
    android:inputType="number"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/generateButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="50dp"
    android:layout_marginRight="12dp"
    android:text="Generate List"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

</LinearLayout>

ActivityList.xml

<!-- activity_list.xml -->

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
```

```
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
```

</LinearLayout>

ActivityList.kt

```
package com.example.practical10
```

```
import android.os.Bundle
```



```

import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class ListActivity : AppCompatActivity() {

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

        val recyclerView = findViewById<RecyclerView>(R.id.recyclerView)
        recyclerView.layoutManager = LinearLayoutManager(this)

        val numberOfItems = intent.getIntExtra("NUMBER_OF_ITEMS", 0)
        val items = List(numberOfItems) { "Item #${it + 1}" }

        recyclerView.adapter = ListAdapter(items)
    }
}

```

ActivityMain.kt

```

package com.example.practical10

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast

class MainActivity : AppCompatActivity() {

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

        val numberInput = findViewById<EditText>(R.id.numberInput)
        val generateButton = findViewById<Button>(R.id.generateButton)

        generateButton.setOnClickListener {
            val numberText = numberInput.text.toString()
            if (numberText.isNotEmpty()) {

```

```

        val number = numberText.toInt()
        val intent = Intent(this, ListActivity::class.java).apply {
            putExtra("NUMBER_OF_ITEMS", number)
        }
        startActivity(intent)
    } else {
        Toast.makeText(this, "Please enter a number",
            Toast.LENGTH_SHORT).show()
    }
}
}
}
}

```

ListAdapter.kt

```
package com.example.practical10
```

```

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

```

```

class ListAdapter(private val items: List<String>) :
    RecyclerView.Adapter<ListAdapter.ViewHolder>() {

    class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val textView: TextView = view.findViewById(android.R.id.text1)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val view = LayoutInflater.from(parent.context)
            .inflate(android.R.layout.simple_list_item_1, parent, false)
        return ViewHolder(view)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.textView.text = items[position]
    }

    override fun getItemCount(): Int = items.size
}

```

practical 11

Develop a dialer application that asks for a mobile number to the user and makes a call to that number on button click.

ActivityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <EditText
        android:id="@+id/phoneNumberInput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter mobile number"
        android:inputType="phone"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/callButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Call" />

</LinearLayout>
```

MainActivity.kt

```
package com.example.practical11

import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

```

import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {

    private lateinit var phoneNumberInput: EditText
    private lateinit var callButton: Button

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

        phoneNumberInput = findViewById(R.id.phoneNumberInput)
        callButton = findViewById(R.id.callButton)

        callButton.setOnClickListener {
            val phoneNumber = phoneNumberInput.text.toString()
            if (phoneNumber.isNotEmpty()) {
                makePhoneCall(phoneNumber)
            } else {
                Toast.makeText(this, "Please enter a valid phone number",
                    Toast.LENGTH_SHORT).show()
            }
        }
    }

    private fun makePhoneCall(phoneNumber: String) {
        if (ContextCompat.checkSelfPermission(this,
            Manifest.permission.CALL_PHONE) !=
            PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                arrayOf(Manifest.permission.CALL_PHONE), 1)
        } else {
            val callIntent = Intent(Intent.ACTION_CALL)
            callIntent.data = Uri.parse("tel:$phoneNumber")
            startActivity(callIntent)
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions:
        Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == 1) {

```

```

        if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            val phoneNumber = phoneNumberInput.text.toString()
            makePhoneCall(phoneNumber)
        } else {
            Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show()
        }
    }
}
}
}

```

practical 12

Develop an SMS manager that list downs all the messages from message box of device. User should also be able to send message through this application.

ActivityMain.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <ListView
        android:id="@+id/smsListView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:dividerHeight="1dp"/>

    <EditText
        android:id="@+id/recipientInput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter recipient number"
        android:inputType="phone"
        android:layout_marginBottom="8dp"/>

    <EditText
        android:id="@+id/messageInput"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```

        android:hint="Enter your message"
        android:layout_marginBottom="8dp"/>

<Button
    android:id="@+id/sendButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Send"/>

</LinearLayout>

```

MainActivity.kt

```

package com.example.practical12

import android.Manifest
import android.content.ContentResolver
import android.content.pm.PackageManager
import android.database.Cursor
import android.net.Uri
import android.os.Bundle
import android.telephony.SmsManager
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {

    private lateinit var smsListView: ListView
    private lateinit var recipientInput: EditText
    private lateinit var messageInput: EditText
    private lateinit var sendButton: Button
    private lateinit var smsAdapter: ArrayAdapter<String>

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

        smsListView = findViewById(R.id.smsListView)
        recipientInput = findViewById(R.id.recipientInput)
    }
}

```

```

messageInput = findViewById(R.id.messageInput)
sendButton = findViewById(R.id.sendButton)

if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_SMS) != PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions(this,
        arrayOf(Manifest.permission.READ_SMS, Manifest.permission.SEND_SMS), 1)
    } else {
        loadSmsInbox()
    }

sendButton.setOnClickListener {
    val recipient = recipientInput.text.toString()
    val message = messageInput.text.toString()
    if (recipient.isNotEmpty() && message.isNotEmpty()) {
        sendSms(recipient, message)
    } else {
        Toast.makeText(this, "Please enter both recipient and message",
Toast.LENGTH_SHORT).show()
    }
}

private fun loadSmsInbox() {
    val smsList = ArrayList<String>()
    val uri: Uri = Uri.parse("content://sms/inbox")
    val contentResolver: ContentResolver = contentResolver
    val cursor: Cursor? = contentResolver.query(uri, null, null, null, null)

    if (cursor != null && cursor.moveToFirst()) {
        do {
            val address = cursor.getString(cursor.getColumnIndexOrThrow("address"))
            val body = cursor.getString(cursor.getColumnIndexOrThrow("body"))
            smsList.add("From: $address\nMessage: $body")
        } while (cursor.moveToNext())
        cursor.close()
    }

    smsAdapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, smsList)
    smsListView.adapter = smsAdapter
}

private fun sendSms(recipient: String, message: String) {

```

```

try {
    val smsManager: SmsManager = SmsManager.getDefault()
    smsManager.sendTextMessage(recipient, null, message, null, null)
    Toast.makeText(this, "SMS sent successfully",
Toast.LENGTH_SHORT).show()
    messageInput.text.clear()
    recipientInput.text.clear()
} catch (e: Exception) {
    Toast.makeText(this, "Failed to send SMS: ${e.message}",
Toast.LENGTH_SHORT).show()
}
}

override fun onRequestPermissionsResult(requestCode: Int, permissions:
Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if (requestCode == 1) {
        if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            loadSmsInbox()
        } else {
            Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show()
        }
    }
}
}
}

```

practical 13

Develop a user registration application which data from user and insert received data in a text file (generated/existing) on the same device.

ActivityMain.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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

```



```
<EditText
    android:id="@+id/name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="12dp"
    android:layout_marginBottom="16dp"
    android:hint="Enter your name"
    android:inputType="textPersonName"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
    android:id="@+id/email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="12dp"
    android:layout_marginBottom="16dp"
    android:hint="Enter your email"
    android:inputType="textEmailAddress"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/name" />
```

```
<EditText
    android:id="@+id/password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="12dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="12dp"
    android:layout_marginBottom="16dp"
    android:hint="Enter your password"
    android:inputType="textPassword"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/email" />
```

```
<Button
```

```
    android:id="@+id/registerButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="20dp"
    android:text="Register"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/password" />
```

</LinearLayout>

MainActivity.kt

```
package com.example.practical13

import android.content.Context
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import java.io.File
import java.io.FileOutputStream

class MainActivity : AppCompatActivity() {

    private lateinit var nameInput: EditText
    private lateinit var emailInput: EditText
    private lateinit var passwordInput: EditText
    private lateinit var registerButton: Button

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

        nameInput = findViewById(R.id.name)
        emailInput = findViewById(R.id.email)
        passwordInput = findViewById(R.id.password)
        registerButton = findViewById(R.id.registerButton)
```

```

registerButton.setOnClickListener {
    val name = nameInput.text.toString()
    val email = emailInput.text.toString()
    val password = passwordInput.text.toString()

    if (name.isNotEmpty() && email.isNotEmpty() && password.isNotEmpty()) {
        saveUserDataToFile(name, email, password)
    } else {
        Toast.makeText(this, "Please fill all the fields",
Toast.LENGTH_SHORT).show()
    }
}

private fun saveUserDataToFile(name: String, email: String, password: String) {
    val fileName = "user_data.txt"
    val fileContents = "Name: $name\nEmail: $email\nPassword: $password\n\n"

    try {
        val fileOutputStream: FileOutputStream = openFileOutput(fileName,
Context.MODE_APPEND)
        fileOutputStream.write(fileContents.toByteArray())
        fileOutputStream.close()
        Toast.makeText(this, "User data saved successfully",
Toast.LENGTH_SHORT).show()

        // Clear the input fields
        nameInput.text.clear()
        emailInput.text.clear()
        passwordInput.text.clear()
    } catch (e: Exception) {
        Toast.makeText(this, "Failed to save user data: ${e.message}",
Toast.LENGTH_SHORT).show()
    }
}
}

```

practical 14

Develop an application that asks the user to select background color of application. The application must remember and apply selected background color every time when user launches the application then after.

ActivityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rootLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <RadioGroup
        android:id="@+id/colorRadioGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="12dp"
        android:orientation="vertical">

        <RadioButton
            android:id="@+id/redRadioButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="12dp"
            android:layout_marginTop="60dp"
            android:layout_marginRight="12dp"
            android:text="Red"
            android:textColor="@android:color/holo_red_dark"/>

        <RadioButton
            android:id="@+id/greenRadioButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="12dp"
            android:layout_marginTop="60dp"
            android:layout_marginRight="12dp"
            android:text="Green"
```

```
        android:textColor="@android:color/holo_green_dark"/>
```

```
        <RadioButton
            android:id="@+id/blueRadioButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="12dp"
            android:layout_marginTop="60dp"
            android:layout_marginRight="12dp"
            android:text="Blue"
            android:textColor="@android:color/holo_blue_dark"/>
    </RadioGroup>
```

```
    <Button
        android:id="@+id/applyButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="60dp"
        android:layout_marginRight="12dp"
        android:text="Apply" />
```

```
</LinearLayout>
```

MainActivity.kt

```
package com.example.practical14

import android.content.Context
import android.graphics.Color
import android.os.Bundle
import android.widget.Button
import android.widget.LinearLayout
import android.widget.RadioButton
import android.widget.RadioGroup
import androidx.appcompat.app.AppCompatActivity
```

```

class MainActivity : AppCompatActivity() {

    private lateinit var colorRadioGroup: RadioGroup
    private lateinit var redRadioButton: RadioButton
    private lateinit var greenRadioButton: RadioButton
    private lateinit var blueRadioButton: RadioButton
    private lateinit var applyButton: Button

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

        colorRadioGroup = findViewById(R.id.colorRadioGroup)
        redRadioButton = findViewById(R.id.redRadioButton)
        greenRadioButton = findViewById(R.id.greenRadioButton)
        blueRadioButton = findViewById(R.id.blueRadioButton)
        applyButton = findViewById(R.id.applyButton)

        loadBackgroundColor()

        applyButton.setOnClickListener {
            val selectedColor = when (colorRadioGroup.checkedRadioButtonId) {
                R.id.redRadioButton -> Color.RED
                R.id.greenRadioButton -> Color.GREEN
                R.id.blueRadioButton -> Color.BLUE
                else -> Color.WHITE
            }
            saveBackgroundColor(selectedColor)
            setAppBackgroundColor(selectedColor)
        }
    }

    private fun loadBackgroundColor() {
        val sharedPreferences = getSharedPreferences("AppPreferences",
            Context.MODE_PRIVATE)
        val backgroundColor = sharedPreferences.getInt("backgroundColor",
            Color.WHITE)
        setAppBackgroundColor(backgroundColor)
    }
}

```

```

        // Set the corresponding radio button
        when (backgroundColor) {
            Color.RED -> redRadioButton.isChecked = true
            Color.GREEN -> greenRadioButton.isChecked = true
            Color.BLUE -> blueRadioButton.isChecked = true
        }
    }

    private fun saveBackgroundColor(color: Int) {
        val sharedPreferences = getSharedPreferences("AppPreferences",
            Context.MODE_PRIVATE)
        val editor = sharedPreferences.edit()
        editor.putInt("backgroundColor", color)
        editor.apply()
    }

    private fun setAppBackgroundColor(color: Int) {
        findViewById<LinearLayout>(R.id.rootLayout).setBackgroundColor(color)
    }
}

```

practical 15

Develop a data oriented application for user data (user id, name, address, contact number) management. The application should provide interface to add, update, delete and list data of user (s). (Hint: use SQLite database).

ActivityMain.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/userIdInput"
        android:layout_width="match_parent"

```

```
    android:layout_height="wrap_content"
    android:hint="User ID"
    android:inputType="number"
    android:layout_marginBottom="8dp"/>
```

```
<EditText
    android:id="@+id/nameInput"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Name"
    android:inputType="textPersonName"
    android:layout_marginBottom="8dp"/>
```

```
<EditText
    android:id="@+id/addressInput"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Address"
    android:inputType="textPostalAddress"
    android:layout_marginBottom="8dp"/>
```

```
<EditText
    android:id="@+id/contactNumberInput"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Contact Number"
    android:inputType="phone"
    android:layout_marginBottom="8dp"/>
```

```
<Button
    android:id="@+id/addButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Add User"
    android:layout_marginBottom="8dp"/>
```

```
<Button
    android:id="@+id/updateButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Update User"
    android:layout_marginBottom="8dp"/>
```



```
<Button
    android:id="@+id/deleteButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Delete User"
    android:layout_marginBottom="8dp"/>
```

```
<ListView
    android:id="@+id/userListView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"/>
```

```
</LinearLayout>
```

MainActivity.kt

```
package com.example.practical15

import android.os.Bundle
import android.widget.*
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var userIdInput: EditText
    private lateinit var nameInput: EditText
    private lateinit var addressInput: EditText
    private lateinit var contactNumberInput: EditText
    private lateinit var addButton: Button
    private lateinit var updateButton: Button
    private lateinit var deleteButton: Button
    private lateinit var userListView: ListView

    private lateinit var dbHelper: DatabaseHelper
    private lateinit var userAdapter: ArrayAdapter<String>

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

```

userIdInput = findViewById(R.id.userIdInput)
nameInput = findViewById(R.id.nameInput)
addressInput = findViewById(R.id.addressInput)
contactNumberInput = findViewById(R.id.contactNumberInput)
addButton = findViewById(R.id.addButton)
updateButton = findViewById(R.id.updateButton)
deleteButton = findViewById(R.id.deleteButton)
userListView = findViewById(R.id.userListView)

dbHelper = DatabaseHelper(this)

loadUsers()

addButton.setOnClickListener {
    val id = userIdInput.text.toString().toInt()
    val name = nameInput.text.toString()
    val address = addressInput.text.toString()
    val contactNumber = contactNumberInput.text.toString()
    if (dbHelper.addUser(id, name, address, contactNumber) > -1) {
        Toast.makeText(this, "User added successfully",
Toast.LENGTH_SHORT).show()
        loadUsers()
    } else {
        Toast.makeText(this, "Failed to add user", Toast.LENGTH_SHORT).show()
    }
}

updateButton.setOnClickListener {
    val id = userIdInput.text.toString().toInt()
    val name = nameInput.text.toString()
    val address = addressInput.text.toString()
    val contactNumber = contactNumberInput.text.toString()
    if (dbHelper.updateUser(id, name, address, contactNumber) > 0) {
        Toast.makeText(this, "User updated successfully",
Toast.LENGTH_SHORT).show()
        loadUsers()
    } else {
        Toast.makeText(this, "Failed to update user",
Toast.LENGTH_SHORT).show()
    }
}

deleteButton.setOnClickListener {

```

```

        val id = userIdInput.text.toString().toInt()
        if (dbHelper.deleteUser(id) > 0) {
            Toast.makeText(this, "User deleted successfully",
Toast.LENGTH_SHORT).show()
            loadUsers()
        } else {
            Toast.makeText(this, "Failed to delete user",
Toast.LENGTH_SHORT).show()
        }
    }
}

private fun loadUsers() {
    val users = dbHelper.getAllUsers()
    val userStrings = users.map { "ID: ${it.id}, Name: ${it.name}, Address: $
${it.address}, Contact: ${it.contactNumber}" }
    userAdapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
userStrings)
    userListView.adapter = userAdapter
}
}

```

User.kt

```
package com.example.practical15
```

```
data class User(val id: Int, val name: String, val address: String, val contactNumber:
String)
```

DatabaseHelper.kt

```
package com.example.practical15
```

```
import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

```

```

class DatabaseHelper(context: Context) : SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        private const val DATABASE_NAME = "UserManagement.db"
        private const val DATABASE_VERSION = 1
        private const val TABLE_NAME = "users"
        private const val COLUMN_ID = "id"
        private const val COLUMN_NAME = "name"
        private const val COLUMN_ADDRESS = "address"
        private const val COLUMN_CONTACT_NUMBER = "contact_number"
    }

    override fun onCreate(db: SQLiteDatabase?) {
        val createTable = ("CREATE TABLE $TABLE_NAME (" +
            "$COLUMN_ID INTEGER PRIMARY KEY," +
            "$COLUMN_NAME TEXT," +
            "$COLUMN_ADDRESS TEXT," +
            "$COLUMN_CONTACT_NUMBER TEXT)")
        db?.execSQL(createTable)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        onCreate(db)
    }

    fun addUser(id: Int, name: String, address: String, contactNumber: String): Long {
        val db = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(COLUMN_ID, id)
        contentValues.put(COLUMN_NAME, name)
        contentValues.put(COLUMN_ADDRESS, address)
        contentValues.put(COLUMN_CONTACT_NUMBER, contactNumber)
        return db.insert(TABLE_NAME, null, contentValues)
    }

    fun updateUser(id: Int, name: String, address: String, contactNumber: String): Int {
        val db = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(COLUMN_NAME, name)
    }

```

```

        contentValues.put(COLUMN_ADDRESS, address)
        contentValues.put(COLUMN_CONTACT_NUMBER, contactNumber)
        return db.update(TABLE_NAME, contentValues, "$COLUMN_ID=?",
            arrayOf(id.toString()))
    }

    fun deleteUser(id: Int): Int {
        val db = this.writableDatabase
        return db.delete(TABLE_NAME, "$COLUMN_ID=?", arrayOf(id.toString()))
    }

    fun getAllUsers(): List<User> {
        val userList = ArrayList<User>()
        val db = this.readableDatabase
        val cursor = db.rawQuery("SELECT * FROM $TABLE_NAME", null)

        if (cursor.moveToFirst()) {
            do {
                val id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_ID))
                val name =
                    cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAME))
                val address =
                    cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_ADDRESS))
                val contactNumber =
                    cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_CONTACT_NUMBER))
                userList.add(User(id, name, address, contactNumber))
            } while (cursor.moveToNext())
        }
        cursor.close()
        return userList
    }
}

```

Practical 16

Develop a data driven user authentication application that asks for user id and password to user. After accepting user id and password, application should check whether entered authentication details exists in data table or not. Display appropriate message on screen.

ActivityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/UserId"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="60dp"
        android:layout_marginRight="12dp"
        android:hint="User Id"
        android:inputType="textEmailAddress"
        android:padding="20dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="12dp"
        android:layout_marginTop="60dp"
        android:layout_marginRight="12dp"
        android:hint="Password"
        android:inputType="textPassword"
        android:padding="20dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/UserId" />

    <Button
        android:id="@+id/loginButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

        android:layout_marginLeft="20dp"
        android:layout_marginTop="60dp"
        android:layout_marginRight="20dp"
        android:enabled="false"
        android:padding="15dp"
        android:text="Log In"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/password" />

<TextView
    android:id="@+id/TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="60dp"
    android:layout_marginRight="20dp"
    android:textColor="@android:color/holo_red_dark"
    android:visibility="gone"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/password" />
</LinearLayout>

```

MainActivity.kt

```

package com.example.practical16

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var userIdInput: EditText
    private lateinit var passwordInput: EditText

```

```

private lateinit var loginButton: Button
private lateinit var resultTextView: TextView

private lateinit var dbHelper: DatabaseHelper

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

    userIdInput = findViewById(R.id.userId)
    passwordInput = findViewById(R.id.password)
    loginButton = findViewById(R.id.loginButton)
    resultTextView = findViewById(R.id.TextView)

    dbHelper = DatabaseHelper(this)

    // Optional: Pre-populate the database with a sample user for testing
    dbHelper.addUser(1, "password123")

    loginButton.setOnClickListener {
        val userId = userIdInput.text.toString().toIntOrNull()
        val password = passwordInput.text.toString()

        if (userId != null && password.isNotEmpty()) {
            if (dbHelper.checkUser(userId, password)) {
                resultTextView.text = "Login successful!"
            }
        } else {
            resultTextView.setTextColor(resources.getColor(android.R.color.holo_green_dark))
            resultTextView.text = "Invalid user ID or password"
        }

        resultTextView.setTextColor(resources.getColor(android.R.color.holo_red_dark))
    }

    } else {
        Toast.makeText(this, "Please enter valid credentials",
            Toast.LENGTH_SHORT).show()
    }
}
}

```


DatabaseHelper.kt

```
package com.example.practical16
```

```
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
```

```
class DatabaseHelper(context: Context) : SQLiteOpenHelper(context,
    DATABASE_NAME, null, DATABASE_VERSION) {
```

```
    companion object {
        private const val DATABASE_NAME = "UserAuthentication.db"
        private const val DATABASE_VERSION = 1
        private const val TABLE_NAME = "users"
        private const val COLUMN_USER_ID = "user_id"
        private const val COLUMN_PASSWORD = "password"
    }
}
```

```
    override fun onCreate(db: SQLiteDatabase?) {
        val createTable = ("CREATE TABLE $TABLE_NAME (" +
            "$COLUMN_USER_ID INTEGER PRIMARY KEY," +
            "$COLUMN_PASSWORD TEXT)")
        db?.execSQL(createTable)
    }
}
```

```
    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        onCreate(db)
    }
}
```

```
    fun addUser(userId: Int, password: String): Long {
        val db = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(COLUMN_USER_ID, userId)
        contentValues.put(COLUMN_PASSWORD, password)
        return db.insert(TABLE_NAME, null, contentValues)
    }
}
```

```

fun checkUser(userId: Int, password: String): Boolean {
    val db = this.readableDatabase
    val cursor: Cursor = db.rawQuery(
        "SELECT * FROM $TABLE_NAME WHERE $COLUMN_USER_ID = ? AND $COLUMN_PASSWORD = ?",
        arrayOf(userId.toString(), password)
    )
    val exists = cursor.count > 0
    cursor.close()
    return exists
}
}

```

Practical 17

Develop an application that contains a spinner and an image. Spinner should contain names of animations. Upon selecting an animation name, selected animation should be applied on image view.

ActivityMain.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <Spinner
        android:id="@+id/animationSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"/>

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="200dp"
        android:layout_height="200dp"

```

```
        android:src="@drawable/facebook"
        android:contentDescription="Image for animation"/>

</LinearLayout>
```

Rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<rotate
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="2000"
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%" />
```

FadeIn.xml

```
<?xml version="1.0" encoding="utf-8"?>
<alpha xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="2000"
    android:fromAlpha="0.0"
    android:toAlpha="1.0"/>
```

SlideUp.xml

```
<?xml version="1.0" encoding="utf-8"?>
<translate
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="2000"
    android:fromYDelta="100%"
    android:toYDelta="0%"/>
```

MainActivity.kt

```
package com.example.practical17
```

```
import android.os.Bundle
import android.view.animation.AnimationUtils
import android.widget.AdapterView
import android.widget.AdapterView.Adapter
import android.widget.AdapterView.OnItemClickListener
import android.widget.AdapterView.OnItemSelectedListener
import android.widget.AdapterView.OnItemSelectedListener
import android.widget.AdapterView.OnItemClickListener
import android.widget.AdapterView.OnItemSelectedListener
import androidx.appcompat.app.AppCompatActivity
```

```
class MainActivity : AppCompatActivity() {
```

```
    private lateinit var animationSpinner: Spinner
    private lateinit var imageView: ImageView
```

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

```
        animationSpinner = findViewById(R.id.animationSpinner)
        imageView = findViewById(R.id.imageView)
```

```
        // Define animations in the spinner
```

```
        val animations = listOf("Rotate", "Fade In", "Slide Up")
        val adapter = ArrayAdapter(this, android.R.layout.simple_spinner_item,
            animations)
```

```
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
```

```
        animationSpinner.adapter = adapter
```

```
        // Set the onItemSelectedListener for the spinner
```

```
        animationSpinner.onItemSelectedListener = object :
```

```
        AdapterView.OnItemSelectedListener {
```

```
            override fun onItemSelected(parent: AdapterView<*>, view:
                android.view.View?, position: Int, id: Long) {
```

```
                when (position) {
```

```
                    0 -> applyAnimation(R.anim.rotate)
```

```
                    1 -> applyAnimation(R.anim.fade_in)
```

```
                    2 -> applyAnimation(R.anim.slide_up)
```

```
                }
```

```
            }
```

```

        override fun onNothingSelected(parent: AdapterView<*>) {
            // Do nothing
        }
    }
}

private fun applyAnimation(animationResource: Int) {
    val animation = AnimationUtils.loadAnimation(this, animationResource)
    imageView.startAnimation(animation)
}
}

```

practical 18

Develop an audio player application having facilities to start, pause and stop audio playback.

ActivityMain.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <Button
        android:id="@+id/startButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start"/>

    <Button
        android:id="@+id/pauseButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pause"

```

```

        android:layout_marginTop="16dp"/>

<Button
    android:id="@+id/stopButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Stop"
    android:layout_marginTop="16dp"/>

</LinearLayout>

```

MainActivity.kt

```

package com.example.practical18

import android.media.MediaPlayer
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var startButton: Button
    private lateinit var pauseButton: Button
    private lateinit var stopButton: Button

    private var mediaPlayer: MediaPlayer? = null
    private var isPaused = false

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

        startButton = findViewById(R.id.startButton)
        pauseButton = findViewById(R.id.pauseButton)
        stopButton = findViewById(R.id.stopButton)

        startButton.setOnClickListener {
            if (mediaPlayer == null) {

```

```

        mediaPlayer = MediaPlayer.create(this, R.raw.sample_audio)
        mediaPlayer?.start()
    } else if (isPaused) {
        mediaPlayer?.start()
        isPaused = false
    }
}

pauseButton.setOnClickListener {
    mediaPlayer?.pause()
    isPaused = true
}

stopButton.setOnClickListener {
    mediaPlayer?.stop()
    mediaPlayer?.release()
    mediaPlayer = null
    isPaused = false
}
}

override fun onDestroy() {
    mediaPlayer?.release()
    super.onDestroy()
}
}

```

practical 19

Develop an application to show contents of specified URL without using native browser. Also provide facility to navigate to previous and next page as well as clear browsing history.

ActivityMain.xml

```
<uses-permission android:name="android.permission.INTERNET"/>
```

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

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

```
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"  
android:orientation="vertical">
```

```
<WebView
```

```
    android:id="@+id/webView"  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1"/>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal">
```

```
<Button
```

```
    android:id="@+id/btnBack"  
    android:layout_width="0dp"  
    android:layout_weight="1"  
    android:text="Back" />
```

```
<Button
```

```
    android:id="@+id/btnForward"  
    android:layout_width="0dp"  
    android:layout_weight="1"  
    android:text="Forward" />
```

```
<Button
```

```
    android:id="@+id/btnClearHistory"  
    android:layout_width="0dp"
```



```
        android:layout_weight="1"
        android:text="Clear History" />
    </LinearLayout>
</LinearLayout>
```

MainActivity.kt

```
package com.example.webviewapp

import android.os.Bundle
import android.webkit.WebChromeClient
import android.webkit.WebView
import android.webkit.WebViewClient
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var webView: WebView

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

        webView = findViewById(R.id.webView)
        val btnBack: Button = findViewById(R.id.btnBack)
        val btnForward: Button = findViewById(R.id.btnForward)
        val btnClearHistory: Button = findViewById(R.id.btnClearHistory)
```

```
// Set up WebView
webView.webViewClient = WebViewClient()
webView.webChromeClient = WebChromeClient()
webView.settings.javaScriptEnabled = true

// Load the specified URL
webView.loadUrl("https://www.example.com")

// Back button functionality
btnBack.setOnClickListener {
    if (webView.canGoBack()) {
        webView.goBack()
    }
}

// Forward button functionality
btnForward.setOnClickListener {
    if (webView.canGoForward()) {
        webView.goForward()
    }
}

// Clear browsing history
btnClearHistory.setOnClickListener {
    webView.clearHistory()
}
}
```

```
// Override the back button to navigate back in the web view
override fun onBackPressed() {
    if (webView.canGoBack()) {
        webView.goBack()
    } else {
        super.onBackPressed()
    }
}
}
```

Practical 20

Develop an application that keeps track of location (coordinates) of device and display values of longitude and latitude on screen.

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.locationtracker">

    <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION"/>

    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION"/>

    <application
        ...
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
```

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

ActivityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/tvLatitude"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Latitude: "
        android:textSize="20sp"/>

    <TextView
        android:id="@+id/tvLongitude"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Longitude: "
        android:textSize="20sp"/>

</LinearLayout>
```

MainActivity.kt

```
package com.example.locationtracker

import android.Manifest
import android.content.pm.PackageManager
import android.location.Location
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import com.google.android.gms.location.*

class MainActivity : AppCompatActivity() {

    private lateinit var fusedLocationClient: FusedLocationProviderClient
    private lateinit var tvLatitude: TextView
    private lateinit var tvLongitude: TextView

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

        tvLatitude = findViewById(R.id.tvLatitude)
        tvLongitude = findViewById(R.id.tvLongitude)

        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
```

```

        // Check for location permission

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
LOCATION_PERMISSION_REQUEST_CODE)

        } else {

            startLocationUpdates()

        }

    }

private fun startLocationUpdates() {

    fusedLocationClient.requestLocationUpdates(LocationRequest.create().apply {

        interval = 10000 // 10 seconds

        fastestInterval = 5000 // 5 seconds

        priority = LocationRequest.PRIORITY_HIGH_ACCURACY

    }, locationCallback, null)

}

private val locationCallback = object : LocationCallback() {

    override fun onLocationResult(locationResult: LocationResult?) {

        locationResult ?: return

        for (location in locationResult.locations) {

            updateLocationUI(location)

        }

    }

}

private fun updateLocationUI(location: Location) {

```

```

        tvLatitude.text = "Latitude: ${location.latitude}"
        tvLongitude.text = "Longitude: ${location.longitude}"
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions:
Array<String>, grantResults: IntArray) {
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if ((grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED)) {
                startLocationUpdates()
            }
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        fusedLocationClient.removeLocationUpdates(locationCallback)
    }

    companion object {
        private const val LOCATION_PERMISSION_REQUEST_CODE = 1
    }
}

```