****

**UNIVERSITI MALAYSIA TERENGGANU**

**FACULTY OF COMPUTER SCIENCE AND MATHEMATICS**

**­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CSM3123 - NATIVE MOBILE PROGRAMMING**

**BANCHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS**

**LAB 3 TASK (1,2,3,4)**

**SEMESTER 5 2024/2025**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PREPARED FOR:**

**DR RABIEI B MAMAT**

**PREPARED BY:**

**NUR EZREENA SHUHADA BT EMRAN**

**S66467**

**Link Github:** <https://github.com/NurEzreena/CSM3123_LAB-NATIVE-PROGRAMMING.git>

**TASK 1 : Working with SharedPreferences**

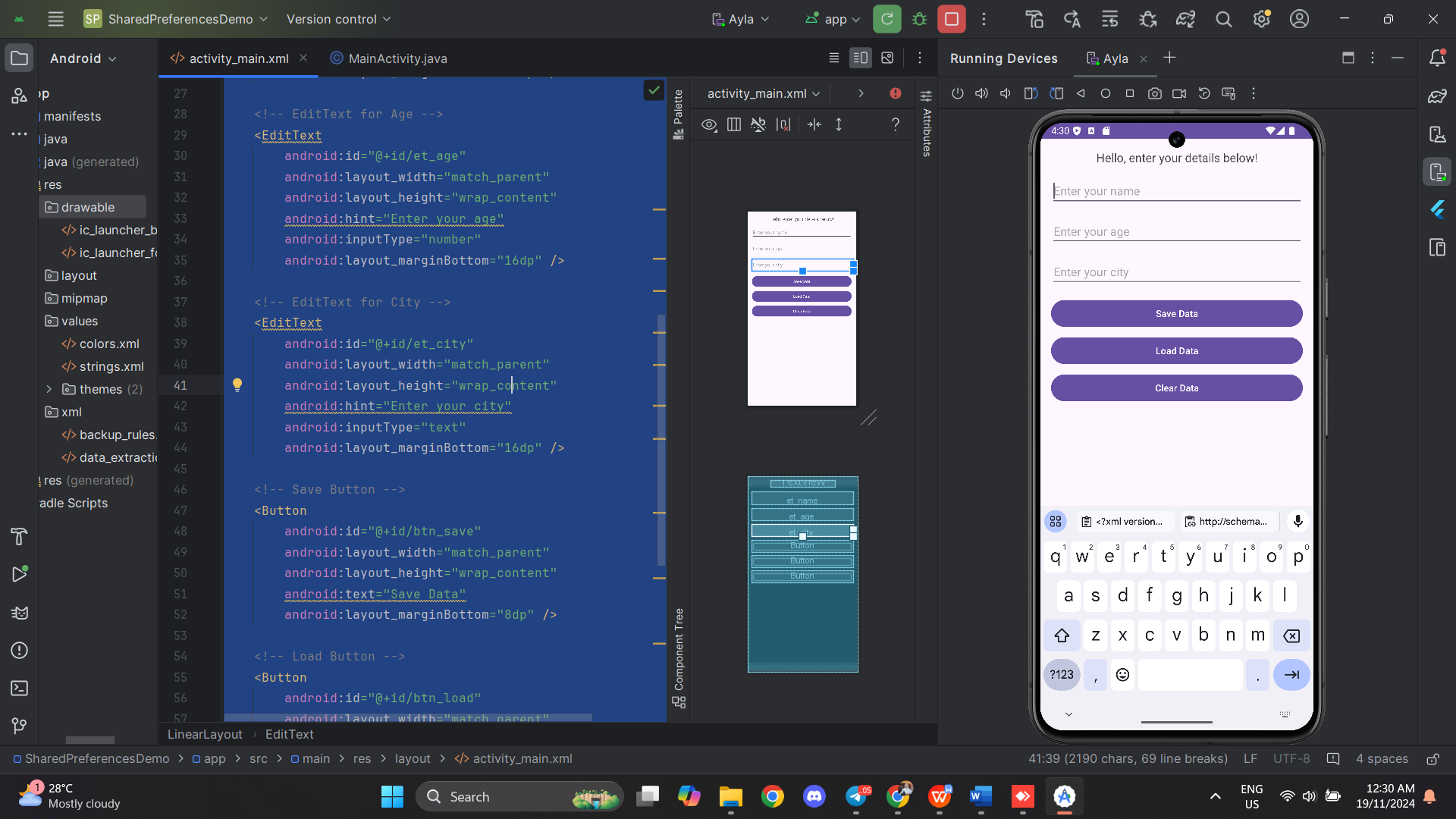
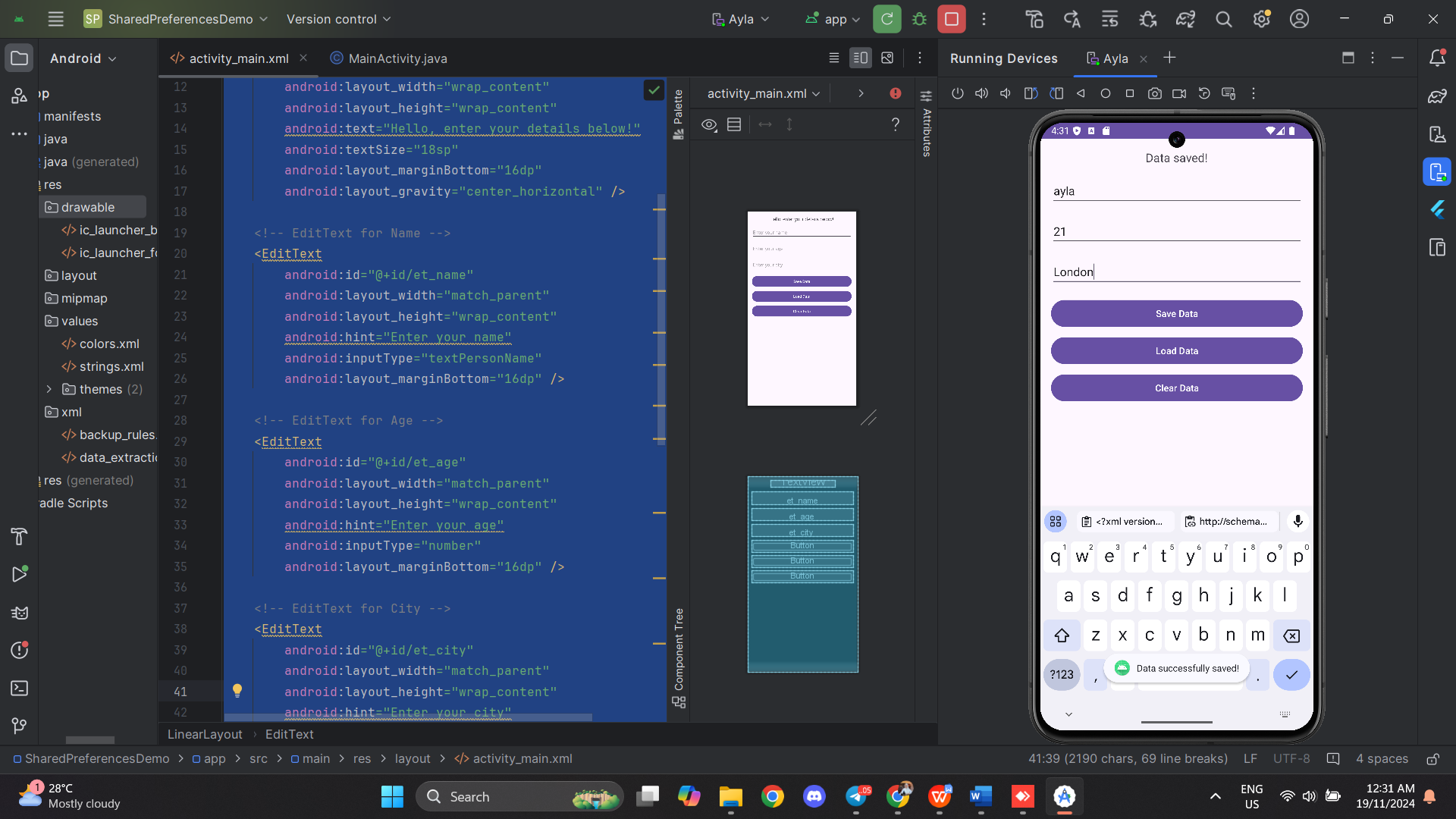
MainActivity.java

|  |
| --- |
| <?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">  <!-- Greeting TextView -->  <TextView  android:id="@+id/tv\_greeting"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello, enter your details below!"  android:textSize="18sp"  android:layout\_gravity="center\_horizontal"  android:layout\_marginBottom="16dp" />  <!-- Name Input -->  <EditText  android:id="@+id/et\_name"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter your name"  android:inputType="textPersonName"  android:layout\_marginBottom="16dp" />  <!-- Age Input -->  <EditText  android:id="@+id/et\_age"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter your age"  android:inputType="number"  android:layout\_marginBottom="16dp" />  <!-- City Input -->  <EditText  android:id="@+id/et\_city"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter your city"  android:inputType="text"  android:layout\_marginBottom="16dp" />  <!-- Save Button -->  <Button  android:id="@+id/btn\_save"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Save Data"  android:background="@drawable/ripple\_button"  android:textColor="#FFFFFF"  android:textStyle="bold"  android:layout\_marginBottom="8dp" />  <!-- Load Button -->  <Button  android:id="@+id/btn\_load"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Load Data"  android:background="@drawable/ripple\_button"  android:textColor="#FFFFFF"  android:textStyle="bold"  android:layout\_marginBottom="8dp" />  <!-- Clear Button -->  <Button  android:id="@+id/btn\_clear"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Clear Data"  android:background="@drawable/ripple\_button"  android:textColor="#FFFFFF"  android:textStyle="bold" />  </LinearLayout> |

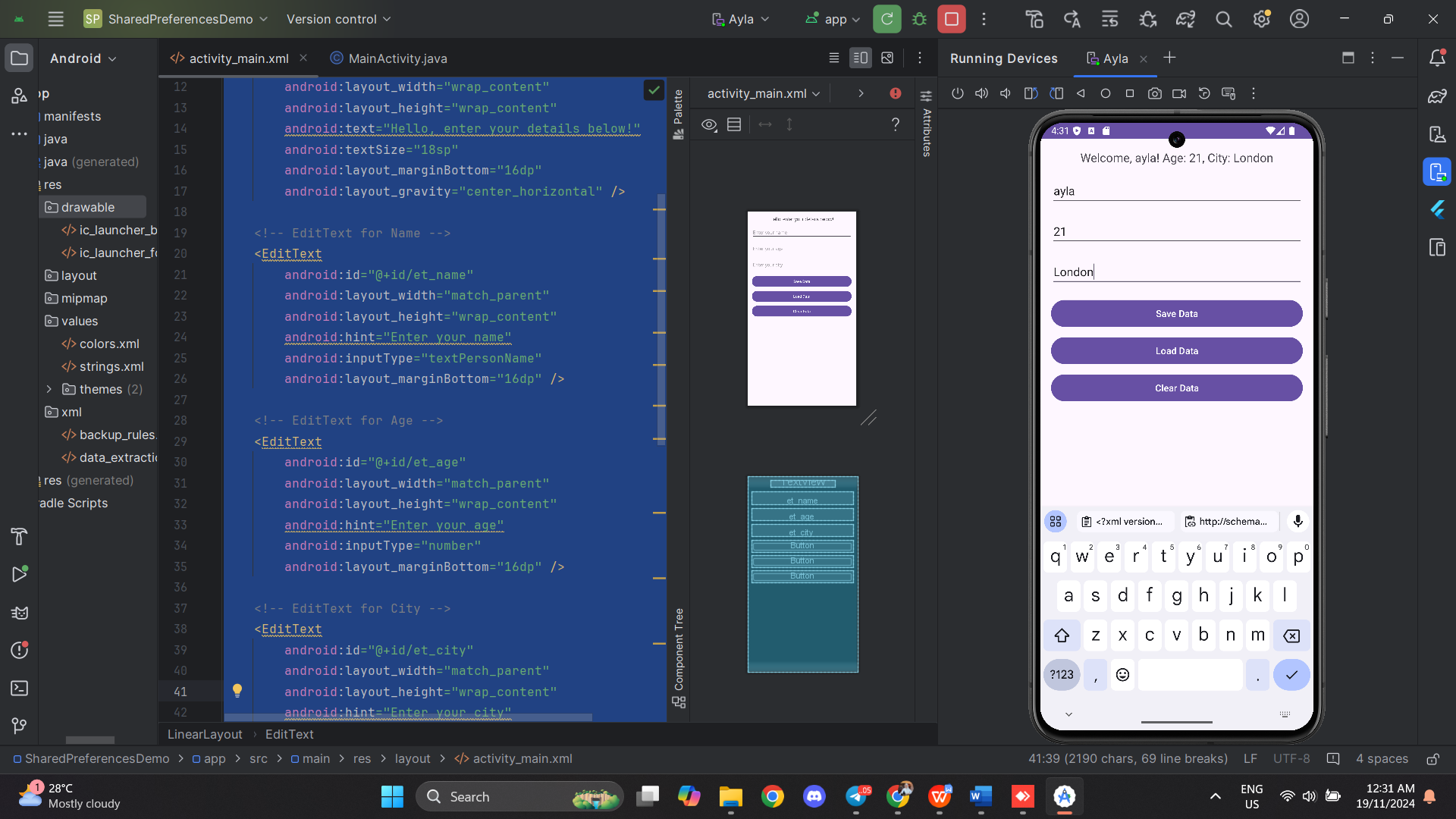
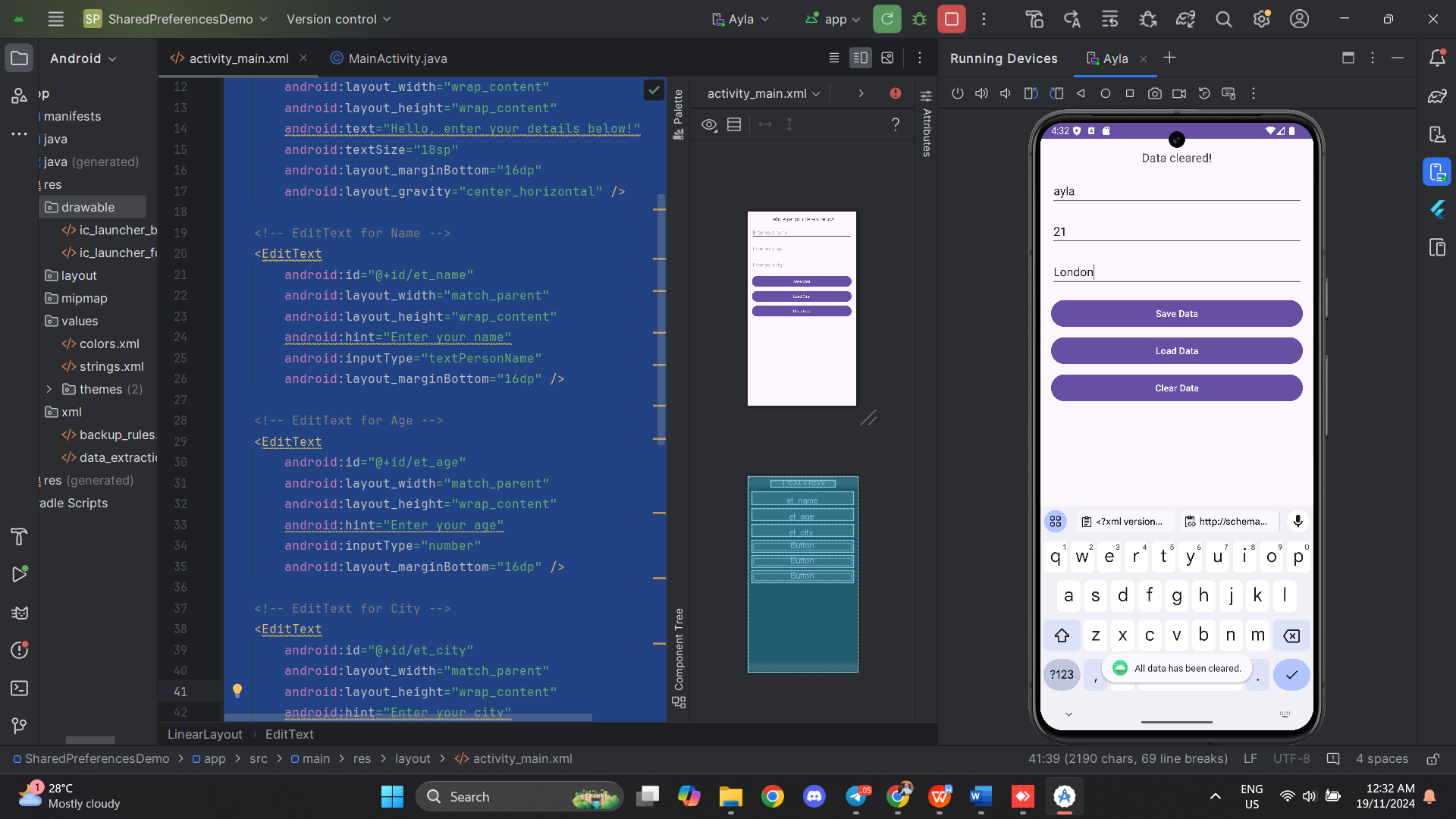
Activity\_main.xml

|  |
| --- |
| <?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 for Greeting -->  <TextView  android:id="@+id/tv\_greeting"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello, enter your details below!"  android:textSize="18sp"  android:layout\_marginBottom="16dp"  android:layout\_gravity="center\_horizontal" />   <!-- EditText for Name -->  <EditText  android:id="@+id/et\_name"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter your name"  android:inputType="textPersonName"  android:layout\_marginBottom="16dp" />   <!-- EditText for Age -->  <EditText  android:id="@+id/et\_age"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter your age"  android:inputType="number"  android:layout\_marginBottom="16dp" />   <!-- EditText for City -->  <EditText  android:id="@+id/et\_city"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter your city"  android:inputType="text"  android:layout\_marginBottom="16dp" />   <!-- Save Button -->  <Button  android:id="@+id/btn\_save"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Save Data"  android:layout\_marginBottom="8dp" />   <!-- Load Button -->  <Button  android:id="@+id/btn\_load"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Load Data"  android:layout\_marginBottom="8dp" />   <!-- Clear Button -->  <Button  android:id="@+id/btn\_clear"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Clear Data" />  </LinearLayout> |

Output:

Save Data

Load Data Clear Data

**Task 2 : Working with SQLite**

MainActivity.kt

|  |
| --- |
| package com.example.sqlitedemo  import android.os.Bundle import android.widget.Button import android.widget.EditText import android.widget.TextView import androidx.appcompat.app.AppCompatActivity import com.example.sqlitedemo.com.example.sqlitedemo.DatabaseHelper  class MainActivity : AppCompatActivity() {   private lateinit var dbHelper: DatabaseHelper   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   dbHelper = DatabaseHelper(this)   val addUserButton = findViewById<Button>(R.id.*btn\_add*)  val displayUsersButton = findViewById<Button>(R.id.*btn\_view*)  val nameEditText = findViewById<EditText>(R.id.*et\_name*)  val ageEditText = findViewById<EditText>(R.id.*et\_age*)  val displayTextView = findViewById<TextView>(R.id.*tv\_result*)   addUserButton.setOnClickListener **{** val name = nameEditText.*text*.toString()  val age = ageEditText.*text*.toString().*toIntOrNull*()  if (!name.*isBlank*() && age != null) {  dbHelper.addUser(name, age)  nameEditText.*text*.clear()  ageEditText.*text*.clear()  }  **}** displayUsersButton.setOnClickListener **{** val users = dbHelper.getAllUsers()  displayTextView.*text* = users.*joinToString*("\n")  **}** } } |

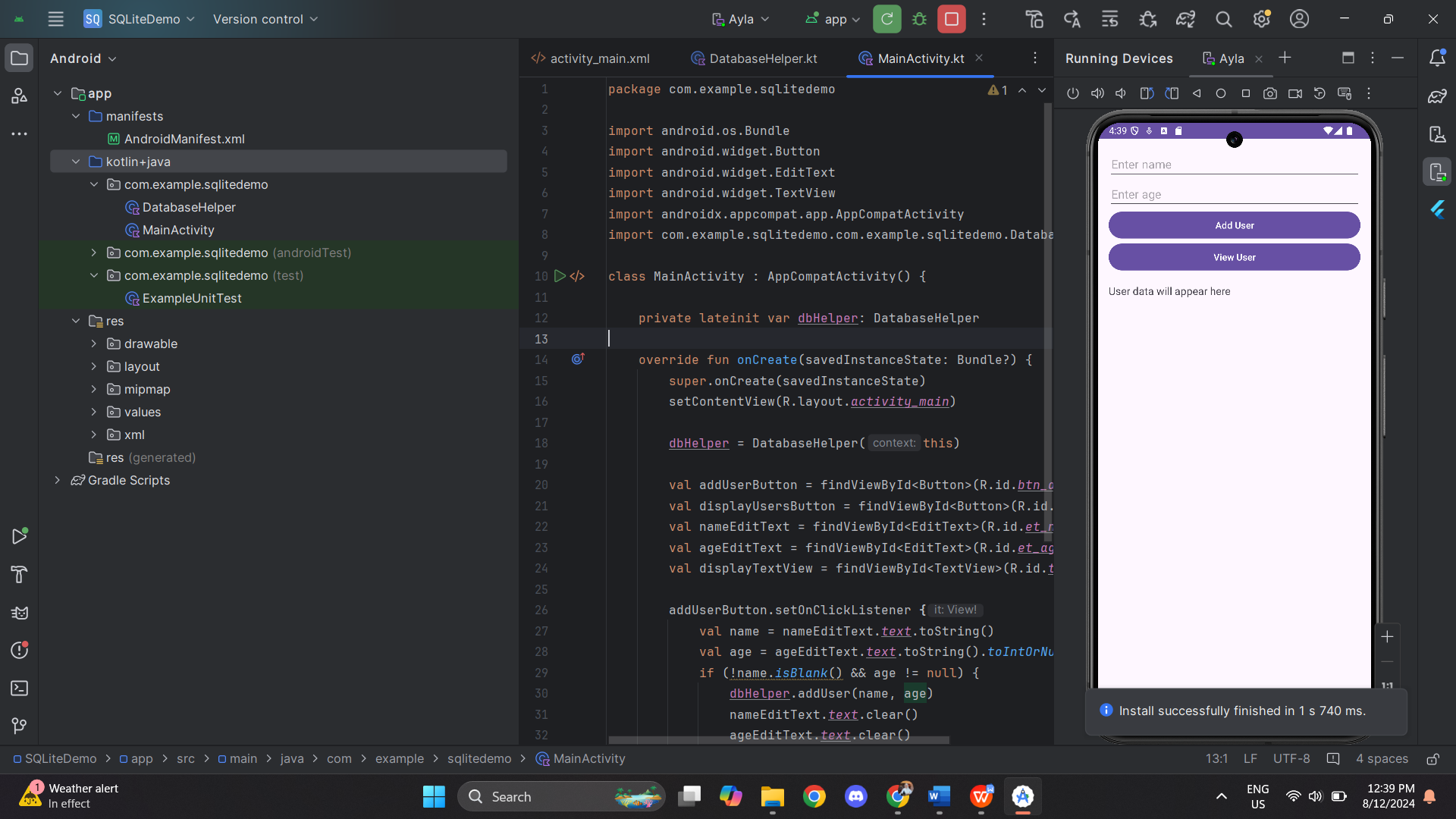
DatabaseHelper.kt

|  |
| --- |
| package com.example.sqlitedemo.com.example.sqlitedemo  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 = "UserDatabase"  private const val DATABASE\_VERSION = 1  private const val TABLE\_USERS = "Users"  private const val COLUMN\_ID = "id"  private const val COLUMN\_NAME = "name"  private const val COLUMN\_AGE = "age"  }   override fun onCreate(db: SQLiteDatabase?) {  val createTable = ("CREATE TABLE $TABLE\_USERS ("  + "$COLUMN\_ID INTEGER PRIMARY KEY AUTOINCREMENT, "  + "$COLUMN\_NAME TEXT, "  + "$COLUMN\_AGE INTEGER)")  db?.execSQL(createTable)  }   override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  db?.execSQL("DROP TABLE IF EXISTS $TABLE\_USERS")  onCreate(db)  }   fun addUser(name: String, age: Int): Boolean {  val db = this.*writableDatabase* val contentValues = ContentValues().*apply* **{** put(COLUMN\_NAME, name)  put(COLUMN\_AGE, age)  **}** val result = db.insert(TABLE\_USERS, null, contentValues)  db.close()  return result != -1L  }   fun getAllUsers(): List<String> {  val userList = ArrayList<String>()  val db = this.*readableDatabase* val cursor = db.rawQuery("SELECT \* FROM $TABLE\_USERS", null)   if (cursor.moveToFirst()) {  do {  val name = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_NAME))  val age = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_AGE))  userList.add("Name: $name, Age: $age")  } while (cursor.moveToNext())  }  cursor.close()  db.close()  return userList  } } |

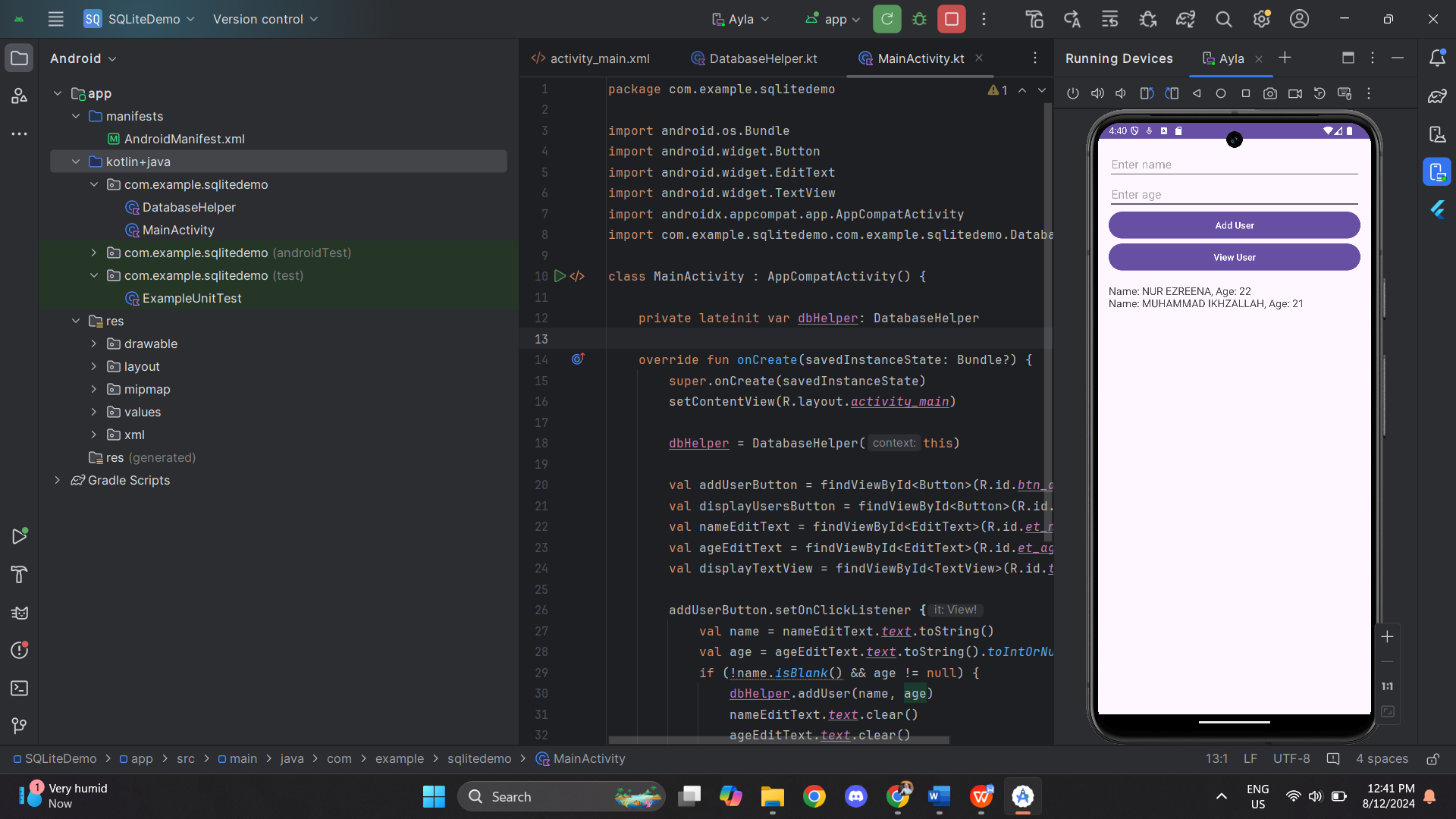
Activity\_main.xml

|  |
| --- |
| <?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 for User Name -->  <EditText  android:id="@+id/et\_name"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter name"  android:inputType="textPersonName" />   <!-- EditText for User Age -->  <EditText  android:id="@+id/et\_age"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter age"  android:inputType="number" />   <!-- Button to Add User -->  <Button  android:id="@+id/btn\_add"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Add User" />   <!-- Button to View User -->  <Button  android:id="@+id/btn\_view"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="View User" />   <!-- TextView to Display User Data -->  <TextView  android:id="@+id/tv\_result"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="User data will appear here"  android:textSize="16sp"  android:paddingTop="16dp" />  </LinearLayout> |

Output:



Key in the data at add user and click button View user



**Exercises**

MainActivity.kt

|  |
| --- |
| package com.example.sqlitedemo  import android.os.Bundle import android.widget.Button import android.widget.EditText import android.widget.TextView import androidx.appcompat.app.AppCompatActivity  class MainActivity : AppCompatActivity() {   private lateinit var dbHelper: DatabaseHelper   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   dbHelper = DatabaseHelper(this)   val addUserButton = findViewById<Button>(R.id.*btn\_add*)  val displayUsersButton = findViewById<Button>(R.id.*btn\_view*)  val updateUserButton = findViewById<Button>(R.id.*btn\_update*)  val deleteUserButton = findViewById<Button>(R.id.*btn\_delete*)  val filterUsersButton = findViewById<Button>(R.id.*btn\_filter*)  val nameEditText = findViewById<EditText>(R.id.*et\_name*)  val ageEditText = findViewById<EditText>(R.id.*et\_age*)  val idEditText = findViewById<EditText>(R.id.*et\_id*)  val displayTextView = findViewById<TextView>(R.id.*tv\_result*)   // Add User  addUserButton.setOnClickListener **{** val name = nameEditText.*text*.toString()  val age = ageEditText.*text*.toString().*toIntOrNull*()  if (!name.*isBlank*() && age != null) {  dbHelper.addUser(name, age)  nameEditText.*text*.clear()  ageEditText.*text*.clear()  }  **}** // Display Users  displayUsersButton.setOnClickListener **{** val users = dbHelper.getAllUsers()  displayTextView.*text* = users.*joinToString*("\n")  **}** // Update User  updateUserButton.setOnClickListener **{** val id = idEditText.*text*.toString().*toIntOrNull*()  val name = nameEditText.*text*.toString()  val age = ageEditText.*text*.toString().*toIntOrNull*()  if (id != null && !name.*isBlank*() && age != null) {  if (dbHelper.updateUser(id, name, age)) {  displayTextView.*text* = "User Updated"  } else {  displayTextView.*text* = "Failed to update user"  }  }  **}** // Delete User  deleteUserButton.setOnClickListener **{** val id = idEditText.*text*.toString().*toIntOrNull*()  if (id != null) {  if (dbHelper.deleteUser(id)) {  displayTextView.*text* = "User Deleted"  } else {  displayTextView.*text* = "Failed to delete user"  }  }  **}** // Filter Users by Age  filterUsersButton.setOnClickListener **{** val minAge = ageEditText.*text*.toString().*toIntOrNull*()  if (minAge != null) {  val filteredUsers = dbHelper.getUsersByAge(minAge)  displayTextView.*text* = filteredUsers.*joinToString*("\n")  }  **}** } } |

DatabaseHelper.kt

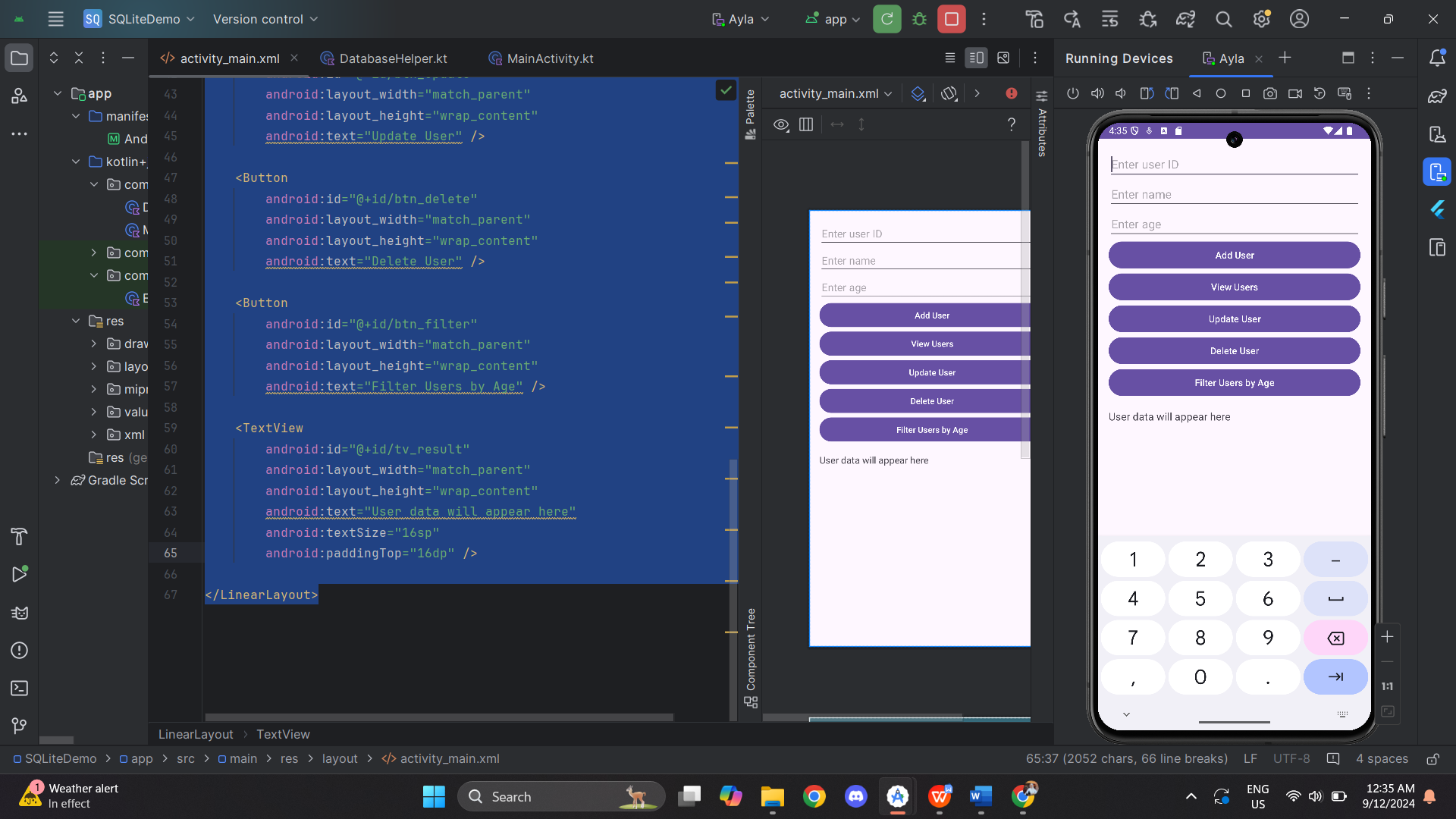
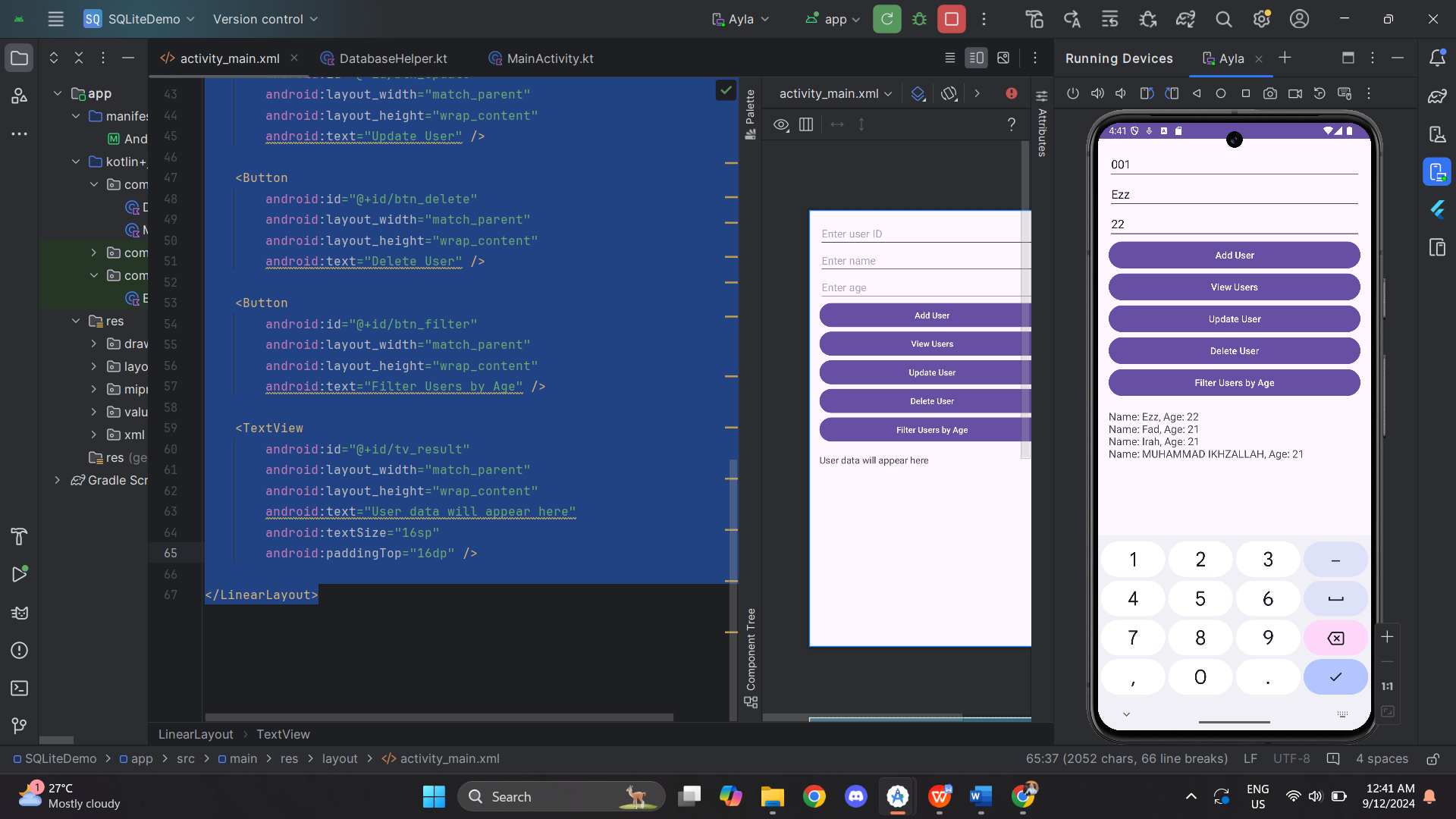
|  |
| --- |
| package com.example.sqlitedemo  import android.content.ContentValues import android.content.Context import android.database.sqlite.SQLiteDatabase import android.database.sqlite.SQLiteOpenHelper  class DatabaseHelper(context: Context) : SQLiteOpenHelper(context, DATABASE\_NAME, null, DATABASE\_VERSION) {   companion object {  private const val DATABASE\_NAME = "UserDatabase"  private const val DATABASE\_VERSION = 1  private const val TABLE\_USERS = "Users"  private const val COLUMN\_ID = "id"  private const val COLUMN\_NAME = "name"  private const val COLUMN\_AGE = "age"  }   override fun onCreate(db: SQLiteDatabase?) {  val createTable = ("CREATE TABLE $TABLE\_USERS (" +  "$COLUMN\_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +  "$COLUMN\_NAME TEXT, " +  "$COLUMN\_AGE INTEGER)")  db?.execSQL(createTable)  }   override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  db?.execSQL("DROP TABLE IF EXISTS $TABLE\_USERS")  onCreate(db)  }   // Add a new user  fun addUser(name: String, age: Int): Boolean {  val db = this.*writableDatabase* val contentValues = ContentValues().*apply* **{** put(COLUMN\_NAME, name)  put(COLUMN\_AGE, age)  **}** val result = db.insert(TABLE\_USERS, null, contentValues)  db.close()  return result != -1L  }   // Get all users sorted alphabetically  fun getAllUsers(): List<String> {  val userList = ArrayList<String>()  val db = this.*readableDatabase* val cursor = db.rawQuery("SELECT \* FROM $TABLE\_USERS ORDER BY $COLUMN\_NAME ASC", null)   if (cursor.moveToFirst()) {  do {  val name = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_NAME))  val age = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_AGE))  userList.add("Name: $name, Age: $age")  } while (cursor.moveToNext())  }  cursor.close()  db.close()  return userList  }   // Update a user's details  fun updateUser(id: Int, name: String, age: Int): Boolean {  val db = this.*writableDatabase* val contentValues = ContentValues().*apply* **{** put(COLUMN\_NAME, name)  put(COLUMN\_AGE, age)  **}** val result = db.update(TABLE\_USERS, contentValues, "$COLUMN\_ID = ?", *arrayOf*(id.toString()))  db.close()  return result > 0  }   // Delete a user by ID  fun deleteUser(id: Int): Boolean {  val db = this.*writableDatabase* val result = db.delete(TABLE\_USERS, "$COLUMN\_ID = ?", *arrayOf*(id.toString()))  db.close()  return result > 0  }   // Filter users by age  fun getUsersByAge(minAge: Int): List<String> {  val userList = ArrayList<String>()  val db = this.*readableDatabase* val cursor = db.rawQuery("SELECT \* FROM $TABLE\_USERS WHERE $COLUMN\_AGE >= ?", *arrayOf*(minAge.toString()))   if (cursor.moveToFirst()) {  do {  val name = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_NAME))  val age = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_AGE))  userList.add("Name: $name, Age: $age")  } while (cursor.moveToNext())  }  cursor.close()  db.close()  return userList  } } |

Activity\_main.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/et\_id"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter user ID"  android:inputType="number" />   <EditText  android:id="@+id/et\_name"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter name"  android:inputType="textPersonName" />   <EditText  android:id="@+id/et\_age"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter age"  android:inputType="number" />   <Button  android:id="@+id/btn\_add"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Add User" />   <Button  android:id="@+id/btn\_view"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="View Users" />   <Button  android:id="@+id/btn\_update"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Update User" />   <Button  android:id="@+id/btn\_delete"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Delete User" />   <Button  android:id="@+id/btn\_filter"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Filter Users by Age" />   <TextView  android:id="@+id/tv\_result"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="User data will appear here"  android:textSize="16sp"  android:paddingTop="16dp" />  </LinearLayout> |

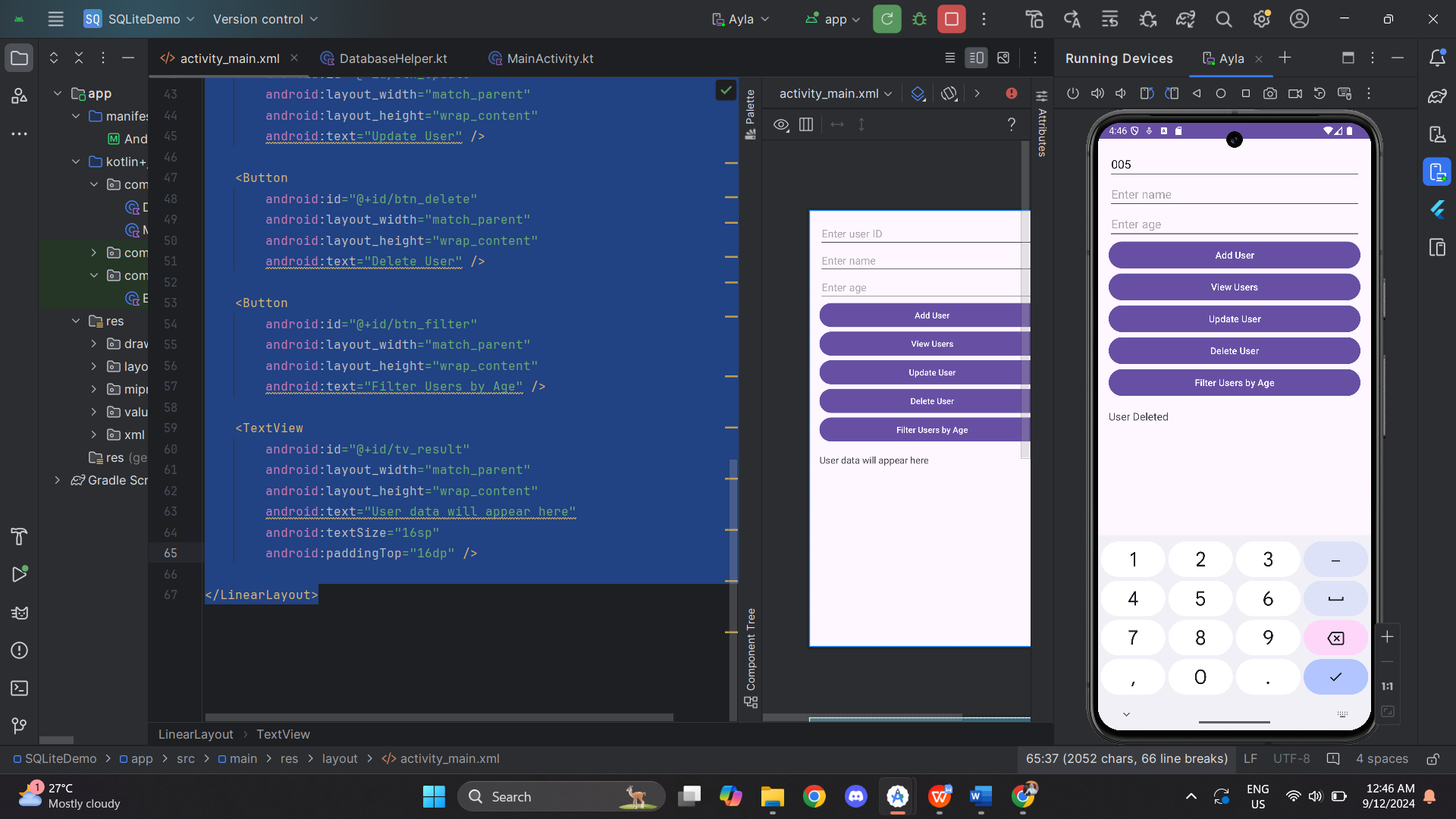
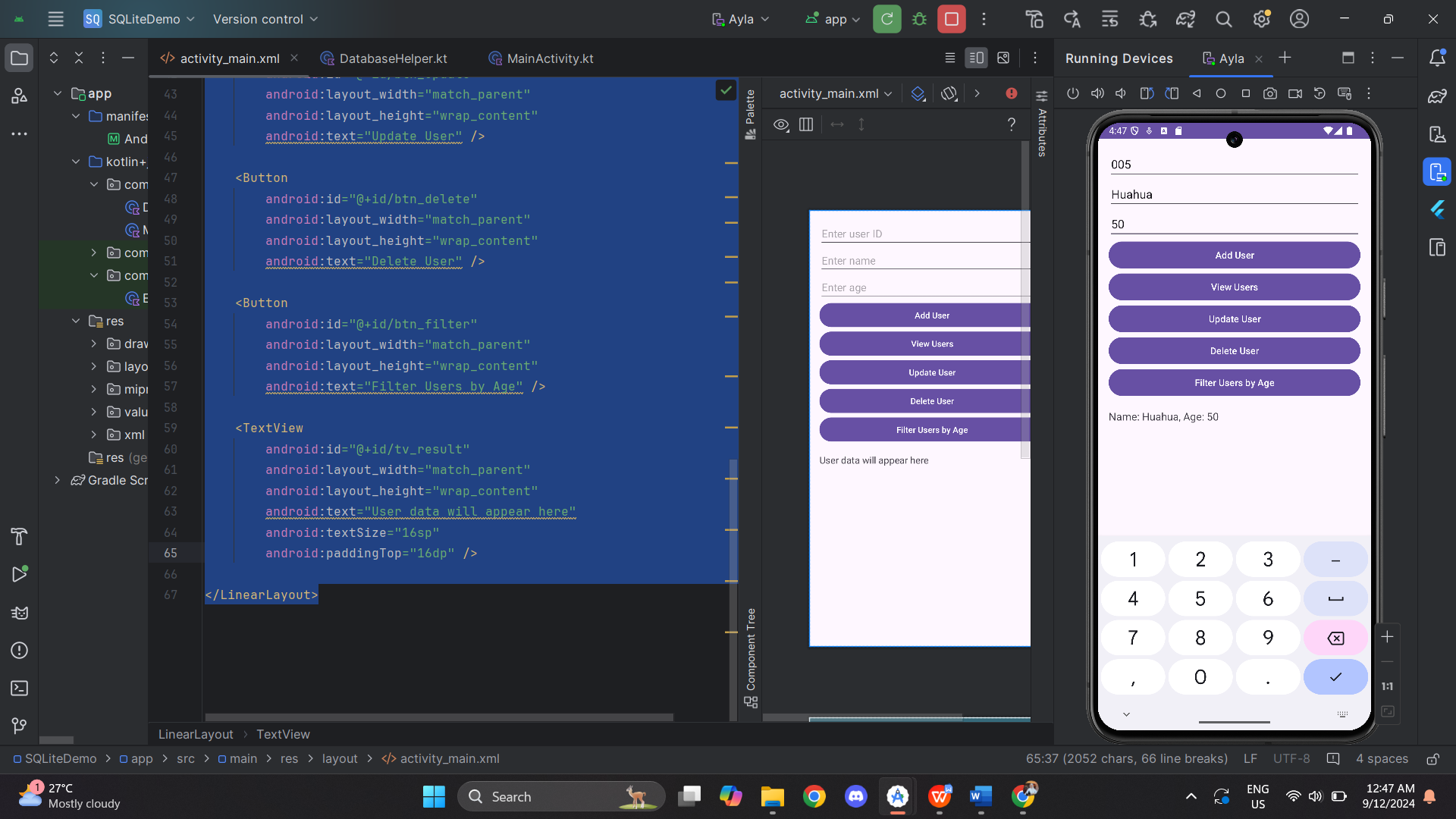
Output:

View User



Filter User by Age

Delete User

**Task 3 : Working with SQLite and RecyclerView**

MainActivity.kt

|  |
| --- |
| package com.example.recyclerviewsqlitedemo  import android.os.Bundle import android.widget.Button import android.widget.EditText import android.widget.Toast import androidx.appcompat.app.AlertDialog import androidx.appcompat.app.AppCompatActivity import androidx.recyclerview.widget.LinearLayoutManager import androidx.recyclerview.widget.RecyclerView  class MainActivity : AppCompatActivity() {   private lateinit var databaseHelper: DatabaseHelper  private lateinit var userAdapter: UserAdapter  private lateinit var recyclerView: RecyclerView  private lateinit var addUserButton: Button   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   // Initialize views  databaseHelper = DatabaseHelper(this)  recyclerView = findViewById(R.id.*recycler\_view*)  addUserButton = findViewById(R.id.*btn\_add\_user*)   // Set up button click listener  addUserButton.setOnClickListener **{** showAddUserDialog()  **}** // Set up RecyclerView  setupRecyclerView()  loadUserData()  }   private fun setupRecyclerView() {  recyclerView.*layoutManager* = LinearLayoutManager(this)  }   private fun loadUserData() {  val users = databaseHelper.getAllUsers()  userAdapter = UserAdapter(users)  recyclerView.*adapter* = userAdapter  }   private fun showAddUserDialog() {  val builder = AlertDialog.Builder(this)  builder.setTitle("Add User")   val inflater = *layoutInflater* val dialogLayout = inflater.inflate(R.layout.*dialog\_add\_user*, null)  builder.setView(dialogLayout)   val etName = dialogLayout.findViewById<EditText>(R.id.*et\_name*)  val etAge = dialogLayout.findViewById<EditText>(R.id.*et\_age*)   builder.setPositiveButton("Add") **{** dialog, \_ **->** val name = etName.*text*.toString()  val age = etAge.*text*.toString().*toIntOrNull*()   if (name.*isNotEmpty*() && age != null) {  databaseHelper.addUser(name, age)  loadUserData()  Toast.makeText(this, "User added", Toast.*LENGTH\_SHORT*).show()  } else {  Toast.makeText(this, "Invalid input", Toast.*LENGTH\_SHORT*).show()  }  dialog.dismiss()  **}** builder.setNegativeButton("Cancel") **{** dialog, \_ **->** dialog.dismiss()  **}** builder.show()  } } |

DatabaseHelper.kt

|  |
| --- |
| package com.example.recyclerviewsqlitedemo  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 = "users.db"  private const val DATABASE\_VERSION = 1   // Table and column names  private const val TABLE\_USERS = "users"  private const val COLUMN\_ID = "id"  private const val COLUMN\_NAME = "name"  private const val COLUMN\_AGE = "age"  }   override fun onCreate(db: SQLiteDatabase) {  // Create the table  val createTable = """  CREATE TABLE $TABLE\_USERS (  $COLUMN\_ID INTEGER PRIMARY KEY AUTOINCREMENT,  $COLUMN\_NAME TEXT NOT NULL,  $COLUMN\_AGE INTEGER NOT NULL  )  """.*trimIndent*()  db.execSQL(createTable)  }   override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {  // Drop the existing table if it exists  db.execSQL("DROP TABLE IF EXISTS $TABLE\_USERS")  onCreate(db)  }   // Add a new user  fun addUser(name: String, age: Int): Long {  val db = *writableDatabase* val values = ContentValues().*apply* **{** put(COLUMN\_NAME, name)  put(COLUMN\_AGE, age)  **}** val result = db.insert(TABLE\_USERS, null, values)  db.close()  return result  }   // Retrieve all users  fun getAllUsers(): List<User> {  val userList = *mutableListOf*<User>()  val db = *readableDatabase* val query = "SELECT \* FROM $TABLE\_USERS"  val cursor = db.rawQuery(query, null)   if (cursor.moveToFirst()) {  do {  val id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_ID))  val name = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_NAME))  val age = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_AGE))  userList.add(User(id, name, age)) // Adjusted for id field  } while (cursor.moveToNext())  }   cursor.close()  db.close()  return userList  }   // Delete a user by ID (optional functionality)  fun deleteUser(id: Int): Int {  val db = *writableDatabase* val result = db.delete(TABLE\_USERS, "$COLUMN\_ID = ?", *arrayOf*(id.toString()))  db.close()  return result  } } |

User.kt

|  |
| --- |
| package com.example.recyclerviewsqlitedemo   data class User (  val id: Int,  val name: String,  val age: Int ) |

UserAdapter.kt

|  |
| --- |
| package com.example.recyclerviewsqlitedemo  import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import android.widget.TextView import androidx.recyclerview.widget.RecyclerView  class UserAdapter(private val userList: List<User>) : RecyclerView.Adapter<UserAdapter.UserViewHolder>() {   override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): UserViewHolder {  val view = LayoutInflater.from(parent.*context*).inflate(R.layout.*user\_item*, parent, false)  return UserViewHolder(view)  }   override fun onBindViewHolder(holder: UserViewHolder, position: Int) {  val user = userList[position]  holder.nameTextView.*text* = user.name  holder.ageTextView.*text* = user.age.toString()  }   override fun getItemCount(): Int {  return userList.size  }   class UserViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  val nameTextView: TextView = itemView.findViewById(R.id.*tv\_name*)  val ageTextView: TextView = itemView.findViewById(R.id.*tv\_age*)  } } |

Activity\_main.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:gravity="center"  tools:context=".MainActivity">   <!-- Button centered -->  <Button  android:id="@+id/btn\_add\_user"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Add User"  android:layout\_gravity="center"  android:layout\_marginTop="100dp"/>   <!-- RecyclerView below the Button -->  <androidx.recyclerview.widget.RecyclerView  android:id="@+id/recycler\_view"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingTop="16dp"/>  </LinearLayout> |

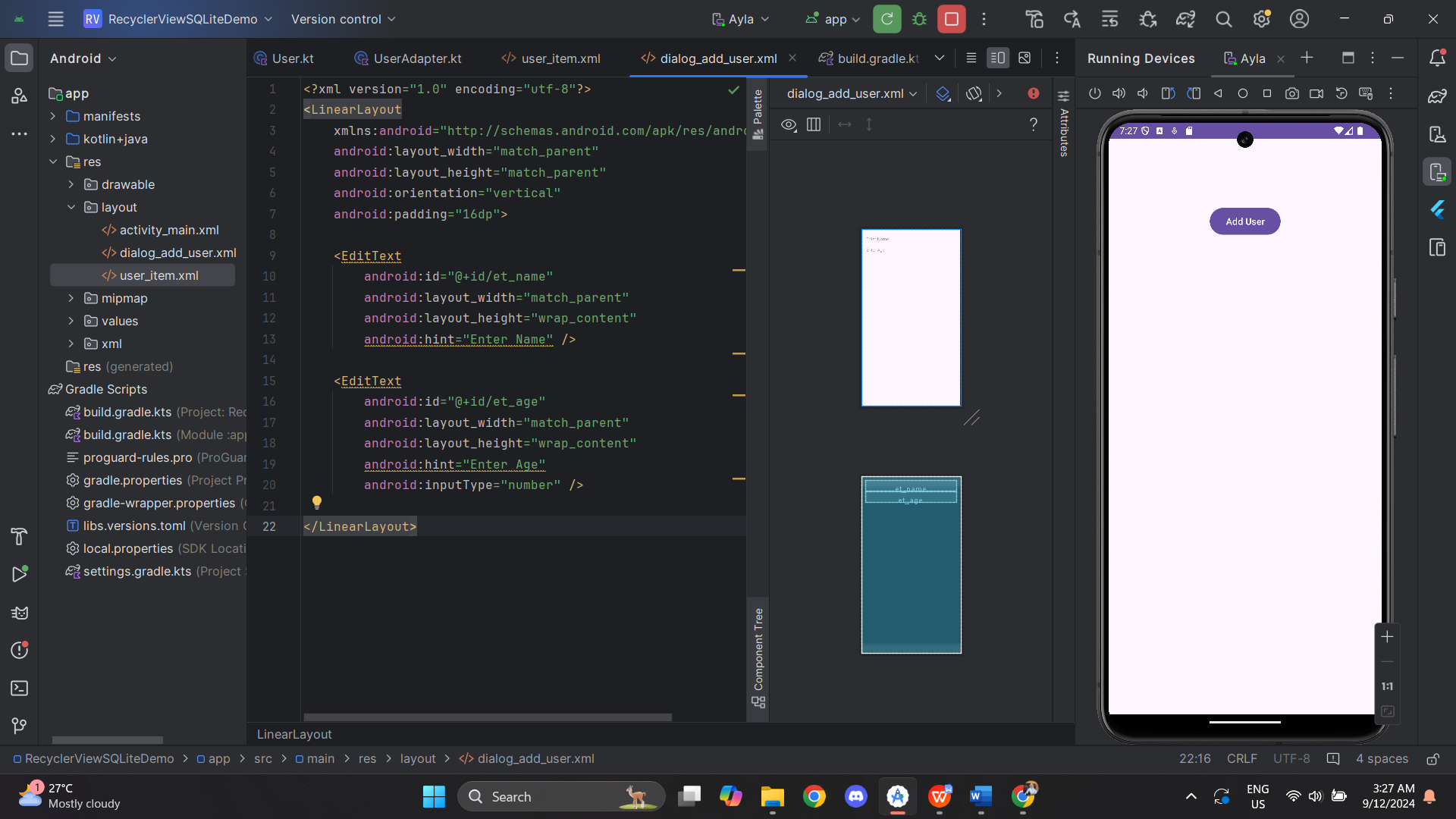
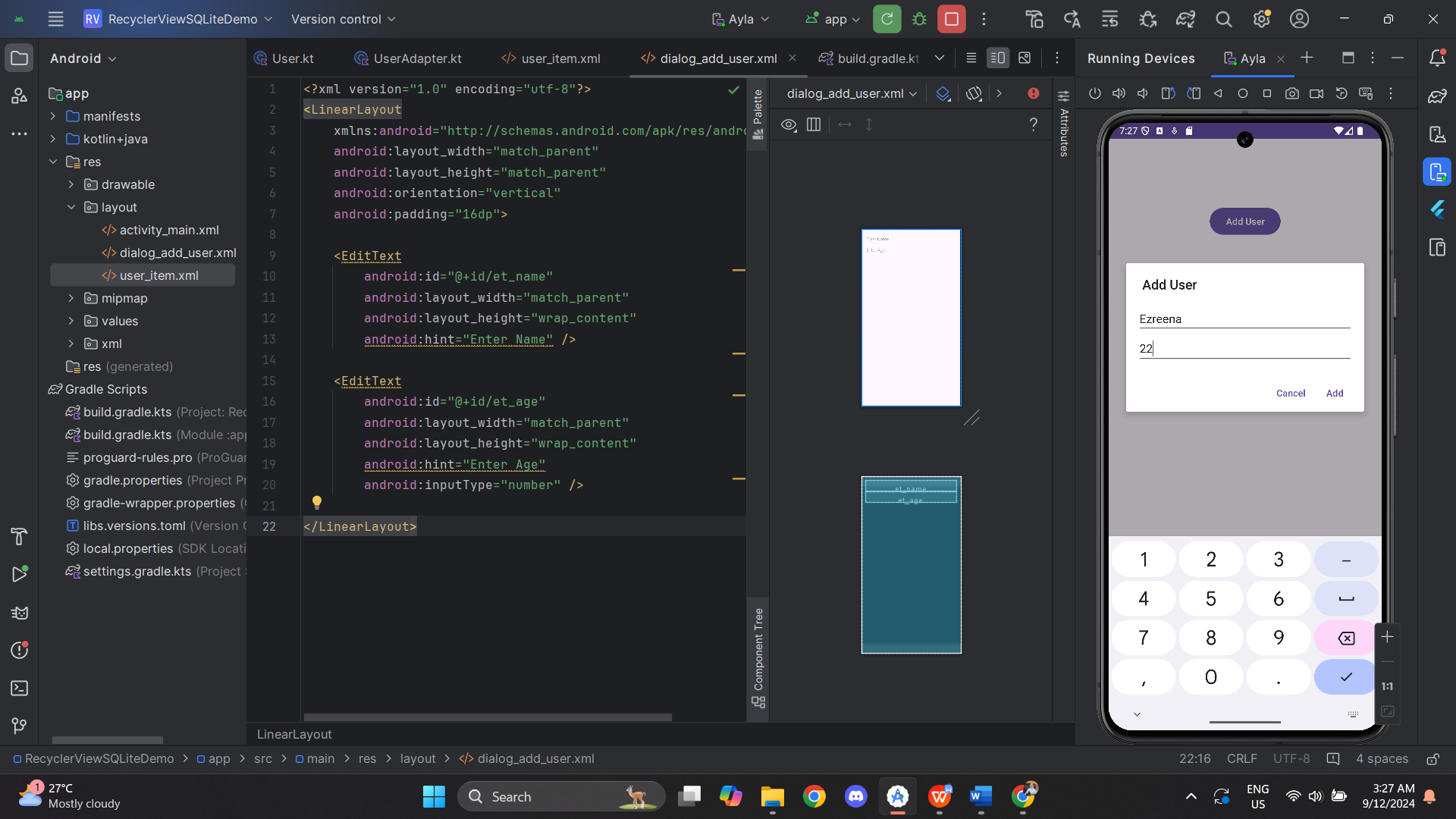
User\_item.xml

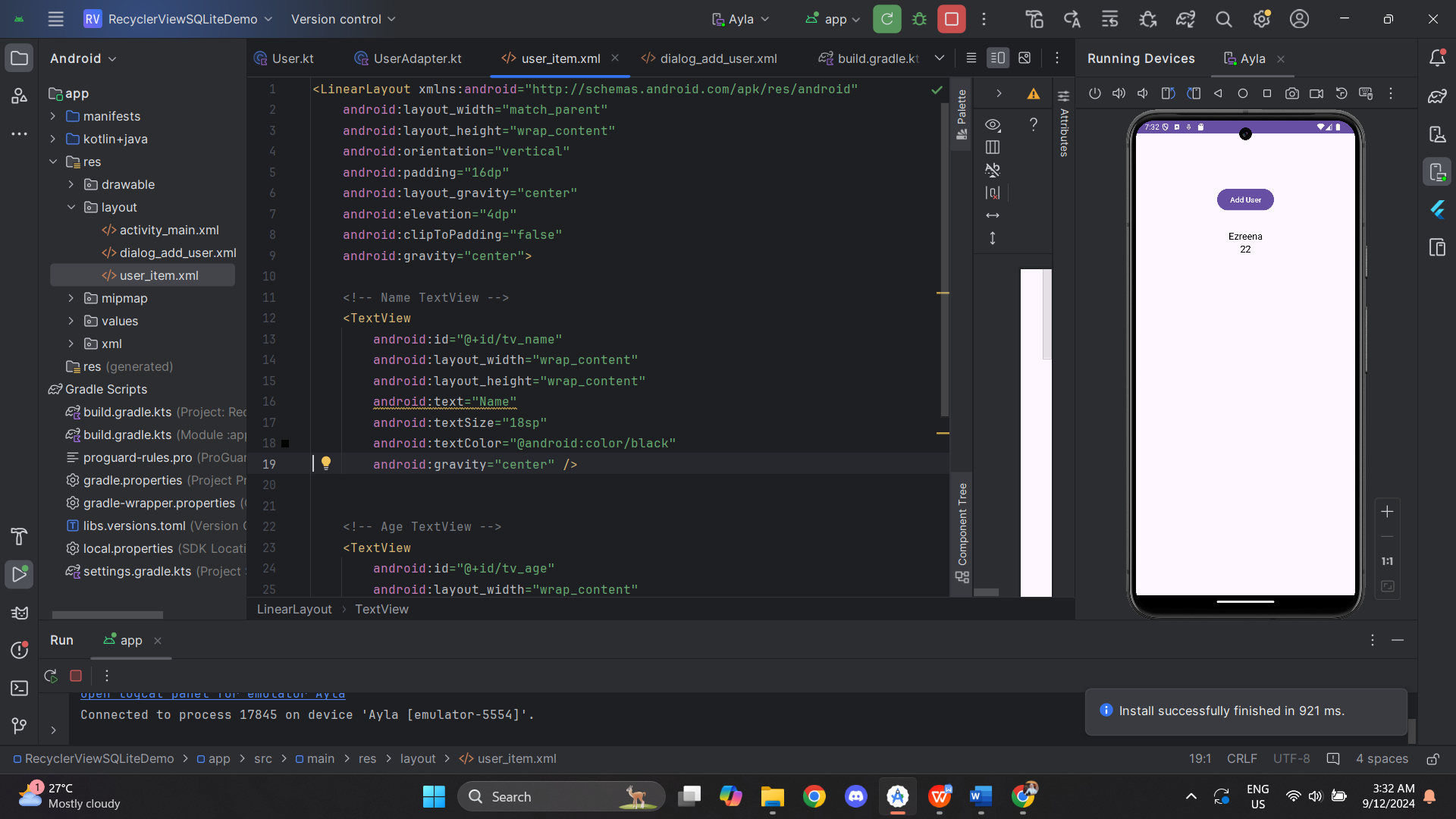
|  |
| --- |
| <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:padding="16dp"  android:layout\_gravity="center"  android:elevation="4dp"  android:clipToPadding="false"  android:gravity="center">     <TextView  android:id="@+id/tv\_name"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Name"  android:textSize="18sp"  android:textColor="@android:color/black"  android:gravity="center" />    <!-- Age TextView -->  <TextView  android:id="@+id/tv\_age"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Age"  android:textSize="18sp"  android:textColor="@android:color/black"  android:fontFamily="sans-serif"  android:gravity="center" /> <!-- Centers the text -->  </LinearLayout> |

Dialog\_add\_user.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/et\_name"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter Name" />   <EditText  android:id="@+id/et\_age"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="Enter Age"  android:inputType="number" />  </LinearLayout> |

Output:



**Task 4 : Working With Fragments**

MainActivity.kt

|  |
| --- |
| package com.example.fragmentexample  import android.os.Bundle import androidx.appcompat.app.AppCompatActivity import androidx.fragment.app.Fragment import androidx.fragment.app.FragmentTransaction  class MainActivity : AppCompatActivity() {   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   // Dynamically add the fragment to the Activity  if (savedInstanceState == null) {  val myFragment = MyFragment()  val transaction: FragmentTransaction = *supportFragmentManager*.beginTransaction()  transaction.replace(R.id.fragment, myFragment)  transaction.commit()  }  } } |

MyFragment.kt

|  |
| --- |
| package com.example.fragmentexample  import android.os.Bundle import android.view.LayoutInflater import android.view.View import android.view.ViewGroup import androidx.fragment.app.Fragment  class MyFragment : Fragment() {   override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  // Inflate the fragment's layout  return inflater.inflate(R.layout.*fragment\_my*, container, false)  } } |

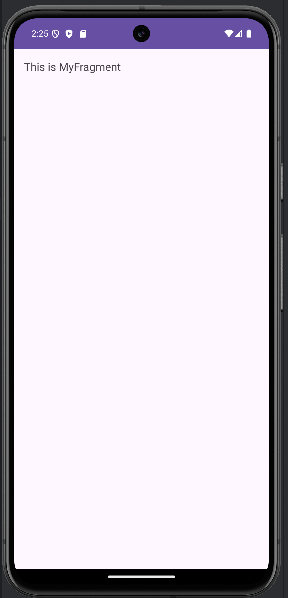
Activity\_main.xml

|  |
| --- |
| <?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">   <!-- Container for dynamically loaded fragments -->  <FrameLayout  android:id="@+id/fragment\_container"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" />  </RelativeLayout> |

Fragment\_my.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:padding="16dp">   <TextView  android:id="@+id/fragment\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="This is MyFragment"  android:textSize="18sp"/>  </LinearLayout> |

Output:



**Handling Fragment Lifecycle**

MainActivity.kt

|  |
| --- |
| package com.example.fragmentexample  import android.os.Bundle  import androidx.appcompat.app.AppCompatActivity  import androidx.fragment.app.FragmentTransaction  class MainActivity : AppCompatActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_main)  // Dynamically add the fragment to the Activity  if (savedInstanceState == null) {  val myFragment = MyFragment()  val transaction: FragmentTransaction = supportFragmentManager.beginTransaction()  transaction.replace(R.id.fragment\_container, myFragment)  transaction.commit()  }  }  } |

MyFragment.kt

|  |
| --- |
| package com.example.fragmentexample  import android.os.Bundle  import android.util.Log  import android.view.LayoutInflater  import android.view.View  import android.view.ViewGroup  import androidx.fragment.app.Fragment  class MyFragment : Fragment() {  private val TAG = "MyFragment"  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  Log.d(TAG, "onCreate")  }  override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  Log.d(TAG, "onCreateView")  return inflater.inflate(R.layout.fragment\_my, container, false)  }  override fun onStart() {  super.onStart()  Log.d(TAG, "onStart")  }  override fun onResume() {  super.onResume()  Log.d(TAG, "onResume")  }  override fun onPause() {  super.onPause()  Log.d(TAG, "onPause")  }  override fun onStop() {  super.onStop()  Log.d(TAG, "onStop")  }  override fun onDestroyView() {  super.onDestroyView()  Log.d(TAG, "onDestroyView")  }  override fun onDestroy() {  super.onDestroy()  Log.d(TAG, "onDestroy")  }  override fun onDetach() {  super.onDetach()  Log.d(TAG, "onDetach")  }  } |

Activity\_main.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">  <!-- Container for dynamically loaded fragments -->  <FrameLayout  android:id="@+id/fragment\_container"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" />  </RelativeLayout> |

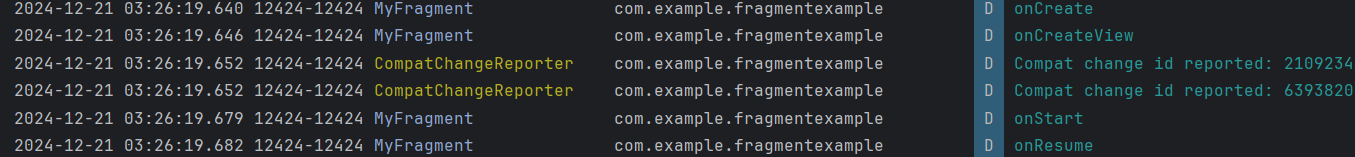
Fragment\_my.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:padding="16dp">  <TextView  android:id="@+id/fragment\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="This is MyFragment"  android:textSize="18sp"/>  </LinearLayout> |

Output:



On LogCat:



**Communicating Between Fragments**

MainActivity.kt

|  |
| --- |
| package com.example.fragmentexample  import android.os.Bundle  import androidx.appcompat.app.AppCompatActivity  class MainActivity : AppCompatActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_main)  // Load MyFragment by default  supportFragmentManager.beginTransaction()  .replace(R.id.fragment\_container, MyFragment())  .commit()  }  fun sendDataToFragment2(data: String) {  val fragment2 = Fragment2()  // Replace MyFragment with Fragment2  supportFragmentManager.beginTransaction()  .replace(R.id.fragment\_container, fragment2)  .commit()  }  } |

MyFragment.kt

|  |
| --- |
| package com.example.fragmentexample  import android.os.Bundle  import android.view.LayoutInflater  import android.view.View  import android.view.ViewGroup  import android.widget.Button  import androidx.fragment.app.Fragment  class MyFragment : Fragment() {  override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  val view = inflater.inflate(R.layout.fragment\_my, container, false)  // Find the button and set its click listener  val button: Button = view.findViewById(R.id.button\_send)  button.setOnClickListener {  (activity as? MainActivity)?.sendDataToFragment2("Hello from MyFragment")  }  return view  }  } |

Fragment2.kt

|  |
| --- |
| package com.example.fragmentexample  import android.os.Bundle  import android.view.LayoutInflater  import android.view.View  import android.view.ViewGroup  import android.widget.Button  import android.widget.TextView  import androidx.fragment.app.Fragment  class Fragment2 : Fragment() {  override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  val view = inflater.inflate(R.layout.fragment2, container, false)  // Get the data from arguments  val message = arguments?.getString("message")  // Display the message in the TextView  val textView: TextView = view.findViewById(R.id.display\_message)  textView.text = message  // Button action  val button: Button = view.findViewById(R.id.button\_action)  button.setOnClickListener {  // Perform any action here when the button is clicked  textView.text = "Action performed!"  }  return view  }  } |

Activity\_main.xml

|  |
| --- |
| <RelativeLayout  xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <FrameLayout  android:id="@+id/fragment\_container"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" />  </RelativeLayout> |

Fragment\_my.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 displaying a message -->  <TextView  android:id="@+id/fragment\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="This is MyFragment"  android:textSize="18sp"  android:textColor="#000000"  android:gravity="center" /> <!-- Optional: Center alignment -->  <!-- Button to send data to MainActivity -->  <Button  android:id="@+id/button\_send"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Send Data to Fragment 2" />  </LinearLayout> |

Fragment2.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"  tools:context=".Fragment2">  <!-- Title TextView -->  <TextView  android:id="@+id/title\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hi welcome my fragment 2!"  android:textSize="22sp"  android:textColor="#333333"  android:fontFamily="sans-serif-medium"  android:layout\_marginTop="32dp"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent" />  <!-- TextView to display the message -->  <TextView  android:id="@+id/display\_message"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="You made it!"  android:textSize="18sp"  android:textColor="#666666"  android:layout\_marginTop="16dp"  android:layout\_marginStart="32dp"  android:layout\_marginEnd="32dp"  app:layout\_constraintTop\_toBottomOf="@id/title\_text"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent" />  <!-- Button to perform an action -->  <Button  android:id="@+id/button\_action"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="click on me"  android:backgroundTint="#6200EE"  android:textColor="#FFFFFF"  android:layout\_marginTop="24dp"  app:layout\_constraintTop\_toBottomOf="@id/display\_message"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> |

Output: