**Pract 9th:**

**Database Programming with SQLite**

**Main Activity.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"

tools:context=".MainActivity">

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/studentId"

android:ems="10"

android:inputType="number"

android:hint="Student ID"/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/studentName"

android:ems="10"

android:inputType="textPersonName"

android:hint="Student Name"/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/studentAge"

android:ems="10"

android:inputType="number"

android:hint="Student Age"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="horizontal">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/addStudent"

android:onClick="addStudent"

android:text="ADD"/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/updateStudent"

android:onClick="updateStudent"

android:text="UPDATE"

android:layout\_marginLeft="20dp"/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/deleteStudent"

android:onClick="deleteStudent"

android:text="DELETE"

android:layout\_marginLeft="20dp"/>

</LinearLayout>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:layout\_marginLeft="130dp"

android:text="Load Details"

android:onClick="loadStudent"/>

<ListView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:id="@+id/studentListView"

android:layout\_weight="1"

android:divider="@android:color/transparent"

android:dividerHeight="5.0sp"/>

</LinearLayout>

**Main Activity.kt**

package com.example.prac9

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.os.Handler

import android.view.View

import android.widget.AdapterView

import android.widget.ArrayAdapter

import android.widget.EditText

import android.widget.ListView

import android.widget.Toast

class MainActivity : AppCompatActivity() {

private lateinit var listView: ListView

private lateinit var studentId: EditText

private lateinit var studentName: EditText

private lateinit var studentAge: EditText

private lateinit var dbHandler: MyDBHandler

private lateinit var adapter: ArrayAdapter<Student>

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

listView= findViewById(R.id.studentListView)

studentId= findViewById(R.id.studentId)

studentAge = findViewById(R.id.studentAge)

studentName =findViewById(R.id.studentName)

dbHandler = MyDBHandler(this)

adapter= ArrayAdapter(this,android.R.layout.simple\_list\_item\_1,dbHandler.loadHandler())

listView.adapter=adapter

listView.onItemClickListener = AdapterView.OnItemClickListener{

\_, \_, position,\_ ->

val student =adapter.getItem(position)

studentId.setText(student?.id.toString())

studentName.setText(student?.name)

studentAge.setText(student?.age.toString())

}

}

fun addStudent(view: View) {

if (studentId.text.toString().isNotEmpty() &&

studentName.text.toString().isNotEmpty() &&

studentAge.text.toString().isNotEmpty()

) {

val id = studentId.text.toString().toInt()

val name = studentName.text.toString()

val age = studentAge.text.toString().toInt()

val student = Student(id, name, age)

val insertId = dbHandler.addHandler(student)

if (insertId == -1L) {

Toast.makeText(this, "Record Already Exists", Toast.LENGTH\_SHORT).show()

} else {

studentId.setText("")

studentAge.setText("")

studentName.setText("")

adapter.add(student)

adapter.notifyDataSetChanged()

Toast.makeText(this, "record added", Toast.LENGTH\_SHORT)

.show()

}

}else{

Toast.makeText(this, "Please fill correct id, name and address", Toast.LENGTH\_SHORT)

.show()

}

}

fun loadStudent(view: View){

adapter.clear()

adapter.addAll(dbHandler.loadHandler())

}

fun updateStudent(view: View) {

if (studentId.text.toString().isNotEmpty() &&

studentName.text.toString().isNotEmpty() &&

studentAge.text.toString().isNotEmpty()

) {

val id = studentId.text.toString().toInt()

val name = studentName.text.toString()

val age = studentAge.text.toString().toInt()

val student = Student(id, name, age)

val rowsAffected = dbHandler.updateHandler(student)

if (rowsAffected > 0) {

adapter.clear()

adapter.addAll(dbHandler.loadHandler())

Toast.makeText(this, "Record Updated", Toast.LENGTH\_SHORT).show()

} else {

Toast.makeText(this, "please fill the correct ID, Name and Age", Toast.LENGTH\_SHORT)

.show()

}

}

}

fun deleteStudent(view: View) {

if (studentId.text.toString().isNotEmpty()) {

val id = studentId.text.toString().toInt()

val rowsAffected = dbHandler.deleteHandler(id)

if (rowsAffected > 0) {

adapter.clear()

adapter.addAll(dbHandler.loadHandler())

studentId.setText("")

studentAge.setText("")

studentName.setText("")

Toast.makeText(this, "Record Deleted", Toast.LENGTH\_SHORT).show()

} else {

Toast.makeText(this, "please fill the correct ID, Name and Age", Toast.LENGTH\_SHORT)

.show()

}

}

}

}

**My DB handler.kt**

package com.example.prac9

import android.content.ContentValues

import android.content.Context

import android.database.sqlite.SQLiteDatabase

import android.database.sqlite.SQLiteOpenHelper

import android.security.identity.AccessControlProfileId

private const val DATABASE\_VERSION=1

private const val DATABASE\_NAME="myDatabase"

private const val TABLE\_NAME="students"

private const val COLUMN\_ID="id"

private const val COLUMN\_NAME="name"

private const val COLUMN\_AGE="age"

class MyDBHandler (context: Context):SQLiteOpenHelper(context, DATABASE\_NAME,null, DATABASE\_VERSION) {

override fun onCreate(db: SQLiteDatabase?) {

val CREATE\_TABLE="CREATE TABLE $TABLE\_NAME($COLUMN\_ID INTEGER PRIMARY KEY,$COLUMN\_NAME TEXT,$COLUMN\_AGE INTEGER)"

db?.execSQL(CREATE\_TABLE)

}

override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {

db?.execSQL("DROP TABLE IF EXISTS $TABLE\_NAME")

onCreate(db)

}

fun loadHandler():List<Student>{

val query = "SELECT \* FROM $TABLE\_NAME"

val db=this.readableDatabase

val cursor = db.rawQuery(query,null)

val students = mutableListOf<Student>()

if (cursor.moveToFirst()){

do{

val id = cursor.getInt(0)

val name = cursor.getString(1)

val age = cursor.getInt(2)

students.add(Student(id,name,age))

}while (cursor.moveToNext())

}

cursor.close()

db.close()

return students

}

fun addHandler(student: Student):Long{

val values = ContentValues()

values.put(COLUMN\_ID,student.id)

values.put(COLUMN\_NAME,student.name)

values.put(COLUMN\_AGE,student.age)

val db = this.writableDatabase

val result = db.insert(TABLE\_NAME,null,values)

db.close()

return result

}

fun updateHandler(student: Student):Int{

val values = ContentValues()

values.put(COLUMN\_NAME,student.name)

values.put(COLUMN\_AGE,student.age)

val db = this.writableDatabase

val result = db.update(TABLE\_NAME,values,"$COLUMN\_ID=?", arrayOf(student.id.toString())

)

db.close()

return result

}

fun deleteHandler(id: Int):Int{

val db = this.writableDatabase

val result= db.delete(TABLE\_NAME,"$COLUMN\_ID=?", arrayOf(id.toString())

)

db.close()

return result

}

}

**Student.kt**

package com.example.prac9

import android.security.identity.AccessControlProfileId

class Student(val id: Int,val name: String,val age: Int) {

override fun toString(): String {

return "ID: $id \nName: $name \nAge: $age"

}

}