Lab04  
  
MainActivity.java:  
package com.dtys.lab04.ui;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
import com.dtys.lab04.R;  
import com.dtys.lab04.data.database.DatabaseHandler;  
import com.dtys.lab04.data.model.Student;  
import com.dtys.lab04.ui.adapter.StudentAdapter;  
  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
  
 private RecyclerView recyclerView;  
 private StudentAdapter adapter;  
 private DatabaseHandler db;  
 private Button btnAdd;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 db = new DatabaseHandler(this);  
  
 recyclerView = findViewById(R.id.*recyclerViewStudents*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
 btnAdd = findViewById(R.id.*btn\_add*);  
  
 // Load dữ liệu ban đầu  
 loadStudents();  
  
 // Khi bấm nút "Thêm sinh viên" → mở AddStudentActivity  
 btnAdd.setOnClickListener(v -> {  
 Intent intent = new Intent(MainActivity.this, AddStudentActivity.class);  
 startActivity(intent);  
 });  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 // Reload lại danh sách mỗi khi quay lại MainActivity  
 loadStudents();  
 }  
  
 private void loadStudents() {  
 List<Student> studentList = db.getAllStudents();  
  
 adapter = new StudentAdapter(this, studentList, student -> {  
 // Khi bấm vào 1 sinh viên → mở StudentDetailActivity  
 Intent intent = new Intent(MainActivity.this, StudentDetailActivity.class);  
 intent.putExtra("student\_id", student.getId());  
 startActivity(intent);  
 });  
  
 recyclerView.setAdapter(adapter);  
 }  
}

DatabaseHandler.java  
package com.dtys.lab04.data.database;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import com.dtys.lab04.data.model.Student;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class DatabaseHandler extends SQLiteOpenHelper {  
 // Database Version  
 private static final int *DATABASE\_VERSION* = 1;  
  
 // Database Name  
 private static final String *DATABASE\_NAME* = "studentManager";  
  
 // Table name  
 private static final String *TABLE\_STUDENTS* = "students";  
  
 // Columns  
 private static final String *KEY\_ID* = "id";  
 private static final String *KEY\_NAME* = "name";  
 private static final String *KEY\_PHONE* = "phone\_number";  
 private static final String *KEY\_EMAIL* = "email";  
  
 public DatabaseHandler(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 // Create table  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String CREATE\_STUDENTS\_TABLE = "CREATE TABLE " + *TABLE\_STUDENTS* + "("  
 + *KEY\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
 + *KEY\_NAME* + " TEXT,"  
 + *KEY\_PHONE* + " TEXT,"  
 + *KEY\_EMAIL* + " TEXT" + ")";  
 db.execSQL(CREATE\_STUDENTS\_TABLE);  
 }  
  
 // Upgrade DB  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_STUDENTS*);  
 onCreate(db);  
 }  
  
 // Insert student  
 public void addStudent(Student student) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(*KEY\_NAME*, student.getName());  
 values.put(*KEY\_PHONE*, student.getPhoneNumber());  
 values.put(*KEY\_EMAIL*, student.getEmail());  
 db.insert(*TABLE\_STUDENTS*, null, values);  
 db.close();  
 }  
  
 // Get single student by ID  
 public Student getStudent(int id) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.query(*TABLE\_STUDENTS*,  
 new String[]{*KEY\_ID*, *KEY\_NAME*, *KEY\_PHONE*, *KEY\_EMAIL*},  
 *KEY\_ID* + "=?",  
 new String[]{String.*valueOf*(id)}, null, null, null);  
  
 if (cursor != null && cursor.moveToFirst()) {  
 Student student = new Student(  
 cursor.getInt(0),  
 cursor.getString(1),  
 cursor.getString(2),  
 cursor.getString(3));  
 cursor.close();  
 return student;  
 }  
 return null;  
 }  
  
 // Get all students  
 public List<Student> getAllStudents() {  
 List<Student> studentList = new ArrayList<>();  
 String selectQuery = "SELECT \* FROM " + *TABLE\_STUDENTS*;  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.rawQuery(selectQuery, null);  
  
 if (cursor.moveToFirst()) {  
 do {  
 Student student = new Student();  
 student.setId(cursor.getInt(0));  
 student.setName(cursor.getString(1));  
 student.setPhoneNumber(cursor.getString(2));  
 student.setEmail(cursor.getString(3));  
 studentList.add(student);  
 } while (cursor.moveToNext());  
 }  
 cursor.close();  
 return studentList;  
 }  
  
 // Update student  
 public int updateStudent(Student student) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(*KEY\_NAME*, student.getName());  
 values.put(*KEY\_PHONE*, student.getPhoneNumber());  
 values.put(*KEY\_EMAIL*, student.getEmail());  
  
 return db.update(*TABLE\_STUDENTS*, values, *KEY\_ID* + "=?",  
 new String[]{String.*valueOf*(student.getId())});  
 }  
  
 // Delete student by ID  
 public void deleteStudent(int id) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*TABLE\_STUDENTS*, *KEY\_ID* + "=?", new String[]{String.*valueOf*(id)});  
 db.close();  
 }  
  
 // Delete all students  
 public void deleteAllStudents() {  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*TABLE\_STUDENTS*, null, null);  
 db.close();  
 }  
}

Student.java  
package com.dtys.lab04.data.model;  
  
public class Student {  
 private int id;  
 private String name;  
 private String phoneNumber;  
 private String email;  
  
 // Constructors  
 public Student() {  
 }  
  
 public Student(int id, String name, String phoneNumber, String email) {  
 this.id = id;  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 this.email = email;  
 }  
  
 public Student(String name, String phoneNumber, String email) {  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 this.email = email;  
 }  
  
 // Getter & Setter  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getPhoneNumber() {  
 return phoneNumber;  
 }  
  
 public void setPhoneNumber(String phoneNumber) {  
 this.phoneNumber = phoneNumber;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 // Dùng cho debug hoặc hiển thị nhanh  
 @Override  
 public String toString() {  
 return "Student{" +  
 "id=" + id +  
 ", name='" + name + '\'' +  
 ", phoneNumber='" + phoneNumber + '\'' +  
 ", email='" + email + '\'' +  
 '}';  
 }  
}

StudentAdapter.java  
package com.dtys.lab04.ui.adapter;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
import com.dtys.lab04.R;  
import com.dtys.lab04.data.model.Student;  
  
import java.util.List;  
  
public class StudentAdapter extends RecyclerView.Adapter<StudentAdapter.StudentViewHolder> {  
  
 private Context context;  
 private List<Student> studentList;  
 private OnItemClickListener listener;  
  
 // Interface để xử lý sự kiện click  
 public interface OnItemClickListener {  
 void onItemClick(Student student);  
 }  
  
 public StudentAdapter(Context context, List<Student> studentList, OnItemClickListener listener) {  
 this.context = context;  
 this.studentList = studentList;  
 this.listener = listener;  
 }  
  
 @NonNull  
 @Override  
 public StudentViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(context).inflate(R.layout.*item\_student*, parent, false);  
 return new StudentViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull StudentViewHolder holder, int position) {  
 Student student = studentList.get(position);  
 holder.bind(student, listener);  
 }  
  
 @Override  
 public int getItemCount() {  
 return studentList.size();  
 }  
  
 // Update lại dữ liệu khi có thay đổi (thêm, xóa, sửa)  
 public void setData(List<Student> newStudentList) {  
 this.studentList = newStudentList;  
 notifyDataSetChanged();  
 }  
  
 // ViewHolder class  
 public static class StudentViewHolder extends RecyclerView.ViewHolder {  
 TextView tvName, tvPhone, tvEmail;  
  
 public StudentViewHolder(@NonNull View itemView) {  
 super(itemView);  
 tvName = itemView.findViewById(R.id.*tv\_name*);  
 tvPhone = itemView.findViewById(R.id.*tv\_phone*);  
 tvEmail = itemView.findViewById(R.id.*tv\_email*);  
 }  
  
 public void bind(final Student student, final OnItemClickListener listener) {  
 tvName.setText(student.getName());  
 tvPhone.setText(student.getPhoneNumber());  
 tvEmail.setText(student.getEmail());  
  
 itemView.setOnClickListener(v -> listener.onItemClick(student));  
 }  
 }  
}

Kết quả:  
A screenshot of a cell phone

AI-generated content may be incorrect.  
A screen shot of a phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A black rectangular frame with a white screen

AI-generated content may be incorrect.