App 8: Develop an Android app to store and retrieve data from SQLite database

MainActivity.java

package com.example.database2;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

// creating variables for our edittext, button and dbhandler

private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;

private Button addCourseBtn, readCourseBtn;

private DBHandler dbHandler;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// initializing all our variables.

courseNameEdt = findViewById(R.id.idEdtCourseName);

courseTracksEdt = findViewById(R.id.idEdtCourseTracks);

courseDurationEdt = findViewById(R.id.idEdtCourseDuration);

courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);

addCourseBtn = findViewById(R.id.idBtnAddCourse);

readCourseBtn = findViewById(R.id.idBtnReadCourse);

// creating a new dbhandler class

// and passing our context to it.

dbHandler = new DBHandler(MainActivity.this);

// below line is to add on click listener for our add course button.

addCourseBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// below line is to get data from all edit text fields.

String courseName = courseNameEdt.getText().toString();

String courseTracks = courseTracksEdt.getText().toString();

String courseDuration = courseDurationEdt.getText().toString();

String courseDescription = courseDescriptionEdt.getText().toString();

// validating if the text fields are empty or not.

if (courseName.isEmpty() && courseTracks.isEmpty() && courseDuration.isEmpty() && courseDescription.isEmpty()) {

Toast.makeText(MainActivity.this, "Please enter all the data..", Toast.LENGTH\_SHORT).show();

return;

}

// on below line we are calling a method to add new

// course to sqlite data and pass all our values to it.

dbHandler.addNewCourse(courseName, courseDuration, courseDescription, courseTracks);

// after adding the data we are displaying a toast message.

Toast.makeText(MainActivity.this, "Course has been added.", Toast.LENGTH\_SHORT).show();

courseNameEdt.setText("");

courseDurationEdt.setText("");

courseTracksEdt.setText("");

courseDescriptionEdt.setText("");

}

});

readCourseBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// opening a new activity via a intent.

Intent i = new Intent(MainActivity.this, ViewCourses.class);

startActivity(i);

}

});

}

}

DBHandler.java

package com.example.database2;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;

public class DBHandler extends SQLiteOpenHelper {

// creating a constant variables for our database.

// below variable is for our database name.

private static final String DB\_NAME = "studentinformation";

// below int is our database version

private static final int DB\_VERSION = 1;

// below variable is for our table name.

private static final String TABLE\_NAME = "StudentInfo";

// below variable is for our id column.

private static final String ID\_COL = "id";

// below variable is for our course name column

private static final String NAME\_COL = "name";

// below variable id for our course duration column.

private static final String DURATION\_COL = "duration";

// below variable for our course description column.

private static final String DESCRIPTION\_COL = "description";

// below variable is for our course tracks column.

private static final String TRACKS\_COL = "tracks";

// creating a constructor for our database handler.

public DBHandler(Context context) {

super(context, DB\_NAME, null, DB\_VERSION);

}

// below method is for creating a database by running a sqlite query

@Override

public void onCreate(SQLiteDatabase db) {

// on below line we are creating

// an sqlite query and we are

// setting our column names

// along with their data types.

String query = "CREATE TABLE " + TABLE\_NAME + " ("

+ ID\_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "

+ NAME\_COL + " TEXT,"

+ DURATION\_COL + " TEXT,"

+ DESCRIPTION\_COL + " TEXT,"

+ TRACKS\_COL + " TEXT)";

// at last we are calling a exec sql

// method to execute above sql query

db.execSQL(query);

}

// this method is use to add new course to our sqlite database.

public void addNewCourse(String courseName, String courseDuration, String courseDescription, String courseTracks) {

// on below line we are creating a variable for

// our sqlite database and calling writable method

// as we are writing data in our database.

SQLiteDatabase db = this.getWritableDatabase();

// on below line we are creating a

// variable for content values.

ContentValues values = new ContentValues();

// on below line we are passing all values

// along with its key and value pair.

values.put(NAME\_COL, courseName);

values.put(DURATION\_COL, courseDuration);

values.put(DESCRIPTION\_COL, courseDescription);

values.put(TRACKS\_COL, courseTracks);

// after adding all values we are passing

// content values to our table.

db.insert(TABLE\_NAME, null, values);

// at last we are closing our

// database after adding database.

db.close();

}

// we have created a new method for reading all the courses.

public ArrayList<CourseModal> readCourses() {

// on below line we are creating a

// database for reading our database.

SQLiteDatabase db = this.getReadableDatabase();

// on below line we are creating a cursor with query to read data from database.

Cursor cursorCourses = db.rawQuery("SELECT \* FROM " + TABLE\_NAME, null);

// on below line we are creating a new array list.

ArrayList<CourseModal> courseModalArrayList = new ArrayList<>();

// moving our cursor to first position.

if (cursorCourses.moveToFirst()) {

do {

// on below line we are adding the data from cursor to our array list.

courseModalArrayList.add(new CourseModal(cursorCourses.getString(1),

cursorCourses.getString(4),

cursorCourses.getString(2),

cursorCourses.getString(3)));

} while (cursorCourses.moveToNext());

// moving our cursor to next.

}

// at last closing our cursor

// and returning our array list.

cursorCourses.close();

return courseModalArrayList;

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

// this method is called to check if the table exists already.

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

onCreate(db);

}

}

activity\_main.xml

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

<LinearLayout

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<!--Edit text to enter course name-->

<EditText

android:id="@+id/idEdtCourseName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:hint="Enter Student Roll No" />

<!--edit text to enter course duration-->

<EditText

android:id="@+id/idEdtCourseDuration"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:hint="Enter Student Name" />

<!--edit text to display course tracks-->

<EditText

android:id="@+id/idEdtCourseTracks"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:hint="Enter Course Name" />

<!--edit text for course description-->

<EditText

android:id="@+id/idEdtCourseDescription"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:hint="Enter Students Marks" />

<!--button for adding new course-->

<Button

android:id="@+id/idBtnAddCourse"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:text="Add Details"

android:textAllCaps="false" />

<!--new button for opening our course list activity-->

<Button

android:id="@+id/idBtnReadCourse"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:text="Read All Details"

android:textAllCaps="false" />

</LinearLayout>

Output:

