Experiment - 10

Student Name: Sandeep Kumar **UID: 20BCS4885**
Branch: BE - CSE **Section/Group: 603/A**
Semester: 6th semester **Subject: MAD LAB**

Aim:

Create an Android application for user registration that stores the user details ina database table.

Requirements:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 4 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Java JDK5 or later version
- **Java Runtime Environment (JRE) 6 or higher.**

Step by Step Implementation:

Step 1: Create a New Project

To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio. Note that select **Java** as the programming language.

Step 2: Adding permissions to access the storage in the AndroidManifest.xml file

Navigate to the **app** > **AndroidManifest.xml** and add the below code to it.

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

Step 3: Working with the activity_main.xml file

Navigate to the **app** > **res** > **layout** > **activity_main.xml** and add the below code to that file. Below is the code for the **activity_main.xml** file.

```
<?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">


    <EditText
        android:id="@+id/idEdtCourseName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter course Name" />


    <EditText
        android:id="@+id/idEdtCourseDuration"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Duration" />


    <EditText
        android:id="@+id/idEdtCourseTracks"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Tracks" />


    <EditText
        android:id="@+id/idEdtCourseDescription"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Description" />


    <Button
        android:id="@+id/idBtnAddCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Add Course"
        android:textAllCaps="false" />

</LinearLayout>
```

Step 4: Creating a new Java class for performing SQLite operations

Navigate to the **app > java > your app's package name > Right-click on it > New > Java class** and name it as **DBHandler** and add the below code to it. Comments are added inside the code to understand the code in more detail.

//Java code

```
package com.abhimanyu.databaseconnect;

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

public class DBHandler extends SQLiteOpenHelper {

    private static final String DB_NAME = "coursedb";
    private static final int DB_VERSION = 1;
    private static final String TABLE_NAME = "mycourses";.
    private static final String ID_COL = "id";
    private static final String NAME_COL = "name";
    private static final String DURATION_COL = "duration";
    private static final String DESCRIPTION_COL = "description";
    private static final String TRACKS_COL = "tracks";

    public DBHandler(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        String query = "CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + NAME_COL + " TEXT,"
                + DURATION_COL + " TEXT,"
                + DESCRIPTION_COL + " TEXT,"
                + TRACKS_COL + " TEXT)";

        db.execSQL(query);
    }

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

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
```

```
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);

        db.insert(TABLE_NAME, null, values);
        db.close();
    }

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

        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```

Step 5: Working with the MainActivity.java file

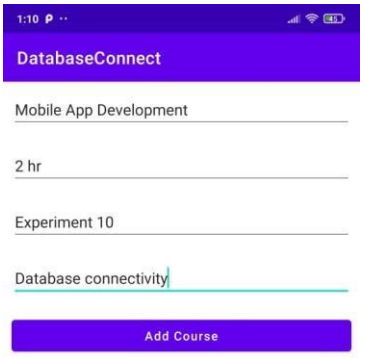
Go to the MainActivity.java **file and refer to the following code. Below is the code for the** MainActivity.java **file.**

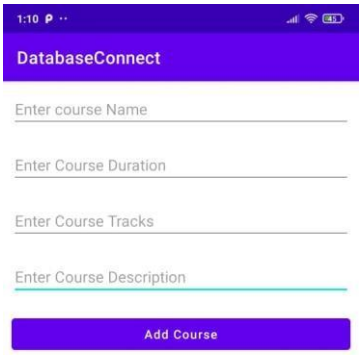
```
package com.abhimanyu.databaseconnect;

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 {


    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
    private DBHandler dbHandler;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        courseNameEdt = findViewById(R.id.idEdtCourseName);
        courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
        courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
        courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
        Button addCourseBtn = findViewById(R.id.idBtnAddCourse);

        dbHandler = new DBHandler(MainActivity.this);


        addCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
            String courseName = courseNameEdt.getText().toString();
            String courseTracks = courseTracksEdt.getText().toString();
            String courseDuration = courseDurationEdt.getText().toString();
            String courseDescription = courseDescriptionEdt.getText().toString();

            if (courseName.isEmpty() && courseTracks.isEmpty() && courseDuration.isEmpty() &&
courseDescription.isEmpty()) {
                Toast.makeText(MainActivity.this, "Please enter all the data..",
Toast.LENGTH_SHORT).show();
                return;
            }

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

            Toast.makeText(MainActivity.this, "Course has been added.",
Toast.LENGTH_SHORT).show();
            courseNameEdt.setText("");
            courseDurationEdt.setText("");
            courseTracksEdt.setText("");
            courseDescriptionEdt.setText("");
        }
    });
  }
}
```

Now run your app and see the output of the app.

Output:





Data is Stored in the Database as shown below: