

الطلاب المشاركين:

سلم التصحيح:

الفهرس:

Contents

[قاعدة البيانات 3](#_Toc187889012)

[تصميم الجدول العلائقيEntity Relationship Diagram ERD : 3](#_Toc187889013)

[الجداول للواصفات والكياناتEntities and Attributes Tables : 4](#_Toc187889014)

[الجداول للعلاقات Relationship Tables: 4](#_Toc187889015)

[السؤال الثاني: 5](#_Toc187889016)

[تصميم قاعدة البيانات باستخدام SQLite: 5](#_Toc187889017)

[خارطة عمل التطبيق: 6](#_Toc187889018)

[الصفوف والواجهات: 7](#_Toc187889019)

[1-النشاط الرئيسي Main Activity: 7](#_Toc187889020)

[الواجهات: 7](#_Toc187889021)

[الأكواد: 8](#_Toc187889022)

[2-نشاط إضافة معلومات المندوب Add Employee Activity: 10](#_Toc187889023)

[الواجهات: 10](#_Toc187889024)

[الأكواد: 11](#_Toc187889025)

[3-النشاط تعبئة المبيعات Add Sales Activity: 17](#_Toc187889026)

[الواجهات: 17](#_Toc187889027)

[الأكواد: 18](#_Toc187889028)

[4-نشاط البحث عن مبيعات المندوب Search Sale Activity: 29](#_Toc187889029)

[الواجهات: 29](#_Toc187889030)

[الأكواد 30](#_Toc187889031)

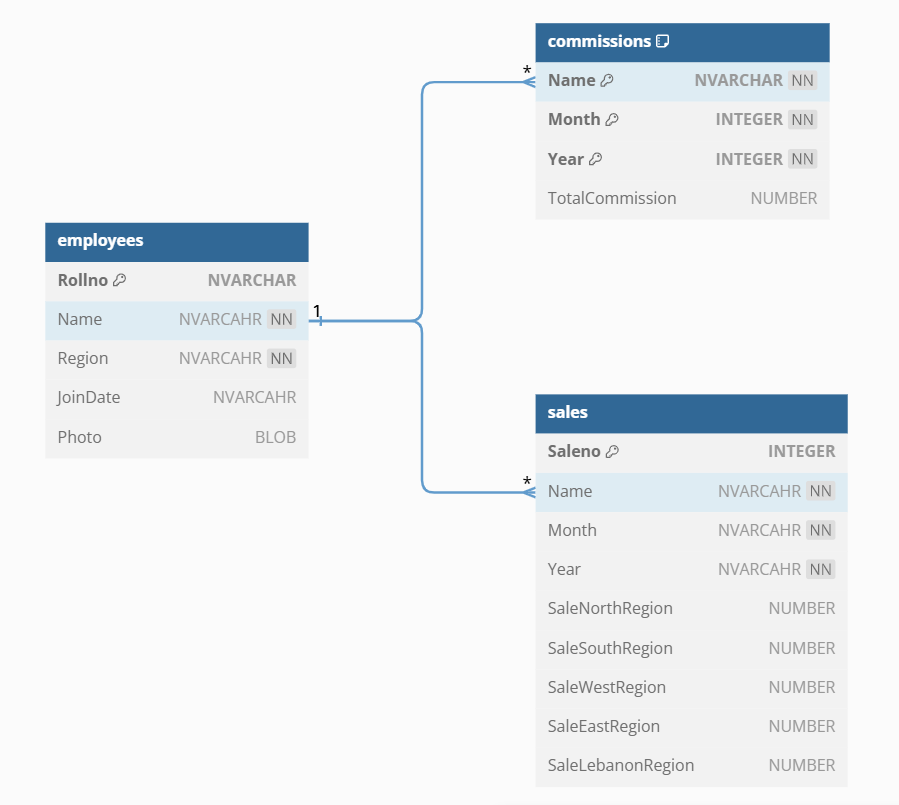
[5-نشاط البحث عن عمولات المندوب Search Commission Activity: 32](#_Toc187889032)

[الواجهات: 32](#_Toc187889033)

[الأكواد: 33](#_Toc187889034)

# قاعدة البيانات:

## تصميم الجدول العلائقيEntity Relationship Diagram ERD :



**حيث أن 1 يعبر عن جانب أحادي(One) العلاقة و \* تعبر عن جانب متعدد العلاقة (Many)**

## الجداول للواصفات والكياناتEntities and Attributes Tables :

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Type | Data Type | Attribute | Entity |
| Simplex | NVARCHAR | Rollno(PK) | employees |
| Simplex | NVARCHAR(Not Null) | Name(UNIQUE) |
| Simplex | NVARCHAR(Not Null) | Region |
| Simplex | NVARCHAR | JoinDate |
| Complex | BLOB | Photo |

تم ارفاق Unique constraint ل Name وذلك لمنع تعدد موظفين بنفس الاسم في نفس الجدول

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Type | Data Type | Attribute | Entity |
| Simple | INTEGER | Saleno(PK) | sales |
| Simple | NVARCHAR(Not Null) | Name(FK) |
| Simple | NVARCHAR(Not Null) | Month |
| Simple | NVARCHAR(Not Null) | Year |
| Simple | NUMBER | SaleNorthRegion |
| Simple | NUMBER | SaleSouthRegion |
| Simple | NUMBER | SaleWestRegion |
| Simple | NUMBER | SaleEastRegion |
| Simple | NUMBER | SaleLebanonRegion |

تم تحديد مفتاح اجنبي على Name وهو مفتاح ربط مع جدول الموظفين

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Type | Data Type | Attribute | Entity |
| Simple | NVARCHAR(Not Null) | Name(PK)(FK) | commissions |
| Simple | INTEGER | Month(PK) |
| Simple | INTEGER | Year(PK) |
| Simple | NUMBER | TotalCommission |

في جدول العمولات تم استخدام طريقة مفاتيح رئيسية مركبة Composite Primary Key: (Name, Month, Year) وذلك لأن هذه المجموعة تعمل على تحديد كل سجل عمولة لموظف بشكل فريد في شهر وسنة محددة وعدم التكرار.

تم تحديد مفتاح اجنبي على Name وهو مفتاح ربط مع جدول الموظفين

## الجداول للعلاقات Relationship Tables:

|  |  |  |  |
| --- | --- | --- | --- |
| Relationship Type | Relationship | Second Entity | First Entity |
| One-to-Many | Has a | sales | employees |
| One-to-Many | Has a | commssions | employees |
| Many-to-One | Belongs to | employees | sales |
| Many-to-One | Belongs to | employees | commissions |

# السؤال الثاني:

## تصميم قاعدة البيانات باستخدام SQLite:

فيما يلي هو جملة SQL في إنشاء قاعدة البيانات

CREATE TABLE IF NOT EXISTS employees **(**

Rollno NVARCHAR PRIMARY KEY**,**

Name NVARCHAR UNIQUE**,**

Region NVARCHAR**,**

JoinDate NVARCHAR**,**

Photo BLOB**);**

CREATE TABLE IF NOT EXISTS sales **(**

Saleno INTEGER PRIMARY KEY AUTOINCREMENT**,**

Name NVARCHAR NOT NULL**,**

Month NVARCHAR NOT NULL**,**

Year NVARCHAR NOT NULL**,**

SaleNorthRegion NUMBER DEFAULT 0**,**

SaleSouthRegion NUMBER DEFAULT 0**,**

SaleWestRegion NUMBER DEFAULT 0**,**

SaleEastRegion NUMBER DEFAULT 0**,**

SaleLebanonRegion NUMBER DEFAULT 0**,**

FOREIGN KEY **(**Name**)** REFERENCES employees**(**Name**)** ON DELETE CASCADE**);**

CREATE TABLE IF NOT EXISTS commissions **(**

Name NVARCHAR NOT NULL**,**

Month INTEGER NOT NULL**,**

Year INTEGER NOT NULL**,**

TotalCommission NUMBER DEFAULT 0**,**

PRIMARY KEY **(**Name**,** Month**,** Year**),**

FOREIGN KEY **(**Name**)** REFERENCES employees**(**Name**)** ON DELETE CASCADE**);**

تم إدراج DELETE CASCADE لمسح كل السجلات من المبيعات والعمولات في حال حذف الموظف.

# خارطة عمل التطبيق:

يتألف التطبيق من خمس واجهات (واجهة رئيسية توصل إلى اربع واجهات فرعية)

يبدأ التطبيق من النشاط الرئيسي Main Activity التي تحتوي على صورة الشركة ومعلومات الطلاب المشاركين والأزرار الرئيسية للتطبيق وهي:

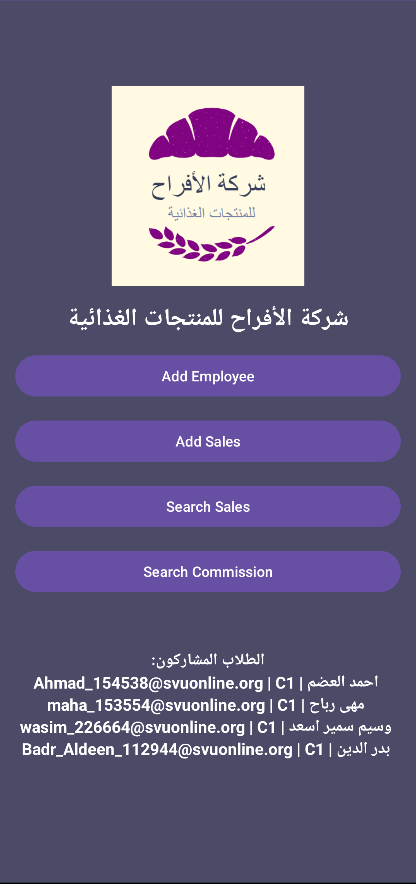
1. زر Add Employee وهو زر يفتح صفحة لإضافة معلومات المندوب (الموظف) ويوجد فيها الViews التالية:
   1. Enter Roll Number: مربع حوار لإدخال المعرف الخاص بالموظف الجديد.
   2. Enter Name: مربع حوار لإدخال اسم المندوب.
   3. Choose Region: قائمة منسدلة لاختيار المنطقة الاساسية التي يبيع فيها المندوب.
   4. Select Date Join: قائمة اختيار تاريخ التوظيف في الشركة.
   5. Choose Photo: زر اختيار صورة والتي ستقوم بتفح المعرض لاختيار صورة منه لتحميلها لقاعدة البيانات.
   6. Add Employee: زر اضافة المندوب وبعد التحقق من المدخلات وصحتها يتم تحميلها لقاعدة البيانات.
   7. Delete Employee: زر حذف المندوب وذلك بعد ادخال المعرف الخاص بالمندوب في مربع الحوار الخاص به لحذفه.
   8. Modify Employee: زر تعديل البيانات الخاصة بالمندوب وذلك بعد ادخال المعرف الخاص بالمندوب الموجود مسبقا وادخال البيانات الجديدة في مربعات الحوار الخاصة بها ليتم تحديثها.
   9. View Employee: زر عرض المعلومات الخاصة بالمندوب بعد إدخال المعرف الخاص بالمندوب في مربع الحوار الخاص به ليظهر صورته في مربع الصورة الخاص ورسالة بمعلوماته.
   10. View all Employees: زر عرض جميع الموظفين المدخلة معلوماتهم في رسالة واحدة.
2. زر Add Sale وهو زر يفتح صفحة لإضافة معلومات المبيعات التي قام بها المندوب في الشهر والسنة المخصصين ويوجد فيها الViews التالية:
   1. ادخل اسمك: مربع حوار مفتوح يتم من خلاله قراءة الاسم المدخل
   2. بحث: زر بحث عن الاسم المدخل في المربع السابق ويتم من خلاله تحصيل المنطقة الخاصة بالمندوب وصورته
   3. الصورة: يتم فيها عرض صورة المندوب بعد الضغط على زر البحث وكانت النتائج موجودة
   4. منطقتك: مربع حوار مغلق يتم فيه إظهار المنطقة الخاصة بالمندوب بعد الضغط على زر البحث
   5. Month: مربع حوار يقبل اعداد بين 12 و 1 وهو المقابل لأشهر السنة
   6. Year: وهو مربع حوار ارقام السنة
   7. المنطقة الشمالية: مربع حوار لإدخال المبلغ للمبيع في المنطقة الشمالية
   8. المنطقة الجنوبية: مربع حوار لإدخال المبلغ للمبيع في المنطقة الجنوبية
   9. المنطقة الشرقية: مربع حوار لإدخال المبلغ للمبيع في المنطقة الشرقية
   10. المنطقة الغربية: مربع حوار لإدخال المبلغ للمبيع في المنطقة الغربية
   11. لبنان: مربع حوار لإدخال المبلغ للمبيع في لبنان
   12. العمولة: زر يتم فيه تخزين المعلومات المدخلة من المبالغ المالية في المناطق الموافقة وبعدها مقارنتها مع المنطقة الاساسية للمندوب لحساب عمولة المندوب من المبيعات ويتم بعدها تخزين المعلومات كاملة

# الصفوف والواجهات:

## النشاط الرئيسي Main Activity:

### الواجهات:

عند تحميل التطبيق تظهر الواجهة الرئيسية للتطبيق التي تحتوي على المكونات الرئيسية له وازار الوصول للنشاطات الأخرى وفيها يكون صورة الشركة المختصة بالتطبيق كالآتي:



**تتكون من صورة الشركة واسمها واربعة ازرار كل منها تفعل النشاط المتعلق بها كالآتي:**

1. **Add Employee Button : الضغط على هذا الزر يقوم بتفعيل onclicklistner الذي يقوم بإظهار النشاط لإضافة معلومات المندوب**
2. **Add Sales Button**: الضغط على هذا الزر يقوم بتفعيل **onclicklistner** الذي يقوم بإظهار النشاط لبحث عن المندوب وإضافة المبيعات الخاصة به
3. **Search Sales Button: الضغط على هذا الزر يقوم بتفعيل onclicklistner الذي يقوم بإظهار النشاط للبحث عن معلومات المبيعات الخاصة بالمندوب**
4. **Search Commission Button: الضغط على هذا الزر يقوم بتفعيل onclicklistner الذي يقوم بإظهار النشاط للبحث عن العمولات الخاصة بالمندوب.**

يوجد في الأخير مربع حوار يحتوي المشاركين في المشروع مع المعرفات والايميلات والصفوف.

### الأكواد:

في ملف MainActivity.java يكون:

public class MainActivity **extends** AppCompatActivity **{**

@Override

protected void onCreate**(**Bundle savedInstanceState**)** **{**

**super.**onCreate**(**savedInstanceState**);**

setContentView**(**R**.**layout**.**activity\_main**);**

//Initialize Components تعريف الأزرار

Button btnAddEmployee **=** findViewById**(**R**.**id**.**btn\_add\_employee**);**

Button btnAddSales **=** findViewById**(**R**.**id**.**btn\_add\_sales**);**

Button btnSearchSales **=** findViewById**(**R**.**id**.**btn\_search\_sales**);**

Button btnSearchCommission **=** findViewById**(**R**.**id**.**btn\_search\_commission**);**

//Start Activity Add Employee بدء نشاط إضافة موظف

btnAddEmployee**.**setOnClickListener**(**view **->** **{**

Intent intent **=** **new** Intent**(**MainActivity**.this,** Add\_employee**.**class**);**

startActivity**(**intent**);**

**});**

//Start Activity Add Sales بدء نشاط إضافة مبيعات

btnAddSales**.**setOnClickListener**(**view **->** **{**

Intent intent **=** **new** Intent**(**MainActivity**.this,** AddSaleActivity**.**class**);**

startActivity**(**intent**);**

**});**

//Start activity Search sales بدء نشاط البحث عن مبيعات المندوب

btnSearchSales**.**setOnClickListener**(**view **->** **{**

Intent intent **=** **new** Intent**(**MainActivity**.this,** SearchSalesActivity**.**class**);**

startActivity**(**intent**);**

**});**

//Start Activity Search for commsission بدء نشاط البحث عن العمولات للمندوب

btnSearchCommission**.**setOnClickListener**(**view **->** **{**

Intent intent **=** **new** Intent**(**MainActivity**.this,** SearchCommissionActivity**.**class**);**

startActivity**(**intent**);**

**});**

**}**

**}**

الأكواد في ملف MainActivity.XML يكون:

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center"

android:padding="16dp"

android:background="#4B4A67">

<!-- شعار الشركة -->

<ImageView

android:id="@+id/companyLogo"

android:layout\_width="200dp"

android:layout\_height="200dp"

android:src="@drawable/logo"

android:contentDescription="Company Logo"

android:layout\_gravity="center" />

<!-- اسم الشركة -->

<TextView

android:id="@+id/companyName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:text="شركة الأفراح للمنتجات الغذائية"

android:textSize="24sp"

android:textStyle="bold"

android:gravity="center"

android:textColor="#FFFFFFFF" />

<!-- أزرار التنقل -->

<Button

android:id="@+id/btn\_add\_employee"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Add Employee"

android:layout\_marginTop="16dp"

/>

<Button

android:id="@+id/btn\_add\_sales"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Add Sales"

android:layout\_marginTop="16dp" />

<Button

android:id="@+id/btn\_search\_sales"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Search Sales"

android:layout\_marginTop="16dp" />

<Button

android:id="@+id/btn\_search\_commission"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Search Commission"

android:layout\_marginTop="16dp" />

<TextView

android:id="@+id/participated\_std"

android:layout\_width="match\_parent"

android:layout\_height="181dp"

android:layout\_marginTop="16dp"

android:gravity="center"

android:text="@string/participated\_std"

android:textSize="16sp"

android:textStyle="bold"

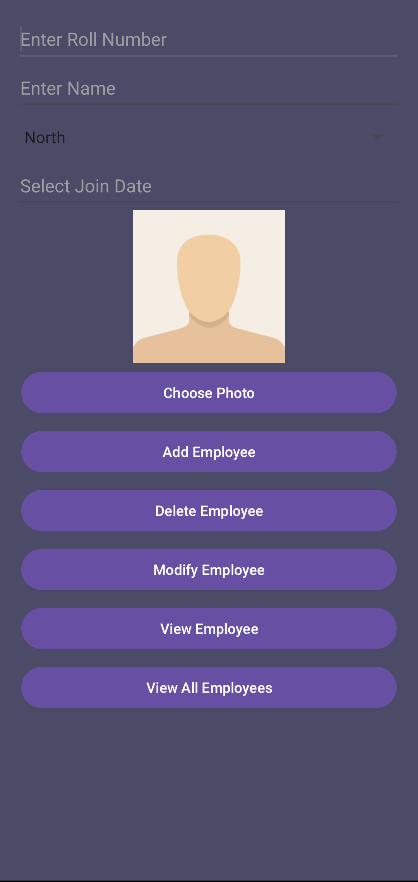
android:textColor="#FFFFFFFF"/>

</LinearLayout>

عند الضغط على زر إضافة موظف يتم تشغيل النشاط الخاص بإضافة معلومات المندوب ويبدأ عمل الصف Add\_employee.java

## نشاط إضافة معلومات المندوب Add Employee Activity:

### الواجهات:

****زر Add Employee وهو زر يفتح صفحة لإضافة معلومات المندوب (الموظف) ويوجد فيها الViews التالية:

1. Enter Roll Number: مربع حوار لإدخال المعرف الخاص بالموظف الجديد.
2. Enter Name: مربع حوار لإدخال اسم المندوب.
3. Choose Region: قائمة منسدلة لاختيار المنطقة الاساسية التي يبيع فيها المندوب.
4. Select Date Join: قائمة اختيار تاريخ التوظيف في الشركة.
5. Choose Photo: زر اختيار صورة والتي ستقوم بتفح المعرض لاختيار صورة منه لتحميلها لقاعدة البيانات.
6. Add Employee: زر اضافة المندوب وبعد التحقق من المدخلات وصحتها يتم تحميلها لقاعدة البيانات.
7. Delete Employee: زر حذف المندوب وذلك بعد ادخال المعرف الخاص بالمندوب في مربع الحوار الخاص به لحذفه.
8. Modify Employee: زر تعديل البيانات الخاصة بالمندوب وذلك بعد ادخال المعرف الخاص بالمندوب الموجود مسبقا وادخال البيانات الجديدة في مربعات الحوار الخاصة بها ليتم تحديثها.
9. View Employee: زر عرض المعلومات الخاصة بالمندوب بعد إدخال المعرف الخاص بالمندوب في مربع الحوار الخاص به ليظهر صورته في مربع الصورة الخاص ورسالة بمعلوماته.
10. View all Employees: زر عرض جميع الموظفين المدخلة معلوماتهم في رسالة واحدة.

#### طريقة العمل:

**يبدأ النشاط عند إدخال المعرف المراد في مربع الحوار الخاص به ثم الاسم واختيار المنطقة من القائمة المنسدلة ثم اخيتار تاريخ انضمام المندوب للشركة ثم يتم الضغط على choose photo ليتم بعدها فتح المعرض لاختيار صورة من المعرض ثم عند الضغط على زر اضافة الموظف يتم ضغط الصورة لان قاعدة البيانات والطرق في عرض الصور فيها شروط تحجيم في العرض لذلك يجب ضغط الصورة compress ثم يتم التحقق من جميع البيانات وحفظها في قاعدة مع اظهار رسالة نجاح**

**عند الضغط على زر الحذف يتحقق من رقم معرف المندوب في مربع الحوار ثم يفعل الطريقة المسؤولة عن حذف المندوب مع سجلاته كاملة من الجداول الأخرى.**

**عند الضغط على زر التعديل يتحقق من رقم معرف المندوب في مربع الحوار ثم يفعل الطريقة المسؤولة عن تعديل معلومات المندوب التي من خلالها يتحقق من باقي مربعات الحوار من المدخلات ليقوم بتحديثها في قاعدة البيانات.**

**عند الضغط على زر عرض الموظف يتم تفعيل طريقة لعرض رسالة صغيرة فيها معلومات الموظف من قاعدة البيانات وتحميل صورته في مربع الصورة المخصص**

**عند الضغط على زر عرض جميع الموظفين يتم تفعيل طريقة لإظهار رسالة فيها جميع الموظفين المسجلين في قاعدة البيانات.**

### الأكواد:

في ملف Add\_employee.java يكون التالي:

public class Add\_employee extends AppCompatActivity {  
 *//declaring variables* EditText editRollno, editName, editDate;  
 Spinner spinnerRegion;  
 ImageView imgPhoto;  
 Button btnAdd, btnDelete, btnModify, btnView, btnViewAll,btnChoosePhoto;  
 SQLiteDatabase db;  
 *//declaring variables for image* private ActivityResultLauncher<Intent> pickImageLauncher;  
 private byte[] imageBytes;  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_add\_employee*);  
  
 *// Initialize the ActivityResultLauncher* editRollno = findViewById(R.id.*ed\_Rollno*);  
 editName = findViewById(R.id.*ed\_Name*);  
 editDate = findViewById(R.id.*ed\_Date*);  
 spinnerRegion = findViewById(R.id.*spinnerRegion*);  
 imgPhoto = findViewById(R.id.*imgPhoto*);  
 btnAdd = findViewById(R.id.*btn\_Add*);  
 btnModify = findViewById(R.id.*btn\_Modify*);  
 btnDelete = findViewById(R.id.*btn\_Delete*);  
 btnView = findViewById(R.id.*btnView*);  
 btnViewAll = findViewById(R.id.*btnViewAll*);  
 btnChoosePhoto = findViewById(R.id.*btnChoosePhoto*);  
 *// (Spinner)* String[] regions = {"North", "South", "West", "East", "Lebanon"};  
 ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_dropdown\_item*, regions);  
 spinnerRegion.setAdapter(adapter);  
 *// Choosing Image using method to open the gallery and select an image then compress it into a byte array* pickImageLauncher = registerForActivityResult(  
 new ActivityResultContracts.StartActivityForResult(),  
 result -> {  
 if (result.getResultCode() == *RESULT\_OK* && result.getData() != null) {  
 Uri selectedImageUri = result.getData().getData();  
 try {  
 Bitmap bitmap = MediaStore.Images.Media.*getBitmap*(getContentResolver(), selectedImageUri); *//open media gallery and select image* imageBytes = compressImage(bitmap); *// Compress the image* imgPhoto.setImageBitmap(bitmap); *//setting image to the imageView* } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 );  
 *// creating employee table* db = openOrCreateDatabase("EmployeeDB", Context.*MODE\_PRIVATE*, null);  
 db.execSQL("CREATE TABLE IF NOT EXISTS employees (" +  
 "Rollno NVARCHAR PRIMARY KEY, " +  
 "Name NVARCHAR UNIQUE, " +  
 "Region NVARCHAR, " +  
 "JoinDate NVARCHAR, " +  
 "Photo BLOB)");  
 *//choose photo button start method to choose photo from database* btnChoosePhoto.setOnClickListener(view -> pickImage());  
 *//add button start method to insert data into database* btnAdd.setOnClickListener(this::onAddClick);  
 *//delete button start method to delete data from database* btnDelete.setOnClickListener(this::onDeleteClick);  
 *//modify button start method to modify data from database* btnModify.setOnClickListener(this::onModifyClick);  
 *//edit date button start method open date view to select date* editDate.setOnClickListener(this::onDateClick);  
 *//view button start method to view data from database* btnView.setOnClickListener(this::onViewClick);  
 *//view all button start method to view all data from database* btnViewAll.setOnClickListener(this::onViewAllClick);  
 *//image view to default image* imgPhoto.setImageResource(R.drawable.*ic\_default\_photo*);  
 }  
 *//Method to compress image as cursor can not display image with high quality because of limit constraints* private byte[] compressImage(Bitmap bitmap) {  
 ByteArrayOutputStream stream = new ByteArrayOutputStream(); *//bit array class to store image* bitmap.compress(Bitmap.CompressFormat.*JPEG*, 50, stream); *// Compress image to 50% of its actual size and quality* return stream.toByteArray(); *//return byte array of image* }  
 *//Method to open gallery and select image* private void pickImage() {  
 Intent intent = new Intent(Intent.*ACTION\_PICK*, MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);  
 pickImageLauncher.launch(intent);  
 } *//Method for adding employee data* private void onAddClick(View view) {  
 *//checking entry validity* if (editRollno.getText().toString().trim().isEmpty() ||  
 editName.getText().toString().trim().isEmpty() ||  
 editDate.getText().toString().trim().isEmpty()) {  
 showMessage("Error", "Please fill all fields.");  
 return;  
 }  
 *//extracting entered data and storing in variables* String rollno = editRollno.getText().toString().trim();  
 String name = editName.getText().toString().trim().toLowerCase(); *//case sensitive for names* String region = spinnerRegion.getSelectedItem().toString();  
 String joinDate = editDate.getText().toString().trim();  
 if (imageBytes == null || imageBytes.length == 0) {  
 showMessage("Error", "Please select an image.");  
 return;  
 }  
 *//SQL statement to insert data into the database* try {  
 db.execSQL("INSERT INTO employees (Rollno, Name, Region, JoinDate, Photo) VALUES (?, ?, ?, ?, ?)",  
 new Object[]{rollno, name, region, joinDate, imageBytes});  
 showMessage("Success", "Employee added successfully!");  
 clearFields();  
 } catch (Exception e) {  
 showMessage("Error", "Failed to add employee: " + e.getMessage());  
 }  
 }  
 *//method to delete employee data* private void onDeleteClick(View view) {  
 *//extract the Rollno entered to delete* String rollno = editRollno.getText().toString().trim();  
 if (rollno.isEmpty()) {  
 showMessage("Error", "Please enter the Roll Number to delete.");  
 return;  
 }  
 *//SQL statement to delete data from the database* db.execSQL("DELETE FROM employees WHERE Rollno = ?", new Object[]{rollno});  
 showMessage("Success", "Employee deleted successfully!");  
 clearFields();  
 }  
 *//method to modify employee data* private void onModifyClick(View view) {  
 *//extract the Rollno entered to modify* String rollno = editRollno.getText().toString().trim();  
 if (rollno.isEmpty()) {  
 showMessage("Error", "Please enter the Roll Number to modify.");  
 return;  
 }  
 *//extract the new data to modify and storing in variables* String name = editName.getText().toString().trim().toLowerCase(); *//case sensitive for names* String region = spinnerRegion.getSelectedItem().toString();  
 String joinDate = editDate.getText().toString().trim();  
 if (imageBytes == null || imageBytes.length == 0) {  
 showMessage("Error", "Please select an image.");  
 return;  
 }  
 *//SQL statement to modify data in the database* try {  
 db.execSQL("UPDATE employees SET Name = ?, Region = ?, JoinDate = ?, Photo = ? WHERE Rollno = ?",  
 new Object[]{name, region, joinDate,imageBytes, rollno});  
 showMessage("Success", "Employee modified successfully!");  
 clearFields();  
 } catch (Exception e) {  
 showMessage("Error", "Failed to Modify employee: " + e.getMessage());  
 }  
 }  
 *//method to view employee data* private void onViewClick(View view) {  
 *//extract the Rollno entered to view* String rollno = editRollno.getText().toString().trim();  
 if (rollno.isEmpty()) {  
 showMessage("Error", "Please enter the Roll Number to view.");  
 return;  
 }  
 *//SQL statement to retrieve data from the database* Cursor cursor = db.rawQuery("SELECT \* FROM employees WHERE Rollno = ?", new String[]{rollno});  
 if (cursor.moveToFirst()) {  
 *//extract the retrieved data and storing in variables* String name = cursor.getString(1);  
 String region = cursor.getString(2);  
 String joinDate = cursor.getString(3);  
 byte[] photo = cursor.getBlob(4);  
 *//display the retrieved data in message* showMessage("Employee Details",  
 "Roll Number: " + rollno +  
 "\nName: " + name +  
 "\nRegion: " + region +  
 "\nJoin Date: " + joinDate);  
 *//display the retrieved image in imageView* if (photo != null && photo.length > 0) {  
 Bitmap bitmap = BitmapFactory.*decodeByteArray*(photo, 0, photo.length);*//convert byte array to bitmap* if (bitmap != null) {  
 imgPhoto.setImageBitmap(bitmap);*//setting image to the imageView* } else {  
 imgPhoto.setImageResource(R.drawable.*ic\_default\_photo*);*//setting default image to the imageView* }  
 } else {  
 imgPhoto.setImageResource(R.drawable.*ic\_default\_photo*);*//setting default image to the imageView* }  
 } else {  
 showMessage("Error", "No employee found with Roll Number: " + rollno);*//display error message if no employee found* }  
 cursor.close();  
 }  
 *//method to view all employee data* private void onViewAllClick(View view) {  
 *//SQL statement to retrieve all data from the database* Cursor cursor = db.rawQuery("SELECT \* FROM employees", null);  
 if (cursor.getCount() == 0) {  
 showMessage("Error", "No employees found.");  
 return;  
 }  
 *//display the retrieved data in message* StringBuilder builder = new StringBuilder();  
 while (cursor.moveToNext()) {  
 String rollno = cursor.getString(0);  
 String name = cursor.getString(1);  
 String region = cursor.getString(2);  
 String joinDate = cursor.getString(3);  
 builder.append("Roll Number: ").append(rollno)  
 .append("\nName: ").append(name)  
 .append("\nRegion: ").append(region)  
 .append("\nJoin Date: ").append(joinDate)  
 .append("\n\n");  
 }  
 cursor.close();  
 showMessage("All Employees", builder.toString());  
 }  
 *//method to open date view to select date* private void onDateClick(View view) {  
 *//open date view to select date* Calendar calendar = Calendar.*getInstance*();  
 int year = calendar.get(Calendar.*YEAR*);  
 int month = calendar.get(Calendar.*MONTH*);  
 int day = calendar.get(Calendar.*DAY\_OF\_MONTH*);  
 *//open date view to select date* DatePickerDialog datePicker = new DatePickerDialog(this, (datePicker1, y, m, d) -> {  
 editDate.setText(y + "-" + (m + 1) + "-" + d);  
 }, year, month, day);  
 datePicker.show();  
 }  
 *//Method to show popup message* private void showMessage(String title, String message) {  
 *//popup message* AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle(title);  
 builder.setMessage(message);  
 builder.setIcon(R.drawable.*ic\_launcher\_foreground*);  
 builder.show();  
 }  
  
 *//Clear Method to clear fields* private void clearFields() {  
 editRollno.setText("");  
 editName.setText("");  
 editDate.setText("");  
 imgPhoto.setImageResource(R.drawable.*ic\_default\_photo*);  
 }  
}

في ملف add\_employee.XML يكون:

*<?*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"  
 android:background="#4B4A67">  
<!-- Empoyee ID -->  
 <EditText  
 android:id="@+id/ed\_Rollno"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Roll Number"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
<!-- Empoyee name -->  
 <EditText  
 android:id="@+id/ed\_Name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Name"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
<!-- region spinner -->  
 <Spinner  
 android:id="@+id/spinnerRegion"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:minHeight="48dp"  
 android:textColor="#FFFFFFFF" />  
<!-- date selection -->  
 <EditText  
 android:id="@+id/ed\_Date"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:focusable="false"  
 android:hint="Select Join Date"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
<!-- employee image view -->  
 <ImageView  
 android:id="@+id/imgPhoto"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 android:layout\_gravity="center"  
 android:src="@drawable/ic\_placeholder"  
 android:scaleType="centerCrop" />  
<!-- button to choose photo -->  
 <Button  
 android:id="@+id/btnChoosePhoto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Choose Photo"  
 android:layout\_margin="5dp"/>  
<!-- button to add employee -->  
 <Button  
 android:id="@+id/btn\_Add"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Add Employee"  
 android:layout\_margin="5dp"  
 />  
<!-- button to delete employee -->  
 <Button  
 android:id="@+id/btn\_Delete"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Delete Employee"  
 android:layout\_margin="5dp"/>  
<!-- button to modify employee -->  
 <Button  
 android:id="@+id/btn\_Modify"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Modify Employee"  
 android:layout\_margin="5dp"/>  
<!-- button to view employee -->  
 <Button  
 android:id="@+id/btnView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="View Employee"  
 android:layout\_margin="5dp"/>  
<!-- button to view all employees -->  
 <Button  
 android:id="@+id/btnViewAll"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="View All Employees"  
 android:layout\_margin="5dp"/>  
</LinearLayout>

##### نهاية النشاط Add Employee

## النشاط تعبئة المبيعات Add Sales Activity:

### الواجهات:



### الأكواد:

في ملف AddSaleActivity.java يكون:

public class AddSaleActivity extends AppCompatActivity {  
 *//declaring variables* EditText ed\_name,ed\_month,ed\_year, ed\_north, ed\_south, ed\_west, ed\_east, ed\_lebanon;  
 Button btn\_profit, btn\_search;  
 Button btn\_view\_sales;  
 Button btn\_view\_commissions;  
 ImageView im\_yourPhoto;  
 TextView tv\_yourRegion;  
 SQLiteDatabase db;  
 MediaPlayer mediaPlayer;  
 *//declaring variables for image* private ActivityResultLauncher<Intent> pickImageLauncher;  
 private byte[] imageBytes;  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_add\_sale*);  
  
 *// Initialize the ActivityResultLauncher* ed\_name = findViewById(R.id.*ed\_name*);  
 ed\_year=findViewById(R.id.*ed\_year*);  
 ed\_month=findViewById(R.id.*ed\_month*);  
 ed\_north = findViewById(R.id.*ed\_north*);  
 ed\_south = findViewById(R.id.*ed\_south*);  
 ed\_west = findViewById(R.id.*ed\_west*);  
 ed\_east = findViewById(R.id.*ed\_east*);  
 ed\_lebanon = findViewById(R.id.*ed\_lebanon*);  
 btn\_profit = findViewById(R.id.*btn\_profit*);  
 btn\_search = findViewById(R.id.*btn\_search*);  
 im\_yourPhoto = findViewById(R.id.*im\_yourPhoto*);  
 tv\_yourRegion = findViewById(R.id.*tv\_yourRegion*);  
 btn\_view\_sales = findViewById(R.id.*btn\_view\_sales*);  
 btn\_view\_commissions = findViewById(R.id.*btn\_view\_commissions*);  
  
 *//Database creation* db = openOrCreateDatabase("EmployeeDB", Context.*MODE\_PRIVATE*, null);  
 *//table employees* db.execSQL("CREATE TABLE IF NOT EXISTS employees (" +  
 "Rollno NVARCHAR PRIMARY KEY, " +  
 "Name NVARCHAR UNIQUE, " +  
 "Region NVARCHAR, " +  
 "JoinDate NVARCHAR, " +  
 "Photo BLOB)");  
 *//table sales* db.execSQL("CREATE TABLE IF NOT EXISTS sales (" +  
 "Saleno INTEGER PRIMARY KEY AUTOINCREMENT, " +  
 "Name NVARCHAR NOT NULL, " +  
 "Month NVARCHAR NOT NULL, " +  
 "Year NVARCHAR NOT NULL, " +  
 "SaleNorthRegion NUMBER DEFAULT 0, " +  
 "SaleSouthRegion NUMBER DEFAULT 0, " +  
 "SaleWestRegion NUMBER DEFAULT 0, " +  
 "SaleEastRegion NUMBER DEFAULT 0, " +  
 "SaleLebanonRegion NUMBER DEFAULT 0, " +  
 "FOREIGN KEY (Name) REFERENCES employees(Name) ON DELETE CASCADE)");  
 *//table commissions* db.execSQL("CREATE TABLE IF NOT EXISTS commissions (" +  
 "Name NVARCHAR NOT NULL, " +  
 "Month INTEGER NOT NULL, " +  
 "Year INTEGER NOT NULL, " +  
 "TotalCommission NUMBER DEFAULT 0, " +  
 "PRIMARY KEY (Name, Month, Year), " +  
 "FOREIGN KEY (Name) REFERENCES employees(Name) ON DELETE CASCADE)");  
 *//media player sound effect for message* mediaPlayer = MediaPlayer.*create*(this, R.raw.*alert\_sound*);  
 *//profit button start method to start method to insert sale into database* btn\_profit.setOnClickListener(this::onProfitClick);  
 *//search button start method to search employee in database* btn\_search.setOnClickListener(this::onSearchClick);  
 *//view sales button start method to view sales from database* btn\_view\_sales.setOnClickListener(view -> viewSales());  
 *//view commissions button start method to view commissions from database* btn\_view\_commissions.setOnClickListener(view -> viewCommissions());  
 *//image view to default image* im\_yourPhoto.setImageResource(R.drawable.*ic\_default\_photo*);  
 }  
 *//Method to search employee in database* @SuppressLint("SetTextI18n")  
 public void onSearchClick(View view) {  
 *//extract the Rollno entered to search* String name = ed\_name.getText().toString().trim().toLowerCase();  
 *//checking entry validity* if (name.isEmpty()) {  
 Toast.*makeText*(this, "Please enter an employee name", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 *//SQL statement to retrieve data from the database* Cursor cursor = db.rawQuery("SELECT Region, Photo FROM employees WHERE Name = ?", new String[]{name});  
 *//checking if employee exists* try {  
 if (cursor.moveToFirst()) {  
 *//extract the retrieved data and storing in variables* String region = cursor.getString(0);  
 byte[] photoBytes = cursor.getBlob(1); *// Get the image as BLOB  
 //display the retrieved data in message* tv\_yourRegion.setText("Region: " + region);  
 *//display the retrieved image in imageView* if (photoBytes != null) {  
 *// Convert byte array to Bitmap* Bitmap bitmap = BitmapFactory.*decodeByteArray*(photoBytes, 0, photoBytes.length);  
 im\_yourPhoto.setImageBitmap(bitmap);*//setting image to the imageView* } else {  
 im\_yourPhoto.setImageResource(R.drawable.*ic\_placeholder*);*//setting default image to the imageView* }  
 } else {  
 *//display error message if no employee found* Toast.*makeText*(this, "Employee not found", Toast.*LENGTH\_SHORT*).show();  
 tv\_yourRegion.setText("Region: Not Found");  
 im\_yourPhoto.setImageResource(R.drawable.*ic\_placeholder*);  
 }  
 } finally {  
 cursor.close();  
 }  
 }  
 *//Method to insert sale into database* public void onProfitClick(View view) {  
 *//checking entry validity* String name = ed\_name.getText().toString().trim().toLowerCase();  
 if (name.isEmpty()) {  
 *//display error message if no employee found* Toast.*makeText*(this, "Please enter an employee name", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 if (!isEmployeeExists(name)) {  
 *//display error message if no employee found* Toast.*makeText*(this, "Employee does not exist. Please add the employee first.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 try {  
 *//extracting data from views* String month = ed\_month.getText().toString().trim();*//extract from month view* String year = ed\_year.getText().toString().trim();*//extract from year view  
 //sales data* double north = Double.*parseDouble*(ed\_north.getText().toString());*//extract from north view* double south = Double.*parseDouble*(ed\_south.getText().toString());*//extract from south view* double west = Double.*parseDouble*(ed\_west.getText().toString());*//extract from west view* double east = Double.*parseDouble*(ed\_east.getText().toString());*//extract from east view* double lebanon = Double.*parseDouble*(ed\_lebanon.getText().toString());*//extract from lebanon view  
 //Calling method to calculate the commission as per the numbers entered* double totalCommission = calculateTotalCommission(name, north, south, west, east, lebanon);  
 *//checking if sales already exists* if (isSalesExists(name, month, year)) {  
 playAlertSound(); *// تشغيل الصوت* showReplaceDialog(name, month, year, totalCommission, north, south, west, east, lebanon);  
 } else {  
 *//calling method to insert data into sales database* insertSales(name, month, year, north, south, west, east, lebanon);  
 *//calling method to insert data into commissions database* insertCommission(name, month, year, totalCommission);  
 Toast.*makeText*(this, "Sales and commission added successfully!", Toast.*LENGTH\_LONG*).show();  
 *//display success message* new AlertDialog.Builder(this)  
 .setTitle("Operation Successful")  
 .setMessage("Sales and commission have been successfully added to the database.\nTotal Commission: " + formatNumber(totalCommission) + " S.P")  
 .setPositiveButton("OK", null)  
 .show();  
 }  
 } catch (NumberFormatException e) {  
 Toast.*makeText*(this, "Please enter valid sales values.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 *//Method to check if employee exists* private boolean isEmployeeExists(String name) {  
 *//SQL statement to retrieve data from the database* Cursor cursor = db.rawQuery("SELECT \* FROM employees WHERE Name = ?", new String[]{name});  
 boolean exists = cursor.getCount() > 0;  
 cursor.close();  
 return exists;  
 }  
 *//Method to insert data into sales database* private void insertSales(String name, String month, String year, double north, double south, double west, double east, double lebanon) {  
 db.execSQL("INSERT INTO sales (Name, Month, Year, SaleNorthRegion, SaleSouthRegion, SaleWestRegion, SaleEastRegion, SaleLebanonRegion) VALUES (?, ?, ?, ?, ?, ?, ?, ?)",  
 new Object[]{name, month, year, north, south, west, east, lebanon});  
 }  
 *//Method to insert data into commissions database* private void insertCommission(String name, String month, String year, double totalCommission) {  
 db.execSQL("INSERT INTO commissions (Name, Month, Year, TotalCommission) VALUES (?, ?, ?, ?)",  
 new Object[]{name, month, year, totalCommission});  
 }  
 *//Method to replace existing data* private void showReplaceDialog(String name, String month, String year, double totalCommission, double north, double south, double west, double east, double lebanon) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 *//display message of replace existing data statement* builder.setTitle("Replace Existing Data?");  
 builder.setMessage("Sales data for this employee already exists for the same month and year. Do you want to replace it?");  
 builder.setPositiveButton("Replace", (dialog, which) -> {  
 *//calling method to update data in sales database* updateSales(name, month, year, north, south, west, east, lebanon);  
 *//calling method to update data in commissions database* updateCommission(name, month, year, totalCommission);  
 *//display success message* showSuccessDialog(totalCommission);  
 });  
 *//display cancel button* builder.setNegativeButton("Cancel", null);  
 builder.show();  
 }  
 *//Method to display success message* private void showSuccessDialog(double totalCommission) {  
 new AlertDialog.Builder(this)  
 .setTitle("Operation Successful")  
 .setMessage("The operation was successful!\nTotal Commission: " + totalCommission + " S.P")  
 .setPositiveButton("OK", null)  
 .show();  
 }  
 *//Method to view sales from database* private void viewSales() {  
 *//SQL statement to retrieve all data from the database* Cursor cursor = db.rawQuery("SELECT \* FROM sales", null);  
 if (cursor.getCount() == 0) {  
 Toast.*makeText*(this, "No sales records found.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 *//display the retrieved data in message* StringBuilder builder = new StringBuilder();  
 while (cursor.moveToNext()) {  
 builder.append("Sale ID: ").append(cursor.getInt(0))  
 .append("\nName: ").append(cursor.getString(1))  
 .append("\nMonth: ").append(cursor.getInt(2))  
 .append("\nYear: ").append(cursor.getInt(3))  
 .append("\nNorth: ").append(formatNumber(cursor.getDouble(4)))  
 .append("\nSouth: ").append(formatNumber(cursor.getDouble(5)))  
 .append("\nWest: ").append(formatNumber(cursor.getDouble(6)))  
 .append("\nEast: ").append(formatNumber(cursor.getDouble(7)))  
 .append("\nLebanon: ").append(formatNumber(cursor.getDouble(8)))  
 .append("\n\n");  
 }  
 cursor.close();  
 *//display the retrieved data in message* new AlertDialog.Builder(this)  
 .setTitle("Sales Records")  
 .setMessage(builder.toString())  
 .setPositiveButton("OK", null)  
 .show();  
 }  
 *// Helper method to format numbers* private String formatNumber(double value) {  
 if (value == (long) value) {  
 return String.*format*("%d", (long) value); *// Format as integer if no decimal part* } else {  
 return String.*format*("%.2f", value); *// Format as fixed-point decimal* }  
 }  
 *//Method to view commissions from database* private void viewCommissions() {  
 *//sql statement to retrieve all data from the database* Cursor cursor = db.rawQuery("SELECT \* FROM commissions", null);  
 if (cursor.getCount() == 0) {  
 Toast.*makeText*(this, "No commission records found.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 *//display the retrieved data in message* StringBuilder builder = new StringBuilder();  
 while (cursor.moveToNext()) {  
 builder.append("Name: ").append(cursor.getString(0))  
 .append("\nMonth: ").append(cursor.getInt(1))  
 .append("\nYear: ").append(cursor.getInt(2))  
 .append("\nTotal Commission: ").append(formatNumber(cursor.getDouble(3)))  
 .append("\n\n");  
 }  
 cursor.close();  
 *//display the retrieved data in message* new AlertDialog.Builder(this)  
 .setTitle("Commission Records")  
 .setMessage(builder.toString())  
 .setPositiveButton("OK", null)  
 .show();  
 }  
 *//Method to check if sales already exists* private boolean isSalesExists(String name, String month, String year) {  
 *//SQL statement to retrieve data from the database* String query = "SELECT \* FROM sales WHERE Name = ? AND Month = ? AND Year = ?";  
 Cursor cursor = db.rawQuery(query, new String[]{name.trim(), month.trim(), year.trim()});  
 *//checking if sales already exists* boolean exists = cursor.getCount() > 0;  
 cursor.close();  
 return exists;  
 }  
 *//Method to update data in sales database* private void updateSales(String name, String month, String year, double north, double south, double west, double east, double lebanon) {  
 *//SQL statement to update data in the database* db.execSQL("UPDATE sales SET SaleNorthRegion = ?, SaleSouthRegion = ?, SaleWestRegion = ?, SaleEastRegion = ?, SaleLebanonRegion = ? " +  
 "WHERE Name = ? AND Month = ? AND Year = ?",  
 new Object[]{north, south, west, east, lebanon, name, month, year});  
 }  
 *//Method to update data in commissions database* private void updateCommission(String name, String month, String year, double totalCommission) {  
 *//SQL statement to update data in the database* db.execSQL("UPDATE commissions SET TotalCommission = ? WHERE Name = ? AND Month = ? AND Year = ?",  
 new Object[]{totalCommission, name, month, year});  
 }  
 *//Method to calculate the total commissions* private double calculateTotalCommission(String name, double north, double south, double west, double east, double lebanon) {  
 String employeeRegion = getEmployeeRegion(name);  
 *//storing data from views to be calculated* double commissionNorth = calculateCommission(north, employeeRegion.equals("North"));  
 double commissionSouth = calculateCommission(south, employeeRegion.equals("South"));  
 double commissionWest = calculateCommission(west, employeeRegion.equals("West"));  
 double commissionEast = calculateCommission(east, employeeRegion.equals("East"));  
 double commissionLebanon = calculateCommission(lebanon, employeeRegion.equals("Lebanon"));  
 *//calculating total commission* double totalCommission=(commissionNorth + commissionSouth + commissionWest + commissionEast + commissionLebanon);  
 totalCommission = Math.*round*(totalCommission);  
 return totalCommission;  
 }  
 *//Method to get employee region* private String getEmployeeRegion(String name) {  
 *//SQL statement to get the region of the entered name* Cursor cursor = db.rawQuery("SELECT Region FROM employees WHERE Name = ?", new String[]{name});  
 if (cursor.moveToFirst()) {  
 String region = cursor.getString(0);  
 cursor.close();  
 return region;  
 }  
 cursor.close();  
 return "Unknown";  
 }  
 *//Method to calculate the commission* private double calculateCommission(double sales, boolean isSameRegion) {  
 *//calculation logic* if (sales <= 100000000) {  
 *//calculation if the sale is less than 100000000 ( if same region > yes commission is 5% no commission is 3%)* return sales \* (isSameRegion ? 0.05 : 0.03);  
 } else {  
 *//calculation if the sale is more than 100000000* return 100000000 \* (isSameRegion ? 0.05 : 0.03) + (sales - 1000000) \* (isSameRegion ? 0.07 : 0.04);  
 }  
 }  
 *//alert sound effect for message* private void playAlertSound() {  
 if (mediaPlayer != null) {  
 mediaPlayer.start();  
 }  
 }  
 *//alert sound effect for message stop* @Override  
 protected void onDestroy() {  
 if (mediaPlayer != null) {  
 mediaPlayer.release();  
 }  
 super.onDestroy();  
 }  
  
}

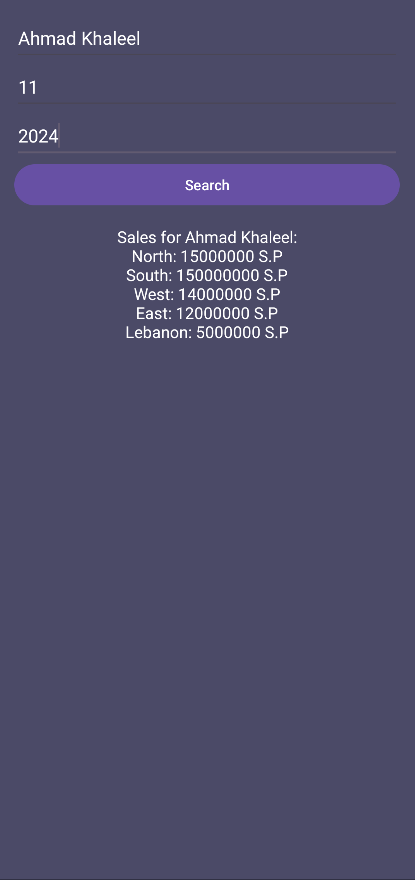
في ملف Activity\_add\_sales.XML يكون:

*<?*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:gravity="center"  
 android:orientation="vertical"  
 android:background="#4B4A67">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:visibility="visible"  
 android:layout\_margin="10dp">  
  
 <ImageView  
 android:id="@+id/im\_yourPhoto"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 tools:srcCompat="@tools:sample/avatars"  
 android:layout\_margin="10dp"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="124dp"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:orientation="horizontal">  
  
 <EditText  
 android:id="@+id/ed\_name"  
 android:layout\_width="126dp"  
 android:layout\_height="60dp"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:hint="@string/hint\_name"  
 android:inputType="text"  
 android:textAlignment="center"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <Button  
 android:id="@+id/btn\_search"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="@string/btn\_search" />  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/tv\_yourRegion"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="@string/tv\_region"  
 android:textAlignment="center"  
 android:textSize="20dp"  
 android:gravity="center"  
 android:textColor="#FFFFFFFF"/>  
  
 </LinearLayout>  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/tv\_date"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="@string/tv\_date"  
 android:textAlignment="center"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_month"  
 android:layout\_width="163dp"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/et\_month"  
 android:inputType="number"  
 android:textAlignment="center"  
 android:textSize="20sp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_year"  
 android:layout\_width="186dp"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="@string/et\_year"  
 android:inputType="number"  
 android:textAlignment="center"  
 android:textSize="20sp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 </LinearLayout>  
  
 <TableLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp">  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/tv\_north"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/tv\_north"  
 android:textAlignment="center"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_north"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:autofillHints="0"  
 android:textAlignment="center"  
 android:hint="@string/hint\_money"  
 android:inputType="number"  
 android:textSize="20sp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/tv\_south"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/tv\_south"  
 android:textAlignment="center"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_south"  
 android:layout\_width="349dp"  
 android:layout\_height="wrap\_content"  
 android:autofillHints="0"  
 android:textAlignment="center"  
 android:hint="@string/hint\_money"  
 android:inputType="number"  
 android:textColorHint="#DDAAAAAA"  
 android:textSize="20sp"  
 android:textColor="#FFFFFFFF"/>  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/tv\_east"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/tv\_east"  
 android:textAlignment="center"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_east"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:autofillHints="0"  
 android:textAlignment="center"  
 android:hint="@string/hint\_money"  
 android:inputType="number"  
 android:textSize="20sp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 >  
  
 <TextView  
 android:id="@+id/tv\_west"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/tv\_west"  
 android:textAlignment="center"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_west"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:autofillHints="0"  
 android:textAlignment="center"  
 android:hint="@string/hint\_money"  
 android:inputType="number"  
 android:textSize="20sp"  
 android:textColor="#FFFFFFFF"  
 android:textColorHint="#DDAAAAAA"/>  
  
 </TableRow>  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/tv\_lebanon"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/tv\_lebanon"  
 android:textAlignment="center"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_lebanon"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:autofillHints="0"  
 android:textAlignment="center"  
 android:hint="@string/hint\_money"  
 android:inputType="number"  
 android:textColorHint="#DDAAAAAA"  
 android:textSize="20sp"  
 android:textColor="#FFFFFFFF"/>  
  
 </TableRow>  
  
 </TableLayout>  
  
 <LinearLayout  
 android:id="@+id/btn\_cost"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <TableLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1">  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" >  
  
 <Button  
 android:id="@+id/btn\_profit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="onclick"  
 android:text="@string/btn\_text"  
 android:layout\_margin="5dp"/>  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" >  
  
 <Button  
 android:id="@+id/btn\_view\_commissions"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="View Commissions"  
 android:layout\_margin="5dp"/>  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" >  
  
 <Button  
 android:id="@+id/btn\_view\_sales"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="View Sales"  
 android:layout\_margin="5dp"/>  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
 </TableLayout>  
  
 </LinearLayout>  
  
</LinearLayout>

##### نهاية النشاط Add Employee

## نشاط البحث عن مبيعات المندوب Search Sale Activity:

### الواجهات:



### الأكواد

في SeachSaleActivity.java يكون:

public class SearchSalesActivity extends AppCompatActivity {  
 *//declaring variables* EditText ed\_name, ed\_month, ed\_year;  
 Button btn\_search;  
 TextView tv\_result;  
 SQLiteDatabase db;  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_search\_sales*);  
 *// Initialize views* ed\_name = findViewById(R.id.*ed\_name*);  
 ed\_month = findViewById(R.id.*ed\_month*);  
 ed\_year = findViewById(R.id.*ed\_year*);  
 btn\_search = findViewById(R.id.*btn\_search*);  
 tv\_result = findViewById(R.id.*tv\_result*);  
 *// Initialize database* db = openOrCreateDatabase("EmployeeDB", *MODE\_PRIVATE*, null);  
 *// Search button click listener to handle search* btn\_search.setOnClickListener(view -> searchSales());  
 }  
 *// Method to search sales* @SuppressLint("SetTextI18n")  
 private void searchSales() {  
 String name = ed\_name.getText().toString().trim().toLowerCase(); *//case sensitive* String month = ed\_month.getText().toString().trim();  
 String year = ed\_year.getText().toString().trim();  
 *// Entered Data Validation* if (name.isEmpty() || month.isEmpty() || year.isEmpty()) {  
 Toast.*makeText*(this, "Please fill all fields", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 *//sql statement to retrieve data from the database* Cursor cursor = db.rawQuery("SELECT \* FROM sales WHERE Name = ? AND Month = ? AND Year = ?",  
 new String[]{name, month, year});  
 *//checking if sales exists* if (cursor.getCount() == 0) {  
 tv\_result.setText("No sales found for the given criteria.");  
 } else {  
 *//extract the retrieved data and storing in variables* StringBuilder result = new StringBuilder();  
 while (cursor.moveToNext()) {  
 String north = cursor.getString(4); *// SaleNorthRegion* String south = cursor.getString(5); *// SaleSouthRegion* String west = cursor.getString(6); *// SaleWestRegion* String east = cursor.getString(7); *// SaleEastRegion* String lebanon = cursor.getString(8); *// SaleLebanonRegion* result.append("Sales for ").append(name).append(":\n")  
 .append("North: ").append(north).append(" S.P\n")  
 .append("South: ").append(south).append(" S.P\n")  
 .append("West: ").append(west).append(" S.P\n")  
 .append("East: ").append(east).append(" S.P\n")  
 .append("Lebanon: ").append(lebanon).append(" S.P\n\n");  
 }  
 tv\_result.setText(result.toString());  
 }  
 cursor.close();  
 }  
}

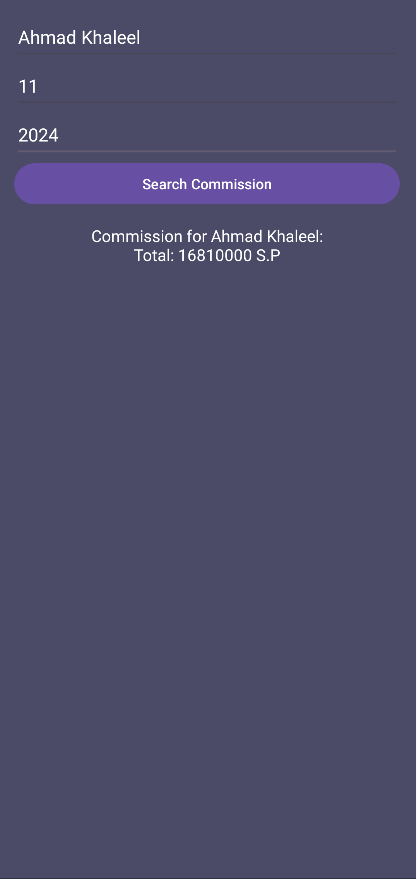
في ملف activity\_seach\_sale.XML يكون:

*<?*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"  
 android:background="#4B4A67">  
  
 <EditText  
 android:id="@+id/ed\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Employee Name"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_month"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Month"  
 android:inputType="number"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_year"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Year"  
 android:inputType="number"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <Button  
 android:id="@+id/btn\_search"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Search" />  
  
 <TextView  
 android:id="@+id/tv\_result"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="16dp"  
 android:textSize="16sp"  
 android:gravity="center"  
 android:text="Results will be displayed here"  
 android:textColor="#FFFFFFFF"/>  
  
</LinearLayout>

##### نهاية النشاط Add Employee

## نشاط البحث عن عمولات المندوب Search Commission Activity:

### الواجهات:



### الأكواد:

في ملف SearchCommission.java يكون:

public class SearchCommissionActivity extends AppCompatActivity {  
 *//declaring variables* EditText ed\_name, ed\_month, ed\_year;  
 Button btn\_search;  
 TextView tv\_result;  
 SQLiteDatabase db;  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_search\_commission*);  
  
 *// Initialize views* ed\_name = findViewById(R.id.*ed\_name*);  
 ed\_month = findViewById(R.id.*ed\_month*);  
 ed\_year = findViewById(R.id.*ed\_year*);  
 btn\_search = findViewById(R.id.*btn\_search*);  
 tv\_result = findViewById(R.id.*tv\_result*);  
  
 *// Initialize database* db = openOrCreateDatabase("EmployeeDB", *MODE\_PRIVATE*, null);  
 db.execSQL("CREATE TABLE IF NOT EXISTS commissions(Name NVARCHAR, Month INTEGER, Year INTEGER, TotalCommission NUMBER)");  
 *// Search button click listener to handle search* btn\_search.setOnClickListener(view -> searchCommission());  
 }  
 *// Method to search commission* @SuppressLint("SetTextI18n")  
 private void searchCommission() {  
 String name = ed\_name.getText().toString().trim().toLowerCase(); *//case sensitive* String monthStr = ed\_month.getText().toString().trim();  
 String yearStr = ed\_year.getText().toString().trim();  
 *// Entered Data Validation* if (name.isEmpty() || monthStr.isEmpty() || yearStr.isEmpty()) {  
 Toast.*makeText*(this, "Please fill all fields", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 int month, year;  
 try {  
 month = Integer.*parseInt*(monthStr);  
 year = Integer.*parseInt*(yearStr);  
 *// Entered Data Validation for month* if (month < 1 || month > 12) {  
 Toast.*makeText*(this, "Invalid month. Enter a value between 1 and 12.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 *// Entered Data Validation for year* if (year < 1900 || year > 2100) { *// تحديد نطاق زمني منطقي للسنة* Toast.*makeText*(this, "Invalid year. Enter a valid year.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 } catch (NumberFormatException e) {  
 *// Entered Data Validation for month and year* Toast.*makeText*(this, "Month and Year must be numeric.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 *//sql statement to retrieve data from the database* Cursor cursor = db.rawQuery("SELECT TotalCommission FROM commissions WHERE Name = ? AND Month = ? AND Year = ?",  
 new String[]{name, String.*valueOf*(month), String.*valueOf*(year)});  
  
 if (cursor.getCount() == 0) {  
 *//display error message if no commission found* tv\_result.setText("No commission found for the given criteria.");  
 } else {  
 *//extract the retrieved data and storing in variables* cursor.moveToFirst();  
 String totalCommission = cursor.getString(0); *// TotalCommission* tv\_result.setText(String.*format*("Commission for %s:\nTotal: %s S.P", name, totalCommission));  
 }  
 cursor.close();  
 }  
}

في ملف Activity\_search\_commission.XML يكون:

*<?*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"  
 android:background="#4B4A67">  
  
 <EditText  
 android:id="@+id/ed\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Employee Name"  
 android:inputType="text"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_month"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Month"  
 android:inputType="number"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <EditText  
 android:id="@+id/ed\_year"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Year"  
 android:inputType="number"  
 android:minHeight="48dp"  
 android:textColorHint="#DDAAAAAA"  
 android:textColor="#FFFFFFFF"/>  
  
 <Button  
 android:id="@+id/btn\_search"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Search Commission" />  
  
 <TextView  
 android:id="@+id/tv\_result"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:textAlignment="center"  
 android:paddingTop="16dp"  
 android:text="Results will be displayed here"  
 android:textColor="#FFFFFFFF"  
 android:textSize="16sp" />  
  
</LinearLayout>

##### نهاية النشاط Add Employee