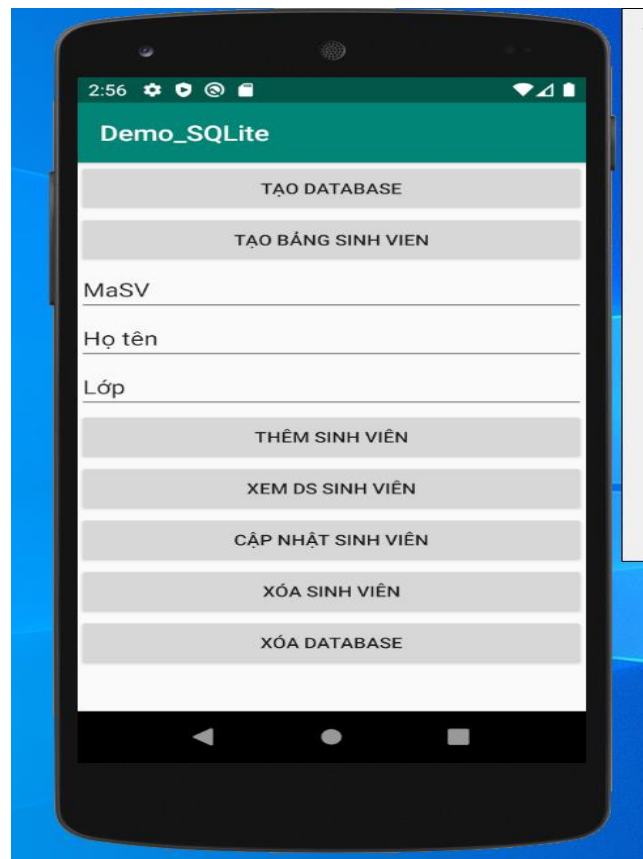


BÀI 9: PHIẾU BÀI TẬP

- Bài tập 1.** Tạo ứng dụng quản lý sinh viên với các chức năng đơn giản (Phần 1)1
- Bài tập 2.** Tạo ứng dụng quản lý sinh viên với các chức năng đơn giản (Phần 2)5
- Bài tập 3.** Phát triển ứng dụng quản lý sinh viên có thêm chức năng login dành cho admin, có thêm ListView để khi thêm sinh viên có thể nhìn thấy được trực quan.....6
- Bài tập 4.** Sinh viên Class Product bao gồm các thuộc tính sau:.....6
- Bài tập 5.** Sinh viên thiết kế ứng dụng quản lý TieemVacXinCovid cho công ty ABC solution. Biết rằng cần có các chức năng sau:.....7
- Bài tập 6.** Viết ứng dụng xemphimonline cho phép người sử dụng đăng ký tài khoản để xem phim. Người quản trị admin thì được thêm bớt sửa xóa phim (trong đề bài để đơn giản hóa cho phép phim sử dụng đường link của bên thứ 3 như youtube).7

Bài tập 1. Tạo ứng dụng quản lý sinh viên với các chức năng đơn giản (Phần 1)

- Thiết kế giao diện ứng dụng
- Import các thư viện cần thiết để thao tác trên SQLite
- Tạo database là bảng tblsv gồm (masv Text PK, tensv Text, lop Text)
- Thực hiện chức năng thêm sinh viên



Hình 1: Giao diện ứng dụng cần thiết kế

Hướng dẫn:

Bước 1 : thiết kế giao diện trên 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">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="btnTaoDB"
        android:text="Tạo database" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tạo bảng Sinh Viên"
```

```
android:onClick="btnTaoBangSV"/>
```

```
<EditText
```

```
    android:id="@+id/edtMasv"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:text="MaSV" />
```

```
<EditText
```

```
    android:id="@+id/edtHoten"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:text="Họ tên" />
```

```
<EditText
```

```
    android:id="@+id/edtLop"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:text="Lớp" />
```

```
<Button
```

```
    android:id="@+id/button3"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Thêm Sinh viên"  
    android:onClick="btnThemSV"/>
```

```
<Button
```

```
    android:id="@+id/button4"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="btnXemdsSV"  
    android:text="Xem ds sinh viên" />
```

```
<Button
```

```
    android:id="@+id/button5"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="btnUpdateSV"  
    android:text="Cập nhật sinh viên" />
```

```
<Button
```

```
    android:id="@+id/button6"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Xóa Sinh viên"  
    android:onClick="btnXoaSV"/>
```

```
<Button
    android:id="@+id/button7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Xóa database"
    android:onClick="btnXoadb"/>
```

```
</LinearLayout>
```

Bước 2: Thêm các thư viện cần thiết để thao tác trên SQLite

```
// thêm thư viện để thao tác với CSDL
import android.view.*;
import android.database.sqlite.*;
```

Bước 3: Tạo code để tạo CSDL và tạo bảng tblsv

```
// khai báo database là đối tượng lớp SQLiteDatabase
SQLiteDatabase database;
EditText edtMasV, edtHoten, edtLop;
// code tạo csdl Demosql.db
// câu lệnh openOrCreateDatabase dùng để mở ra nếu có sẵn hoặc tạo mới nếu chưa có
public void btnTaoDB(View v) {
    database=openOrCreateDatabase("Demosql.db", MODE_PRIVATE, null);
}
//sử dụng câu lệnh SQL tạo bảng
// lưu ý SQLite chỉ thao tác trên file lên sử dụng string sql và thực thi bằng câu lệnh execSQL
public void btnTaoBangSV(View v) {
    String sql="CREATE table tblsv (masv TEXT primary key,tensv TEXT,lop TEXT) ";
    database.execSQL(sql);
}
```

Bước 4: Tạo code để thêm sinh viên vào trong bảng tblsv

```
public void btnThemSV(View v) {
    // lấy dữ liệu từ giao diện vào các biến
    edtMasV=findViewById(R.id.edtMasv);
    edtHoten=findViewById(R.id.edtHoten);
    edtLop=findViewById(R.id.edtLop);
    // sử dụng ContentValues để lưu trữ các giá trị
    ContentValues sv= new ContentValues();
    // sử dụng phương thức put để truyền giá trị
    sv.put("masv", edtMasV.getText().toString());
    sv.put("tensv", edtHoten.getText().toString());
    sv.put("lop", edtLop.getText().toString());
    // sử dụng database.insert để chèn vào bảng
    // nếu kq trả về = -1 là thêm không thành công trái lại đã thêm thành công
    if(database.insert("tblsv", null, sv) != -1) {
        Toast.makeText(this, "Thêm sv thành công", Toast.LENGTH_LONG).show();
    }
    else
    {

```

```
Toast.makeText(this, "CÓ LỖI", Toast.LENGTH_LONG).show();  
}  
}
```

Bài tập 2. Tạo ứng dụng quản lý sinh viên với các chức năng đơn giản (Phần 2)

Ở nội dung bài tập 2, chúng ta phát triển bài tập 1 thêm các chức năng:

- Xem danh sách sinh viên
- Cập nhật sinh viên
- Xóa sinh viên
- Xóa CSDL

Hướng dẫn:

a. Để xem danh sách sinh viên chúng ta thực thi code cho nút lệnh xem dssv

```
public void btnXemdssv(View v){  
    // Cursor dùng để lưu kết quả truy vấn  
    Cursor  
c=database.query("tblsv", null, null, null, null, null, null);  
    // gán biến data rỗng để lưu kết quả từ truy vấn và hiển  
thị  
    String data="";  
    // chuyển con trỏ về bản ghi đầu tiên để duyệt từ đầu  
c.moveToFirst();  
    // vòng lặp duyệt trong cursor kết thúc khi c ở sau bản  
ghi cuối  
    while (!c.isAfterLast()){  
        // gán kết quả cho data  
        data+=c.getString(0)+"-" +c.getString(1)+"-  
"+c.getString(2)+"\n";  
        // tiến thêm 1 bản ghi  
        c.moveToNext();  
    }  
    // đóng cursor lại  
c.close();  
    // hiện thị dssv  
    Toast.makeText(this, data, Toast.LENGTH_LONG).show();  
}
```

b. Để cập nhật sinh viên chúng ta cài đặt code như sau:

```
public void btnUpdateSv(View v){  
    // dùng contentValues lưu trữ giá trị  
    ContentValues sv =new ContentValues();  
    // cập nhật tensv  
    sv.put("tensv", edtHoten.getText().toString());  
    // sử dụng câu lệnh update có điều kiện tại masv  
truyền vào  
    if(database.update("tblsv", sv, "masv=?", new  
String[] {edtMaSV.getText().toString()})!=0){  
        Toast.makeText(this, "Cập nhật thành  
công", Toast.LENGTH_LONG).show();  
    }
```

```
    }  
    else  
        Toast.makeText(this, "Có  
lỗi", Toast.LENGTH_LONG).show();  
    }
```

c. Để xóa sinh viên chúng ta cài đặt code như sau:

```
public void btnXoaSV(View v) {  
    // xóa sv nếu có masv bằng masv truyền vào  
    if(database.delete("tblsv", "masv=?", new  
String[] {edtMaSV.getText().toString()}) != 0) {  
        Toast.makeText(this, "Xóa sv thành  
công", Toast.LENGTH_LONG).show();  
    }  
    else  
        Toast.makeText(this, "Có  
lỗi", Toast.LENGTH_LONG).show();  
    }
```

d. Để xóa csdl, chúng ta cài đặt code như sau :

```
public void btnXoadb(View v) {  
    // sử dụng câu lệnh deleteDatabase để xóa db  
    if(deleteDatabase("Demosql.db")) {  
        Toast.makeText(this, "Xóa db thành  
công", Toast.LENGTH_LONG).show();  
    }  
    else  
        Toast.makeText(this, "Có  
lỗi", Toast.LENGTH_LONG).show();  
    }
```

Bài tập 3. Phát triển ứng dụng quản lý sinh viên có thêm chức năng login dành cho admin, có thêm ListView để khi thêm sinh viên có thể nhìn thấy được trực quan.

Bài tập 4. Sinh viên Class Product bao gồm các thuộc tính sau:

- Id
- Name
- Price
- Image

Thực hiện Custom ListView sau cho hiển thị được Image, Name và Price của Product trên ListView.

Sinh viên tạo Class ProductDAO để xem, thêm, xóa và sửa Product.

Sinh viên sử dụng SQLite để tạo CSDL QLBH gồm bảng Product và bảng Account (uId - PK, UserName, Password),

Thiết kế ứng dụng cho phép login vào, sau khi login thành công sử dụng class ProductDAO thực hiện các thao tác :

- Thêm sản phẩm
- Xóa sản phẩm
- Cập nhật sản phẩm
- Tìm kiếm sản phẩm (theo tên)

Bài tập 5. Sinh viên thiết kế ứng dụng quản lý TieemVacXinCovid cho công ty ABC solution. Biết rằng cần có các chức năng sau:

- Cập nhật thông tin nhân viên
- Admin phải login vào hệ thống
- Kiểm tra thống kê tình trạng tiêm của nhân viên
- Đưa ra danh sách các nhân viên cần tiêm (chưa tiêm hoặc mũi tiêm gần nhất quá 6 tháng)

Bài tập 6. Viết ứng dụng xemphimonline cho phép người sử dụng đăng ký tài khoản để xem phim. Người quản trị admin thì được thêm bớt sửa xóa phim (trong đề bài để đơn giản hóa cho phép phim sử dụng đường link của bên thứ 3 như youtube).

Người sử dụng có các chức năng:

- Đánh giá phim (* từ 1-5)
- Đưa danh sách phim vào phim yêu thích
- Tìm kiếm phim
- Xem tiếp các bộ phim chưa xem xong (phim nhiều tập)