

TRƯỜNG ĐẠI HỌC BÀ RỊA – VŨNG TÀU

KHOA KỸ THUẬT - CÔNG NGHỆ

Logo

Description automatically generated

BÁO CÁO MÔN LẬP TRÌNH ỨNG DỤNG DI ĐỘNG VÀ GAME 2

ĐỀ TÀI: PHPMySQL

# 

|  |  |  |
| --- | --- | --- |
| Trình độ đào tạo | : | Đại Học |
| Tên sinh viên | : | Nguyễn Văn Sơn |
| Mã sinh viên | : | 20030016 |
| Lớp | : | DH20LT |
| Khoa | : | Kỹ thuật – Công nghệ |
| Giảng viên hướng dẫn | : | Nguyễn Thị Minh Nương |

# **MỤC LỤC**

[**MỤC LỤC** 2](#_Toc120008052)

[**A.** **LÝ THUYẾT** 4](#_Toc120008053)

[**I.** **Ngôn ngữ PHP** 4](#_Toc120008054)

[**1.** **Giới thiệu** 4](#_Toc120008055)

[**2.** **Cách thức hoạt động** 4](#_Toc120008056)

[**3.** **Một số câu lệnh cơ bản.** 4](#_Toc120008057)

[a) Cú pháp 4](#_Toc120008058)

[b) Kiểu dữ liệu và biến 5](#_Toc120008059)

[c) Các phép toán số học 6](#_Toc120008060)

[d) Các phép toán so sánh 6](#_Toc120008061)

[e) Các cấu trúc rẽ nhánh và vòng lặp 7](#_Toc120008062)

[f) Mảng 8](#_Toc120008063)

[g) Lớp và đối tượng trong PHP 9](#_Toc120008064)

[**II.** **Hệ quản trị CSDL MySQL** 9](#_Toc120008065)

[**1.** **Giới thiệu hệ quản trị CSDL MySQL** 9](#_Toc120008066)

[**2.** **PHP kết nối với MySQL** 11](#_Toc120008067)

[**3.** **Cài đặt và cấu hình XAMPP** 14](#_Toc120008068)

[**III.** **PHPmySql với Android** 16](#_Toc120008069)

[**B.** **Thực hành** 17](#_Toc120008070)

[**I.** **Giao diện** 17](#_Toc120008071)

[**1.** **Drawable** 18](#_Toc120008072)

[a) Themes 18](#_Toc120008073)

[b) Colors 18](#_Toc120008074)

[c) card\_shape.xml 18](#_Toc120008075)

[d) behind\_card.xml 19](#_Toc120008076)

[e) dialog\_shape.xml 19](#_Toc120008077)

[f) icon 19](#_Toc120008078)

[**2.** **layout** 19](#_Toc120008079)

[a) main\_activity.xml 19](#_Toc120008080)

[b) Card.xml 21](#_Toc120008081)

[c) dialog.xml 23](#_Toc120008082)

[d) dialog\_delete.xml 24](#_Toc120008083)

[**II.** **Code** 25](#_Toc120008084)

[**1.** **Chuẩn bị** 25](#_Toc120008085)

[**2.** **API php** 26](#_Toc120008086)

[**3.** **Code java** 29](#_Toc120008087)

[a) Cấu hình retrofit và các phương thức gọi api 29](#_Toc120008088)

[b) Lớp Furniture 30](#_Toc120008089)

[c) FurnitureAdapter 31](#_Toc120008090)

[d) MainActivity 32](#_Toc120008091)

# **LÝ THUYẾT**

## **Ngôn ngữ PHP**

### **Giới thiệu**

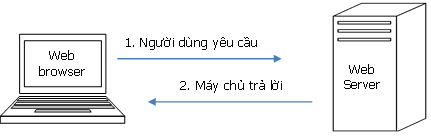
PHP là ngôn ngữ xử lý ở phía Server, là 1 công cụ rất mạnh trong việc phát triển web có tính tương tác

PHP được sử dụng rất phổ biến, hoàn toàn miễn phí và là đối thủ cạnh tranh đáng gờm của Microsoft ASP.

PHP được phát triển từ một sản phẩm có tên là PHP/FI. PHP/FI do Rasmus Lerdorf tạo ra năm 1995, ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng.

PHP/FI, viết tắt từ "Personal Home Page/Forms Interpreter", bao gồm một số các chức năng cơ bản cho PHP như ta đã biết đến chúng ngày nay. Nó có các biến kiểu như Perl, thông dịch tự động các biến của form và cú pháp HTML nhúng. Cú pháp này giống như của Perl, mặc dù hạn chế hơn nhiều, đơn giản và có phần thiếu nhất quán.

### **Cách thức hoạt động**



Khi người dùng muốn truy cập một tập tin PHP, họ phải gửi yêu cầu thông qua việc truy cập vào đường dẫn đến tập tin. Sau đó, máy chủ web sẽ tiếp nhận yêu cầu và biên dịch tập tin PHP thành mã HTML thuần túy rồi gửi trả về trình duyệt của người dùng.

### **Một số câu lệnh cơ bản.**

#### Cú pháp

- Được giới hạn bởi cặp kí hiệu <?php …. ?> hoặc <? …. ?>

-Các biến được bắt đầu bằng dấu $ và không cần khai báo trước kiểu dữ liệu.

-Mỗi câu lệnh được kết thúc bởi dấu ;

-Chú thích được đặt trong cặp kí hiệu /\* …. \*/ với đoạn chú thích hoặc // và # với dòng chú thích.

-Về cú pháp các từ khóa và ngôn ngữ, PHP tương tự hầu hết các ngôn ngữ lập trình bậc cao có cú pháp kiểu C, C++, Java, Perl …

#### Kiểu dữ liệu và biến

**-PHP có ba kiểu dữ liệu cơ bản: interger, double và string**

+Integer là kiểu số nguyên chiếm 4byte bộ nhớ, giá trị nằm trong khoảng -2 tỷ đến +2 tỷ

+Double là kiểu số thực, phạm vi biểu diễn nằm trong khoảng (-10^-308, 10^308)

+Kiểu dữ liệu string bao gồm các kí tự, chuỗi kí tự và con số.

**-Trong PHP, không cần khai báo kiểu dữ liệu khi khai báo biến. Mọi biến số trong PHP đều bắt đầu bằng ký hiệu $**

PHP không có khái niệmTRUE và FALSE mà hiểu TRUE là giá trị 1 và FALSE là giá trị 0.

Ví dụ

$a = 1; // a là biến có kiểu dữ liệu int

$b = 1.5; // b là biến có kiểu dữ liệu double

$c = “banking academy”; // c là biến có kiểu dữ liệu string

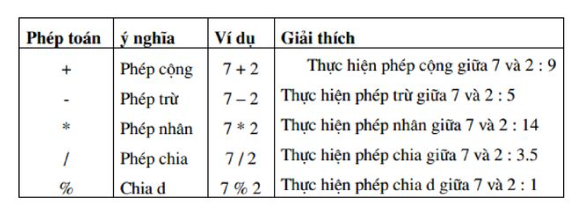
Có thể ép kiểu trong PHP theo cách sau

$a = 15,5; // Biến a đang có kiểu dữ liệu double

$b = (int)$a; // Biến b có kiểu dữ liệu int, giá trị = 15

$c = (string) $b; // Biến c có kiểu dữ liệu string, giá trị = “15”

#### Các phép toán số học



#### Các phép toán so sánh

Table

Description automatically generated

#### Các cấu trúc rẽ nhánh và vòng lặp

**-Cấu trúc rẽ nhánh if – else:**

if( biểu thức logic){

//Khối lệnh thực hiện nếu biểu thức đúng

}else{

// Khối lệnh thực hiện nếu biểu thức sai

}

**-Cấu trúc rẽ nhánh switch-case:**

switch ( biểu thức )

{

case gt1:

câu lệnh 1;

break;

case gt2:

câu lệnh 2;

…………..

default:

câu lệnh n;

}

**-Vòng lặp for:**

for( biểu thức 1; biểu thức 2 (logic); biểu thức 3)

{

//Khối lệnh

}

**-Vòng lặp while:**

while( biểu thức )

{

//Khối lệnh

}

#### Mảng

**-Mảng một chiều**

Khai báo

$mang = array(“gt1”, “gt2”, ….. );

=> $mang[0] = “gt1”, $mang[1] = “gt2” …..

Truy xuất vào giá trị mảng:Một phần tử của mảng có 2 thành phần: chỉ số (index hoặc key) và giá trị (value):

$mang[<chỉ số>] = giá trị

Chỉ số của mảng có thể là số hoặc chuỗi, các giá trị của mảng không cần phải cùng kiểu giá trị

Ví dụ:

$mang[1] = 100;

$mang[“hoten”] = “Nguyễn Văn Nam”;

$mang[“tuoi”] = 19;

$mang[] = “Banking”; // PHP sẽ tự động gán chỉ số là số tăng dần

**-Duyệt mảng bằng vòng lặp foreach**

Cú pháp

$mang = array(“hoten” => “Nam”, “tuoi” => 18, “quequan” => “Hà Nội”);

foreach($mang as $key => $value){

print\_r($key);printr\_r($value);

}

$key: Các chỉ số của lần lượt từ phần tử đầu đến phần tử cuối

$value: Các giá trị của màng tương ứng lần lượt từ phần tử đầu đến phần tử cuối

Có thể không cần $key trong vòng lặp

#### Lớp và đối tượng trong PHP

Khai báo lớp

class <tên lớp>{

private $attribute1;

private $attribute 2; // Định nghĩa các thuộc tính trong lớp

function \_\_construct([params]){

//Phương thức khởi tạo

}

function method1(params …) // Định nghĩa các phương thức

{

//Khối lệnh xử lý trong method1

}

}

## **Hệ quản trị CSDL MySQL**

### **Giới thiệu hệ quản trị CSDL MySQL**

Là một hệ quản trị CSDL quan hệ mã nguồn mở.

Năm 2013, MySQL được đánh giá là hệ quản trị CSDL được sử dụng nhiều thứ 2 trên thế giới.

MySQL là hệ thống cơ sở dữ liệu tiêu chuẩn cho Website đảm bảo đáp ứng được cho dữ liệu và số lượng người dùng lớn (như Facebook, Twitter, và Wikipedia).

**So sánh MySQL vs SQLite**

|  |  |
| --- | --- |
| MySQL | SQLite |
| Được phát triển bởi Oracle vào tháng 5 năm 1995. | Do D. Richard Hipp phát triển vào tháng 8 năm 2000. |
| MySQL được phát triển bằng ngôn ngữ C và C++. | SQLite chỉ được phát triển bằng ngôn ngữ C. |
| MySQL yêu cầu một máy chủ cơ sở dữ liệu để hoạt động. Do đó, nó tuân theo kiến ​​trúc client/server. | SQLite không yêu cầu máy chủ chạy. Do đó, nó không có máy chủ. |
| Nó có thể xử lý nhiều kết nối cùng một lúc. | Nó chỉ có thể xử lý một kết nối tại một thời điểm. |
| Nó có khả năng mở rộng cao và có thể xử lý một lượng lớn dữ liệu rất hiệu quả. | Nó chỉ có thể xử lý một tập dữ liệu nhỏ nếu khối lượng dữ liệu tăng lên làm giảm hiệu suất của nó. |
| Nó yêu cầu không gian lớn trong bộ nhớ để hoạt động (khoảng 600 Mb). | Nó chỉ yêu cầu một số KB dung lượng vì nó rất nhẹ khoảng (250Kb-300Kb). |
| MySQL hỗ trợ nhiều môi trường người dùng. | SQLite không hỗ trợ môi trường nhiều người dùng. |

**So sánh MySQL vs Firebase**

|  |  |
| --- | --- |
| Nền tảng phát triển ứng dụng từ Google. | Cơ sở dữ liệu SQL mã nguồn mở |
| Miễn phí để bắt đầu Trả tiền khi bạn tiếp tục sau đó | miễn phí |
| Không phải mã nguồn mở | Mã nguồn mở |
| Quản lý dữ liệu theo dạng key-value , biểu đồ hoặc tài liệu. | MySQL dựa trên bảng và có các lược đồ được xác định trước. Thích hợp xử lý dữ liệu phức tạp. |
| Firebase là cơ sở dữ liệu NoSQL đồng bộ hóa và lưu trữ dữ liệu theo thời gian thực. | MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở dựa trên SQL |
| Hỗ trợ ít ngôn ngữ hơn so với MySQL: Java, JavaScript, Objective-C… | Các ngôn ngữ lập trình được MySQL hỗ trợ nhiều hơn những ngôn ngữ mà Firebase hỗ trợ. Các ngôn ngữ bao gồm Ada, C, C#, C++, Delphi, Java, PHP, Python… |

### **PHP kết nối với MySQL**

PHP 5 trở về sau có thể kết nối MySQL theo 2 cách:

-MySQLi extension (chữ "i" là viết tắt của improved)

-PDO (PHP Data Objects)

PDO có thể kết nối với 12 hệ quản trị CSDL khác nhau, trong khi MySQLi chỉ làm việc với MySQL.

Nếu dự án cần trao đổi dữ liệu giữa nhiều hệ quản trị CSDL khác nhau thì PDO sẽ hợp lý hơn

Cả 2 phương pháp đều hướng đối tượng, nhưng MySQLi hỗ trợ cả hướng thủ tục

Cả 2 phương pháp đều chống SQL Injection tốt.

**Kết nối MySQL với MySQLi theo hướng đối tượng**

<?php

$servername = "localhost";

$username = "username";

$password = "password";

// Create connection

$conn = new mysqli($servername, $username, $password);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully";

?>

**Kết nối MySQL với MySQLi theo hướng thủ tục**

<?php

$servername = "localhost";

$username = "username";

$password = "password";

// Create connection

$conn = mysqli\_connect($servername, $username, $password);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

echo "Connected successfully";

?>

**Kết nối MySQL với PDO**

<?php

$servername = "localhost";

$username = "username";

$password = "password";

try {

$conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);

// set the PDO error mode to exception

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

echo "Connected successfully";

} catch(PDOException $e) {

echo "Connection failed: " . $e->getMessage();

}

?>

**Một số hàm thường dùng khi làm việc với MySQL trong PHP**

|  |  |
| --- | --- |
| prepare() | khai báo câu truy vấn |
| bindparam() | truyền đối số vào trong câu truy vấn |
| bindresult() | truyền kết quả truy vấn vào trong các biến |
| excute() | thực thi câu truy vấn |
| fetch() | duyệt từng dòng của kết quả truy vấn |

**Truy vấn dữ liệu với MySQLi**

Text

Description automatically generated

**Text

Description automatically generatedThêm dữ liệu**

**Xóa sửa tương tự chỉ cần thay câu lệnh.**

### **Cài đặt và cấu hình XAMPP**

Xampp là chương trình tạo máy chủ Web (Web Server) trên máy tính cá nhân (Localhost) được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như PHPmyadmin. Xampp được download và sử dụng miễn phí tại:[đây](https://www.apachefriends.org/download.html)

Yêu cầu hệ thống : + 64 MB RAM (RECOMMENDED) MB free fixed disk + Windows 98, ME + Windows NT, 2000, XP (RECOMMENDED)

Graphical user interface, application

Description automatically generated

Thư mục gốc (Web Root) để chứa các trang web mặc nhiện của Xampp là ..\xampp\htdocs. Tất cả các Website hoặc thư mục chứa Website đều phải được đặt vào thư mục Web Root (htdocs) thì mới hoạt động được.

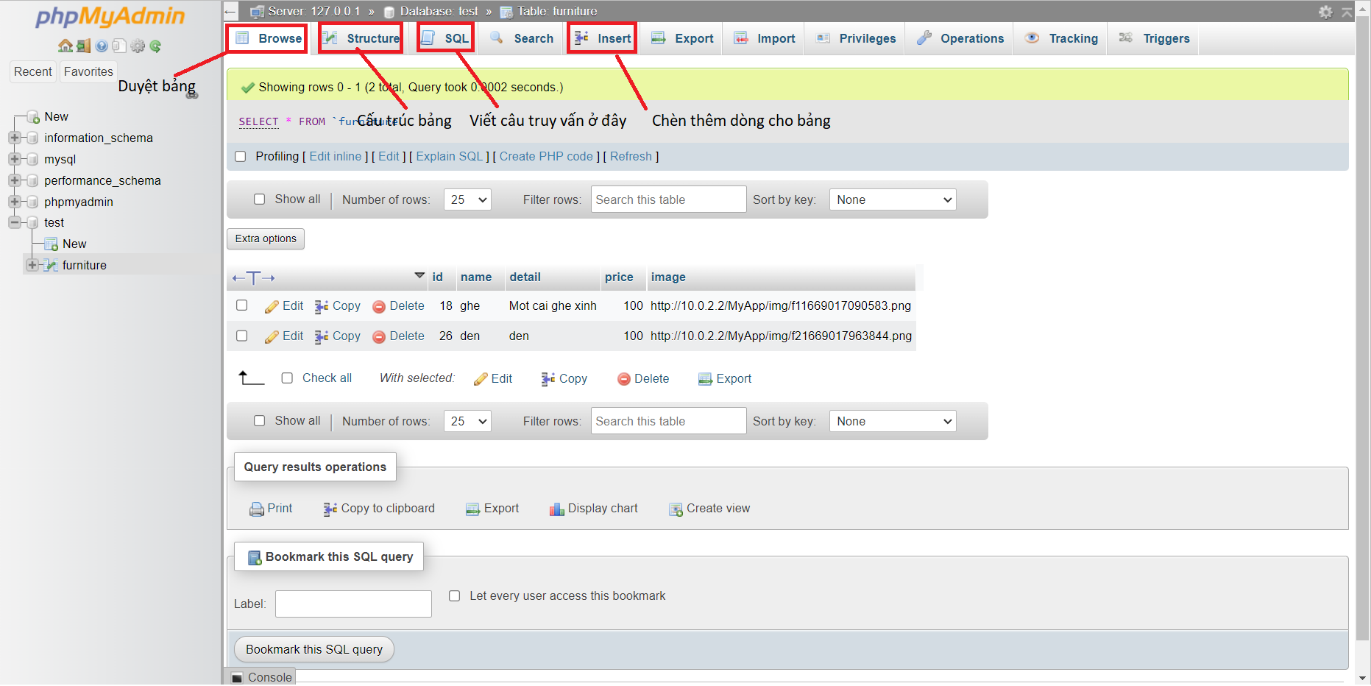
Graphical user interface, application

Description automatically generated

Chọn nút admin ở dòng MySQL để khởi động phpMyAdmin

Graphical user interface, application, Word

Description automatically generated



## **PHPmySql với Android**

Việc tiến hành Response và xử lý những dữ liệu trả về trong Android đôi khi phức tạp và khiến code trở nên rắc rối khó đọc. Để giải quyết vấn đền này 1 số thư viện ra đời, trong đó Retrofit là một ví dụ

Retrofit là một thư viện giúp phân tích cú pháp phản hồi API dễ dàng và được xử lý tốt hơn để sử dụng trong ứng dụng.

Retrofit ngày càng được phổ biến hơn với nhiều ưu điểm mà nó đem lại, ví dụ như là hiệu năng, thời gian hoàn thành task nhanh hơn các thư viện phổ biến khác như AsyncTask và Volley.

Table

Description automatically generated

# **Thực hành**

## **Giao diện**

Graphical user interface, text, application

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

### **Drawable**

#### Themes

<style name="Theme.PhpMySql" parent="Theme.AppCompat.DayNight.NoActionBar">

#### Colors

<color name="body\_color">#dad3c8</color>  
<color name="card\_color">#e9e5de</color>

#### card\_shape.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape android:shape="rectangle" xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@color/card\_color"/>  
 <corners android:radius="16dp"/>  
</shape>

#### behind\_card.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape android:shape="rectangle" xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@color/black"/>  
 <corners android:radius="16dp"/>  
</shape>

#### dialog\_shape.xml

<?xml version="1.0" encoding="utf-8"?>  
<shape android:shape="rectangle" xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@color/body\_color"/>  
 <corners android:radius="16dp"/>  
</shape>

#### icon

Text

Description automatically generated

### **layout**

#### main\_activity.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".MainActivity"  
 android:background="@color/body\_color">  
  
 <TextView  
 android:id="@+id/myshop"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Furniture Shop"  
 android:textColor="@color/black"  
 android:textSize="32sp"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:layout\_marginStart="40dp"  
 android:layout\_marginTop="50dp"/>  
  
 <EditText  
 android:id="@+id/search"  
 android:layout\_width="316dp"  
 android:layout\_height="50dp"  
 app:layout\_constraintTop\_toBottomOf="@id/myshop"  
 android:layout\_marginTop="12dp"  
 android:background="@drawable/card\_shape"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:hint="Search furniture..."  
 android:paddingLeft="16dp"/>  
  
 <ImageView  
 android:layout\_width="32dp"  
 android:layout\_height="32dp"  
 android:src="@drawable/ic\_baseline\_search\_24"  
 app:layout\_constraintEnd\_toEndOf="@id/search"  
 app:layout\_constraintTop\_toTopOf="@id/search"  
 android:layout\_margin="10dp"/>  
  
  
 <ListView  
 android:id="@+id/listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="560dp"  
 app:layout\_constraintTop\_toBottomOf="@id/search"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:divider="@null"  
 android:layout\_marginTop="15dp"  
 />  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/add"  
 android:layout\_width="64dp"  
 android:layout\_height="64dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:cardBackgroundColor="@color/black"  
 app:cardCornerRadius="32dp"  
 android:layout\_margin="32dp"  
 app:layout\_constraintBottom\_toBottomOf="parent">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:src="@drawable/ic\_baseline\_add\_24"  
 android:padding="12dp"/>  
 </androidx.cardview.widget.CardView>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

#### Card.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
  
 <ImageView  
 android:id="@+id/img"  
 android:layout\_width="180dp"  
 android:layout\_height="150dp"  
 android:scaleType="fitXY"  
 android:elevation="5dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:layout\_marginStart="32dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:src="@drawable/f1"/>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/cardbehind"  
 android:layout\_width="match\_parent"  
 android:layout\_height="120dp"  
 android:layout\_marginLeft="40dp"  
 android:layout\_marginRight="40dp"  
 android:layout\_marginTop="36dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:background="@drawable/behind\_card">  
  
 <LinearLayout  
 android:layout\_width="30dp"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:paddingRight="8dp">  
  
 <ImageView  
 android:id="@+id/close"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/ic\_baseline\_close\_24"  
 android:layout\_marginBottom="8dp"/>  
  
  
 <ImageView  
 android:id="@+id/edit"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/ic\_baseline\_drive\_file\_rename\_outline\_24"  
 android:layout\_marginBottom="8dp"/>  
 <ImageView  
 android:id="@+id/delete"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/ic\_baseline\_delete\_outline\_24"  
 android:layout\_marginBottom="8dp"/>  
  
  
 </LinearLayout>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/cardfront"  
 android:layout\_width="match\_parent"  
 android:layout\_height="120dp"  
 android:layout\_marginLeft="40dp"  
 android:layout\_marginRight="40dp"  
 android:layout\_marginTop="36dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:background="@drawable/card\_shape">  
  
 <TextView  
 android:id="@+id/name"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="20sp"  
 android:text="Sofa chair"  
 android:textColor="@color/black"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:textAlignment="textStart"  
 android:layout\_marginTop="12dp"/>  
  
 <TextView  
 android:id="@+id/detail"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="12sp"  
 android:text="Lorem Ipsum is simply dummy text."  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:paddingRight="8dp"  
 app:layout\_constraintTop\_toTopOf="@id/name"  
 android:textAlignment="textStart"  
 android:layout\_marginTop="32dp"/>  
  
 <TextView  
 android:id="@+id/price"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="20sp"  
 android:text="$180"  
 android:textColor="@color/black"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="@id/detail"  
 android:textAlignment="textStart"  
 android:layout\_marginTop="36dp"/>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

#### dialog.xml

<?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="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="48dp"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:background="@drawable/dialog\_shape"  
 android:gravity="center">  
  
  
 <TextView  
 android:id="@+id/type\_dialog"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Add new furniture"  
 android:textAlignment="center"  
 android:textColor="@color/black"  
 android:textSize="25sp"  
 android:textStyle="bold"/>  
  
 <ImageView  
 android:id="@+id/dialog\_img"  
 android:layout\_width="88dp"  
 android:layout\_height="88dp"  
 android:src="@drawable/ic\_baseline\_camera\_alt\_24"  
 android:layout\_marginTop="12dp"/>  
  
 <EditText  
 android:id="@+id/dialog\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="12dp"  
 android:hint="Name"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp"  
 android:background="@drawable/card\_shape"/>  
 <EditText  
 android:id="@+id/dialog\_detail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="12dp"  
 android:hint="Detail"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp"  
 android:background="@drawable/card\_shape"/>  
 <EditText  
 android:id="@+id/dialog\_price"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="12dp"  
 android:hint="Price"  
 android:inputType="numberDecimal"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp"  
 android:background="@drawable/card\_shape"/>  
  
 <Button  
 android:id="@+id/dialog\_add"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:text="add"  
 android:textStyle="bold"  
 android:layout\_marginTop="20dp"  
 android:textColor="@color/white"  
 android:background="@drawable/behind\_card"/>  
  
 <Button  
 android:id="@+id/dialog\_cancel"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:text="CANCEL"  
 android:textStyle="bold"  
 android:layout\_marginTop="12dp"  
 android:textColor="@color/white"  
 android:background="@drawable/behind\_card"/>  
  
</LinearLayout>

#### dialog\_delete.xml

<?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="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="48dp"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:background="@drawable/dialog\_shape"  
 android:gravity="center">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="20sp"  
 android:text="Are you sure?"  
 android:textColor="@color/black"  
 android:textStyle="bold"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center">  
  
 <Button  
 android:id="@+id/dialog\_yes"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="50dp"  
 android:text="YES"  
 android:textStyle="bold"  
 android:layout\_marginTop="20dp"  
 android:textColor="@color/white"  
 android:background="@drawable/behind\_card"  
 android:layout\_margin="20dp"/>  
  
 <Button  
 android:id="@+id/dialog\_no"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="50dp"  
 android:text="CANCEL"  
 android:textStyle="bold"  
 android:layout\_marginTop="20dp"  
 android:textColor="@color/white"  
 android:background="@drawable/behind\_card"  
 android:layout\_margin="20dp"/>  
 </LinearLayout>  
  
</LinearLayout>

## **Code**

### **Chuẩn bị**

Import 4 thư viện trong build.gradle

implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
// Để gọi api php từ server  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
// convert json object sang java object và ngược lại  
implementation 'com.squareup.retrofit2:converter-scalars:2.9.0'  
// convert string của php sang string của java  
implementation 'com.squareup.picasso:picasso:2.8'  
// lấy ảnh từ url truyền vào imageview

Xin quyền internet và đọc bộ nhớ

<uses-permission android:name="android.permission.INTERNET"/>  
<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

Thêm 2 thuộc tính cho thẻ application

android:usesCleartextTraffic="true"   
android:requestLegacyExternalStorage="true"

### **API php**

**Kết nối database**

<?php

 /\*\*

  \*

  \*/

 class Database

 {

    private $servername = "localhost";

    private $username = "root";

    private $password = "";

    private $conn = null;

     function \_\_construct()

     {

     }

     function connect(){

         $this->conn = new mysqli($this->servername, $this->username, $this->password, 'test');

        if ($this->conn->connect\_error) {

          die("Connection failed: " . $this->connect\_error);

        }

        return $this->conn;

     }

 }

?>

**Lấy dữ liệu**

<?php

    include\_once('./DbConnect.php');

    class Furniture{

        function \_\_construct($id, $name, $detail, $price, $image){

            $this->id = $id;

            $this->name = $name;

            $this->detail = $detail;

            $this->price = $price;

            $this->image = $image;

        }

    }

    $Database = new Database();

    $conn = $Database->connect();

    $array = array();

    $stmt = $conn->prepare("SELECT \* FROM furniture");

    $stmt->execute();

    $stmt->bind\_result($id, $name, $detail, $price, $image);

    while($stmt->fetch()){

        array\_push($array, new Furniture($id, $name, $detail, $price, $image));

    }

    echo json\_encode($array);

?>

**Thêm dữ liệu**

<?php

    include\_once('./DbConnect.php');

    if(isset($\_POST['name']) || isset($\_POST['detail']) || isset($\_POST['price']) || isset($\_POST['image'])){

        $Database = new Database();

        $conn = $Database->connect();

        $stmt = $conn->prepare('INSERT INTO furniture (name, detail, price, image) VALUES(?,?,?,?)');

        $stmt->bind\_param("ssss", $\_POST['name'],$\_POST['detail'],$\_POST['price'],$\_POST['image']);

        if($stmt->execute()){

            echo 'Insert Sucess';

        }else{

            echo 'Inser Failed' + $stmt->error;

        }

    }

?>

**Sửa dữ liệu**

<?php

    include\_once('./DbConnect.php');

    if(isset($\_POST['name']) || isset($\_POST['detail']) || isset($\_POST['price']) || isset($\_POST['image'])){

        $Database = new Database();

        $conn = $Database->connect();

        $stmt = $conn->prepare('UPDATE furniture set name = ?, detail = ?, price = ?, image = ? WHERE ID = ?');

        $stmt->bind\_param("sssss", $\_POST['name'],$\_POST['detail'],$\_POST['price'],$\_POST['image'], $\_POST['id']);

        if($stmt->execute()){

            echo 'Update Sucess';

        }else{

            echo 'Update Failed' + $stmt->error;

        }

    }

?>

**Xóa dữ liệu**

<?php

    include\_once('./DbConnect.php');

    if(isset($\_POST['id'])){

        $Database = new Database();

        $conn = $Database->connect();

        $stmt = $conn->prepare('DELETE from furniture WHERE ID = ?');

        $stmt->bind\_param("s", $\_POST['id']);

        if($stmt->execute()){

            echo 'Delete Sucess';

        }else{

            echo 'Delete Failed' + $stmt->error;

        }

    }

?>

**Upload hình ảnh lên server**

<?php

    $file\_path = "img/";

    $file\_path = $file\_path.basename($\_FILES['uploaded\_file']['name']);

    if (move\_uploaded\_file($\_FILES['uploaded\_file']['tmp\_name'], $file\_path)) {

        echo $\_FILES['uploaded\_file']['name'];

    }else{

        echo "Failed";

    }

 ?>

### **Code java**

#### Cấu hình retrofit và các phương thức gọi api

public interface Retrofit {  
 String *url* = "http://10.0.2.2/MyApp/";  
  
 Gson *gson* = new GsonBuilder().setLenient().create();  
  
  
 Retrofit *retrofit* = new retrofit2.Retrofit.Builder()  
 .baseUrl(*url*)  
 .addConverterFactory(ScalarsConverterFactory.*create*())  
 .addConverterFactory(GsonConverterFactory.*create*(*gson*))  
 .build()  
 .create(Retrofit.class);  
  
  
 @GET("GetFurniture.php")  
 Call<List<Furniture>> getFurnitures();  
  
 @FormUrlEncoded  
 @POST("InsertFurniture.php")  
 Call<String> insertFuniture(@Field("name") String name,  
 @Field("detail") String detail,  
 @Field("price") Float price,  
 @Field("image") String image);  
  
 @Multipart  
 @POST("UploadImage.php")  
 Call<String> uploadImage(@Part MultipartBody.Part photo);  
  
 @FormUrlEncoded  
 @POST("UpdateFurniture.php")  
 Call<String> updateFuniture(@Field("name") String name,  
 @Field("detail") String detail,  
 @Field("price") Float price,  
 @Field("image") String image,  
 @Field("id") int id);  
  
 @FormUrlEncoded  
 @POST("DeleteFurniture.php")  
 Call<String> deleteFuniture(@Field("id") int id);  
}

#### Lớp Furniture

public class Furniture {  
  
@SerializedName("id")  
@Expose  
private Integer id;  
@SerializedName("name")  
@Expose  
private String name;  
@SerializedName("detail")  
@Expose  
private String detail;  
@SerializedName("price")  
@Expose  
private Integer price;  
@SerializedName("image")  
@Expose  
private String image;  
  
public Integer getId() {  
return id;  
}  
  
public void setId(Integer id) {  
this.id = id;  
}  
  
public String getName() {  
return name;  
}  
  
public void setName(String name) {  
this.name = name;  
}  
  
public String getDetail() {  
return detail;  
}  
  
public void setDetail(String detail) {  
this.detail = detail;  
}  
  
public Integer getPrice() {  
return price;  
}  
  
public void setPrice(Integer price) {  
this.price = price;  
}  
  
public String getImage() {  
return image;  
}  
  
public void setImage(String image) {  
this.image = image;  
}  
  
  
  
}

#### FurnitureAdapter

public class FurnitureAdapter extends ArrayAdapter<Furniture> {  
  
 MainActivity context;  
 int resource;  
  
 public FurnitureAdapter(MainActivity context, int resource) {  
 super(context, resource);  
 this.context = context;  
 this.resource = resource;  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 View view = LayoutInflater.*from*(context).inflate(resource,null);  
  
 ImageView imageView = view.findViewById(R.id.*img*);  
 TextView name = view.findViewById(R.id.*name*);  
 TextView detail = view.findViewById(R.id.*detail*);  
 TextView price = view.findViewById(R.id.*price*);  
 ConstraintLayout layoutFront = view.findViewById(R.id.*cardfront*);  
 ConstraintLayout layoutBehind = view.findViewById(R.id.*cardbehind*);  
 ImageView close = view.findViewById(R.id.*close*);  
 ImageView edit = view.findViewById(R.id.*edit*);  
 ImageView delete = view.findViewById(R.id.*delete*);  
  
  
 Furniture f = getItem(position);  
  
 if(!f.getImage().equals("")){  
 Picasso.*get*().load(f.getImage()).into(imageView);  
 }  
 name.setText(f.getName());  
 detail.setText(f.getDetail());  
 price.setText("$"+f.getPrice());  
  
 view.setOnLongClickListener(v->{  
 layoutFront.animate().translationX(-44);  
 layoutBehind.animate().translationX(44);  
 return false;  
 });  
  
 close.setOnClickListener(v->{  
 layoutFront.animate().translationX(0);  
 layoutBehind.animate().translationX(0);  
 });  
  
 edit.setOnClickListener(v->{  
 layoutFront.animate().translationX(0);  
 layoutBehind.animate().translationX(0);  
 context.OpenDiaLog(false, f);  
 });  
  
 delete.setOnClickListener(v->{  
 context.OpenDialogDelete(f.getId());  
 });  
  
 return view;  
 }  
}

#### MainActivity

package com.example.phpmysql;  
  
import androidx.activity.result.ActivityResult;  
import androidx.activity.result.ActivityResultCallback;  
import androidx.activity.result.ActivityResultLauncher;  
import androidx.activity.result.contract.ActivityResultContract;  
import androidx.activity.result.contract.ActivityResultContracts;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.cardview.widget.CardView;  
import androidx.core.app.ActivityCompat;  
  
import android.Manifest;  
import android.app.Activity;  
import android.app.Dialog;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.CountDownTimer;  
import android.provider.MediaStore;  
import android.view.Gravity;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.Window;  
import android.view.WindowManager;  
import android.view.animation.AnimationUtils;  
import android.widget.AbsListView;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.squareup.picasso.Picasso;  
  
import java.io.File;  
import java.io.InputStream;  
import java.util.ArrayList;  
import java.util.List;  
  
import okhttp3.MediaType;  
import okhttp3.MultipartBody;  
import okhttp3.RequestBody;  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
public class MainActivity extends AppCompatActivity {  
  
 ListView listView;  
 CardView cardView;  
 ImageView image;  
 Dialog dialog;  
 FurnitureAdapter furnitureAdapter;  
 int REQUEST\_CODE\_PMFOLDER = 123;  
 String realpath;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 listView = findViewById(R.id.*listview*);  
 cardView = findViewById(R.id.*add*);  
  
 furnitureAdapter = new FurnitureAdapter(this, R.layout.*card*);  
  
 LoadData();  
  
 listView.setAdapter(furnitureAdapter);  
  
 cardView.setOnClickListener(v ->{  
 OpenDiaLog(true, null);  
 });  
 }  
  
 private void LoadData() {  
 furnitureAdapter.clear();  
 Call<List<Furniture>> call = Retrofit.*retrofit*.getFurnitures();  
  
 call.enqueue(new Callback<List<Furniture>>() {  
 @Override  
 public void onResponse(Call<List<Furniture>> call, Response<List<Furniture>> response) {  
 furnitureAdapter.addAll(response.body());  
 }  
  
 @Override  
 public void onFailure(Call<List<Furniture>> call, Throwable t) {  
 Toast.*makeText*(MainActivity.this, t.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 furnitureAdapter.notifyDataSetChanged();  
 }  
  
  
 public void OpenDiaLog(Boolean isAdd, Furniture f) {  
 realpath = "";  
  
 CreateDialog(R.layout.*dialog*);  
  
 Button cancel = dialog.findViewById(R.id.*dialog\_cancel*);  
 TextView typeDialog = dialog.findViewById(R.id.*type\_dialog*);  
 Button add = dialog.findViewById(R.id.*dialog\_add*);  
 image = dialog.findViewById(R.id.*dialog\_img*);  
 EditText edtname = dialog.findViewById(R.id.*dialog\_name*);  
 EditText edtdetail = dialog.findViewById(R.id.*dialog\_detail*);  
 EditText edtprice = dialog.findViewById(R.id.*dialog\_price*);  
  
 cancel.setOnClickListener(v->{  
 CloseDialog();  
 });  
  
 if(!isAdd){  
 typeDialog.setText("Update Furniture");  
 add.setText("Update");  
 }  
  
 if(f != null){  
 edtname.setText(f.getName());  
 Picasso.*get*().load(f.getImage()).into(image);  
 edtdetail.setText(f.getDetail());  
 edtprice.setText(f.getPrice().toString());  
 }  
  
 add.setOnClickListener(v->{  
 String name = edtname.getText().toString();  
 String detail = edtdetail.getText().toString();  
 String price = edtprice.getText().toString();  
 if(!isAdd){  
 if(realpath.equals("")){  
 UpdateFurniture(name, detail, Float.*parseFloat*((price)),f.getImage(), f.getId());  
 }else{  
 UpdateFurniture(name, detail, Float.*parseFloat*((price)),UploadImage(), f.getId());  
 }  
 }else{  
 if(name.trim().equals("")){  
 edtname.setError("Please input name");  
 }else if(detail.trim().equals("")){  
 edtdetail.setError("Please input detail");  
 }else if(price.trim().equals(""))  
 {  
 edtprice.setError("Please input price");  
 }else{  
 AddFurniture(name, detail, Float.*parseFloat*((price)));  
 }  
 }  
 });  
  
 image.setOnClickListener(v->{  
 ActivityCompat.*requestPermissions*(MainActivity.this,  
 new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*},  
 REQUEST\_CODE\_PMFOLDER);  
 });  
  
 dialog.show();  
 }  
  
 private String UploadImage() {  
 if (!realpath.equals("")){  
 File file = new File(realpath);  
 String filePath = file.getAbsolutePath();  
  
  
 String[] arrayFileName = filePath.split("\\.");  
 filePath = arrayFileName[0] + System.*currentTimeMillis*() + "." + arrayFileName[1];  
 String[] fileNameArray = filePath.split("/");  
 String filePathSv = Retrofit.*url*+ "img/" + fileNameArray[fileNameArray.length - 1];  
  
 RequestBody requestBody = RequestBody.*create*(MediaType.*parse*("multipart/form-data"),file);  
  
 MultipartBody.Part multipartBody = MultipartBody.Part.*createFormData*("uploaded\_file",filePath,requestBody);  
  
 Call<String> call = Retrofit.*retrofit*.uploadImage(multipartBody);  
 call.enqueue(new Callback<String>() {  
 @Override  
 public void onResponse(Call<String> call, Response<String> response) {  
  
 }  
  
 @Override  
 public void onFailure(Call<String> call, Throwable t) {  
 Toast.*makeText*(MainActivity.this, "Upload image fail: "+ t.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 return filePathSv;  
 }  
 return "";  
 }  
  
 public void CreateDialog(int layout){  
 dialog = new Dialog(this);  
 dialog.requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 dialog.setContentView(layout);  
 dialog.setCancelable(false);  
  
 final Window window = dialog.getWindow();  
 window.setLayout(AbsListView.LayoutParams.*MATCH\_PARENT*, AbsListView.LayoutParams.*WRAP\_CONTENT*);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
  
 WindowManager.LayoutParams wlp = window.getAttributes();  
 wlp.gravity = Gravity.*CENTER*;  
 window.addFlags(WindowManager.LayoutParams.*FLAG\_DIM\_BEHIND*);  
 window.setDimAmount(0.5f);  
 window.setAttributes(wlp);  
  
 ((ViewGroup)dialog.getWindow().getDecorView())  
 .getChildAt(0).startAnimation(AnimationUtils.*loadAnimation*(  
 this,android.R.anim.*slide\_in\_left*));  
 }  
  
  
  
 public void CloseDialog() {  
 ((ViewGroup)dialog.getWindow().getDecorView())  
 .getChildAt(0).startAnimation(AnimationUtils.*loadAnimation*(  
 MainActivity.this,android.R.anim.*slide\_out\_right*));  
  
 CountDownTimer countDownTimer = new CountDownTimer(400,400) {  
 @Override  
 public void onTick(long millisUntilFinished) {  
  
 }  
  
 @Override  
 public void onFinish() {  
 dialog.dismiss();  
 }  
 };  
 countDownTimer.start();  
 }  
  
  
 private void AddFurniture(String name, String detail, Float price) {  
 Call<String> call = Retrofit.*retrofit*.insertFuniture(name, detail, price, UploadImage());  
 call.enqueue(new Callback<String>() {  
 @Override  
 public void onResponse(Call<String> call, Response<String> response) {  
 if(response.body().equals("Insert Sucess")){  
 Toast.*makeText*(MainActivity.this, "Add success", Toast.*LENGTH\_SHORT*).show();  
 CloseDialog();  
 LoadData();  
 }else{  
 Toast.*makeText*(MainActivity.this, "Add failed", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
  
 @Override  
 public void onFailure(Call<String> call, Throwable t) {  
 Toast.*makeText*(MainActivity.this, "Add failed: "+ t.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 }  
  
 private void UpdateFurniture(String name, String detail, Float price,String image, int id) {  
 Call<String> call = Retrofit.*retrofit*.updateFuniture(name,detail,price,image,id);  
 call.enqueue(new Callback<String>() {  
 @Override  
 public void onResponse(Call<String> call, Response<String> response) {  
 if(response.body().equals("Update Sucess")){  
 Toast.*makeText*(MainActivity.this, "Update success", Toast.*LENGTH\_SHORT*).show();  
 CloseDialog();  
 LoadData();  
 }else{  
 Toast.*makeText*(MainActivity.this, "Update failed", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onFailure(Call<String> call, Throwable t) {  
 Toast.*makeText*(MainActivity.this, "Update failed: "+ t.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 public void OpenDialogDelete(int id){  
 CreateDialog(R.layout.*dialog\_delete*);  
  
 Button yes = dialog.findViewById(R.id.*dialog\_yes*);  
 Button no = dialog.findViewById(R.id.*dialog\_no*);  
  
 yes.setOnClickListener(v->{  
 DeleteFurniture(id);  
 CloseDialog();  
 LoadData();  
 });  
  
 no.setOnClickListener(v->{  
 CloseDialog();  
 });  
  
 dialog.show();  
 }  
  
 public void DeleteFurniture(int id){  
 Call<String> call = Retrofit.*retrofit*.deleteFuniture(id);  
 call.enqueue(new Callback<String>() {  
 @Override  
 public void onResponse(Call<String> call, Response<String> response) {  
 if(response.body().equals("Delete Sucess")){  
 Toast.*makeText*(MainActivity.this, "Delete Success", Toast.*LENGTH\_SHORT*).show();  
 }else{  
 Toast.*makeText*(MainActivity.this, "Delete Failed", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onFailure(Call<String> call, Throwable t) {  
 Toast.*makeText*(MainActivity.this, "Delete Failed: "+ t.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 if (requestCode == REQUEST\_CODE\_PMFOLDER && permissions.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*){  
 Intent intent = new Intent(Intent.*ACTION\_PICK*);  
 intent.setType("image/\*");  
 someActivityResultLauncher.launch(intent);  
 }else{  
 Toast.*makeText*(getApplicationContext(),"Ban chua cap quyen chon hinh anh",Toast.*LENGTH\_LONG*).show();  
 }  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 }  
  
  
 ActivityResultLauncher<Intent> someActivityResultLauncher = registerForActivityResult(  
 new ActivityResultContracts.StartActivityForResult(),  
 new ActivityResultCallback<ActivityResult>() {  
 @Override  
 public void onActivityResult(ActivityResult result) {  
 if (result.getResultCode() == Activity.*RESULT\_OK*) {  
 // There are no request codes  
 Intent data = result.getData();  
 Uri uri = data.getData();  
 realpath = getRealPathFromURI(uri);  
  
 try {  
 InputStream inputStream = getContentResolver().openInputStream(uri);  
 Bitmap bitmap = BitmapFactory.*decodeStream*(inputStream);  
 image.setImageBitmap(bitmap);  
 }catch (Exception e){  
 e.printStackTrace();  
 }  
  
 }  
 }  
 });  
  
 public String getRealPathFromURI (Uri contentUri) {  
 String path = null;  
 String[] proj = { MediaStore.MediaColumns.*DATA* };  
 Cursor cursor = getContentResolver().query(contentUri, proj, null, null, null);  
 if (cursor.moveToFirst()) {  
 int column\_index = cursor.getColumnIndexOrThrow(MediaStore.MediaColumns.*DATA*);  
 path = cursor.getString(column\_index);  
 }  
 cursor.close();  
 return path;  
 }  
}