

# BÁO CÁO BÀI TẬP

Môn học: Bảo mật Web và Ứng dụng

Bài tập buổi 10:

Android với kết nối SSL/TLS

GV: Nghi Hoàng Khoa

Ngày báo cáo: 24/05/2023

**Nhóm: Pengu**

## 1. THÔNG TIN CHUNG:

Lớp: NT213.N21.ANTT

STT	Họ và tên	MSSV	Email
1	Phạm Phúc Đức	20520162	20520162@gm.uit.edu.vn
2	Nguyễn Hoàng Phúc	20520277	20520277@gm.uit.edu.vn
3	Nguyễn Đức Tấn	20520751	20520751@gm.uit.edu.vn
4	Nguyễn Nhật Hiếu Trung	20520830	20520830@gm.uit.edu.vn

## 2. NỘI DUNG THỰC HIỆN:<sup>1</sup>

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	<a href="#">Xây dựng 1 ứng dụng Android sử dụng kết nối SSL/TLS</a>	100%	Cả nhóm

<sup>1</sup> Ghi nội dung công việc, các kịch bản trong bài Thực hành

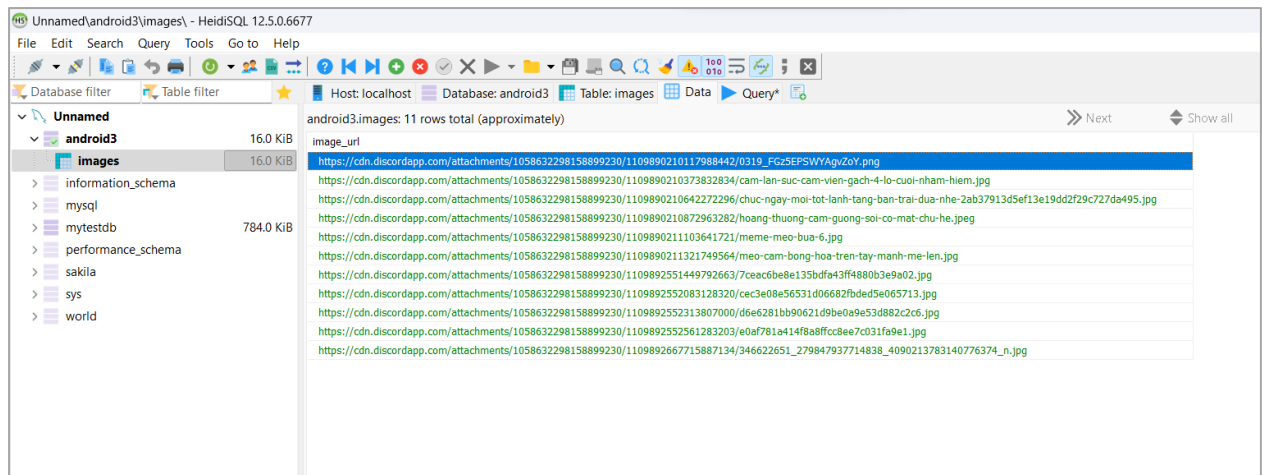


# BÁO CÁO CHI TIẾT

## 1. Xây dựng 1 ứng dụng Android sử dụng kết nối SSL/TLS

Ứng dụng xây dựng: App xem ảnh từ link lấy được trong MySQL database với server sử dụng Apache24

### 1.1. Tạo database có lưu các link ảnh:



Hình 1: Tạo database với các dữ liệu cần

Database gồm 1 bảng images và 1 cột image\_url.

### 1.2. Tạo file GetImage.php lấy dữ liệu từ MySQL:

```

1 <?php
2 $hostname = "10.0.139.42";
3 $username = "pengu";
4 $password = "123456";
5 $database = "android3";
6
7 try {
8     $conn = new PDO("mysql:host=$hostname;dbname=$database", $username, $password);
9     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10
11     // Thực hiện truy vấn cơ sở dữ liệu để lấy danh sách hình ảnh
12     $query = "SELECT image_url FROM images";
13     $stmt = $conn->prepare($query);
14     $stmt->execute();
15     $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
16
17     // Trả về danh sách liên kết hình ảnh dưới dạng JSON
18     echo json_encode($result);
19 } catch (PDOException $e) {
20     echo "Lỗi kết nối cơ sở dữ liệu: " . $e->getMessage();
21 }
22 ?>
23

```

Hình 2: Lấy dữ liệu từ database

Giải thích nội dung trong file .php:

Sau khi thực hiện query “**SELECT image\_url FROM images**”, dữ liệu sẽ được in ra dưới dạng JSON như dưới:

```

[{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109890210117988442/0319_FGz5EPSWYAgyZoY.png"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109890210373832834/cam-lan-suc-cam-vien-gach-4-lo-cuoi-nham-hiem.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109890210642272296/chuc-ngay-moi-tot-lanh-tang-ban-trai-dua-nhe-2ab37913d5ef13e19dd2f29c727da495.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109890210872963282/hoang-thuong-cam-guonng-soi-co-mat-chu-he.jpeg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109890211103641721/meme-meo-bua-6.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109890211321749564/meo-cam-bong-hoa-tren-tay-manh-me-len.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109892551449792663/7ceac6be8e135bdfa43ff4880b3e9a02.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109892552083128320/cec3e08e56531d06682fbded5e065713.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109892552313807000/d6e6281bb90621d9be0a9e53d882c2c6.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109892552561283203/e0af781a414f8a8ffcc8ee7c031fa9e1.jpg"},
{"image_url":"https://cdn.discordapp.com/attachments/1058632298158899230/1109892667715887134/346622651_279847937714838_4090213783140776374_u.jpg"}]

```

Hình 3: Dữ liệu in ra từ lệnh `echo json_encode($result);`

### 1.3. Lấy và xử lý dữ liệu

```

client.newCall(request).enqueue(new Callback() {
    @Override
    public void onResponse(Call call, Response response) throws IOException {
        if (response.isSuccessful()) {
            try {
                String jsonData = response.body().string();
                JSONArray imageUrls = new JSONArray(jsonData);
                for (int i = 0; i < imageUrls.length(); i++) {
                    JSONObject imageObject = imageUrls.getJSONObject(i);
                    String imageUrl = imageObject.getString("image_url");
                    try {
                        imageUrl = imageUrl.replace(target: "\\/", replacement: "/");
                        URI uri = new URI(imageUrl);
                        Uri imageUri = Uri.parse(uri.toString());
                        datalist.add(imageUri);
                    } catch (URISyntaxException e) {
                        e.printStackTrace();
                    }
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }
}

```

Hình 4: Xem thông tin chữ kí của các file apk

Việc xử lý dữ liệu sẽ diễn ra như sau:

- Các dữ liệu nhận được sẽ trở thành 1 chuỗi với lệnh **response.body().string()** và sau đó trở thành danh sách với lệnh **JSONArray(jsonData)**.
- Tiếp đến vòng lặp for sẽ duyệt qua từng phần tử để lấy liên kết hình ảnh nằm trong đó, **imageObject.getString("image\_url")** sẽ lấy các giá trị của trường "image\_url" và dữ liệu này sẽ được lọc bỏ các dấu \ do Trong JSON, khi một chuỗi chứa các ký tự đặc biệt như / xuất hiện, thì chuỗi sẽ được trình bày dưới dạng \\ giúp cho việc phân tích cú pháp JSON không bị sai.
- Cuối cùng các liên kết hoàn chỉnh sẽ được thêm vào datalist và đây chính là danh sách các liên kết hình ảnh chúng ta sẽ mở tại MainActivity

Tiếp đến, cài đặt 1 số thứ cần thiết:

- **permission:** chúng ta cần phân quyền truy cập Internet cho ứng dụng, mặc định khi được thêm vào, ứng dụng sẽ có quyền truy cập internet mà không cần phải xác nhận từ người dùng:

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

- **Dependency:** Picasso giúp cho việc hiển thị hình ảnh từ các liên kết còn Bouncy Castle dùng cho việc xác thực chữ ký

```
implementation 'com.squareup.picasso:picasso:2.71828'
```

```
implementation 'org.bouncycastle:bcprov-jdk15on:1.68'
```

**Bước 1:** Tạo chứng chỉ tự ký với openssl và BKS keystore cho chứng chỉ tự ký để dùng trong ứng dụng Android với keytool:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Program Files\OpenSSL-Win64\bin> openssl genkey -algorithm RSA -out D:\keystore\private.key
.....+++++
.....+++++
PS C:\Program Files\OpenSSL-Win64\bin> openssl req -new -key private.key -out D:\keystore\csr.pem
Can't open private.key for reading, No such file or directory
34359836736:error:02001002:system library:fopen:No such file or directory:crypto/bio/bss_file.c:69:fopen('private.key','r')
34359836736:error:2006D080:BIIO routines:BIIO_new_file:no such file:crypto/bio/bss_file.c:76:
unable to load Private Key
PS C:\Program Files\OpenSSL-Win64\bin> openssl req -new -key D:\keystore\private.key -out D:\keystore\csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:VN
State or Province Name (full name) [Some-State]:HoChiMinh
Locality Name (eg, city) []:TD
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Pengu.Có
Organizational Unit Name (eg, section) []:PenguClown
Common Name (e.g. server FQDN or YOUR name) []:10.0.139.42
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Hình 5: Tạo các file .key và .pem

```
PS C:\Program Files\OpenSSL-Win64\bin> openssl req -new -x509 -sha256 -key D:\keystore\private.key -in D:\keystore\csr.pem -out D:\keystore\ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:VN
State or Province Name (full name) [Some-State]:HoChiMinh
Locality Name (eg, city) []:ThuDuc
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Pengu.IT
Organizational Unit Name (eg, section) []:Penguuu
Common Name (e.g. server FQDN or YOUR name) []:10.0.139.42
Email Address []:
PS C:\Program Files\OpenSSL-Win64\bin>
```

Hình 6: Tạo file .crt từ 2 file trên

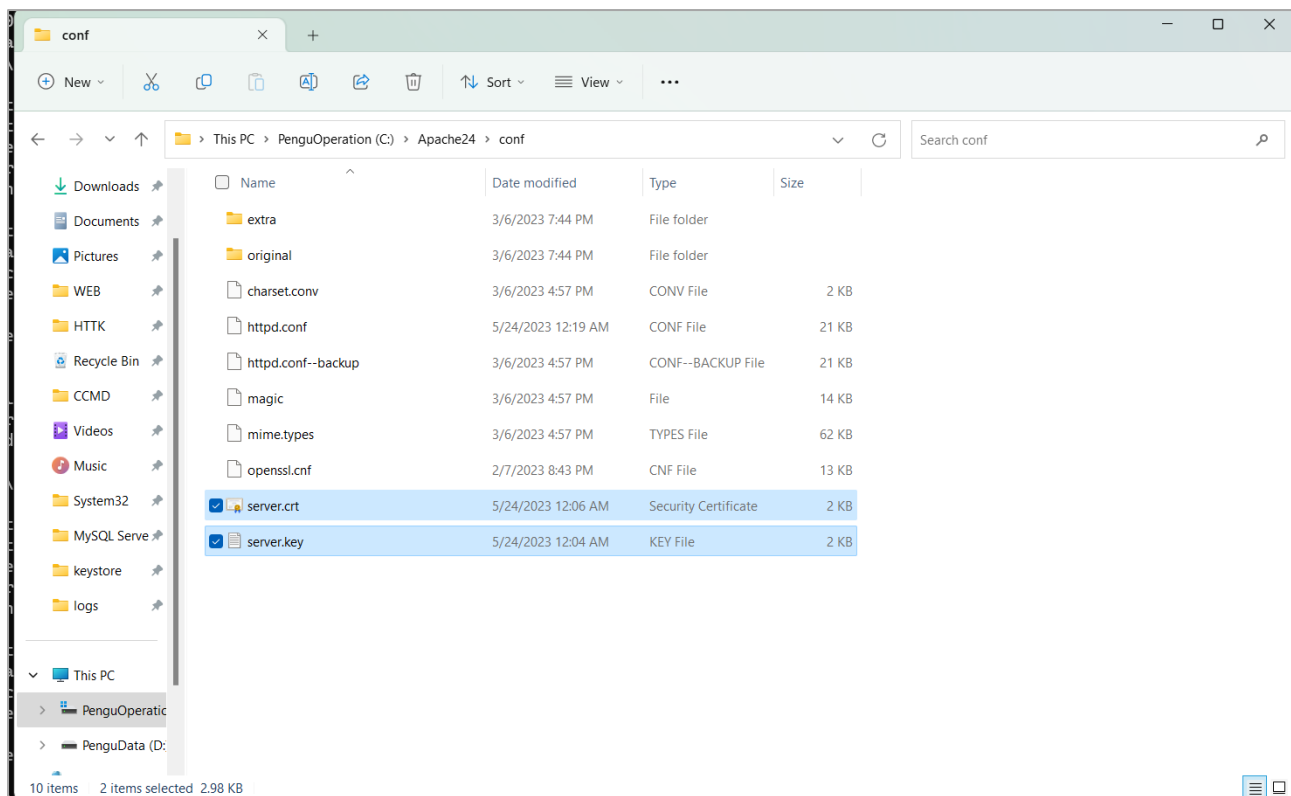
**Bước 2:** Tạo file bks với mật khẩu là penguwu và đưa thẳng vào trong tệp res/raw của thư mục project:

```
PS D:\keystore> keytool -import -v -trustcacerts -alias penguincertificate -file ca.crt -keystore C:\Users\Pengu\Desktop\Android2\viewMeme\app\src\main\res\raw\selfsignedcerts\penguincertificate.bks -provider org.bouncycastle.jce.provider.BouncyCastleProvider -providerpath "C:\Users\Pengu\Desktop\Android2\bcprov-jdk18on-173.jar" -storepass penguwu
Owner: CN=10.0.139.42, OU=Penguwu, O=Pengu.IT, L=ThuDuc, ST=HoChiMinh, C=VN
Issuer: CN=10.0.139.42, OU=Penguwu, O=Pengu.IT, L=ThuDuc, ST=HoChiMinh, C=VN
Serial number: 5105249f643b5c234e02dc4154cf22984e2bcddb
Valid from: Wed May 24 00:06:35 ICT 2023 until: Fri Jun 23 00:06:35 ICT 2023
Certificate fingerprints:
  SHA1: 12:D3:6A:2F:48:01:E0:C4:E7:05:3A:F1:FE:04:E9:EC:08:0C:C5:E4
  SHA256: F3:34:1A:3C:1D:E5:64:BB:F5:97:5F:54:98:4E:AF:BB:CC:9D:74:66:32:7B:D5:7A:80:D9:B6:98:6B:AD:99
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 6D B9 7C FA F3 DD 92 A1 47 F4 7C C5 65 35 E4 DF m.....G...e5..
0010: E7 0F 48 B3 ..H.
]
]
#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
CA:true
PathLen:2147483647
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 6D B9 7C FA F3 DD 92 A1 47 F4 7C C5 65 35 E4 DF m.....G...e5..
0010: E7 0F 48 B3 ..H.
]
]
Trust this certificate? [no]: y
```

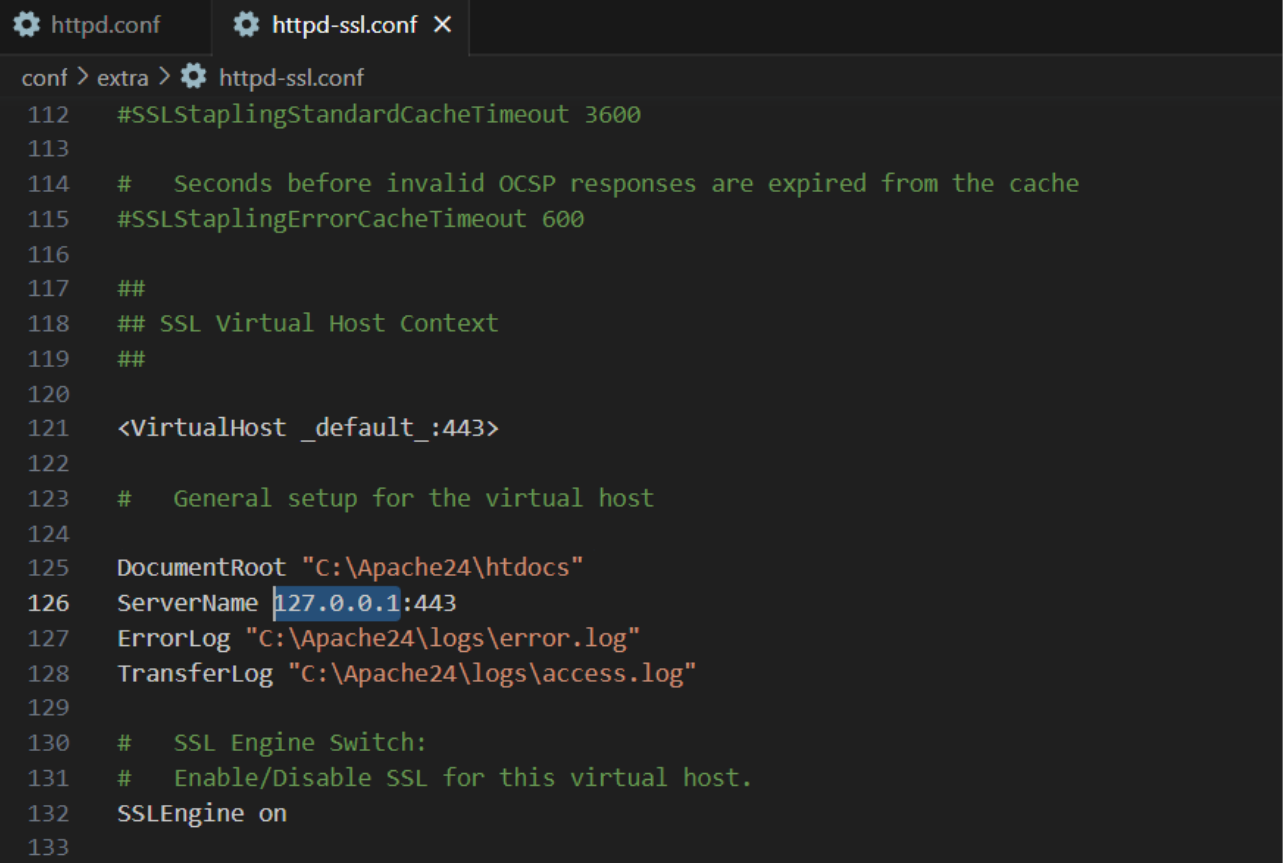
Hình 7: Sử dụng keytool tạo file bks

**Bước 3:** Đưa 2 file .crt và .pem vào thư mục C:/Apache24/conf và đều đổi tên thành server để bớt được 1 bước sửa đường dẫn tới chứng chỉ trong file http-ssl.conf:



Hình 8: Thay đổi tên và vị trí của 2 tệp đã tạo





```

conf > extra > httpd-ssl.conf
112 #SSLStaplingStandardCacheTimeout 3600
113
114 #   Seconds before invalid OCSP responses are expired from the cache
115 #SSLStaplingErrorCacheTimeout 600
116
117 ##
118 ## SSL Virtual Host Context
119 ##
120
121 <VirtualHost _default_:443>
122
123 #   General setup for the virtual host
124
125 DocumentRoot "C:\Apache24\htdocs"
126 ServerName 127.0.0.1:443
127 ErrorLog "C:\Apache24\logs\error.log"
128 TransferLog "C:\Apache24\logs\access.log"
129
130 #   SSL Engine Switch:
131 #   Enable/Disable SSL for this virtual host.
132 SSLEngine on
133

```

Hình 9: Thay đổi servername thành 127.0.0.1

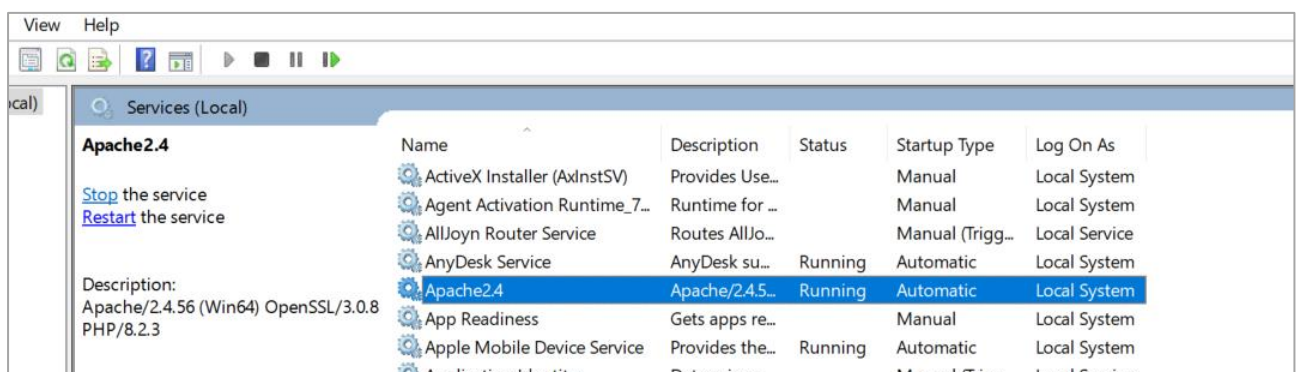
**Bước 4:** Sau đó uncomment dòng dưới cùng với các module cần thiết trong file httpd.conf:

```

conf > httpd.conf
514 # Virtual hosts
515 #Include conf/extra/httpd-vhosts.conf
516
517 # Local access to the Apache HTTP Server Manual
518 #Include conf/extra/httpd-manual.conf
519
520 # Distributed authoring and versioning (WebDAV)
521 #Include conf/extra/httpd-dav.conf
522
523 # Various default settings
524 #Include conf/extra/httpd-default.conf
525
526 # Configure mod_proxy_html to understand HTML4/XHTML1
527 <IfModule proxy_html_module>
528 Include conf/extra/proxy-html.conf
529 </IfModule>
530
531 # Secure (SSL/TLS) connections
532 Include conf/extra/httpd-ssl.conf
533 #
534 # Note: The following must must be present to support
535 #       starting without SSL on platforms with no /dev/random equivalent
536 #       but a statically compiled-in mod_ssl.
537 #
538 <IfModule ssl_module>
539 SSLRandomSeed startup builtin
540 SSLRandomSeed connect builtin
541 </IfModule>
542 #PHP8 Configuration
543 LoadModule php_module "c:/php82/php8apache2_4.dll"
544 AddHandler application/x-httpd-php .php
545 AddType application/x-httpd-php .php
546 PHPIniDir "c:/php82"
    
```

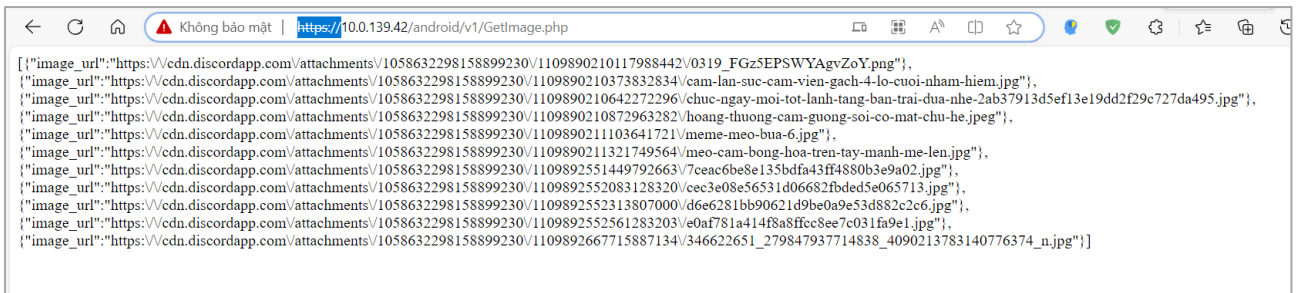
Hình 10: Uncomment các cài đặt cần thiết

Restart Apache trong service để lưu cài đặt:



Hình 11: Khởi động lại Apache

Có vẻ đã thành công, hmmm:

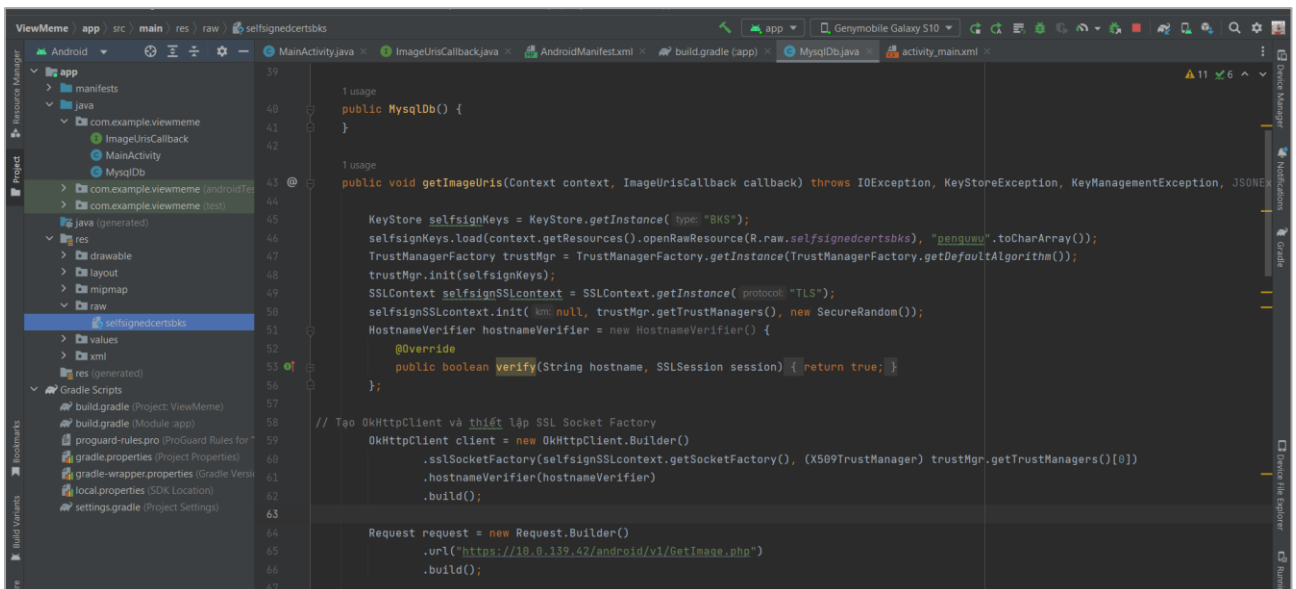


Hình 12: Thử truy cập vào đường link ban đầu với https://

**Bước 5:** Hàm đọc hình ảnh theo vị trí trong mảng sử dụng Picasso đã thêm trong dependency ở trên:

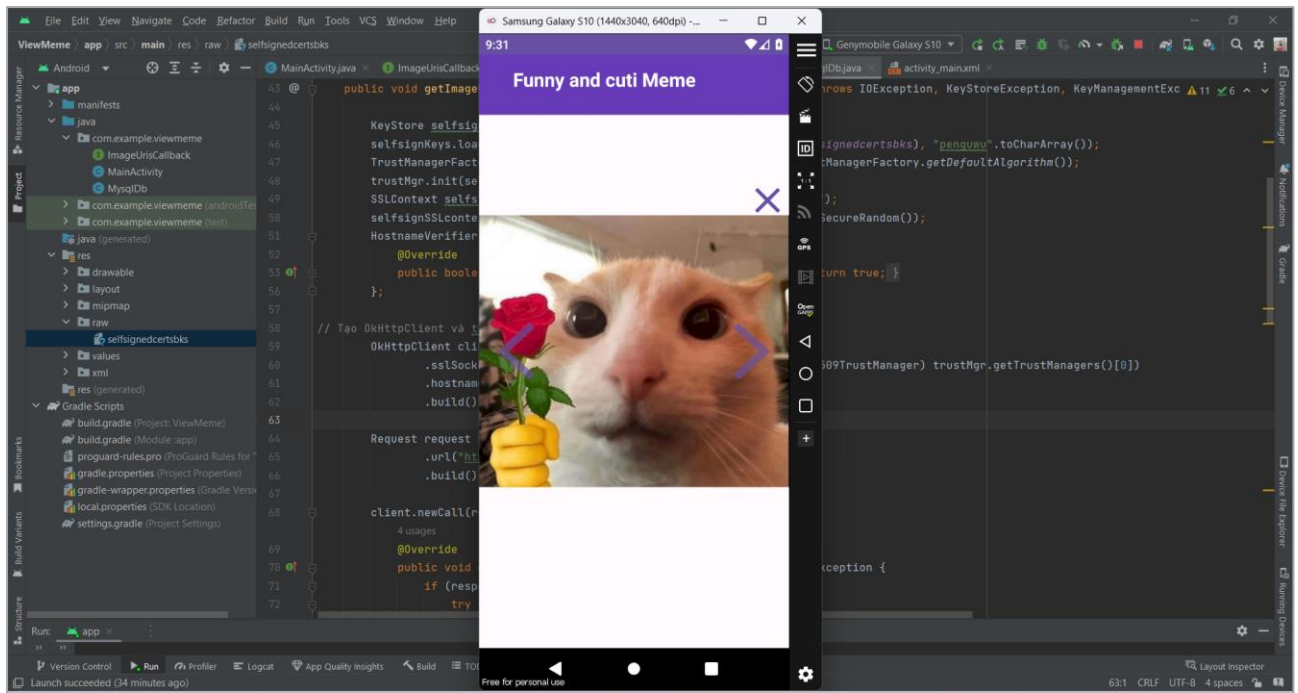


Hình 13: Hàm giúp hiển thị hình ảnh ra ImageView



Hình 14: Sử dụng chứng chỉ đã thêm trong kết nối SSL/TLS

Kết quả khi chạy:



Hình 15: Ứng dụng chạy thành công

Video demo:

<https://youtu.be/N8SYi0AMq2U>