

activity_form.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"
    android:orientation="vertical"
    android:layout_margin="20sp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:orientation="vertical"
        android:layout_height="match_parent">

        <EditText
            android:id="@+id/t1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter Name"
            android:inputType="textPersonName"
            android:minHeight="48dp" />

        <EditText
            android:id="@+id/t2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter Email"
            android:inputType="textEmailAddress"
            android:minHeight="48dp" />

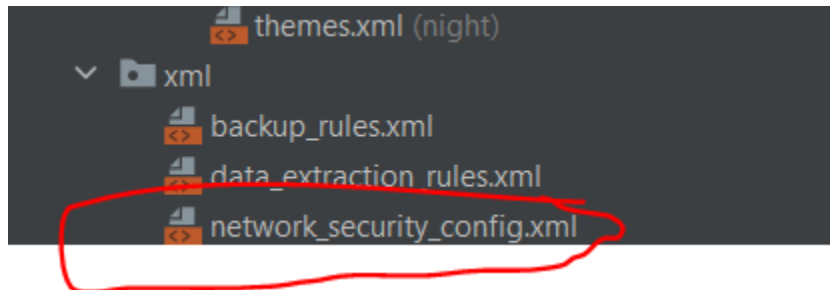
        <EditText
            android:id="@+id/t3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter Password"
            android:inputType="textPassword"
            android:minHeight="48dp" />

        <Button
            android:id="@+id/SignUpBtn"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sign up"/>
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```



network_security_config.xml

```

<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">https://10.0.2.2</domain>
    </domain-config>
</network-security-config>

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.prac10">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:networkSecurityConfig="@xml/network_security_config"
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="prac10"

```

Refer above pic for adding the code

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

```
android:networkSecurityConfig="@xml/network_security_config"
```

```
android:usesCleartextTraffic="true"
```

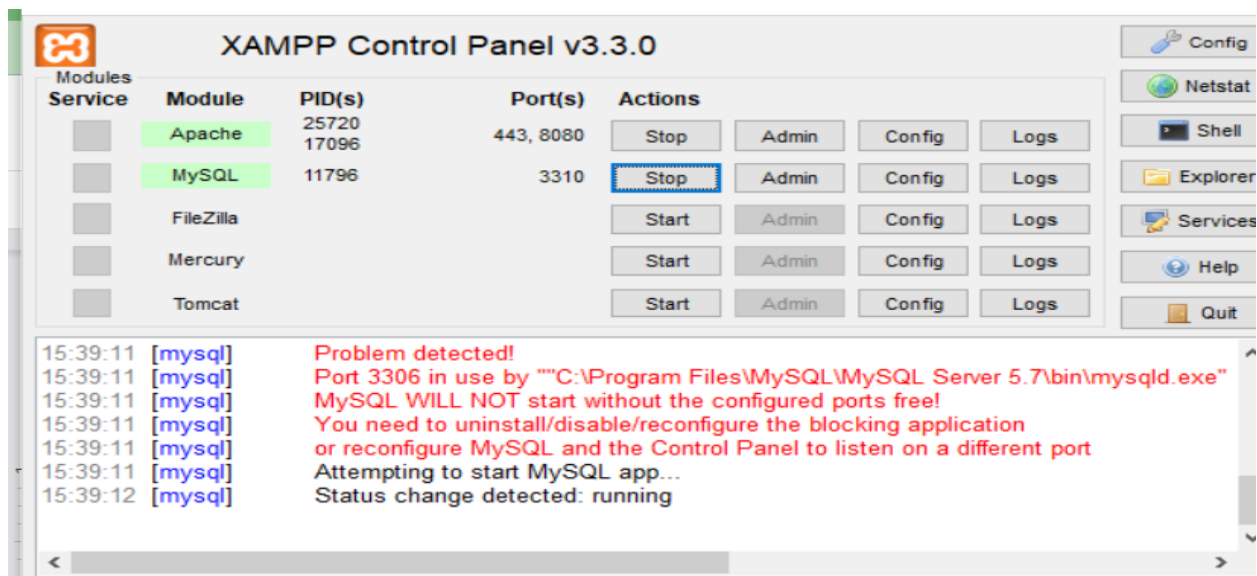
build.gradle(:app)

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
```

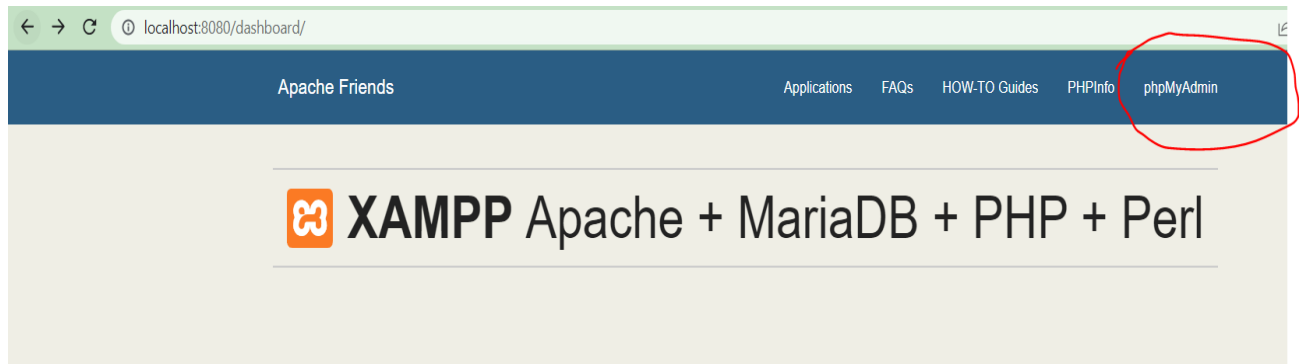
```
implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
```

```
dependencies {  
  
    implementation 'androidx.appcompat:appcompat:1.5.1'  
    implementation 'com.google.android.material:material:1.7.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'  
    implementation 'com.android.volley:volley:1.2.1'  
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
    implementation 'com.squareup.retrofit2:converter-gson:2.5.0'  
}
```

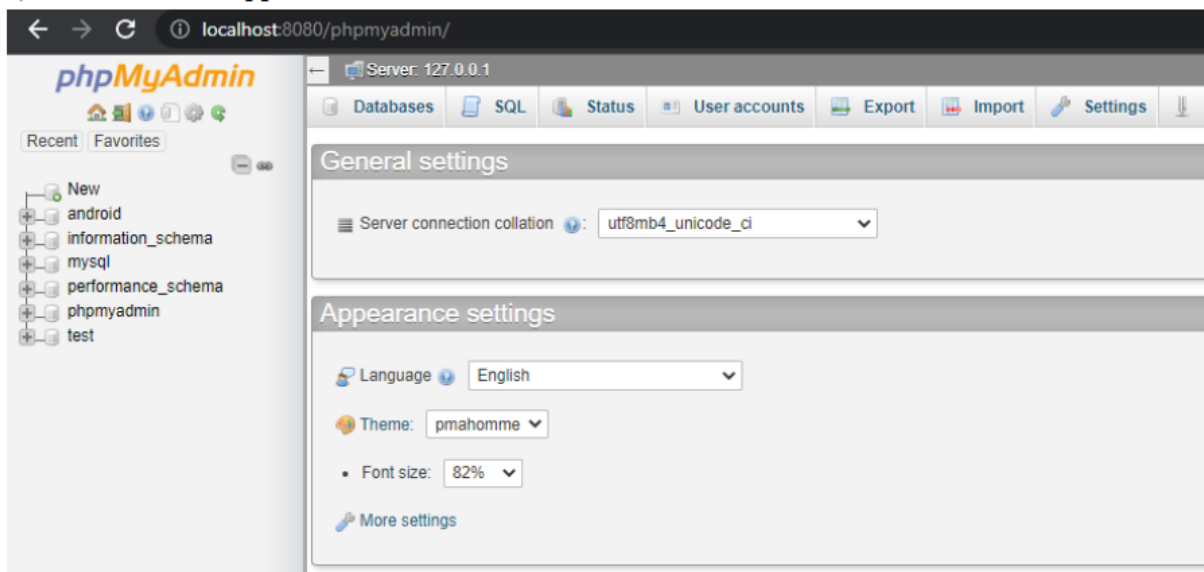
Xampp should be there and turned on



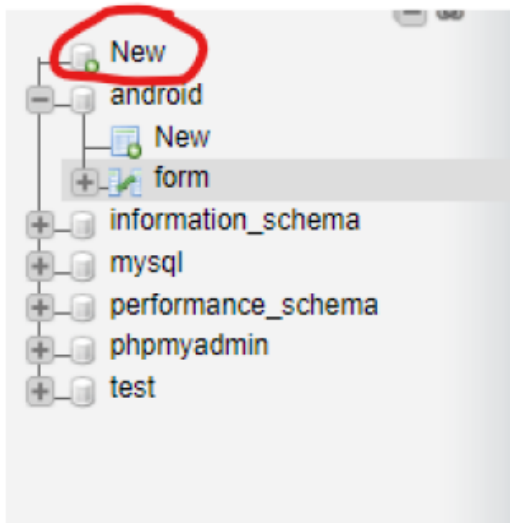
Goto localhost:8080 in your Browser→Click on phpmyadmin



3) Below window appears



4) Click on New



5) Enter database name and click Create

Databases

Create database ⓘ

android Collation ▼ Create

6) Enter Name and enter number of the columns and click Go

Create table ⓘ

Name: Number of columns:

Go

7) Enter details as shown below


The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a tree view of databases: New, android, information_schema, mysql, performance_schema, phpmyadmin, and test. The main area displays the 'form' table structure. At the top, there's a navigation bar with tabs: Browse, Structure, SQL, Search, Insert, Export, Import, and P. Below this, the table name 'form' is entered, followed by 'Add 1 column(s)' and a 'Go' button. The table structure is shown in a table with columns: Name, Type, Length/Values, Default, and Collation. Three columns are defined: 'name' (VARCHAR, 50, None), 'email' (VARCHAR, 50, None), and 'password' (VARCHAR, 50, None). Below the table structure, there are fields for 'Table comments:', 'Collation:', and 'Storage engine:'. At the bottom, there's a 'PARTITION definition:' link.

8) Click on create below image window is shown

The screenshot shows the 'Table structure' window in phpMyAdmin. At the top, there's a navigation bar with tabs: Browse, Structure, SQL, Search, Insert, and Export. Below this, there are two buttons: 'Table structure' and 'Relation view'. The main area displays a table with columns: #, Name, Type, Collation, Attributes, Null, Default, and Comment. Three columns are listed: 1 name varchar(50) latin1_swedish_ci No None, 2 email varchar(50) latin1_swedish_ci No None, and 3 password varchar(50) latin1_swedish_ci No None. Below the table, there's a toolbar with buttons: Check all, With selected: Browse, Change, Drop, and Pr. At the bottom, there's a 'Print' button, a 'Propose table structure' button, a 'Track table' button, a 'Move columns' button, and an 'Im' button. At the very bottom, there's a 'Indexes' button.

Goto xampp folder→htdocs→Create Folder PHP→Signup.php

> This PC > Local Disk (C:) > xampp > htdocs > PHP

Name	Date modified	Type	Size
 Signup	17-11-2022 19:32	PHP Source File	1 KB

Write code Signup.php in notepad and save it with .php extension

Signup.php

```
<?php
$conn = mysqli_connect("localhost","root","","android");
$name=trim($_POST['name']);
$email=trim($_POST['email']);
$password=trim($_POST['password']);
if($conn)
{
    $qry = "select * from form where email='$email'";
    $result = mysqli_query($conn,$qry);
    if(mysqli_num_rows($result)>0)
    {
        $response['message']="exist";
        echo json_encode($response);
    }
    else
    {
        $qry = "insert into form(name,email,password)values('$name','$email','$password')";
        if(mysqli_query($conn,$qry))
        {
            $response['message']="success";
        }
        else{
            $response['message']="failed";
        }
        echo json_encode($response);
    }
}
else{
    $response['message']="failed";
    echo json_encode($response);
}
```

```
}  
?>
```

FormActivity.java

```
package com.example.prac10;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
public class FormActivity extends AppCompatActivity {  
    EditText t1,t2,t3;  
    Button SignUpBtn;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_form);  
        t1=(EditText)findViewById(R.id.t1);  
        t2=(EditText)findViewById(R.id.t2);  
        t3=(EditText)findViewById(R.id.t3);  
        SignUpBtn=(Button)findViewById(R.id.SignUpBtn);  
        SignUpBtn.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                processSignUp();  
            }  
        });  
    }  
    void processSignUp(){  
        String name = t1.getText().toString();  
        String email = t2.getText().toString();  
        String password = t3.getText().toString();
```



```

public ResponseModel(String message) {
    this.message = message;
}
public ResponseModel() {}
public String getMessage() {
    return message;
}
public void setMessage(String message) {
    this.message = message;
}
}

```

Controller.java

```

package com.example.prac10;

```

```

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import javax.security.cert.CertificateException;

```

```

import okhttp3.OkHttpClient;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
public class Controller {

```

```

    private static final String url="https://10.0.2.2/PHP/Signup.php/";
    private static Controller clientobject;
    private static Retrofit retrofit;
    Controller() {
        retrofit=new Retrofit.Builder()

```

```

        .baseUrl(url).addConverterFactory(GsonConverterFactory.create()).client(getUnsafeOkHttpClient()).build();
    }
    public static synchronized Controller getInstance() {
        if(clientobject==null)
            clientobject=new Controller();
    }
}

```

```

        return clientobject;
    }
    Apiset getapi() {
        return retrofit.create(Apiset.class);
    }

    private static OkHttpClient getUnsafeOkHttpClient() {
        try {
            // Create a trust manager that does not validate certificate chains
            final TrustManager[] trustAllCerts = new TrustManager[] {
                new X509TrustManager() {
                    @Override
                    public void checkClientTrusted(java.security.cert.X509Certificate[] chain, String
authType) {
                        }

                    @Override
                    public void checkServerTrusted(java.security.cert.X509Certificate[] chain, String
authType) {
                        }

                    @Override
                    public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                        return new java.security.cert.X509Certificate[] {};
                    }
                }
            };

            // Install the all-trusting trust manager
            final SSLContext sslContext = SSLContext.getInstance("SSL");
            sslContext.init(null, trustAllCerts, new java.security.SecureRandom());
            // Create an ssl socket factory with our all-trusting manager
            final SSLSocketFactory sslSocketFactory = sslContext.getSocketFactory();

            OkHttpClient.Builder builder = new OkHttpClient.Builder();
            builder.sslSocketFactory(sslSocketFactory);
            builder.hostnameVerifier(new HostnameVerifier() {
                @Override
                public boolean verify(String hostname, SSLSession session) {
                    return true;
                }
            });

```

```

        }
    });

    OkHttpClient okHttpClient = builder.build();
    return okHttpClient;
} catch (Exception e) {
    throw new RuntimeException(e);
}
}
}
}

```

ApiSet.java

```

package com.example.prac10;

import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.POST;

public interface ApiSet {
    @FormUrlEncoded
    @POST("/PHP/Signup.php")
    Call<ResponseModel> verifyuser(
        @Field("name") String name,
        @Field("email") String email,
        @Field("password") String password
    );
}

```