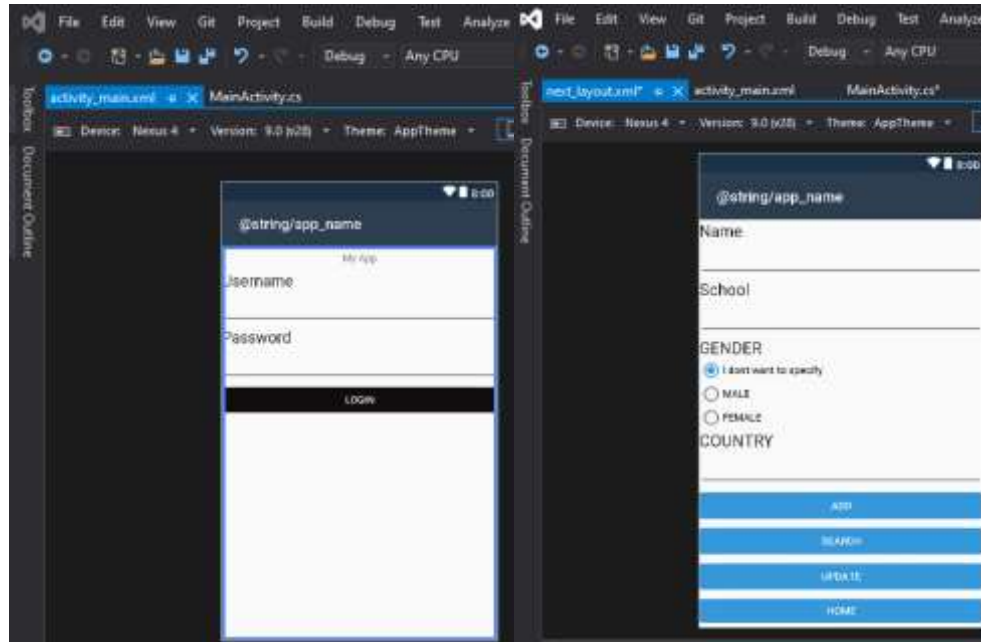
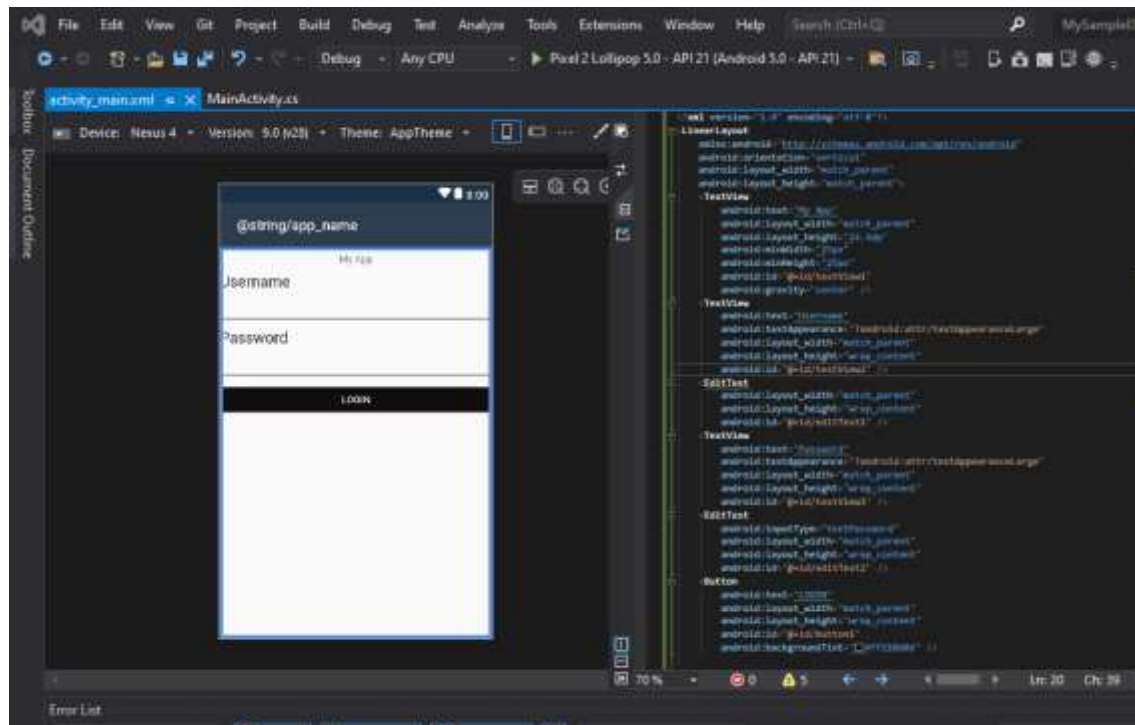


This handout will show the basic setup and coding explanation for our REST Approach in Xamarin. Note that this is a locally hosted setup.

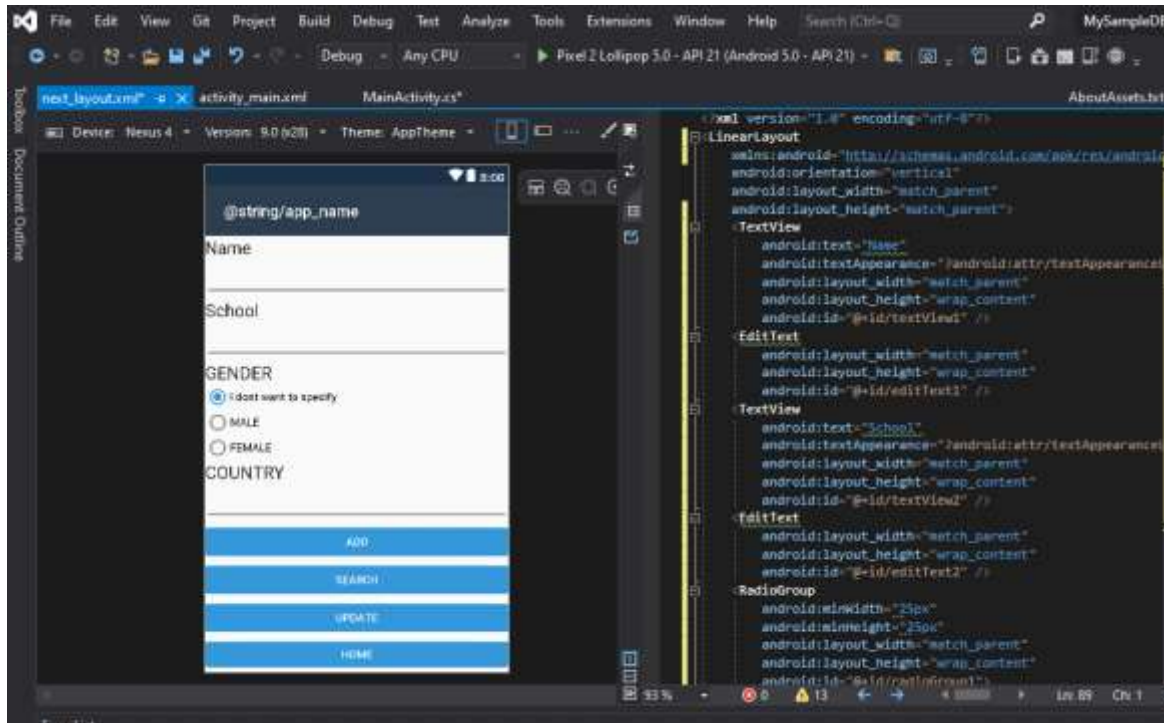
For the Mobile App, we created two(2) layouts: **activity_main** (for login) and **next_layout** (for the transactions)



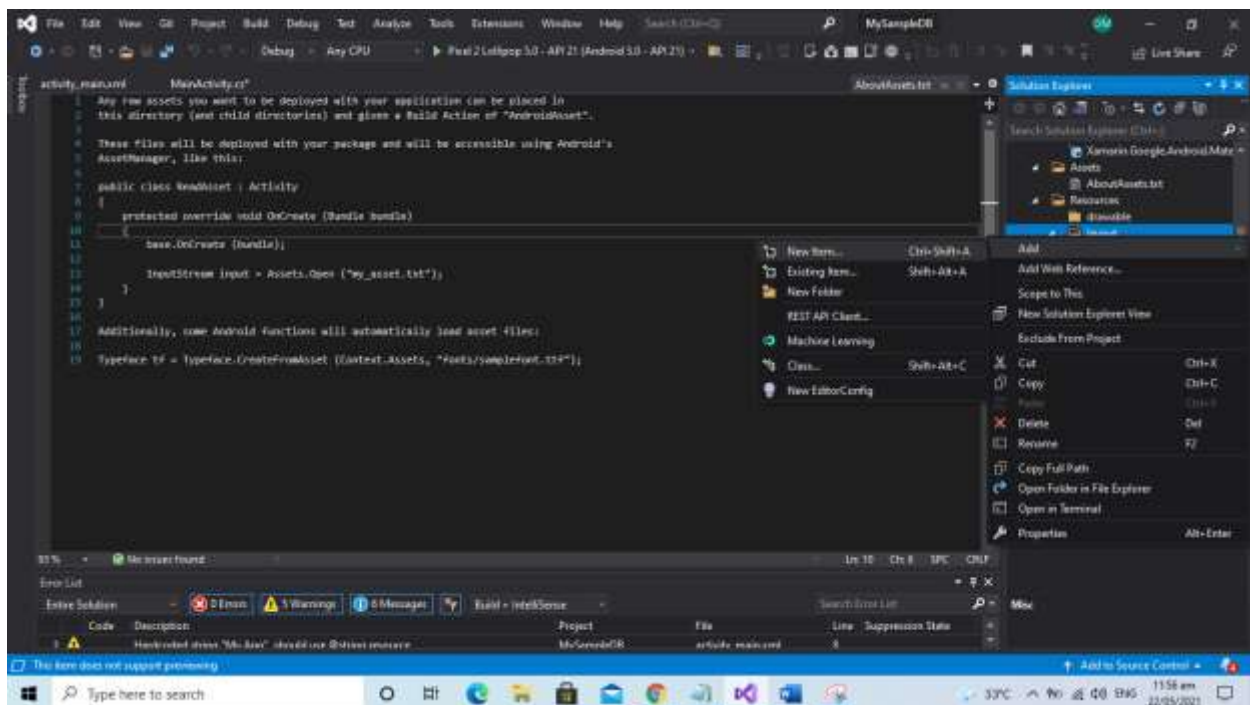
In **activity_main**, we have **TextView**, **EditText (PlainText)** for username, **EditText(Password)** for the password, and a **Button**



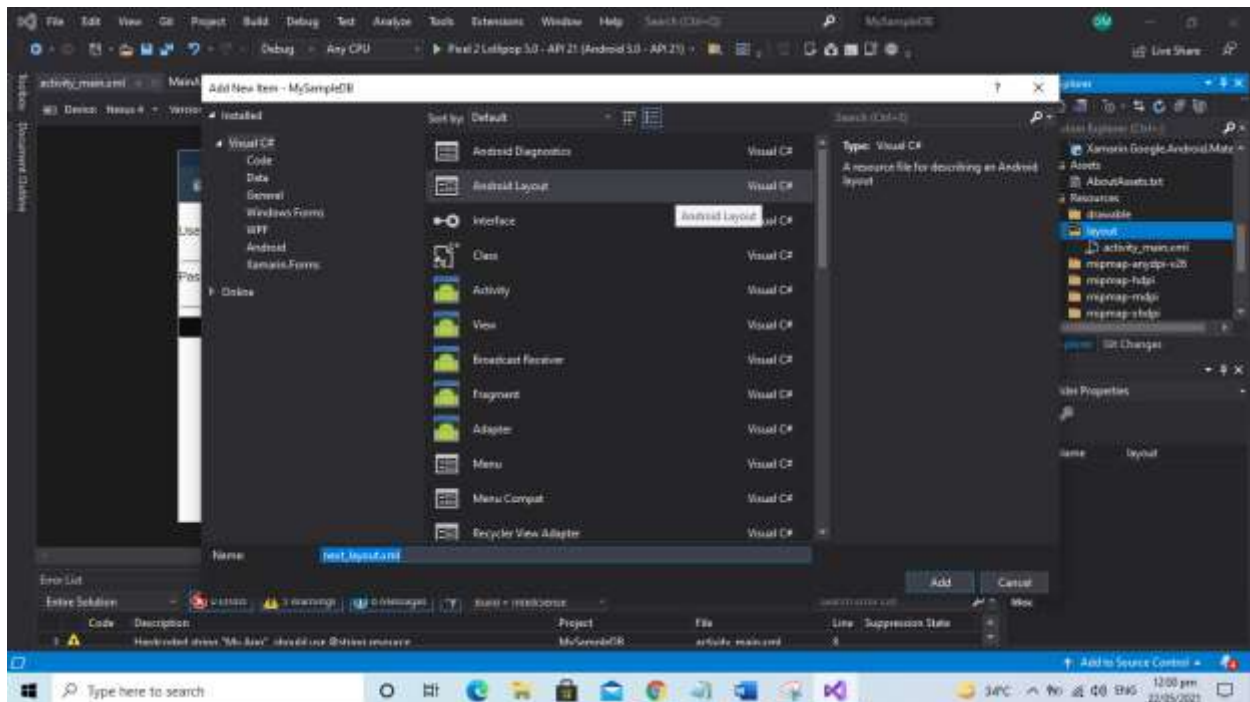
In next_layout, we used TextViews, EditText(for Name, School), GroupRadio for Gender, AutoComplete EditText for Country and Buttons



For those who are just starting with layouts, Here is how to create a new layout for the design of your mobile app. In the solution explorer, Right click in the layout folder under Resource Folder. Select New Item.



A window will appear for you to select Android Layout. Name it (in my case, I named it next_layout)

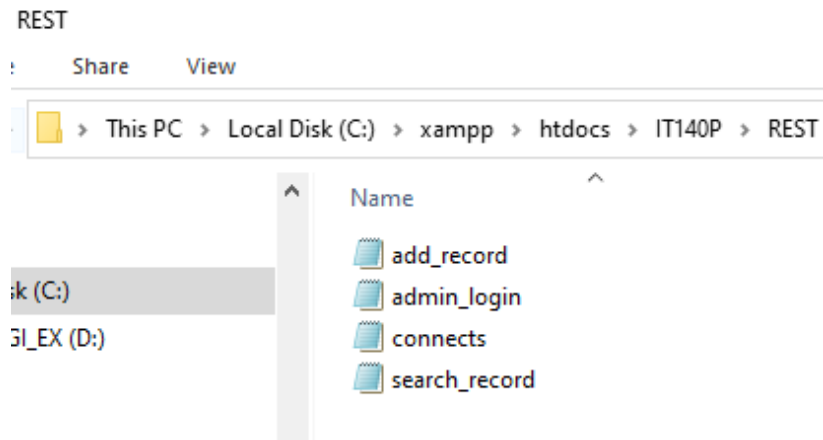


Here are the XML Codes for the two layouts

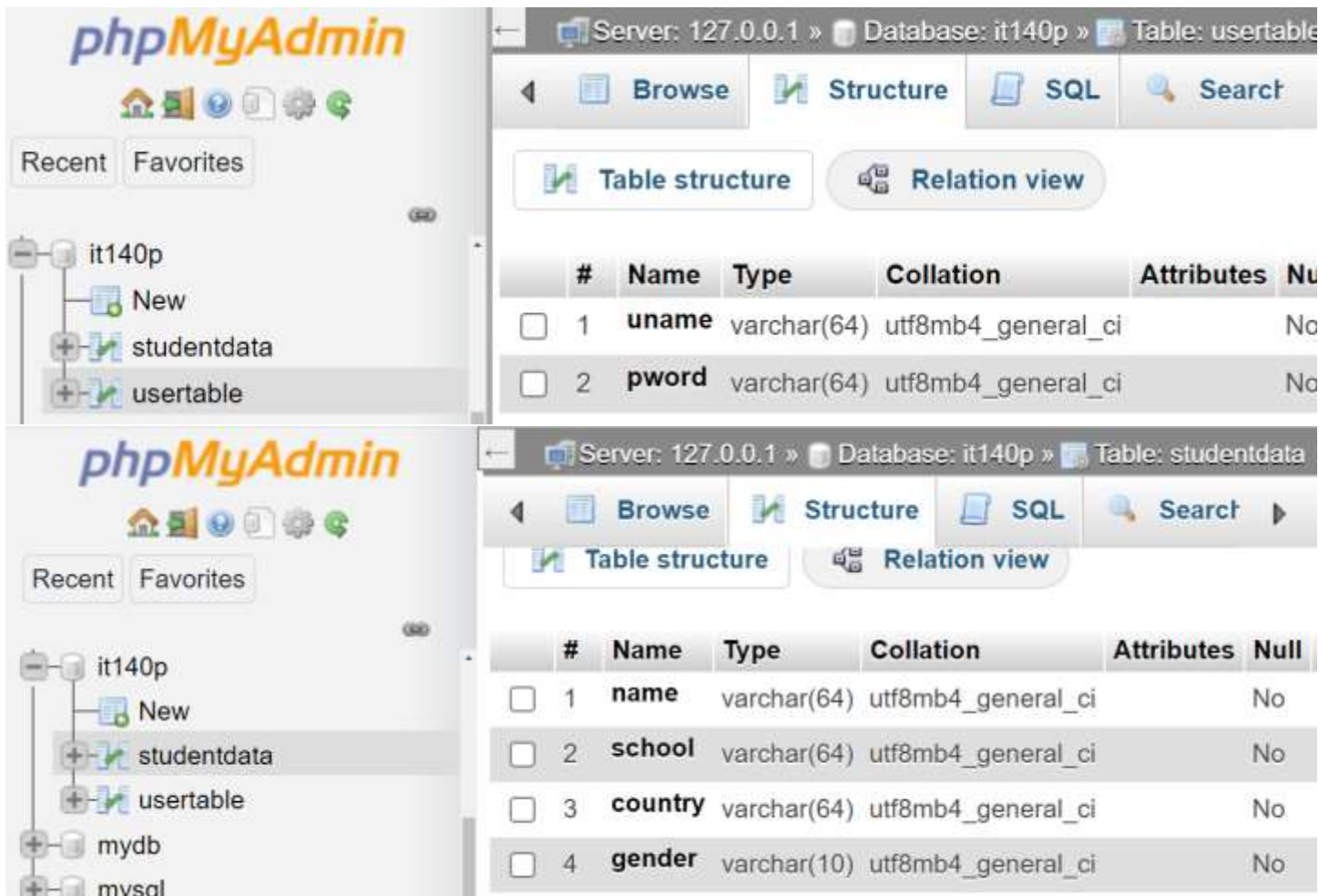
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:text="My App"
        android:layout_width="match_parent"
        android:layout_height="32.5dp"
        android:minWidth="25px"
        android:minHeight="25px"
        android:id="@+id/textView1"
        android:gravity="center" />
    <TextView
        android:text="Username"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView2" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText1" />
    <TextView
        android:text="Password"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView3" />
    <EditText
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText2" />
    <Button
        android:text="LOGIN"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button1"
        android:backgroundTint="#ff110d0d" />
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:text="Name"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView1" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText1" />
    <TextView
        android:text="School"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView2" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editText2" />
    <RadioGroup
        android:minWidth="25px"
        android:minHeight="25px"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/radioGroup1">
        <TextView
            android:text="GENDER"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/textView3" />
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="I dont want to specify"
            android:id="@+id/radioButton1" />
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="MALE"
            android:id="@+id/radioButton2" />
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="FEMALE"
            android:id="@+id/radioButton3" />
    </RadioGroup>
    <TextView
        android:text="COUNTRY"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView4" />
    <AutoCompleteTextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/autoCompleteTextView1" />
    <Button
        android:text="ADD"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button1" />
    <Button
        android:text="SEARCH"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button2" />
    <Button
        android:text="UPDATE"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button3" />
    <Button
        android:text="HOME"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button4" />
</LinearLayout>
```

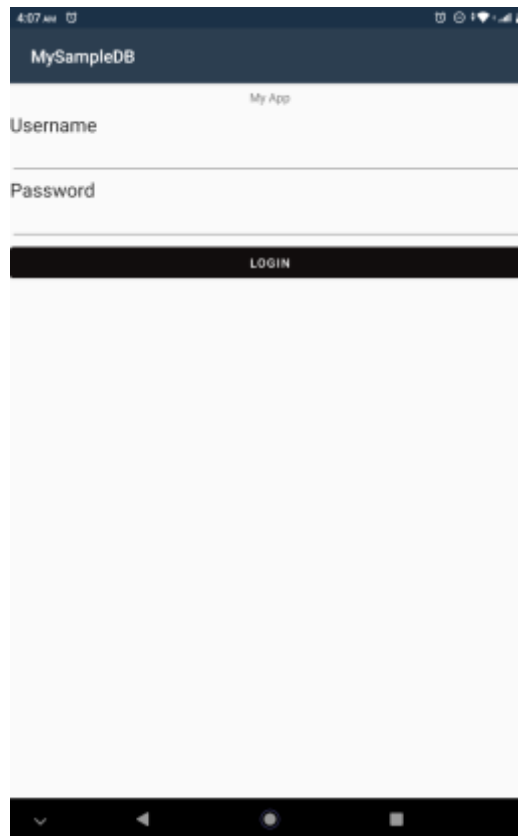
Our REST Web Service API's
My set-up is located herein



This is the database and table structure we used in our Example.
We have two tables: usertable and studentdata



Coding For Login in Xamarin



This is the MainActivity.cs

```
public class MainActivity : AppCompatActivity
```

```
{
```

```
    TextView text1, text2;
    EditText edit1, edit2;
    Button btn1;
    HttpResponseMessage response;
    HttpRequest request;
    String res = "", str = "", uname = "", pword = "";
```

Declaration of widgets and variables

HttpResponseMessage and **HttpRequest** are responsible for the REST Transactions

```
1 reference
```

```
protected override void OnCreate(Bundle savedInstanceState)
```

```
{
```

```
    base.OnCreate(savedInstanceState);
    Xamarin.Essentials.Platform.Init(this, savedInstanceState);
    // Set our view from the "main" layout resource
    SetContentView(Resource.Layout.activity_main);
```

We set layout of MainActivity to be activity_layout

```
    text1 = FindViewById<TextView>(Resource.Id.textView1);
    text2 = FindViewById<TextView>(Resource.Id.textView2);
    edit1 = FindViewById<EditText>(Resource.Id.editText1);
    edit2 = FindViewById<EditText>(Resource.Id.editText2);
    btn1 = FindViewById<Button>(Resource.Id.button1);
```

We instantiate widgets

And set btn to have Click Event

```
    btn1.Click += this.Login;
```

```
}
```

Login Method called by btn.Click

```

public void Login(object sender, EventArgs e)
{
    pword = edit2.Text;
    uname = edit1.Text;
    request = (HttpWebRequest)WebRequest.Create("http://192.168.1.11/IT140P/REST/admin_login.php?uname="+ uname +" &pword=" + pword);
    response = (HttpWebResponse)request.GetResponse();
    StreamReader reader = new StreamReader(response.GetResponseStream());
    res = reader.ReadToEnd();
    Toast.MakeText(this, res, ToastLength.Long).Show();

    if (res.Contains("OK!"))
    {
        Intent i = new Intent(this, typeof(NextActivity));
        i.PutExtra("Name", uname); //baka gamitin ung name sa kabilang Activity
        StartActivity(i);
    }
}

```

We create **HttpWebResponse** to get response from **admin_login** and fetch it back to our mobile app

Response is either "Failed" or "OK"

We instantiate **HttpWebRequest** to create request to **admin_login.php** located in htdocs. **My laptop IP is 192.168.1.11.** Change it accordingly in your setup

We send parameters **uname** and **pword** which we got from editText for username and password

*admin_login - Notepad
File Edit Format View Help

<?php

include_once('connects.php');

\$username = \$_GET['uname'];
\$password = \$_GET['pword'];

\$result = mysqli_query(\$con,"SELECT * FROM usertable WHERE uname = '\$username' AND pword = '\$password'");

```

if(!$row = mysqli_fetch_assoc($result))
{
    echo "Failed!";
}
else
{
    echo "OK!";
}
mysqli_close($con);

```

?>

This is the admin_login webservice API

It GET 'uname' and 'pword' from Mobile App, then makes query from our database if it is existing or not

```

if (res.Contains("OK!"))
{
    Intent i = new Intent(this, typeof(NextActivity));
    i.PutExtra("Name", uname); //baka gamitin ung name sa kabilang Activity
    StartActivity(i);
}

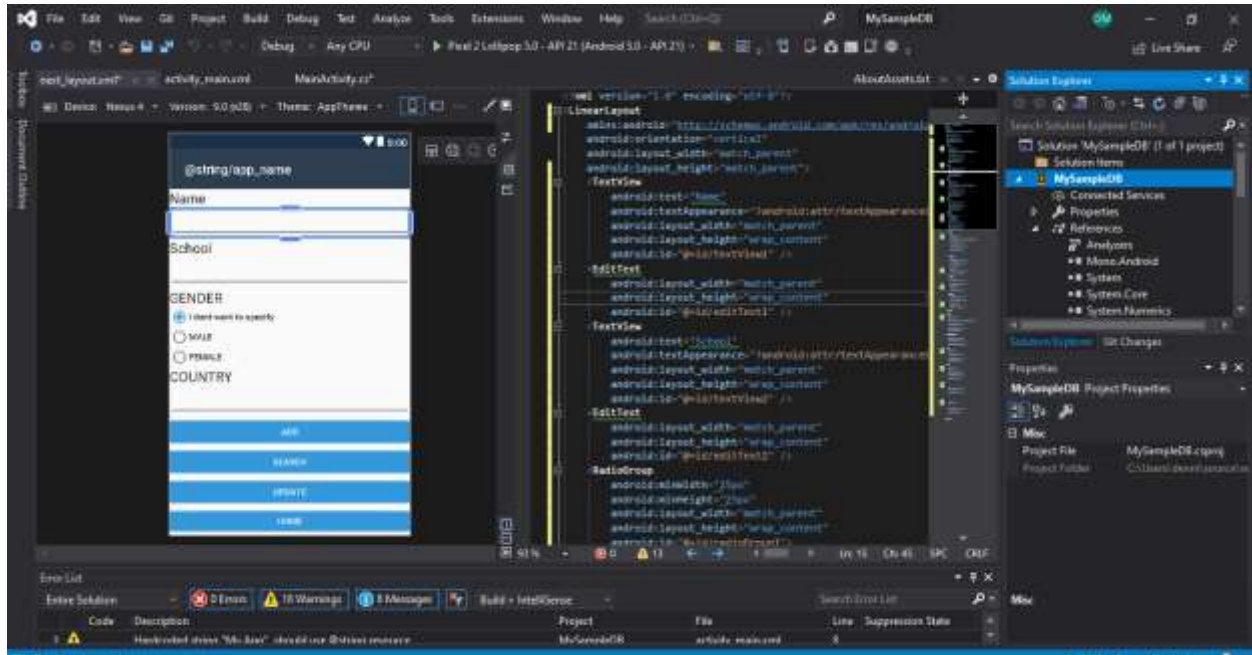
```

An intent was also set in Login.

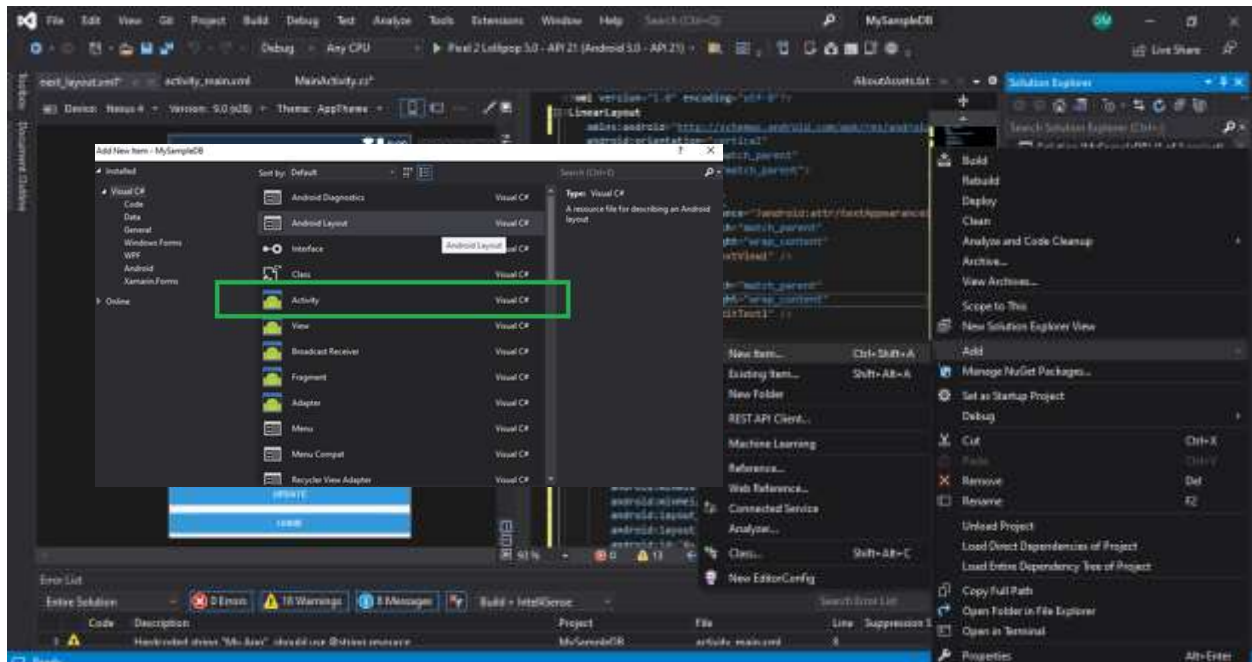
If the Response has "OK" (which is the string we echo from admin_login.php) then it will move to NextActivity (for transaction)

Coding for Transaction in Xamarin (NextActivity)

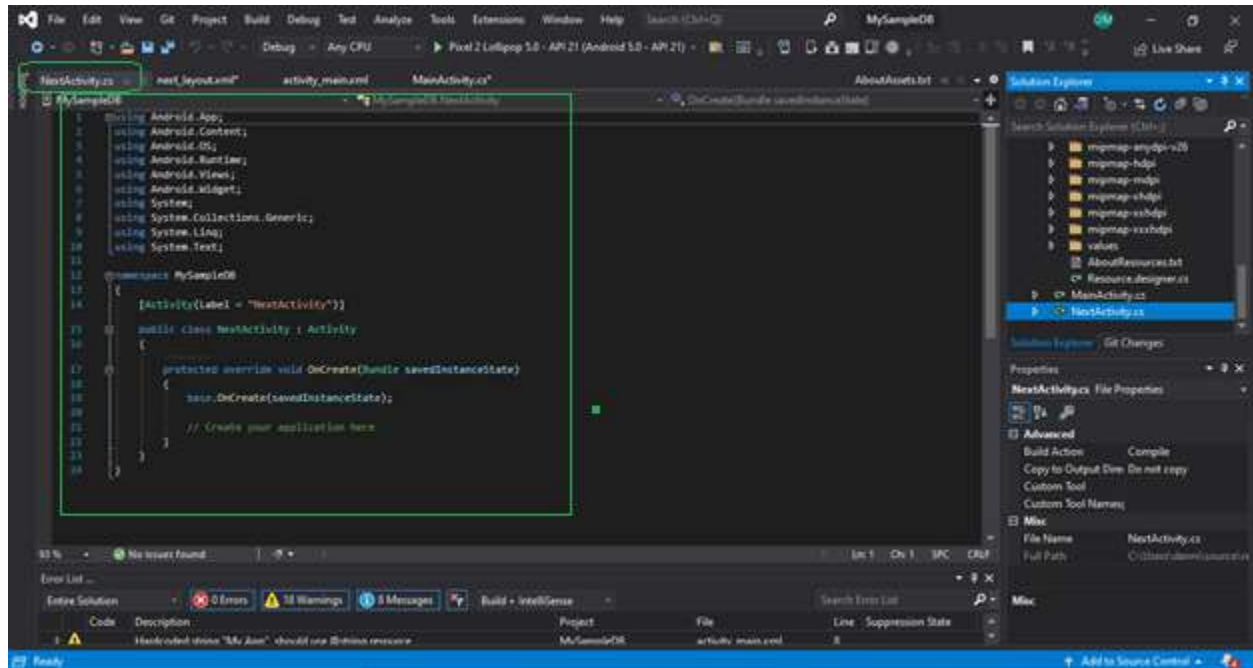
For those who are just starting out, here is how to add a New Activity:
In the Solution Explorer, right click on the Project Name (here it is MySampleDB)



Then Add->New Item, then Select Activity, then Name the Activity (I named it NextActivity)



Here is now the NextActivity. Now we can code here.



..And this is the layout(next_layout) for our NextActivity

4:07 AM

Name

School

GENDER

☒ I don't want to specify

☐ MALE

☐ FEMALE

COUNTRY

ADD

SEARCH

UPDATE

HOME

Coding NextActivity.cs

Lets go directly ro the Oncreate. Make sure the next_layout is set to SetContentView

```
using Android.App;
using Android.Content;
using Android.OS;
using Android.Widget;
using System;
using System.IO;
using System.Net;
using System.Text.Json;

namespace MySampleDB
{
    [Activity(Label = "NextActivity")]
    1 reference
    public class NextActivity : Activity
    {
        EditText editName, editSchool;
        Button btnAdd, btnSearch, btnUpdate, btnHome;
        RadioGroup gender;
        AutoCompleteTextView autoCompleteCountry;
        HttpResponseMessage response;
        HttpWebRequest request;
        String name = "", school = "", country = "", selectedGender = "", res = "", str = "", login_name = "";
    }
}
```

```
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);
    //set layout
    SetContentView(Resource.Layout.next_layout);
    //get name of who login through Intent
    login_name = Intent.GetStringExtra("Name");

    //instantiate widgets
    editName = FindViewById<EditText>(Resource.Id.editText1);
    editSchool = FindViewById<EditText>(Resource.Id.editText2);
    btnAdd = FindViewById<Button>(Resource.Id.button1);
    btnSearch = FindViewById<Button>(Resource.Id.button2);
    btnUpdate = FindViewById<Button>(Resource.Id.button3);
    btnHome = FindViewById<Button>(Resource.Id.button4);
    //set RadioGroup
    gender = FindViewById<RadioGroup>(Resource.Id.radioGroup1);
    gender.CheckedChange += myRadioGroup_CheckedChange;
    //set AutoComplete
    autoCompleteCountry = FindViewById<AutoCompleteTextView>(Resource.Id.autoCompleteTextView1);
    var country = new string[] { "Cambodia", "Indonesia", "Philippines", "Thailand", "Singapore" };
    ArrayAdapter adapter = new ArrayAdapter<string>(this, Android.Resource.Layout.SimpleSpinnerItem, country);
    autoCompleteCountry.Adapter = adapter;

    btnAdd.Click += this.AddRecord;
    btnHome.Click += this.BackHome;
    btnSearch.Click += this.SearchRecord;
}
```

We instantiate widgets

We set btnAdd, btnHome, btnSearch to have Click Event the REST Transactions

We set CheckedChange Event for gender RadioGroup

We setup autoComplete for Country TextView to automatically suggest countries found in the Array

```
1 reference
void myRadioGroup_CheckedChange(object sender, RadioGroup.CheckedChangeEventArgs e)
{
    int checkedItemId = gender.CheckedRadioButtonId;
    RadioButton checkedRadioButton = FindViewById<RadioButton>(checkedItemId);
    selectedGender = checkedItemId.ToString();
    gender.Check(checkedItemId);
}
```

We created CheckedChange function for RadioGroup for gender so we can validate which is checked. Checked or Selected item is stored in selectedGender variable as integer

AddRecord called by btnAdd.Click

```

public void AddRecord(object sender, EventArgs e)
{
    name = editName.Text;
    school = editSchool.Text;
    country = autoCompleteCountry.Text;

    request = (HttpWebRequest)WebRequest.Create("http://192.168.1.11/IT140P/REST/add_record.php?name=" + name + "&school=" + school + "&country=" + country + "&gender=" + selectedGender);
    response = (HttpWebResponse)request.GetResponse();
    StreamReader reader = new StreamReader(response.GetResponseStream());
    res = reader.ReadToEnd();
    Toast.MakeText(this, res, ToastLength.Long).Show();
}

```

We create **HttpWebResponse** to get response from **add_record** and fetch it back to our mobile app

We instantiate **HttpWebRequest** to create request to **add_record.php** located in htdocs

We send parameters to **add_record.php** : name,school,country, and gender which we got from the widgets

add_record - Notepad

File Edit Format View Help

<?php

```

include_once('connects.php');
$name = $_GET['name'];
$school = $_GET['school'];
$country = $_GET['country'];
$gender = $_GET['gender'];

```

```

$result = mysqli_query($con,"INSERT INTO studentdata (name,school,country,gender) VALUES
('$name','$school','$country','$gender')");
echo "Data Inserted";

```

?>

This is the add_record webservice API

It **GET** name, school, country, age from Mobile App, then insert it to our database

SearchRecord Method called by btnSearch.Click

```

public void SearchRecord(object sender, EventArgs e)
{
    name = editName.Text;
    request = (HttpWebRequest)WebRequest.Create("http://192.168.1.11/IT140P/REST/search_record.php?name=" + name );
    response = (HttpWebResponse)request.GetResponse();
    res = response.ProtocolVersion.ToString();
    StreamReader reader = new StreamReader(response.GetResponseStream());
    var result = reader.ReadToEnd();
    //parse result to json then kunin ang root element
    using JsonDocument doc = JsonDocument.Parse(result);
    JsonElement root = doc.RootElement;

    var u1 = root[0]; //isa lang naman result ng query natin so 0 lang.kapag marami, mag loop kayo

    //isa isahing kunin ang value na na-search
    string searchedname = u1.GetProperty("name").ToString();
    string searchedschool = u1.GetProperty("school").ToString();
    string searchedcountry = u1.GetProperty("country").ToString();
    int searchedgender = Convert.ToInt32(u1.GetProperty("gender").ToString());

    //i-set sa widget ang value na nasearch
    editName.Text = searchedname;
    editSchool.Text = searchedschool;
    autoCompleteCountry.Text = searchedcountry;
    gender.Check(searchedgender);
    Toast.MakeText(this, searchedgender.ToString(), ToastLength.Long).Show();
}

```

The response of **search_record** is **JSON encoded**. So we parse the result to JSON.

We instantiate **HttpWebRequest** to create request to **search_record.php** located in htdocs

We send parameter to **search_record..php**: (name) which we got from the widgets

NOTE: we expect a SINGLE Record to be fetched only. For demo purpose, I did not use ID for primary key and the Program has no Validation yet

search_record - Notepad

File Edit Format View Help

<?php

```

include_once('connects.php');
$name = $_GET['name'];

```

```

$query = "SELECT * FROM `studentdata` WHERE name LIKE '%" . $name . "%'";
$check=mysqli_query($con,$query);
$row=mysqli_num_rows($check);
$array = array();

```

```

if($check == FALSE) {
    echo " ".$row." "; // TODO: better error handling
}

```

```

while($row=mysqli_fetch_array($check))
{
    $array[] = $row;
}
echo json_encode($array);

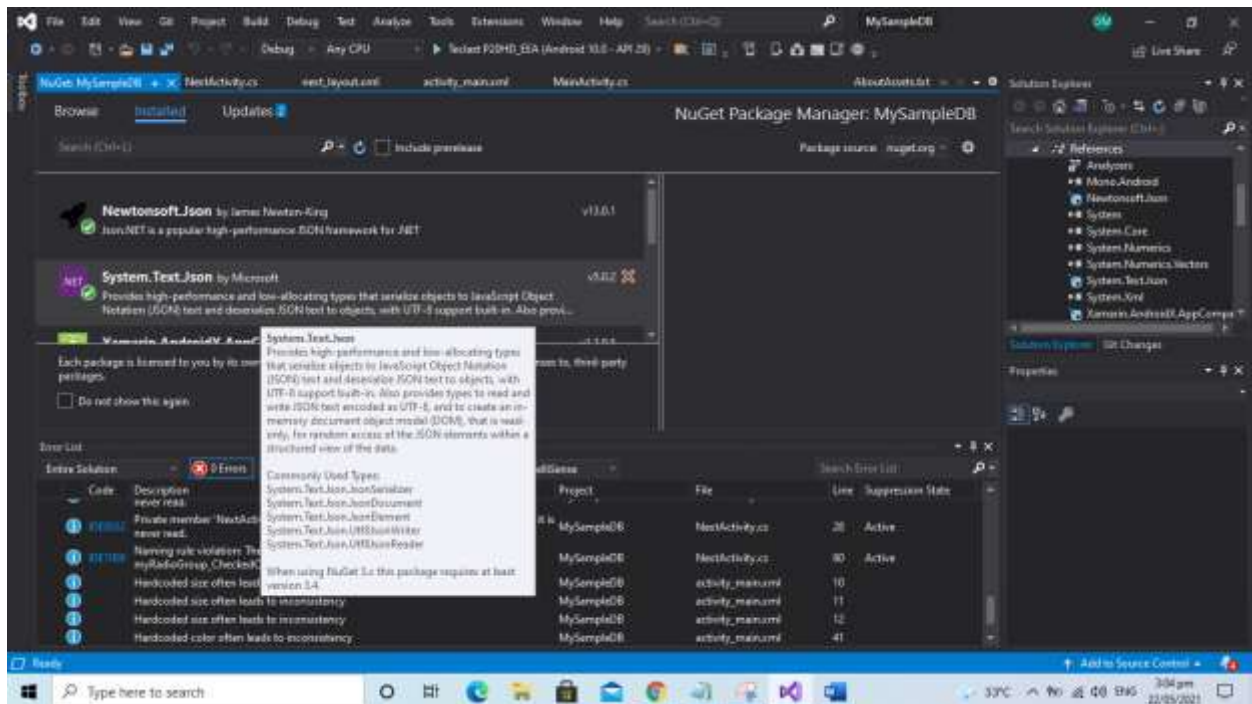
```

?>

This Web Service GET name from Mobile App. It then searches for the record. Since we expect to get Name, Gender, School, and Country from the Table, we encoded it to JSON so we can PASS everything back to Mobile App

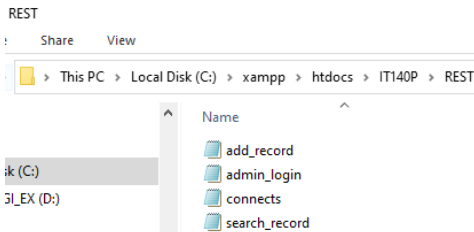
IT140P Module 5: REST Approach via Xamarin Lab Discussion Guide

Note: For JsonDocument and all Json Serialization and Deserialization to work in Xamarin, Go to Solution Explorer, Right click references and check and install System.Text.Json from NuGet

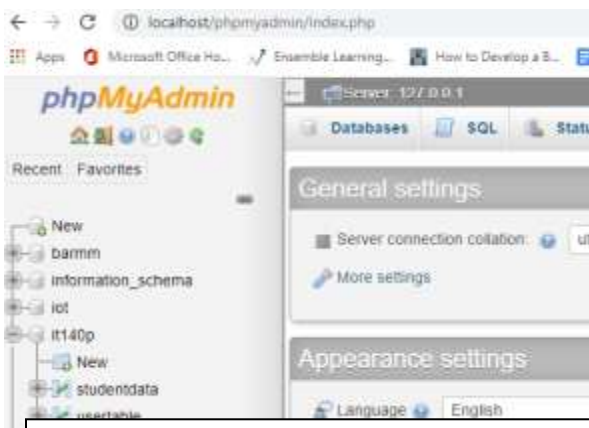


A quick setup summary for those who are NEW with this

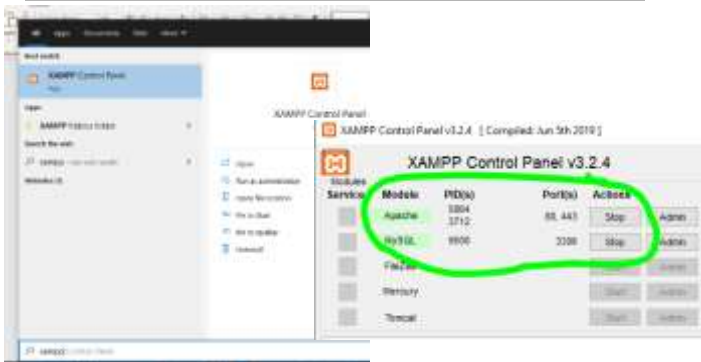
In your laptop create at put the webservice PHP Files in **htdocs**. In my setup, I placed everything in **C:/xampp/htdocs/IT140P/REST**



Create the database using **phpMyAdmin** which can be accessed in **localhost/phpMyAdmin/index.php**.



Make sure that **Apache** and **MySQL** are started and running too



In Xamarin Coding, the **HttpWebRequest** allows your mobile App to interact and send/receive data to the REST webservices in your computer. Make sure to put your laptop's IP Address in the actual URI



```

//Will call Login(object sender, EventArgs e)
{
    password = edit2.Text;
    username = edit1.Text;
    request = (HttpWebRequest)WebRequest.Create("http://192.168.1.1/IT140P/REST/admin_login.php?username=" + username + "&password=" + password);
    response = (HttpWebResponse)request.GetResponse();
    Stream reader = new StreamReader(response.GetResponseStream());
    res = reader.ReadToEnd();
    Toast.MakeText(this, res, ToastLength.Long).Show();
}

```

To get your IP address, go to command prompt and type **ipconfig**. Copy the IPv4 Address and use that in your **HttpWebRequest**



For All Code Listing that you can test, kindly refer to the uploaded Project File I created in Visual Studio