Rock Paper Scissors w/ Firebase and Authentication

- Create a project like we normally did in our android classes, but change the minimum API to 16 instead of 15. This is because firebase does not allow it to go below 15. Take note of the package name as you will need it for setting up the firebase for your android app.

1. Set-up Firebase on your application

1.1. Google firebase and click on the first link and get your own database created.

1.2. After you created your database, you will see an android icon and you click it. It will take you to another page and tell you how to set it up on your application.

1.3. The package name is required for the first step, so copy paste it in and continue, you may fill in the optional boxes if desired. Then click next when you’re ready to move on. These steps may change, so I would strongly recommend following the procedure online.

1.4. This Step requires you to download google-services.json file and put it in your solution. Once you download it go to android studio and switch from android to project. Open up your project folder then your app folder, and drag and drop google-services.json file into their. Then click next when you have completed this step.

1.5. This next step requires you to input a classpath and implementation. So in the project level folder there is a build.grade, open it and in buildscripts dependencies put classpath ‘com.google.gms:google-services:4.3.3’. After that, open up app folder and open build.gradle, at the top add ‘apply plugin: ‘com.google.gms.google-services’’ and down below in the dependencies add ‘implementation ‘com.google.firebase:firebase-analytics:17.2.2’’. Once done, hit the sync now button. Once that is done click next on your browser.

1.6. Next on your browser it will tell you to run you app and check if it connects to firebase. You need to create an emulator, I used nougat 25 w/ Galaxy Nexus. Then run it and once the running is completed and your app is working on your emulator, go back to your web browser and it should say that you successfully connected then proceed to your console.

2. Designing the Log-in / Registration pages, and creating new users onto firebase (Followed youtube video)

2.1. In your solution, create two empty activities, one called Login Activity and the other one called Register Activity.

2.2. Now it is time to customize your both pages, you will need a three fields for username, email, password for register, but for log in you will need two of those fields which are email and password. A button to submit, and one textview to change between both activites. I copied and pasted from one activity to another and changed some words and removed what was not required.

Here is my code:

Activity\_register.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorPrimary"  
 android:orientation="vertical"  
 tools:context=".RegisterActivity">  
 <TextView  
 android:layout\_marginTop="60sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/register\_name"  
 android:textSize="30sp"  
 android:layout\_marginBottom="10sp"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/register\_message" />  
 <EditText  
 android:id="@+id/register\_user\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/user\_message"  
 android:textColor="@color/textWhiteColor"  
 android:layout\_marginTop="60sp"  
 android:layout\_marginLeft="40sp"  
 android:layout\_marginRight="40sp"  
 android:layout\_marginBottom="20sp"/>  
 <EditText  
 android:id="@+id/register\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/email\_message"  
 android:textColor="@color/textWhiteColor"  
 android:layout\_marginLeft="40sp"  
 android:layout\_marginRight="40sp"  
 android:layout\_marginBottom="20sp"/>  
 <EditText  
 android:id="@+id/register\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"

android:inputType="textPassword"  
 android:hint="@string/password\_message"  
 android:textColor="@color/textWhiteColor"  
 android:layout\_marginLeft="40sp"  
 android:layout\_marginRight="40sp"  
 android:layout\_marginBottom="60sp"/>  
 <Button  
 android:id="@+id/create\_new\_account\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="80sp"  
 android:layout\_marginRight="80sp"  
 android:text="@string/new\_account\_button"/>  
 <TextView  
 android:id="@+id/return\_log\_in"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:textAlignment="center"  
 android:layout\_marginTop="6sp"  
 android:text="@string/return\_account\_button"/>  
</LinearLayout>

Activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/colorPrimary"  
 android:orientation="vertical"  
 tools:context=".RegisterActivity">  
 <TextView  
 android:layout\_marginTop="60sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/log\_in\_name"  
 android:textSize="30sp"  
 android:layout\_marginBottom="10sp"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/log\_in\_message"  
 android:layout\_marginBottom="10sp"/>  
 <EditText  
 android:id="@+id/log\_in\_email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/email\_message"  
 android:textColor="@color/textWhiteColor"  
 android:layout\_marginLeft="40sp"  
 android:layout\_marginRight="40sp"  
 android:layout\_marginBottom="20sp"/>  
 <EditText  
 android:id="@+id/log\_in\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"

android:inputType="textPassword"  
 android:hint="@string/password\_message"  
 android:textColor="@color/textWhiteColor"  
 android:layout\_marginLeft="40sp"  
 android:layout\_marginRight="40sp"  
 android:layout\_marginBottom="60sp"/>  
 <Button  
 android:id="@+id/log\_into\_account\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="80sp"  
 android:layout\_marginRight="80sp"  
 android:text="@string/log\_in\_account\_button"/>  
 <TextView  
 android:id="@+id/return\_register"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:textAlignment="center"  
 android:layout\_marginTop="6sp"  
 android:text="@string/return\_register\_button"/>  
</LinearLayout>

2.3. Once you are done, it is time to start communicating with firebase to be able to register new accounts and log into your account. Click Tools and click firebase, a tab will appear to the right of the screen, click on Authentication and then email and password authentication. Once their click connect to firebase, sign into your google account and click allow. Go back to your solution and click sync.

2.4. Next Step, click on button, it will give you a warning of what will be added and accept it. It will add a dependency to your project, app, build.gradle.

2.5. Go to your RegisterActivity and create global variables for the username, email, and password also for the register button and the return to log in page textview. You will have to make a global variable for fire base authentication which is FirebaseAuth fAuth for example. Then in on create, findviewbyid for each of those. Also for authentication you will need to do fAuth = FirebaseAuth.getInstanct(); in on create. Here is my code:

package ca.nait.wteljega1.rockpaperscissorsfirebase;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
  
import com.google.firebase.auth.FirebaseAuth;  
  
public class RegisterActivity extends AppCompatActivity  
{  
 EditText registerUsername, registerEmail, registerPassword;  
 Button registerButton;  
 TextView returnLogin;  
 FirebaseAuth fAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
  
 registerUsername = findViewById(R.id.*register\_user\_name*);  
 registerEmail = findViewById(R.id.*register\_email*);  
 registerPassword = findViewById(R.id.*register\_password*);  
 returnLogin = findViewById(R.id.*return\_log\_in*);  
 registerButton = findViewById(R.id.*create\_new\_account\_button*);  
  
 // Declare Fire base Authentication  
 fAuth = FirebaseAuth.*getInstance*();  
 }  
}

2.6. Next we got to put an onclicklistener on the button within the oncreate(). We have done this with many others buttons. So what we do is

registerButton.setOnClickListener(this);

We will noticed an error, now we right click and make the method onclick and do the usual stuff like switch (view.getId()) and the case for the button. Now let grab the values text and string like so

String email = registerEmail.getText().toString().trim();  
String password = registerPassword.getText().toString().trim();

2.7. Now it is time to valiadate, I want to make sure email and password have something written in them and password has a minimum of 8 characters, so what I did in the case for the register button is

if (TextUtils.*isEmpty*(email))  
{  
 registerEmail.setError("Email is Required");  
 return;  
}  
  
if (TextUtils.*isEmpty*(password))  
{  
 registerPassword.setError("Password is Required");  
 return;  
}  
if (password.length() < 8)  
{  
 registerPassword.setError("Password must have 8 or more Characters");

return;  
}

TextUtils is simply a set of utility functions to do operations on String objects. In fact, the whole class doesn't have any instance fields or methods. Everything is static. Consider it like a container to group functions with a text-based semantic. Many of them could have been methods of a String inherited class or CharSequence inherited class.

2.8. Know it is time to create the account, so what we have to do go through firebase authentication. Below there is the method that I used to create my users, this requires email and password, you will notice an error on the addOnComplete Listener. The addOnCompleteListener is another method that is used to validate if the account was created successfully or not.

This is still in the OnClick method

fAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(this);

@Override  
public void onComplete(@NonNull Task<AuthResult> task)  
{  
 if (task.isSuccessful())  
 {  
 Toast.*makeText*(this, "User Created", Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(getApplicationContext(), MainActivity.class));  
 }  
 else  
 {  
 Toast.*makeText*(this, "Error: " + task.getException().getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
}

2.9. Now it is time to check if you are logged in, when you click the registration page, if you are logged in it should take you to the mainactivity. To validate this, in oncreate() we used fAuth.getCurrentUser() != null to check if the user is signed in or not.

if (fAuth.getCurrentUser() != null)  
{  
 startActivity(new Intent(getApplicationContext(), MainActivity.class));  
 finish();  
}

2.10. With that done, now we have to add in the android manifest file internet permissions

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

We need to switch our starting activity around since my solution did not use main activity, its simple

<activity android:name=".LoginActivity"></activity>  
<activity android:name=".MainActivity" />  
<activity android:name=".RegisterActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
</activity>

I just switched around the .MainActivity and the .RegisterActivity

Once that is done go to your firebase, go to console, then authentication and sign-in method and you should see email/password. Click it and enable it.

With that done, now you should be able to run your app and be able to create an account.

Note: If you are getting this error: cannot access InternalTokenProvider

class file for com.google.firebase.internal.InternalTokenProvider not found, switch to your project open application folder, app and in your build.gradle dependencies, update both of these implementation to their latest versions, these were the latest versions for me:

implementation 'com.google.firebase:firebase-analytics:17.3.0'

implementation 'com.google.firebase:firebase-auth:19.3.0'

3. Logging out of your account and logging into your account (followed youtube video)

3.1. With registering your account done, it is time to logout of the account, go to your activity\_main.xml and make a button like so, we code like button list usual here and add a onClick, this is required for creating the method:

<Button  
 android:id="@+id/log\_out\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="right"  
 android:layout\_margin="8sp"  
 android:text="@string/log\_out\_button"

android:onClick="logout" />

Then switch to MainActivity Java class and it is time to code the logout method, the public void is called logout because the onClick of the button is called logout. This is just a different way of making a switch with cases in an onClick method with listeners.

public void logout (View view)  
{  
 FirebaseAuth.*getInstance*().signOut();  
 startActivity(new Intent(getApplicationContext(), LoginActivity.class));  
 finish();  
}

With that done, you should be able to logout, it should take you to the loginActivity after clicking the button.

3.2. Now it is time to log into your account that you registered, to begin we declare global variables and in the oncreate method do findviewbyid like the picture below. Also have to do FirebaseAuth like we did for the register page.

EditText logEmail, logPassword;  
Button logInButton;  
TextView returnRegister;  
FirebaseAuth fAuth;  
  
@Override  
protected void onCreate(Bundle savedInstanceState)  
{  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
   
 logEmail = findViewById(R.id.*log\_in\_email*);  
 logPassword = findViewById(R.id.*log\_in\_password*);  
 logInButton = findViewById(R.id.*log\_into\_account\_button*);  
 returnRegister = findViewById(R.id.*return\_register*);  
   
 fAuth = FirebaseAuth.*getInstance*();  
}

3.3. With that done, now we have to set a onclicklistener for the logInButton and create the onClick Method with a switch and then a case for the button.

In Oncreate() method

logInButton.setOnClickListener(this);

alt enter and click the second option and grab the onclick

@Override  
public void onClick(View view)  
{  
 switch (view.getId())  
 {  
 case R.id.*log\_into\_account\_button*:  
 {  
  
 }  
 }  
}

3.4. This next part is simple, we just copy and paste the validation from the register page.

String email = logEmail.getText().toString().trim();  
String password = logPassword.getText().toString().trim();  
  
if (TextUtils.*isEmpty*(email))  
{  
 logEmail.setError("Email is Required");  
 return;  
}  
  
if (TextUtils.*isEmpty*(password))  
{  
 logPassword.setError("Password is Required");  
 return;  
}  
if (password.length() < 8)  
{  
 logPassword.setError("Password must have 8 or more Characters");

return;  
}

I Change my global variable names, so I had to just change that for this page.

3.5. With that done now we authenticate the user. We use firebase authentication and pass in the email and password to validate.

fAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(this);

I added in addOnCompleteListener like what I did for registering, alt enter and the second option and get the onComplete method. I just copied and pasted and simply changed the toast message if it is successful.

@Override  
public void onComplete(@NonNull Task<AuthResult> task)  
{  
 if (task.isSuccessful())  
 {  
 Toast.*makeText*(this, "User Logged In", Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(getApplicationContext(), MainActivity.class));  
 }  
 else  
 {  
 Toast.*makeText*(this, "Error: " + task.getException().getMessage(), Toast.*LENGTH\_SHORT*).show();

Toast.*makeText*(this, "Register your account if you are not registered yet" , Toast.*LENGTH\_LONG*).show();  
 }  
}

3.6. That is all that is required, the log in should work now and the logout should work as well once your logged in, the one part that has been missed is the textview that’s allows you to switch between Login activity and register activity.

3.7. In the OnCreate() method on both Login and Register activities I make a onClickListener.

RegisterActivity.java

returnLogin.setOnClickListener(this);

LoginActivity.java

returnRegister.setOnClickListener(this);

now in the onclick methods switch create another case and grab the id, and make a startactivity for both of them leading to the opposite pages (from register to login and from login to register)

RegisterActivity.java

case R.id.*return\_log\_in*:  
{  
 startActivity(new Intent(getApplicationContext(), LoginActivity.class));  
}

LoginActivity.java

case R.id.*return\_register*:  
{  
 startActivity(new Intent(getApplicationContext(), RegisterActivity.class));  
}

This allows you to switch between both the login and register.

Once last thing I did was I switch my launching page to be the loginActivity in the androidmanefest file.

<activity android:name=".RegisterActivity"></activity>  
<activity android:name=".MainActivity" />  
<activity android:name=".LoginActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
</activity>

Now you should be able to run your android app and you should be able to log in into your account that you created and signout, and switch between both the log in and register pages.

4. Making the rock paper scissors game (Followed a youtube video for this part)

4.1. With all the firebase signing in and out done with, it is time to start making the rock paper scissors. Let start by making a xml for it, I put two imagesview, textviews, and three buttons which are rock paper scissors, I also put pictures of rock, paper and scissors in my drawable folder and added a background as well into the drawable

Strings.xml

<string name="player\_choice">Players Choice</string>  
<string name="npc\_choice">Computer(NPC) Choice</string>  
<string name="score">Score</string>  
<string name="track\_score">Player: 0 Computer: 0</string>  
<string name="rock\_button">Rock</string>  
<string name="paper\_button">Paper</string>  
<string name="scissor\_button">Scissors</string>

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/images"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/log\_out\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="end"  
 android:layout\_margin="8sp"  
 android:text="@string/log\_out\_button"  
 android:onClick="logout" />  
  
 <TextView  
 android:layout\_marginTop="10sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/player\_choice"  
 android:textSize="20sp"/>  
 <ImageView  
 android:id="@+id/player\_image"  
 android:layout\_marginTop="10sp"  
 android:layout\_width="200sp"  
 android:layout\_height="100sp"  
 android:layout\_gravity="center\_horizontal"  
 android:src="@drawable/rock"/>  
 <TextView  
 android:layout\_marginTop="15sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/npc\_choice"  
 android:textSize="20sp"/>  
 <ImageView  
 android:id="@+id/npc\_image"  
 android:layout\_marginTop="10sp"  
 android:layout\_width="200sp"  
 android:layout\_height="100sp"  
 android:layout\_gravity="center\_horizontal"  
 android:src="@drawable/rock"/>  
 <TextView  
 android:layout\_marginTop="20sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/score"  
 android:textSize="18sp"/>  
 <TextView  
 android:id="@+id/tv\_track\_score"  
 android:layout\_marginTop="15sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="@string/track\_score"  
 android:textSize="18sp"/>  
 <LinearLayout  
 android:layout\_marginTop="20sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:orientation="horizontal">  
 <Button  
 android:id="@+id/rock\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/rock\_button"/>  
 <Button  
 android:id="@+id/paper\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/paper\_button"/>  
 <Button  
 android:id="@+id/scissors\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/scissor\_button"/>  
 </LinearLayout>  
</LinearLayout>

4.2. It is time to code in the main activity java class, we need to declare global variables for both image views, the score text view, and the three buttons and also findviewbyid for each of those in the onCreate method. Here is what I have

For global variables

Button rock\_button, paper\_button, scissors\_button;  
TextView tv\_score;  
ImageView image\_player, image\_npc;

Findviewbyid in the onCreate()

rock\_button = (Button) findViewById(R.id.*rock\_button*);  
paper\_button = (Button) findViewById(R.id.*paper\_button*);  
scissors\_button = (Button) findViewById(R.id.*scissors\_button*);  
tv\_score = (TextView) findViewById(R.id.*tv\_track\_score*);  
image\_player = (ImageView) findViewById(R.id.*player\_image*);  
image\_npc = (ImageView) findViewById(R.id.*npc\_image*);

4.3. Now it is time to set onclicklistener for each of the buttons, like we did previously with other buttons. Here is my code: (Refer back to Step 3.3 for what to do)

In onCreate()

rock\_button.setOnClickListener(this);  
paper\_button.setOnClickListener(this);  
scissors\_button.setOnClickListener(this);

The OnClick Method

@Override  
public void onClick(View view)  
{  
 switch (view.getId())  
 {  
 case R.id.*rock\_button*:  
 {  
  
 }  
 case R.id.*paper\_button*:  
 {  
  
 }  
 case R.id.*scissors\_button*:  
 {  
  
 }  
 }  
}

4.4. Now let’s start by settings their default images rock is rock and so on to do this we take the image view of the player and set image resource to the drawable rock, we repeat this with paper and scissors. Example:

image\_player.setImageResource(R.drawable.*rock*);

4.5. With that done we will create another method to pass in the string that was selected so for the example I will show rock, they are copy and paste you just pass in the string that is input to paper and scissors.

computerChoice("rock");

private void computerChoice(String Player\_Selection)  
{  
  
}

This is how your cases should look like at this point of time

case R.id.*rock\_button*:  
{  
 image\_player.setImageResource(R.drawable.*rock*);  
 computerChoice("rock");

break;  
}  
case R.id.*paper\_button*:  
{  
 image\_player.setImageResource(R.drawable.*paper*);  
 computerChoice("paper");

break;  
}  
case R.id.*scissors\_button*:  
{  
 image\_player.setImageResource(R.drawable.*scissors*);  
 computerChoice("scissors");

break;  
}

4.6. This next part, in the computerChoice method above, we create a randomizer and make, intilize a string and make the random generator produce a number between 1-3. With the random number generator choosing between 1, 2 and 3, we will now make 1 = rock, 2 = paper and 3 = scissors, the String computer\_choice is what will store those values and set the picture of the computers choice.

String computer\_choice = "";  
Random random = new Random();  
  
// choose between 1 2 or 3  
int computer\_choice\_number = random.nextInt(3) + 1;

if (computer\_choice\_number == 1)  
{  
 computer\_choice = "rock";  
 image\_npc.setImageResource(R.drawable.*rock*);  
}  
if (computer\_choice\_number == 2)  
{  
 computer\_choice = "paper";  
 image\_npc.setImageResource(R.drawable.*paper*);  
}  
if (computer\_choice\_number == 3)  
{  
 computer\_choice = "scissors";  
 image\_npc.setImageResource(R.drawable.*scissors*);  
}

4.7. With that done, now we need to determine who wins, loses or draw. We will do a lot of if statements to determine who is the winner. I also declared two global int variables to keep track of the score of the player and computer.

int playerScore, computerScore = 0;

Here are what my if statements look like to determine if it is a draw, win or loss

// Determining who is the winner, loser or if its a draw  
if (computer\_choice == player\_selection)  
{  
 Toast.*makeText*(this, "Draw, Nobody Wins!", Toast.*LENGTH\_SHORT*).show();  
}  
else if (player\_selection == "rock" && computer\_choice == "scissors")  
{  
 playerScore++;  
 Toast.*makeText*(this, "Rock Obliterates Scissors! You Win!", Toast.*LENGTH\_SHORT*).show();  
}  
else if (player\_selection == "rock" && computer\_choice == "paper")  
{  
 computerScore++;  
 Toast.*makeText*(this, "Paper Covers Rock! Computer Wins!", Toast.*LENGTH\_SHORT*).show();  
}  
else if (player\_selection == "paper" && computer\_choice == "rock")  
{  
 playerScore++;  
 Toast.*makeText*(this, "Paper Covers Rock! You Win!", Toast.*LENGTH\_SHORT*).show();  
}  
else if (player\_selection == "paper" && computer\_choice == "scissors")  
{  
 computerScore++;  
 Toast.*makeText*(this, "Paper Gets Cut By Scissors! Computer Wins", Toast.*LENGTH\_SHORT*).show();  
}  
else if (player\_selection == "scissors" && computer\_choice == "rock")  
{  
 computerScore++;  
 Toast.*makeText*(this, "Scissors Gets Crushed By Rock! Computer Wins!", Toast.*LENGTH\_SHORT*).show();  
}  
else if (player\_selection == "scissors" && computer\_choice == "paper")  
{  
 playerScore++;  
 Toast.*makeText*(this, "Scissors Cuts Paper! You Win!", Toast.*LENGTH\_SHORT*).show();  
}

4.8. With that done you are now able to play the game, but we need a way to update the score. Now we return to ours Onclick method and within our cases we set our text view to change and keep updating the score. We need to put the same message for all three cases. Here is what it should look like for all three cases

tv\_score.setText("Player: " + Integer.*toString*(playerScore) + " Computer: " + Integer.*toString*(computerScore));

With that done, you have completed my app. You are now able to play rock paper scissors and have authentication on it from firebase as well.