

ICP10

CHENNA SRINIVAS – 16307995

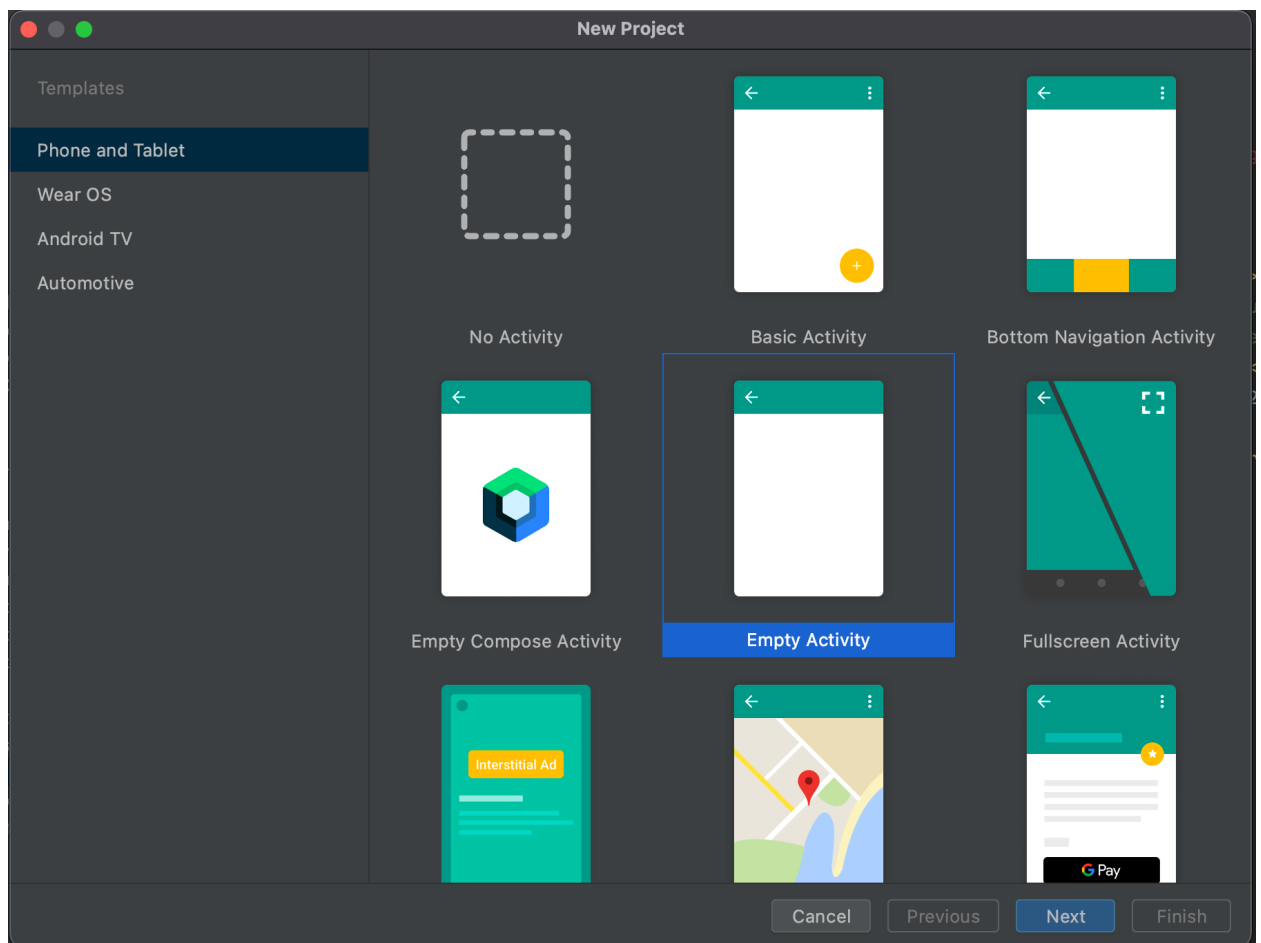
https://github.com/sc7vc/Web_Applications/tree/main/Mobile/ICP10

MAMJALA MANISHA – 16307984

<https://github.com/mmg6m/Web/tree/main/Mobile/ICP10>

In this ICP we are creating Mobile app which fetches the Github user data by performing API calls.

1. Open the Android Studio and click on New Project and select empty activity as below



2. Give our project name as ICP10 as below and click on Finish

New Project

Empty Activity

Creates a new empty activity

Name: ICP10

Package name: com.example.icp10

Save location: /Users/srchenna/Documents/Web_Applications/Mobile/ICP102

Language: Java

Minimum SDK: API 21: Android 5.0 (Lollipop)

Information: Your app will run on approximately **98.0%** of devices.
[Help me choose](#)

☐ Use legacy android.support libraries [?](#)
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

Buttons: Cancel, Previous, Next, Finish

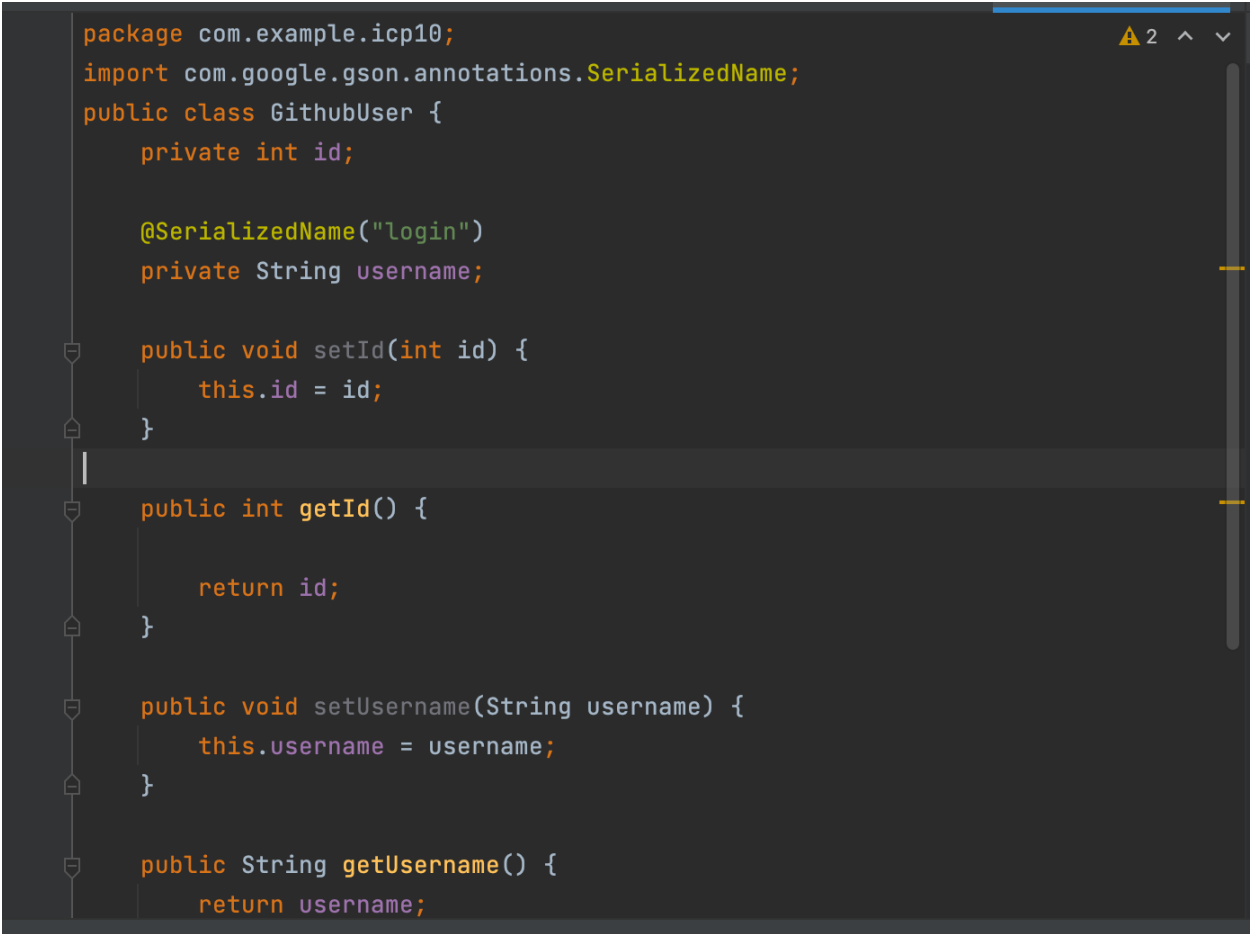
3. We should add below code to give Internet permission to our application in Androidmanifest.xml

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

4. In build.gradle add below snippet for retrofit configuration, which converts API calls into objects

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
```

5. Create GithubUser.java file as below

A screenshot of an IDE window showing the code for GithubUser.java. The code is written in Java and includes package, import, class, and method declarations. The IDE has a dark theme and a sidebar on the left with icons for Explorer, Search, and Run and Debug. The code is as follows:

```
package com.example.icp10;
import com.google.gson.annotations.SerializedName;
public class GithubUser {
    private int id;

    @SerializedName("login")
    private String username;

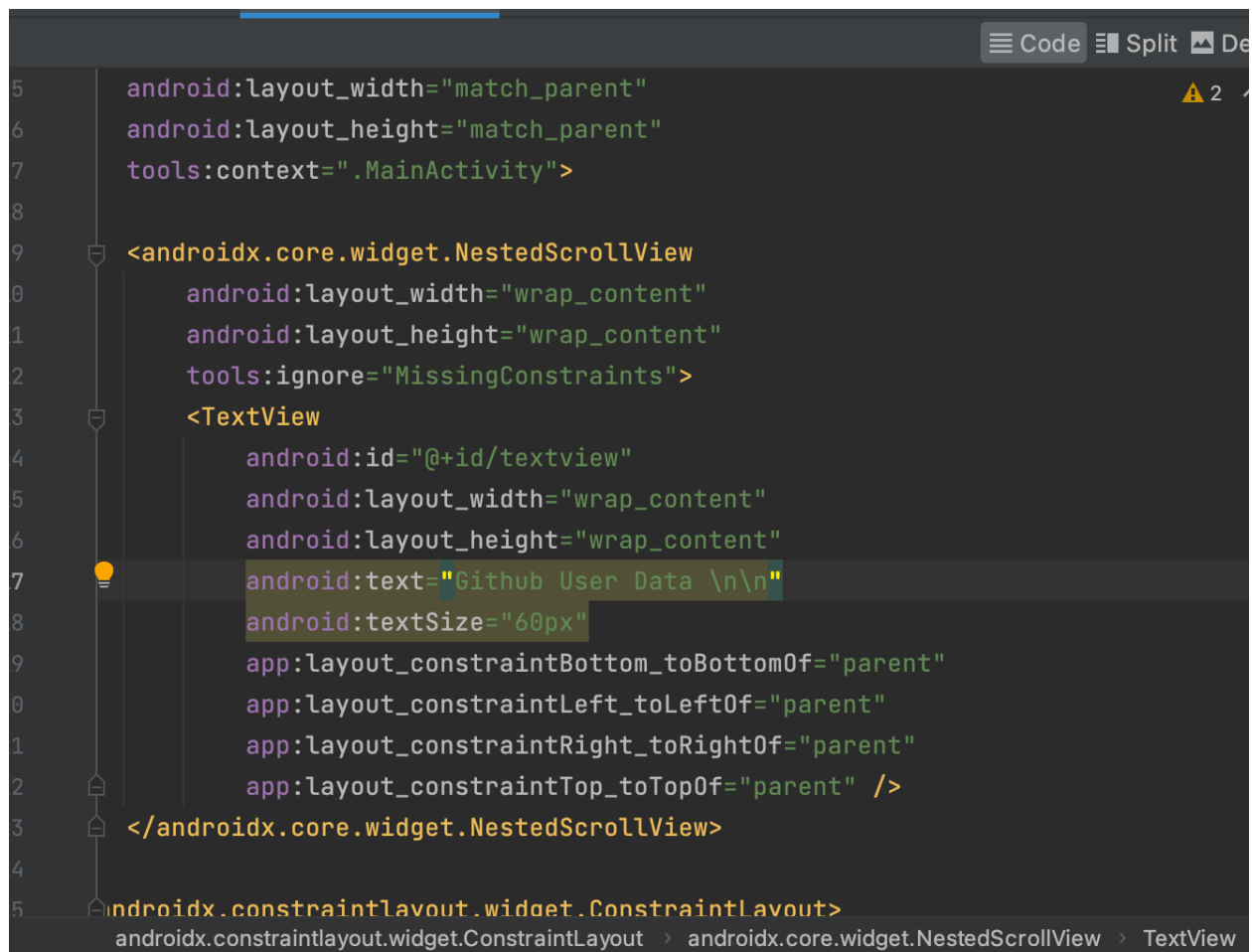
    public void setId(int id) {
        this.id = id;
    }

    public int getId() {
        return id;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getUsername() {
        return username;
    }
}
```

6. Create activity_main.xml as below



```
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <androidx.core.widget.NestedScrollView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         tools:ignore="MissingConstraints">
13         <TextView
14             android:id="@+id/textview"
15             android:layout_width="wrap_content"
16             android:layout_height="wrap_content"
17             android:text="Github User Data \n\n"
18             android:textSize="60px"
19             app:layout_constraintBottom_toBottomOf="parent"
20             app:layout_constraintLeft_toLeftOf="parent"
21             app:layout_constraintRight_toRightOf="parent"
22             app:layout_constraintTop_toTopOf="parent" />
23     </androidx.core.widget.NestedScrollView>
24
25 </androidx.constraintlayout.widget.ConstraintLayout>
```

androidx.constraintlayout.widget.ConstraintLayout > androidx.core.widget.NestedScrollView > TextView

7. We can use retrofit as below

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://api.github.com/")
    .addConverterFactory(GsonConverterFactory.create())
    .build();
```

```
ApiCollections apiCollections = retrofit.create(ApiCollections.class);

//Getting github Users Data
Call<List<GithubUser>> usersCall = apiCollections.getData();

usersCall.enqueue(new Callback<List<GithubUser>>() {
```

8. If the API call is successful we can check it as below and iterate over all the objects received

```
if (response.isSuccessful()) {  
    List<GithubUser> allUsers = response.body();  
  
    for (GithubUser user: allUsers) {
```

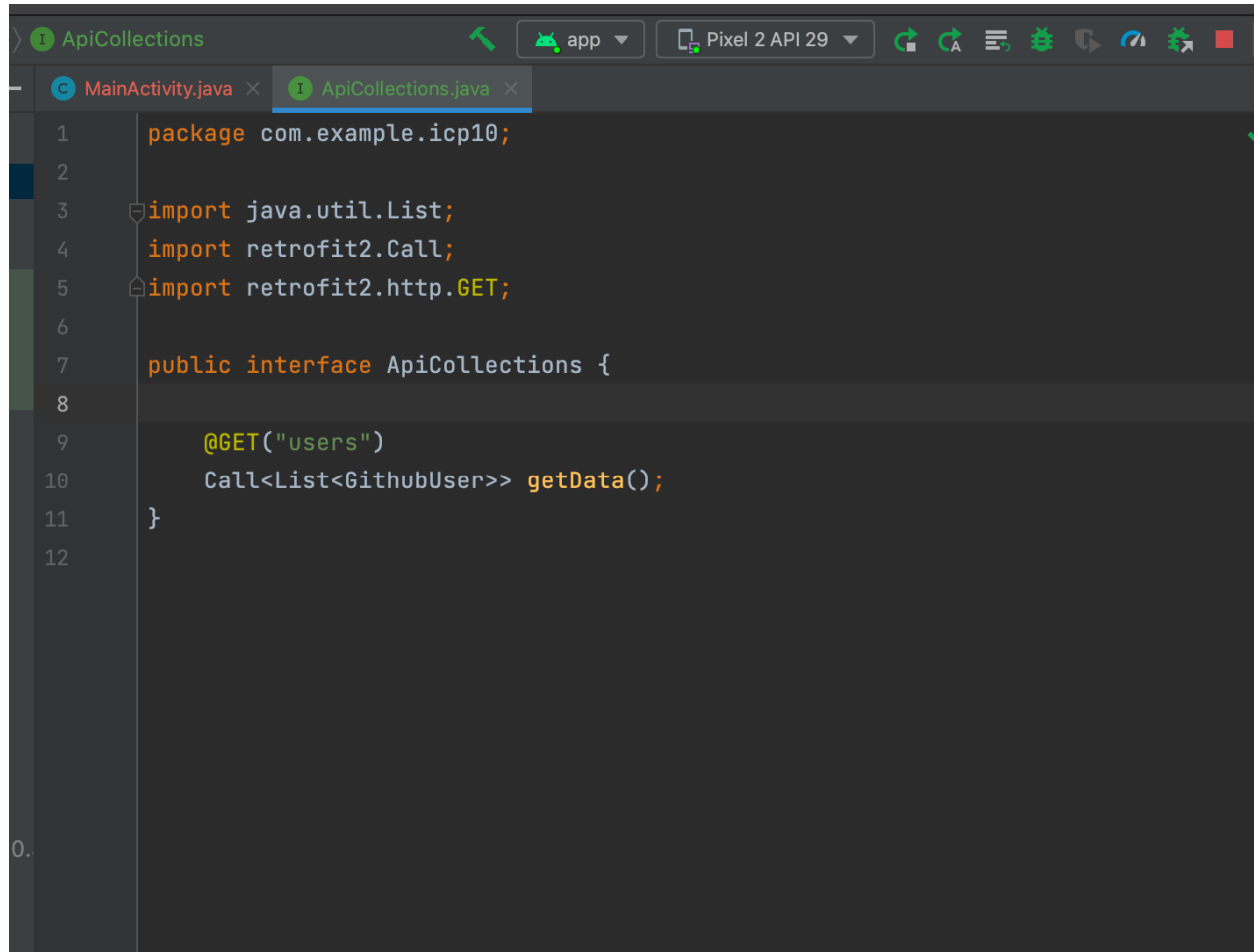
9. We can append the user data which is user_id and user_name as below and display it on phone as below

```
//appending User data  
userData += "User_ID: " + user.getId() + "\n";  
  
userData += "User_Name: " + user.getUsername() + "\n\n";  
  
//appending User data to display view  
textView.append(userData);
```

10. We can also write error responses as below

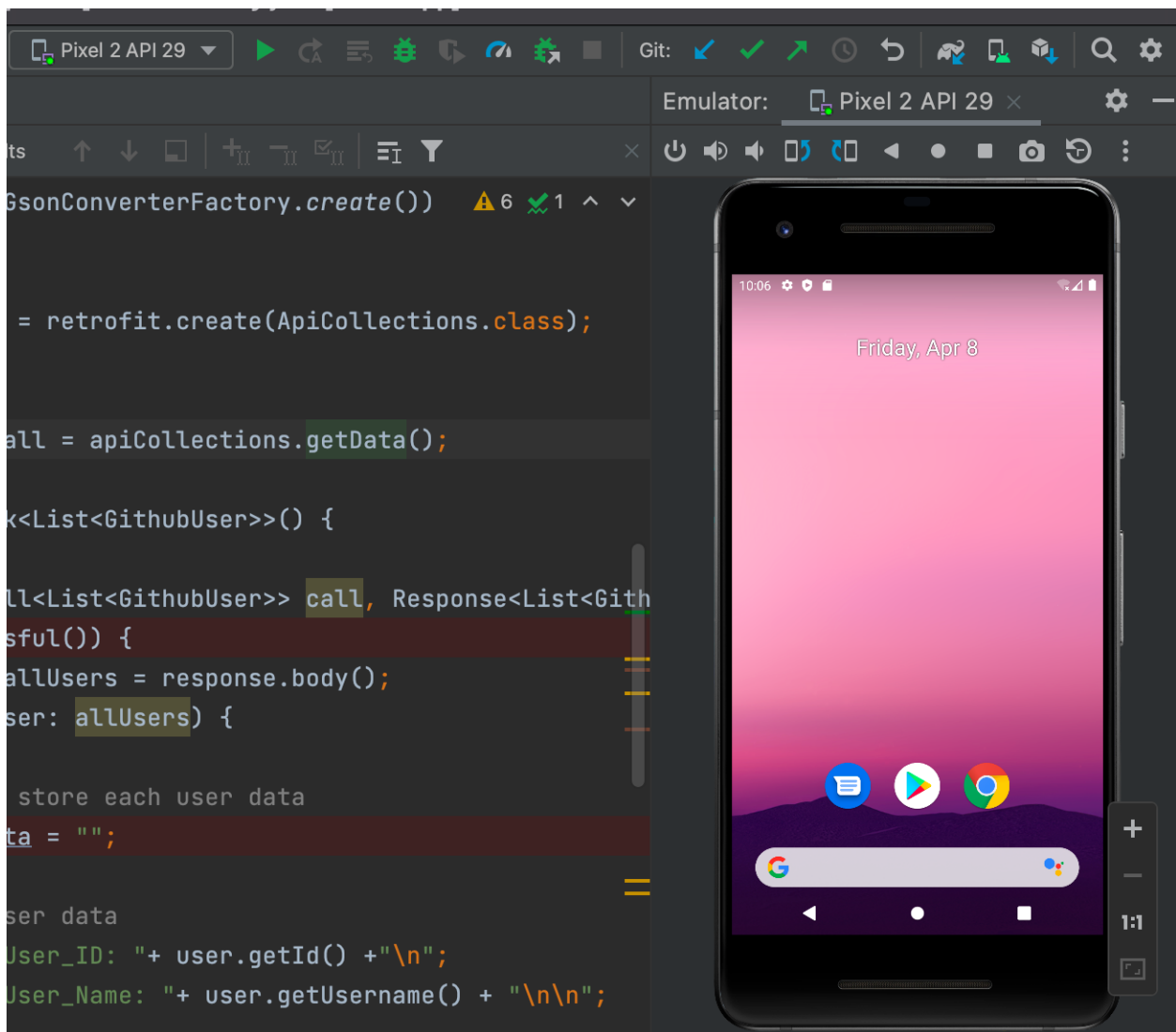
```
@Override  
public void onFailure(Call<List<GithubUser>> call, Throwable t) {  
    Toast.makeText(MainActivity.this, "data Failed", Toast.LENGTH_SHORT);  
}
```

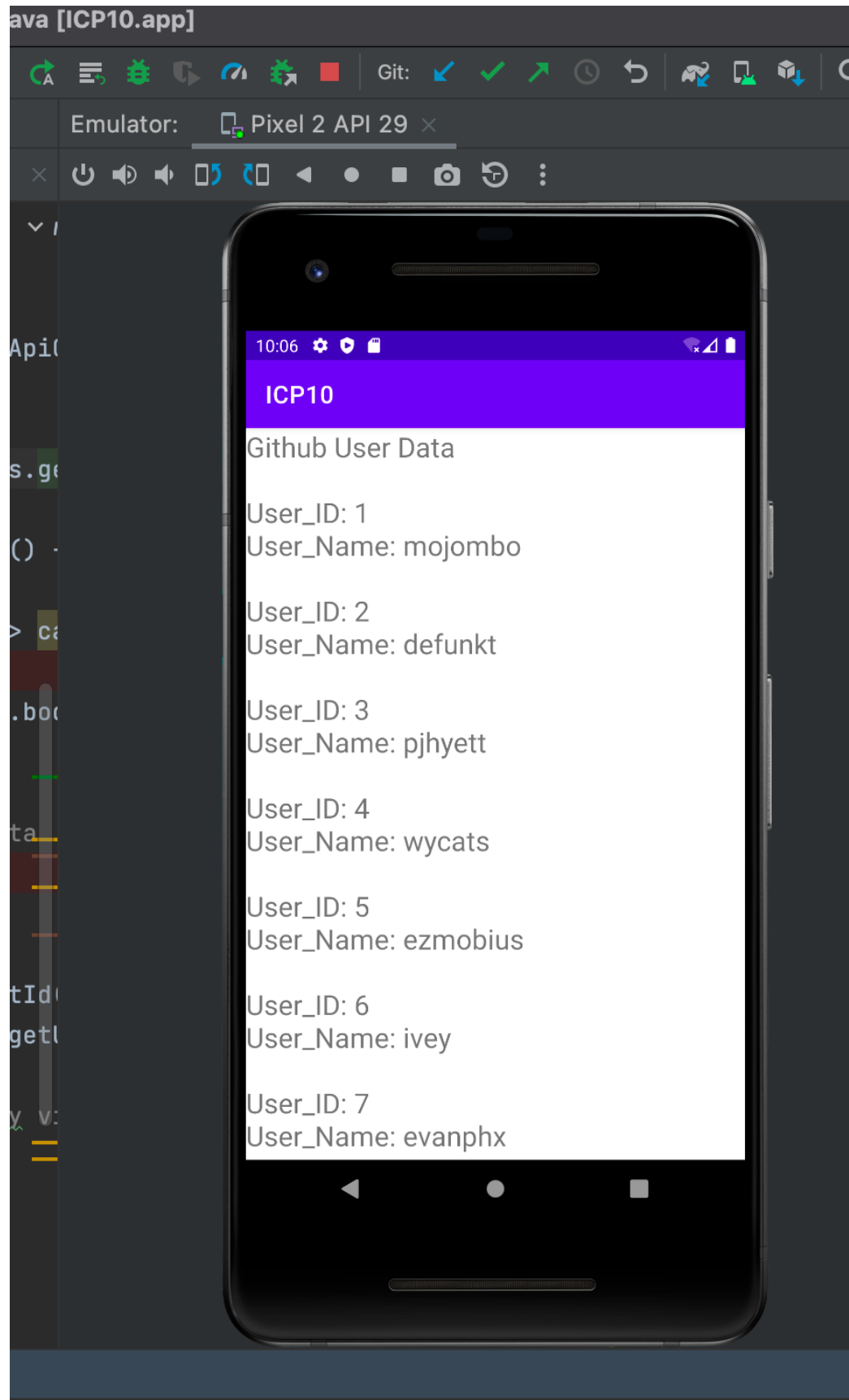
11. We should create ApiCollections interface as below to perform GET api call

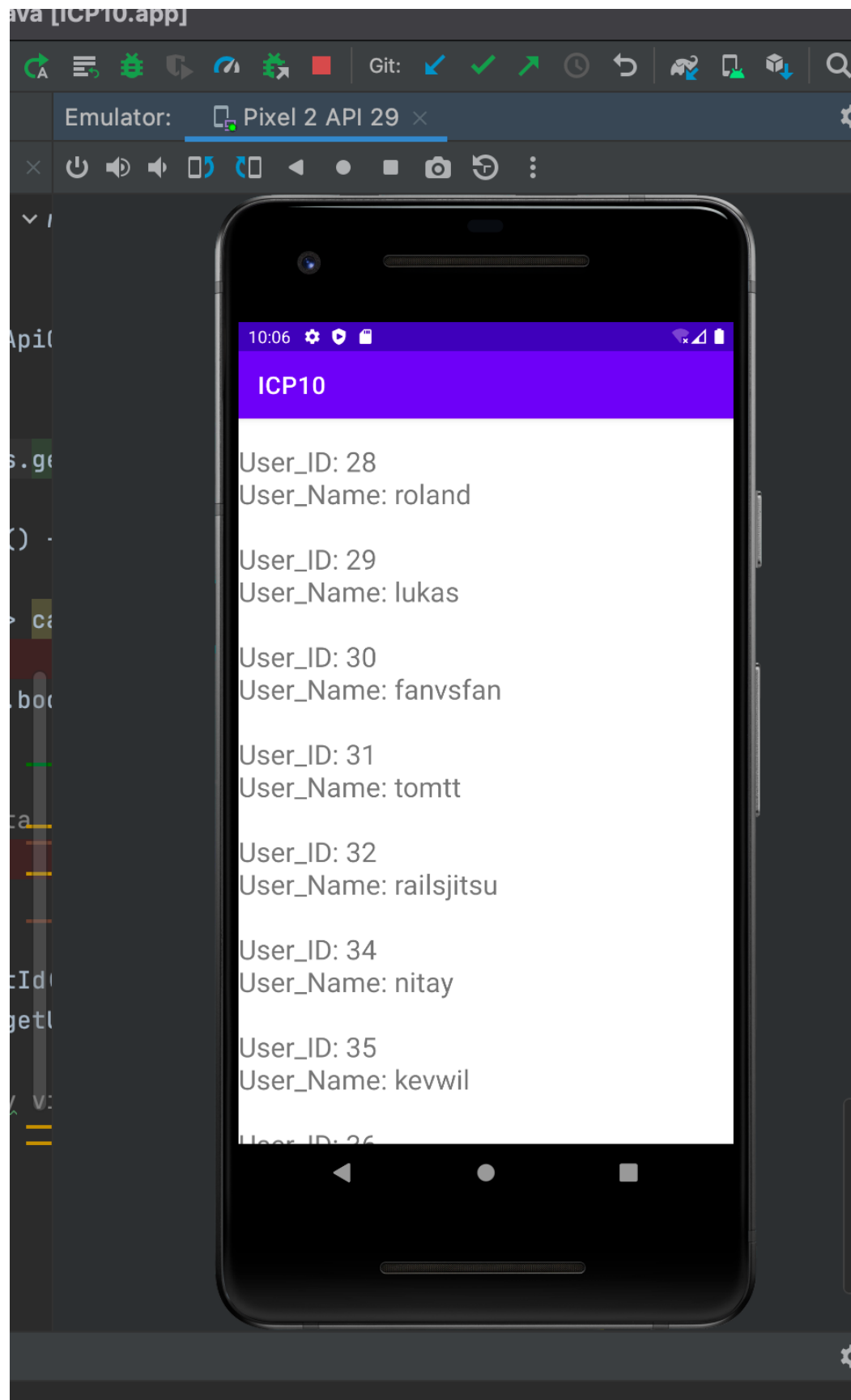


```
1 package com.example.icp10;
2
3 import java.util.List;
4 import retrofit2.Call;
5 import retrofit2.http.GET;
6
7 public interface ApiCollections {
8
9     @GET("users")
10     Call<List<GithubUser>> getData();
11 }
12
```

12. Now lets run the project and see the output







Contribution:

Both of us had performed tasks equally

Conclusion:

It's fun to learn Android Mobile app development using Android Studio. We have learned to various things like providing internet access to our app, performing GET operations, projecting users data.

Challenges:

Faced some difficulties while providing internet access to our app, but we have figured out later.