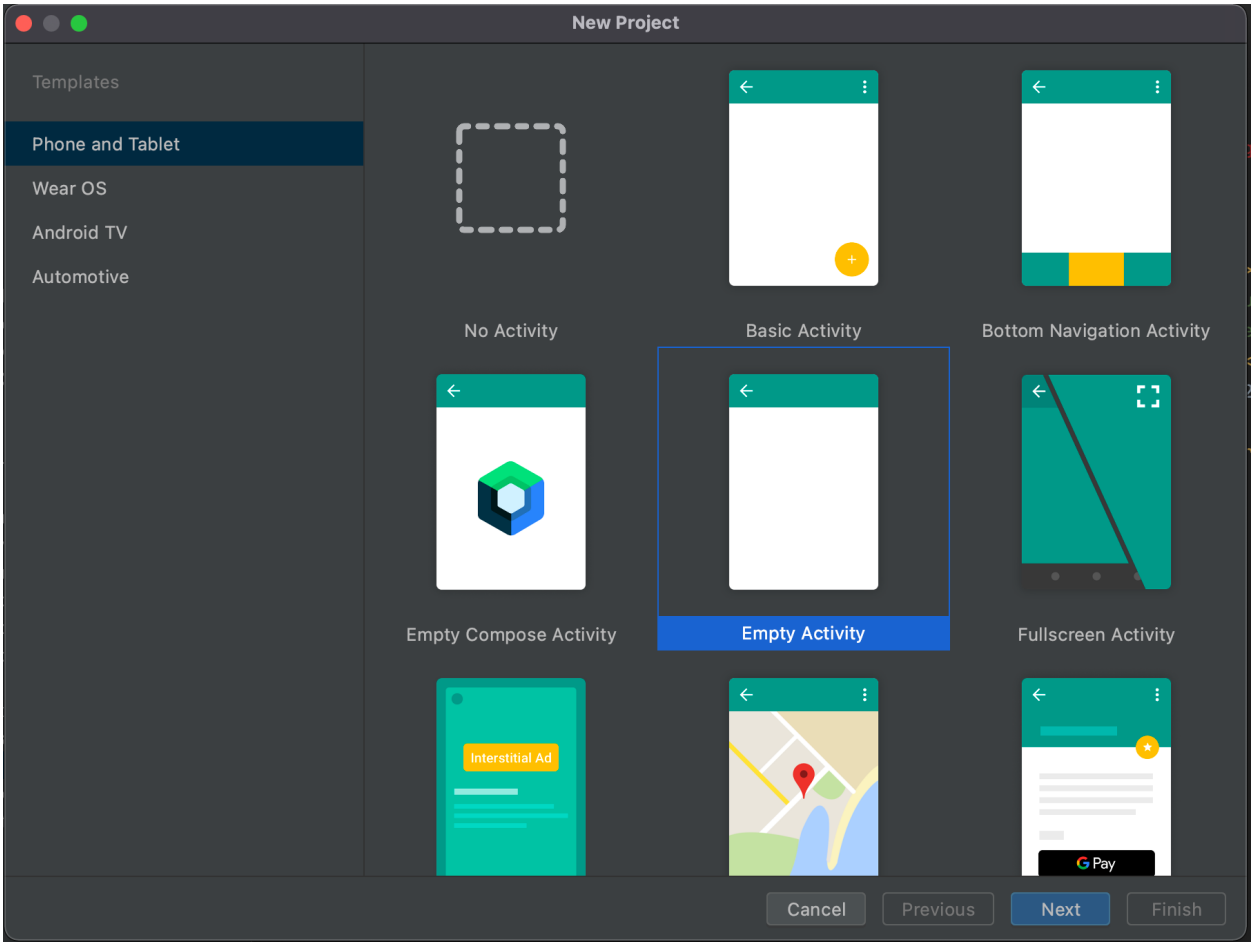
ICP10

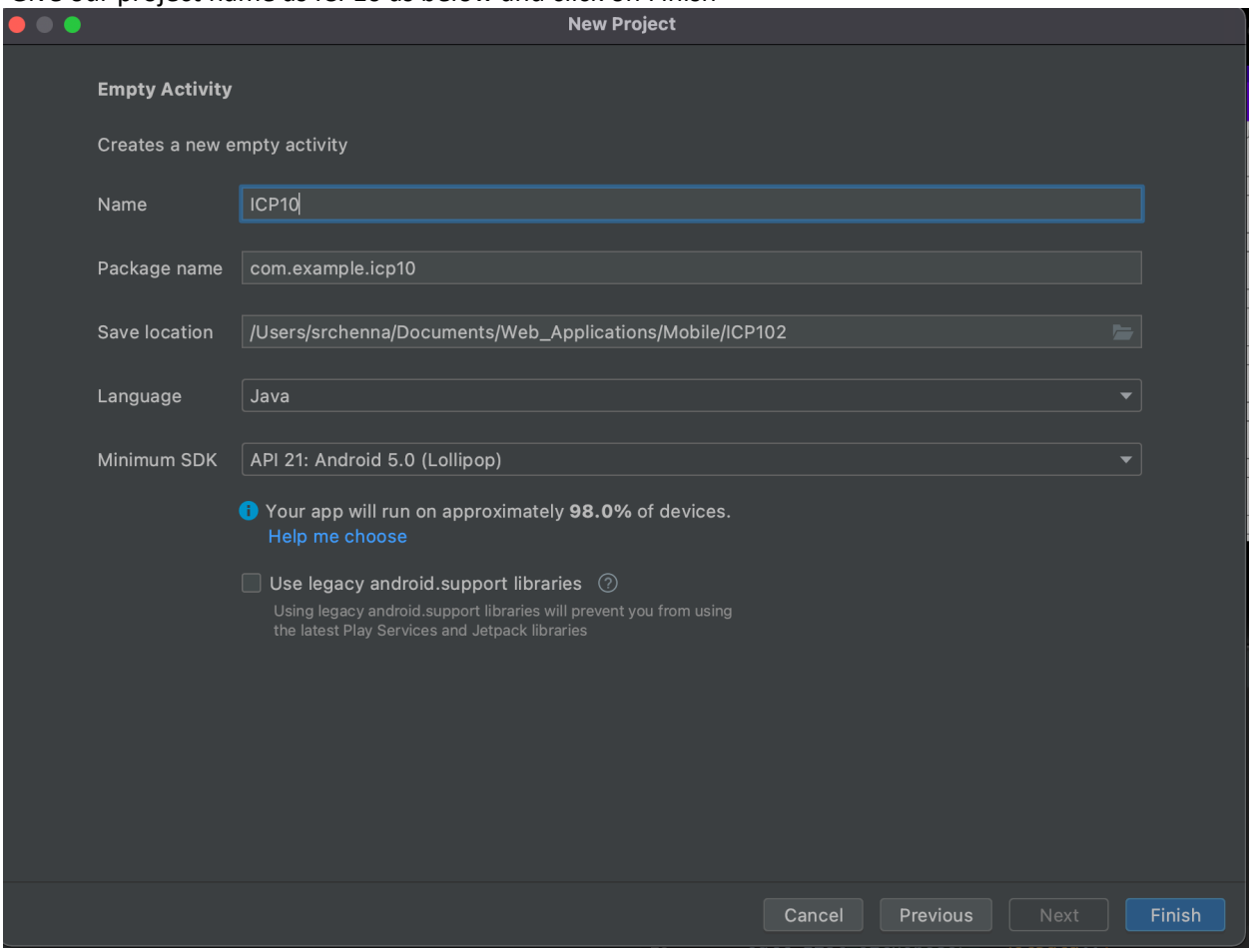
CHENNA SRINIVAS – 16307995
https://github.com/sc7vc/Web_Applications/tree/main/Mobile/ICP10

MAMJALA MANISHA – 16307984
https://github.com/mmg6m/Web/tree/main/Mobile/ICP10

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



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

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

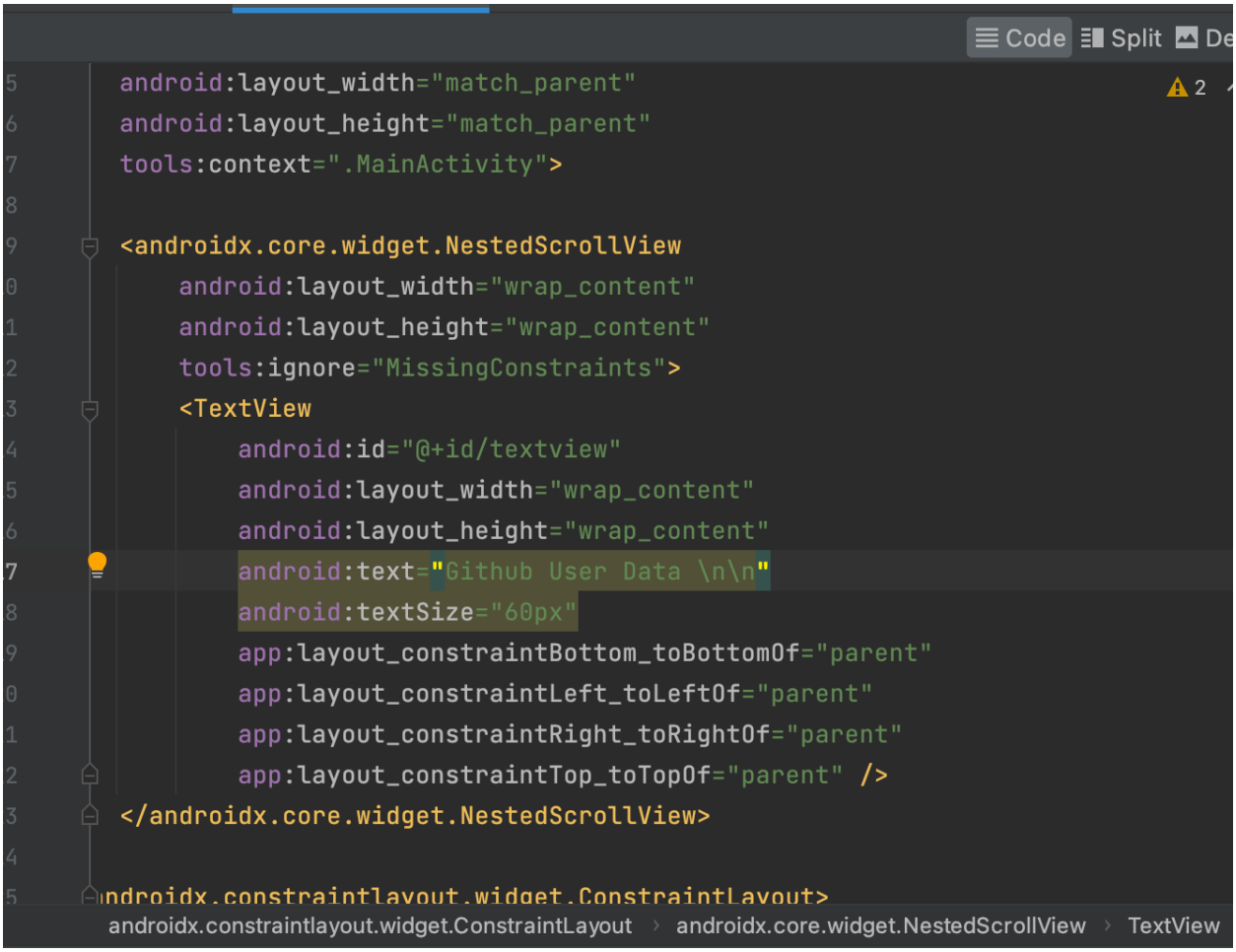
4. In build.gradle add below snippet for retrofit configuration, which converts API calls into objects .

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

5. Create GithubUser.java file as below

```java
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



```
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<androidx.core.widget.NestedScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:ignore="MissingConstraints">
    <TextView
        android:id="@+id/textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Github User Data \n\n"
        android:textSize="60px"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.core.widget.NestedScrollView>

androidx.constraintlayout.widget.ConstraintLayout
```

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) {
```

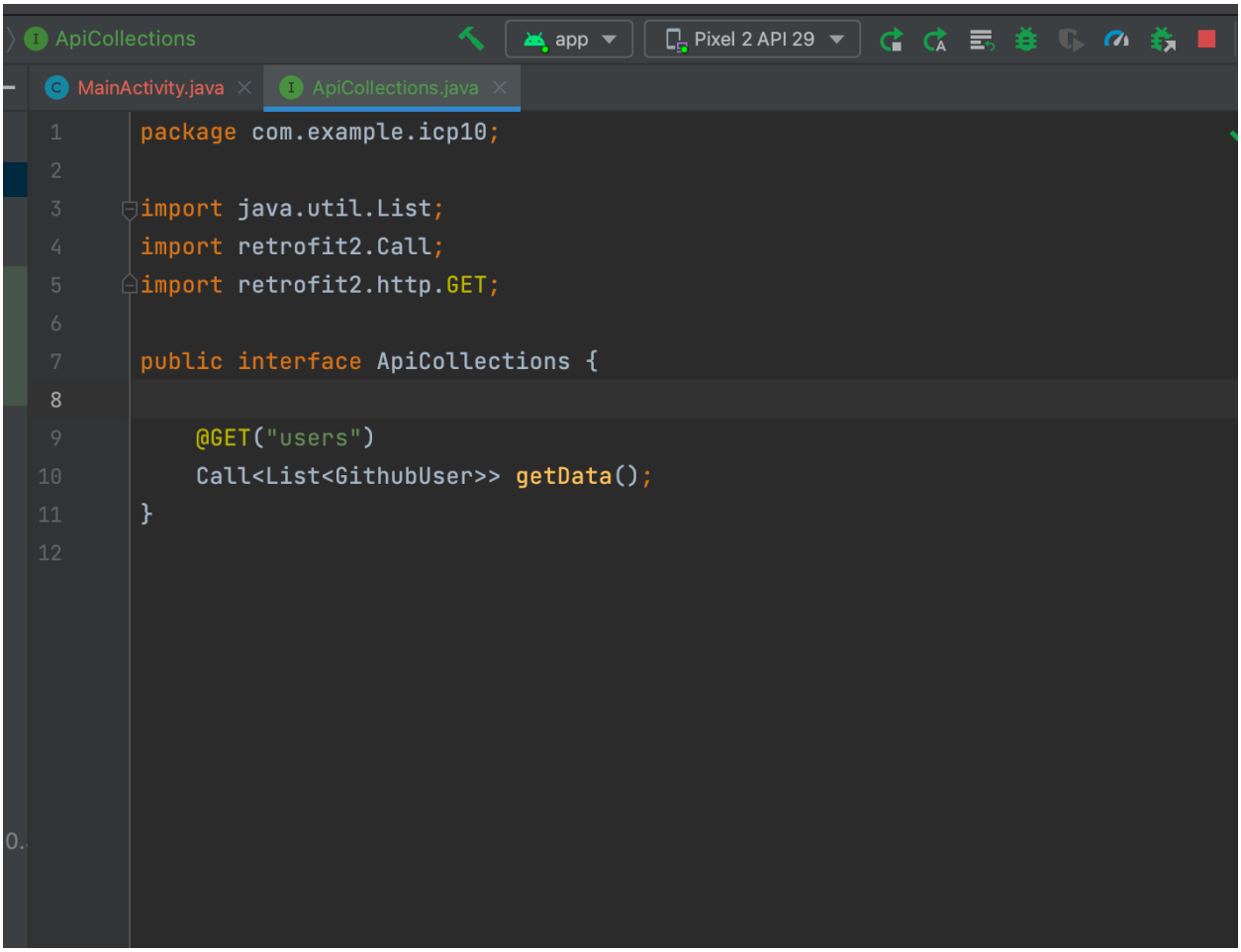
9. We can append the user data which is user_id and user_name as below and display it on phone as below

```
//appending User data
userData += "User_ID: "+ user.getId() +"\n";

userData += "User_Name: "+ user.getUsername() + "\n\n";


//appending User data to dissplay view
textView.append(userData);
```

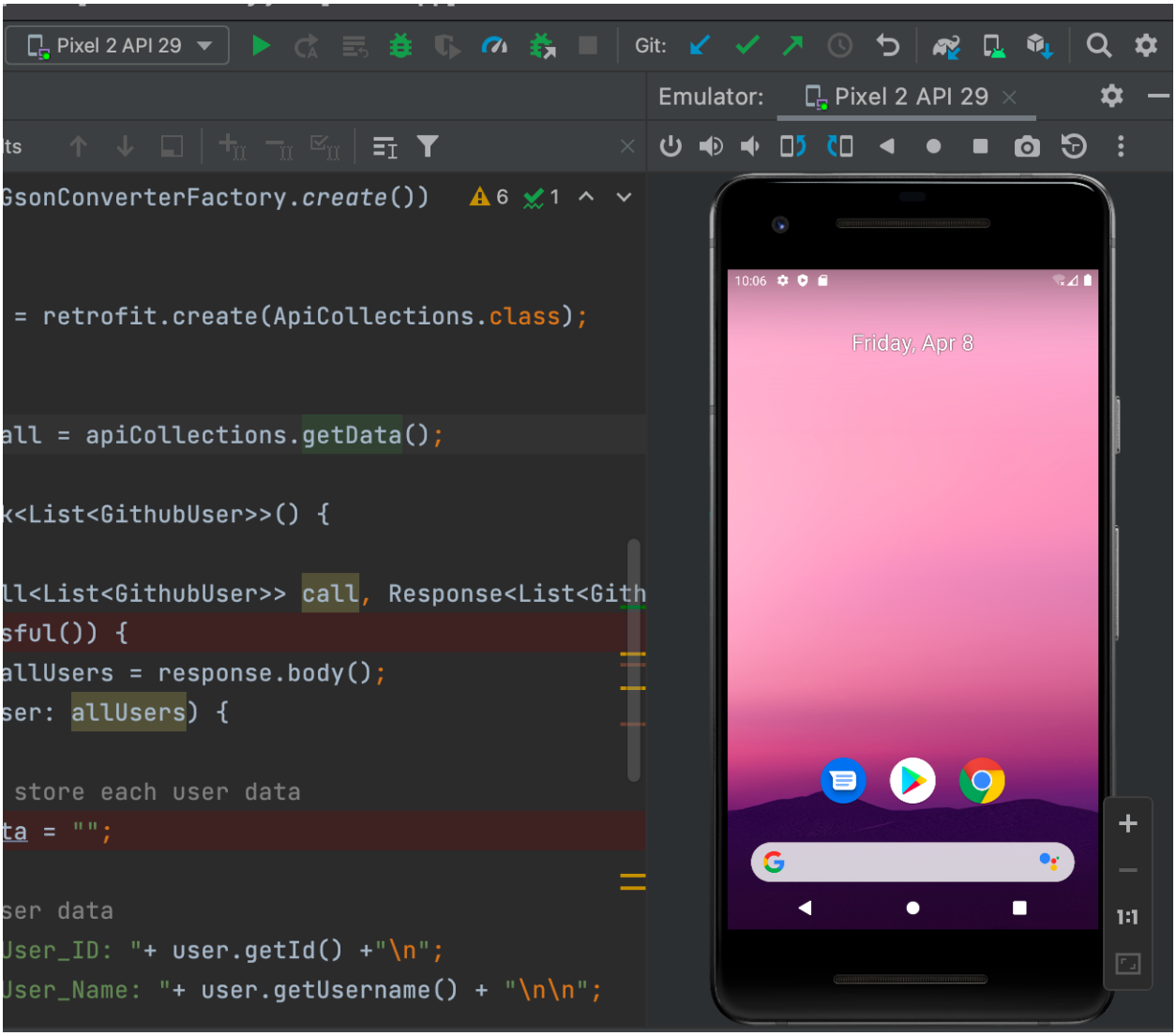
10. We can also write error responses as below

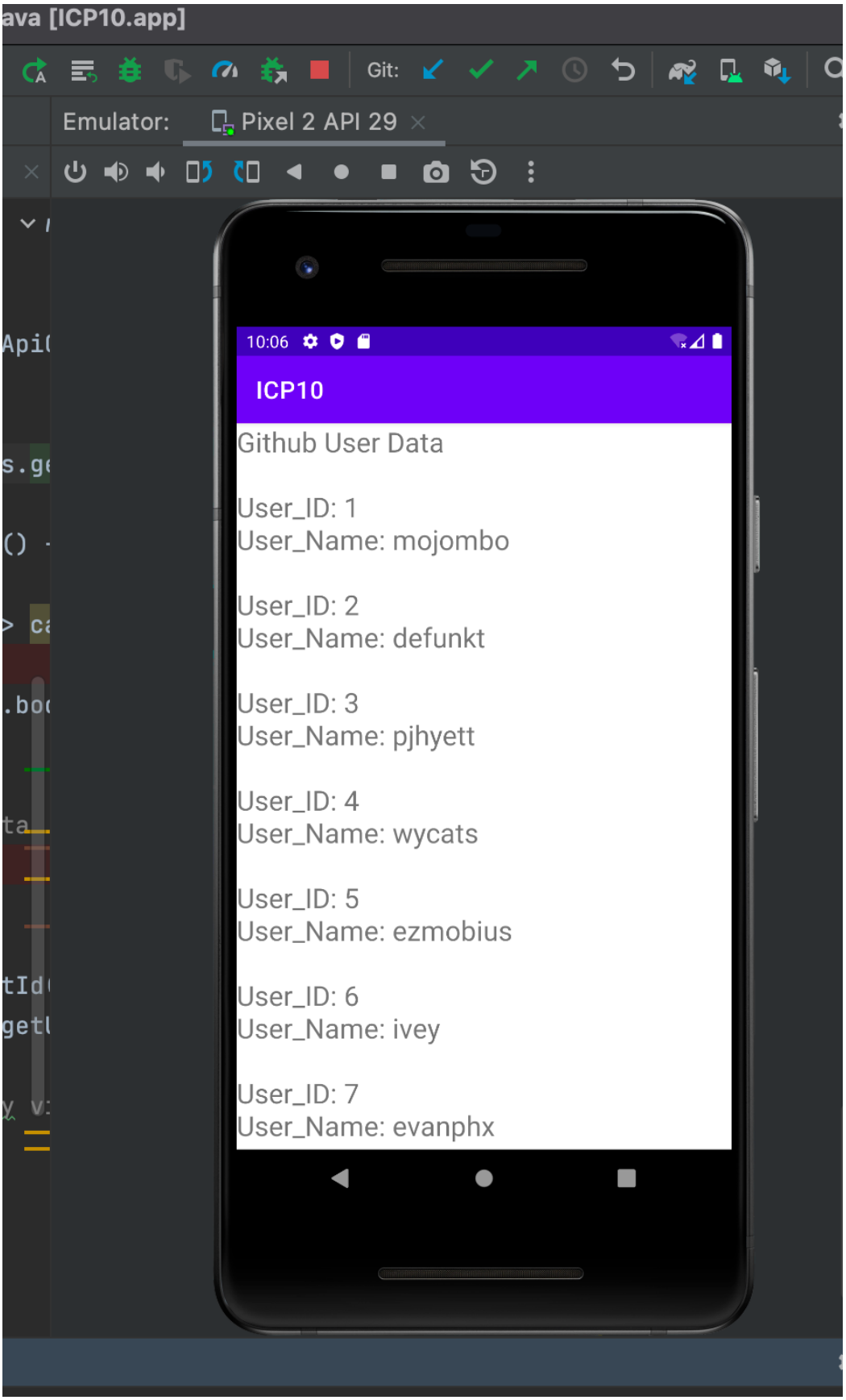
```
@Override
public void onFailure(Call<List<GithubUser>> call, Throwable t) {
    Toast.makeText(MainActivity.this, "data Failed", Toast.LENGTH_SHORT);
}
```

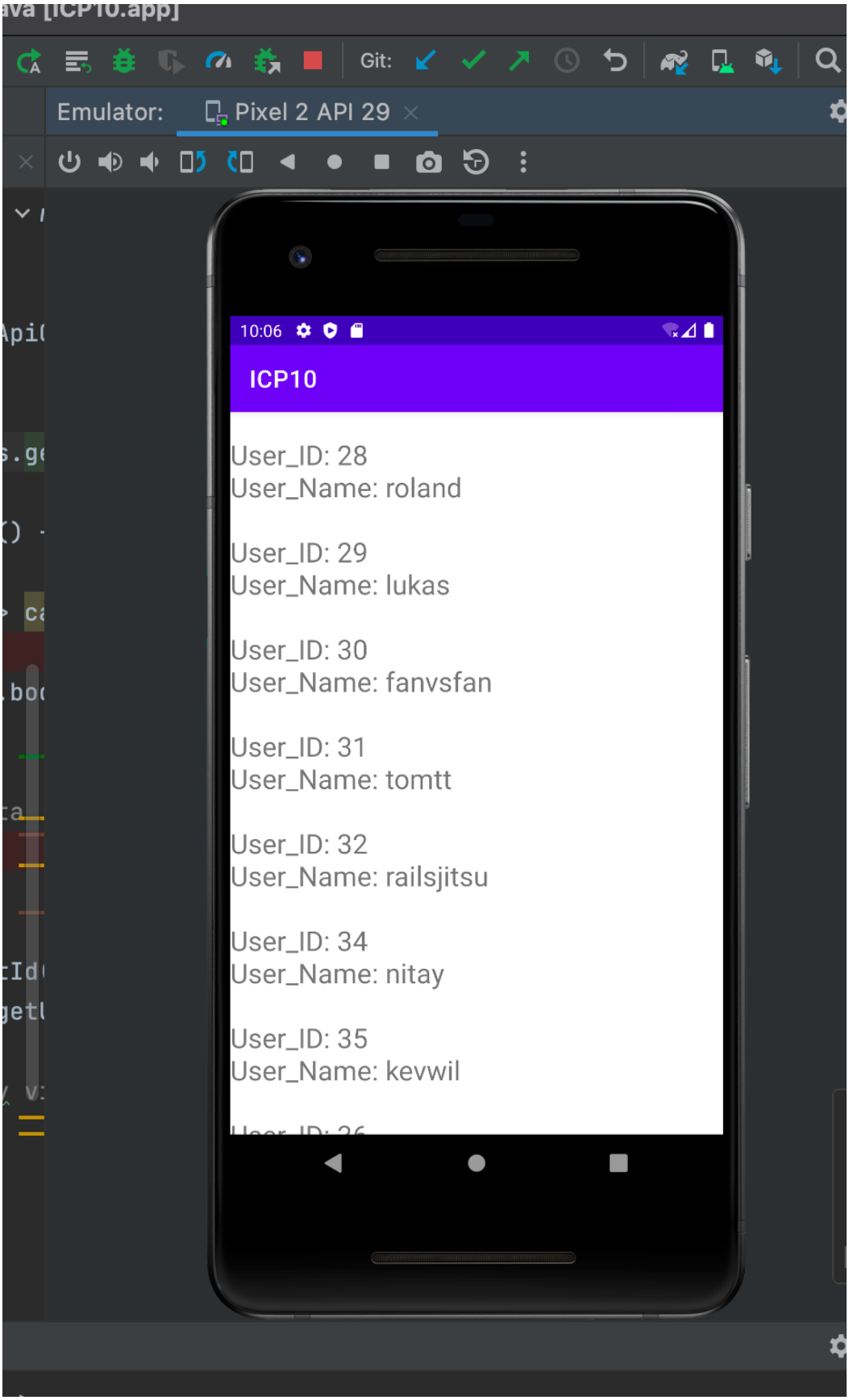
11. We should create ApiCollections interface as below to perform GET api call



12. Now lets run the project and see the output




ava [ICP10.app]
Git:
Emulator:
Pixel 2 API 29
10:06
ICP10
Github User Data
User_ID: 1
User_Name: mojombo
User_ID: 2
User_Name: defunkt
User_ID: 3
User_Name: pjhyett
User_ID: 4
User_Name: wycats
User_ID: 5
User_Name: ezmobius
User_ID: 6
User_Name: ivey
User_ID: 7
User_Name: evanphx


Emulator: Pixel 2 API 29
10:06
ICP10
User_ID: 28
User_Name: roland
User_ID: 29
User_Name: lukas
User_ID: 30
User_Name: fanvsfan
User_ID: 31
User_Name: tomtt
User_ID: 32
User_Name: railsjitsu
User_ID: 34
User_Name: nitay
User_ID: 35
User_Name: kevwil

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.