ICP – 10 (RESTful services, List View, Adapter, and Recycling)

Team details:

Sarath Chandra Kolisetty – sk83g@umsystem.edu

Github link: https://github.com/sarath98-lab/spring-2022/tree/main/Mobile/ICP10/ICP10

Eeshwara Sai Tota – ettkv@umsystem.edu

Github link: https://github.com/SaiKicks/WebMobile-Spring2022/tree/main/mobile/ICP10

Description:

In this ICP, we return Github user ID and userame from the API https://api.github.com using Android Studio.

Procedure:

Following files have been manipulated to achieve the desired result.

1. **AndroidManifest.xml**

   Permission request has been added in this file to get required access from the internet. Following lines of code are included in this file,

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

2. **Build.graddle**

   We add the retrospect configuration to the project in this file. The following lines are added to achieve configuration.

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

3. **User.java**

   User class is created with two private variables id and username. Since the variables are retrieved from JSON, the names of variables are written same as the JSON keys. Annotations can be given to provide alternate names. The class is shown below.

```java
public class User {

    private int id;

    @SerializedName("login")
    private String userName;

    public int getId() { return id; }

    public String getUserName() { return userName; }
}
```

4. **ApiCollections.java**

An interface ApiCollections is created for making GET request. So, the GET annotation is used upon the method and required parameter to be fetched from Github is sent along. Here we are also recycling and list functionality as we are calling data in form of array. The obtained output will be in the form of JSON.

```java
package com.example.icp10;

import ...

public interface ApiCollections {

    @GET("users")
    Call<List<User>> getData();

}
```

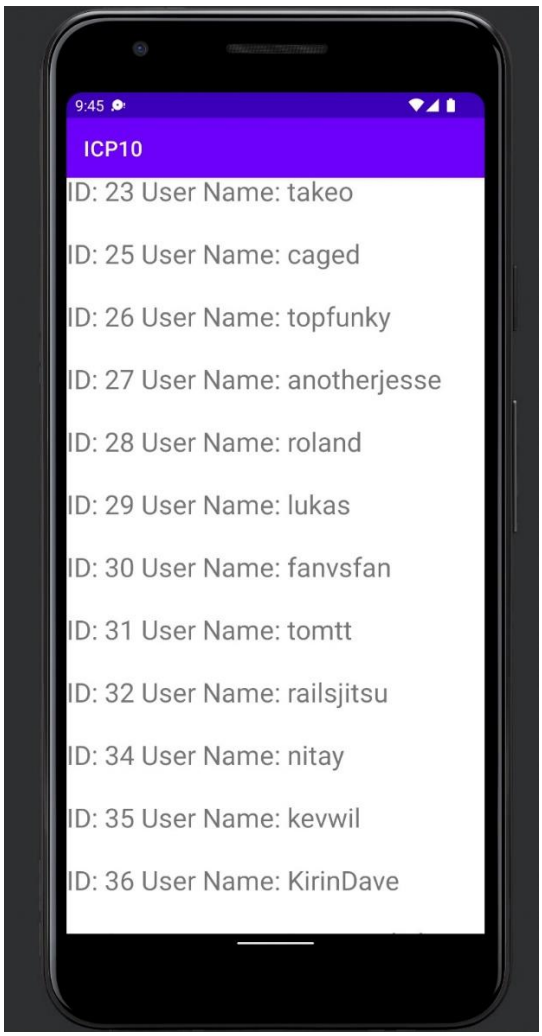
5. **MainActivity.java**

(a) Firstly, a TextView variable textview is created.
(b) Then, retrofit builder is created. In which gihub URL is passed. Add Converter Factory is also added for deserializing response using GSON.
(c) An instance is created for ApiCollections class.

(d) Retrofit is used to download and process the API data and send data to UI using orResponse or orFailure.
(e) ID and name are displayed using the textview field.

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textView = findViewById(R.id.textView);

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

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

    Call<List<User>> usersCall = apiCollections.getData();

    usersCall.enqueue(new Callback<List<User>>() {
        @Override
        public void onResponse(Call<List<User>> call, Response<List<User>> response) {
            if(response.isSuccessful()){

                List<User> users = response.body();

                for(User user: users) {

                    String data = "";

                    data += "ID: " + user.getId() + " ";
                    data += "User Name: " + user.getUserName() + "\n\n";

                    textView.append(data);
                }

            }
        }
```

Output Page



Learning from the lesson

In this ICP, we have learned how to use REST API in the Android studio.

Contribution

Equal Contribution.