

PRACTICAL: 11

AIM:

Creating an application that provides sign in to Google using your Google Account.

THEORY:

Google+ sign-in lets users sign in to your Android app with their existing Google account and get their profile information like name, email, profile pic and other details. By integrating google plus login in your apps, you can get all the user details in one shot. Not only login, you can do other things like posting to their g+ account, getting list of circles, friends list and lot more. The major advantage of integrating G+ login is, you can drive more users to your app by providing quicker & easiest way of registration process.

CODE:

Program: 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:padding="16dp"
    tools:context=".MainActivity" >

    <LinearLayout
        android:id="@+id/llProfile"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:orientation="horizontal"
        android:weightSum="3"
        android:visibility="gone">

        <ImageView
            android:id="@+id/imgProfilePic"
            android:layout_width="80dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:orientation="vertical"
```

```
        android:layout_weight="2" >

        <TextView
            android:id="@+id/txtName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="5dp"
            android:textSize="20dp" />

        <TextView
            android:id="@+id/txtEmail"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="5dp"
            android:textSize="18dp" />
    </LinearLayout>
</LinearLayout>

<com.google.android.gms.common.SignInButton
    android:id="@+id/btn_sign_in"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"/>

<Button
    android:id="@+id/btn_sign_out"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/btn_logout_from_google"
    android:visibility="gone"
    android:layout_marginBottom="10dp"/>

<Button
    android:id="@+id/btn_revoke_access"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/btn_revoke_access"
    android:visibility="gone" />

</LinearLayout>
```

Program: MainActivity.java

```
package com.example.prac_11;

import android.app.ProgressDialog;
import android.content.Intent;
```

```
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.bumptech.glide.Glide;
import com.bumptech.glide.load.engine.DiskCacheStrategy;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.OptionalPendingResult;
import com.google.android.gms.common.api.ResultCallback;
import com.google.android.gms.common.api.Status;
public class MainActivity extends AppCompatActivity implements
    View.OnClickListener,
    GoogleApiClient.OnConnectionFailedListener{

    private static final String TAG = MainActivity.class.getSimpleName();
    private static final int RC_SIGN_IN = 007;
    private GoogleApiClient mGoogleApiClient;
    private ProgressDialog mProgressDialog;
    private SignInButton btnSignIn;
    private Button btnSignOut, btnRevokeAccess;
    private LinearLayout llProfileLayout;
    private ImageView imgProfilePic;
    private TextView txtName, txtEmail;

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

        btnSignIn = (SignInButton) findViewById(R.id.btn_sign_in);
```

```
btnSignIn = (Button) findViewById(R.id.btn_sign_in);
btnRevokeAccess = (Button) findViewById(R.id.btn_revoke_access);
llProfileLayout = (LinearLayout) findViewById(R.id.llProfile);
imgProfilePic = (ImageView) findViewById(R.id.imgProfilePic);
txtName = (TextView) findViewById(R.id.txtName);
txtEmail = (TextView) findViewById(R.id.txtEmail);

btnSignIn.setOnClickListener(this);
btnSignIn.setOnClickListener(this);
btnRevokeAccess.setOnClickListener(this);

GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestEmail()
    .build();

mGoogleApiClient = new GoogleApiClient.Builder(this)
    .enableAutoManage(this, this)
    .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
    .build();

// Customizing G+ button
btnSignIn.setSize(SignInButton.SIZE_STANDARD);
btnSignIn.setScopes(gso.getScopeArray());
}

@Override
public void onClick(View v) {
    int id = v.getId();
    switch (id) {
        case R.id.btn_sign_in:
            signIn();
            break;

        case R.id.btn_sign_out:
            signOut();
            break;

        case R.id.btn_revoke_access:
            revokeAccess();
            break;
    }
}
```

```
}  
private void signIn() {  
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);  
    startActivityForResult(signInIntent, RC_SIGN_IN);  
}  
private void signOut() {  
    Auth.GoogleSignInApi.signOut(mGoogleApiClient).setResultCallback(  
        new ResultCallback<Status>() {  
            @Override  
            public void onResult(Status status) {  
                updateUI(false);  
            }  
        });  
}  
private void revokeAccess() {  
    Auth.GoogleSignInApi.revokeAccess(mGoogleApiClient).setResultCallback(  
        new ResultCallback<Status>() {  
            @Override  
            public void onResult(Status status) {  
                updateUI(false);  
            }  
        });  
}  
private void handleSignInResult(GoogleSignInResult result) {  
    Log.d(TAG, "handleSignInResult:" + result.isSuccess());  
    if (result.isSuccess()) {  
        // Signed in successfully, show authenticated UI.  
        GoogleSignInAccount acct = result.getSignInAccount();  
  
        Log.e(TAG, "display name: " + acct.getDisplayName());  
  
        String personName = acct.getDisplayName();  
        String personPhotoUrl = acct.getPhotoUrl().toString();  
        String email = acct.getEmail();  
  
        Log.e(TAG, "Name: " + personName + ", email: " + email  
            + ", Image: " + personPhotoUrl);  
  
        txtName.setText(personName);  
        txtEmail.setText(email);  
        Glide.with(getApplicationContext()).load(personPhotoUrl)  
            .thumbnail(0.5f)
```

```

        .crossFade()
        .diskCacheStrategy(DiskCacheStrategy.ALL)
        .into(imgProfilePic);

        updateUI(true);
    } else {
        // Signed out, show unauthenticated UI.
        updateUI(false);
    }
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        handleSignInResult(result);
    }
}

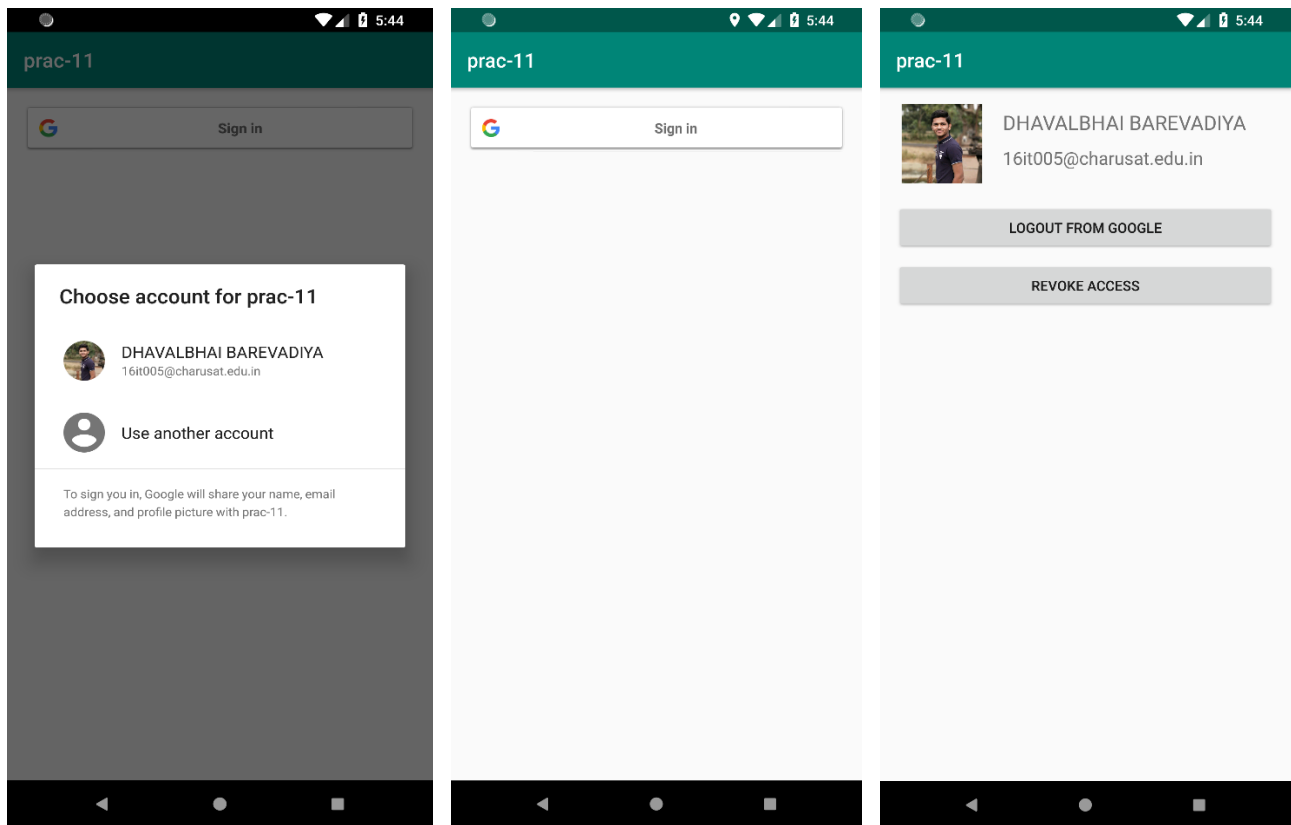
@Override
public void onStart() {
    super.onStart();

    OptionalPendingResult<GoogleSignInResult> opr =
Auth.GoogleSignInApi.silentSignIn(mGoogleApiClient);
    if (opr.isDone()) {
        // If the user's cached credentials are valid, the OptionalPendingResult will be "done"
        // and the GoogleSignInResult will be available instantly.
        Log.d(TAG, "Got cached sign-in");
        GoogleSignInResult result = opr.get();
        handleSignInResult(result);
    } else {
        // If the user has not previously signed in on this device or the sign-in has expired,
        // this asynchronous branch will attempt to sign in the user silently. Cross-device
        // single sign-on will occur in this branch.
        showProgressDialog();
        opr.setResultCallback(new ResultCallback<GoogleSignInResult>() {
            @Override
            public void onResult(GoogleSignInResult googleSignInResult) {
                hideProgressDialog();
                handleSignInResult(googleSignInResult);
            }
        });
    }
}

```

```
        }
    });
}
}
@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    // An unresolvable error has occurred and Google APIs (including Sign-In) will not
    // be available.
    Log.d(TAG, "onConnectionFailed:" + connectionResult);
}
private void showProgressDialog() {
    if (mProgressDialog == null) {
        mProgressDialog = new ProgressDialog(this);
        mProgressDialog.setMessage(getString(R.string.loading));
        mProgressDialog.setIndeterminate(true);
    }

    mProgressDialog.show();
}
private void hideProgressDialog() {
    if (mProgressDialog != null && mProgressDialog.isShowing()) {
        mProgressDialog.hide();
    }
}
private void updateUI(boolean isSignedIn) {
    if (isSignedIn) {
        btnSignIn.setVisibility(View.GONE);
        btnSignOut.setVisibility(View.VISIBLE);
        btnRevokeAccess.setVisibility(View.VISIBLE);
        llProfileLayout.setVisibility(View.VISIBLE);
    } else {
        btnSignIn.setVisibility(View.VISIBLE);
        btnSignOut.setVisibility(View.GONE);
        btnRevokeAccess.setVisibility(View.GONE);
        llProfileLayout.setVisibility(View.GONE);
    }
}
}
```

OUTPUT:**CONCLUSION:**

We have successfully created android application in which we can login using google account.