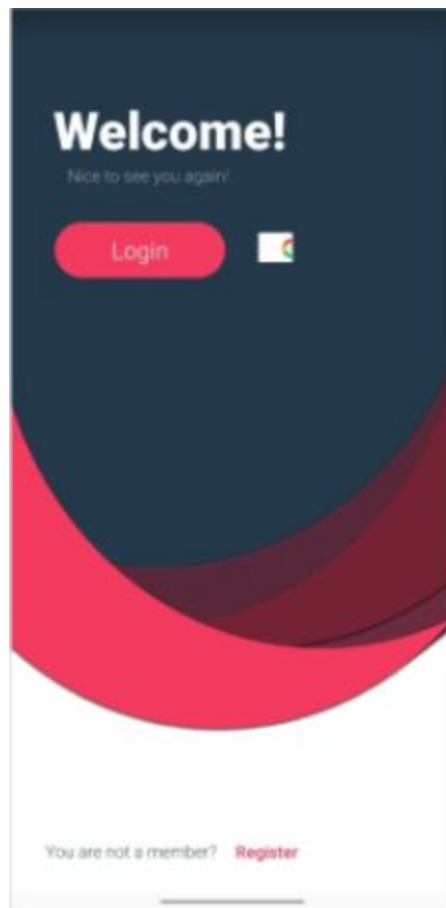


# Bloggging App

## FULL CODE

The following are XML and Java file code used for the App

## 1.welcome Screen



```
package com.example.sapp.Activities;
import android.content.Intent;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import com.example.sapp.R;
import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
```

```

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;

public class MainActivity extends AppCompatActivity {
    private static final int RC_SIGN_IN = 2;
    Button loginButton, reg;
    SignInButton gsign;
    FirebaseAuth mAuth;
    GoogleSignInClient mGoogleSignInClient;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        loginButton = findViewById(R.id.loginButton);
        gsign = findViewById(R.id.googlebtn);
        mAuth = FirebaseAuth.getInstance();
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(MainActivity.this, LoginActivity.class);
                startActivity(intent);
            }
        });

        reg = findViewById(R.id.regb);
        reg.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, RegisterActivity.class);
                startActivity(intent);
            }
        });
        GoogleSignInOptions gso = new
        GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestIdToken(getString(R.string.default_web_client_id))
            .requestEmail()
            .build();
        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

        gsign.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                signIn();
            }
        }
        if(mAuth.getCurrentUser() != null){
            FirebaseUser user = mAuth.getCurrentUser();
            updateUI();
        }
    }

    private void signIn() {

```

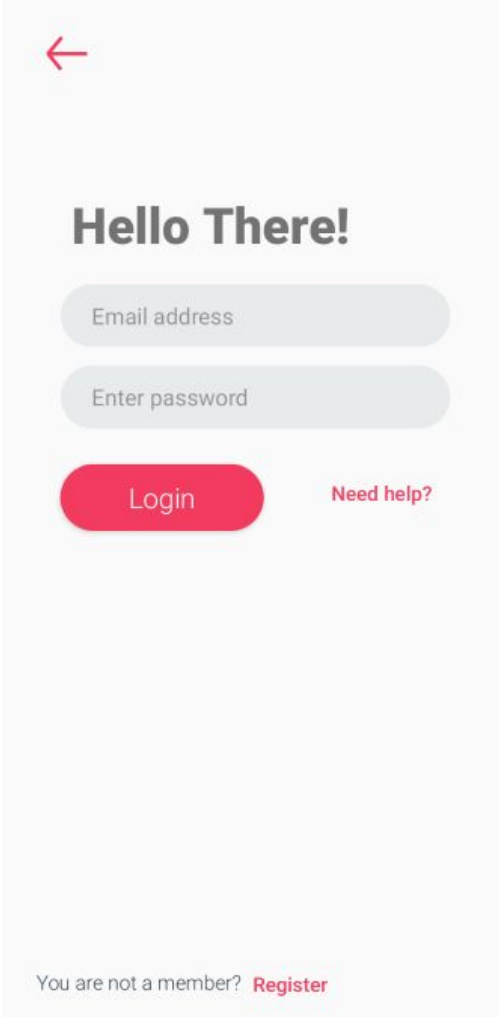
```

        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
        startActivityForResult(signInIntent, RC_SIGN_IN);
    }
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == RC_SIGN_IN) {
            Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
            try {
                GoogleSignInAccount account = task.getResult(ApiException.class);
                firebaseAuthWithGoogle(account);
            } catch (ApiException e) {
                e.printStackTrace();
            }
        }
    }
    private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
        Log.d("TAG", "firebaseAuthWithGoogle:" + acct.getId());
        AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);
        mAuth.signInWithCredential(credential)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        // Sign in success, update UI with the signed-in user's information
                        Log.d("TAG", "signInWithCredential:success");
                        FirebaseUser user = mAuth.getCurrentUser();
                        updateUI();
                    } else {
                        // If sign in fails, display a message to the user.
                        Log.w("TAG", "signInWithCredential:failure", task.getException());
                        Toast.makeText(MainActivity.this, "Falied", Toast.LENGTH_SHORT).show();
                    }
                }
            });
    }

    private void updateUI() {
        Intent intent = new Intent(MainActivity.this, Homedrawer.class);
        startActivity(intent);
        finish();
    }
    @Override
    public void onStart() {
        super.onStart();
        FirebaseUser user = mAuth.getCurrentUser();
        if(user!=null)
        {
            updateUI();
        }
    }
}

```

## 2.Login IN



A mobile app login screen with a light gray background. At the top left is a red back arrow. Below it is the text "Hello There!" in bold. There are two rounded rectangular input fields: the first is labeled "Email address" and the second is labeled "Enter password". Below the input fields is a red rounded rectangular button labeled "Login". To the right of the "Login" button is the text "Need help?". At the bottom of the screen is the text "You are not a member? Register", where "Register" is in red.

```
package com.example.sapp.Activities;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.example.sapp.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
```

```

import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
public class LoginActivity extends AppCompatActivity {
    ImageView backbtn;
    Button reg;
    private EditText userMail,userPassword;
    private Button btnLogin;
    private FirebaseAuth mAuth;
    private Intent HomeActivity;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        backbtn=findViewById(R.id.backbtn);
        backbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
        userMail = findViewById(R.id.editText);
        userPassword = findViewById(R.id.editText2);
        btnLogin = findViewById(R.id.loginBtn);
        mAuth=FirebaseAuth.getInstance();
        HomeActivity=new Intent(this, com.example.sapp.Activities.Homedrawer.class);
        reg=findViewById(R.id.regb);
        reg.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {

                Intent intent = new Intent(LoginActivity.this, RegisterActivity.class);
                startActivity(intent);
            }
        });
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final String mail= userMail.getText().toString();
                final String password= userPassword.getText().toString();
                if (mail.isEmpty() || password.isEmpty()){
                    showMessage("Please fill all the fields");
                }
                else{
                    signIn(mail,password);
                }
            }
        });
    }
    private void signIn(String mail, String password) {
        mAuth.signInWithEmailAndPassword(mail,password).addOnCompleteListener(new
        OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

```

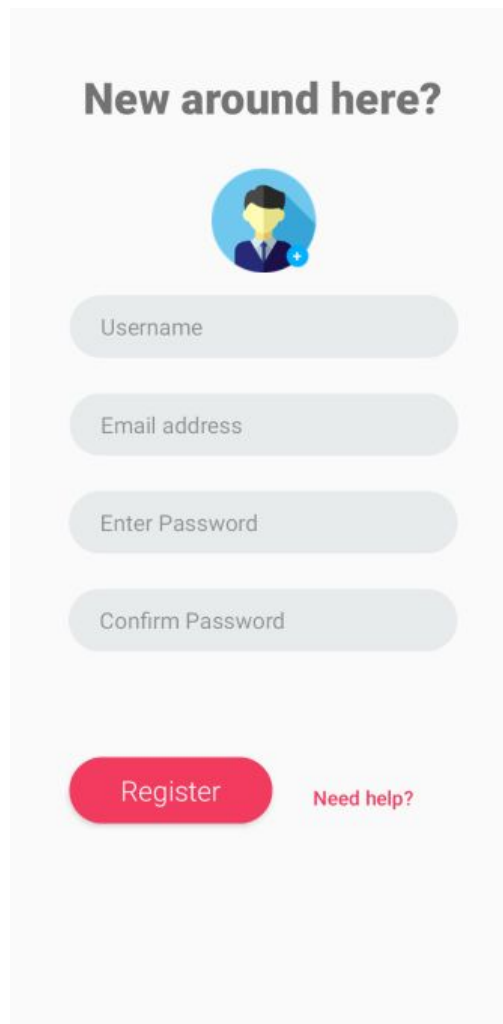
```

        if(task.isSuccessful()){
            updateUI();
        }
        else{
            showMessage(task.getException().getMessage());
        }
    }
    });
}

private void updateUI() {
    startActivity(HomeActivity);
    finish();
}
private void showMessage(String text) {
    Toast.makeText(getApplicationContext(),text,Toast.LENGTH_LONG).show();
}
@Override
protected void onStart(){
    super.onStart();
    FirebaseUser user = mAuth.getCurrentUser();
    if(user!=null)
    {
        updateUI();
    }
}
}

```

### 3. RegisterActivity



The image shows a mobile app registration screen. At the top, the text "New around here?" is displayed in a bold, dark gray font. Below this text is a circular icon featuring a stylized person with dark hair and a blue suit, set against a light blue background with a small white plus sign. Underneath the icon are four rounded rectangular input fields, each with a light gray border and a light gray background. The labels for these fields are "Username", "Email address", "Enter Password", and "Confirm Password", all in a light gray font. At the bottom of the form is a prominent red rounded rectangular button with the word "Register" in white text. To the right of this button is a link labeled "Need help?" in a red font.

```
package com.example.sapp.Activities;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
```



```

import com.example.sapp.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.UserProfileChangeRequest;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
public class RegisterActivity extends AppCompatActivity {
    ImageView ImgUserPhoto;
    static int PRegCode=1;
    static int REQUESCODE=1;
    Uri pickedImgUri;
    private EditText userEmail, userPassword,userPassword2,userName;
    private Button regBtn;
    private FirebaseAuth mAuth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        userEmail=findViewById(R.id.editText4);
        userPassword=findViewById(R.id.editText7);
        userPassword2=findViewById(R.id.editText8);
        userName=findViewById(R.id.editText3);
        regBtn=findViewById(R.id.button);
        mAuth=FirebaseAuth.getInstance();
        regBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                final String email=userEmail.getText().toString();
                final String password=userPassword.getText().toString();
                final String password2=userPassword2.getText().toString();
                final String name=userName.getText().toString();
                if(email.isEmpty() || name.isEmpty() || password.isEmpty() || password2.isEmpty() ||
!password.equals(password2)) {
                    showMessage("Please fill all the fields");
                    regBtn.setVisibility(View.VISIBLE);
                }
                else{
                    //Toast.makeText(getApplicationContext(),"step1",Toast.LENGTH_LONG).show();
                    CreateUserAccount(email,name,password);
                }
            }
        });
        ImgUserPhoto =findViewById(R.id.img);
        ImgUserPhoto.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (Build.VERSION.SDK_INT>=22){
                    checkAndRequestForpermission();
                }
            }
        });
    }
}

```

```

        else
        {
            openGallery();
        }
    }
    });

}

private void CreateUserAccount(String email, final String name, String password) {
    //Toast.makeText(getApplicationContext(),"step2",Toast.LENGTH_LONG).show();
    mAuth.createUserWithEmailAndPassword(email,password)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(task.isSuccessful()){
                    showMessage("Account Created");
                    updateUserInfo(name,pickedImgUri,mAuth.getCurrentUser());
                }
                else{
                    showMessage("Account creation failed");
                }
            }
        });
}

private void updateUserInfo(final String name, Uri pickedImgUri, final FirebaseUser
currentUser){
    StorageReference mStorage =
    FirebaseStorage.getInstance().getReference().child("users_photos");
    final StorageReference imageFilePath=
    mStorage.child(pickedImgUri.getLastPathSegment());
    imageFilePath.putFile(pickedImgUri).addOnSuccessListener(new
    OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            imageFilePath.getDownloadUrl().addOnSuccessListener(new
            OnSuccessListener<Uri>() {
                @Override
                public void onSuccess(Uri uri) {
                    UserProfileChangeRequest profileUpdate = new
                    UserProfileChangeRequest.Builder().setDisplayName(name).setPhotoUri(uri).build();
                    currentUser.updateProfile(profileUpdate).addOnCompleteListener(new
                    OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {

                            if(task.isSuccessful()){
                                showMessage("Register Completed");
                                updateUI();
                            }
                        }
                    });
                }
            }
        }
    });
}

```

```

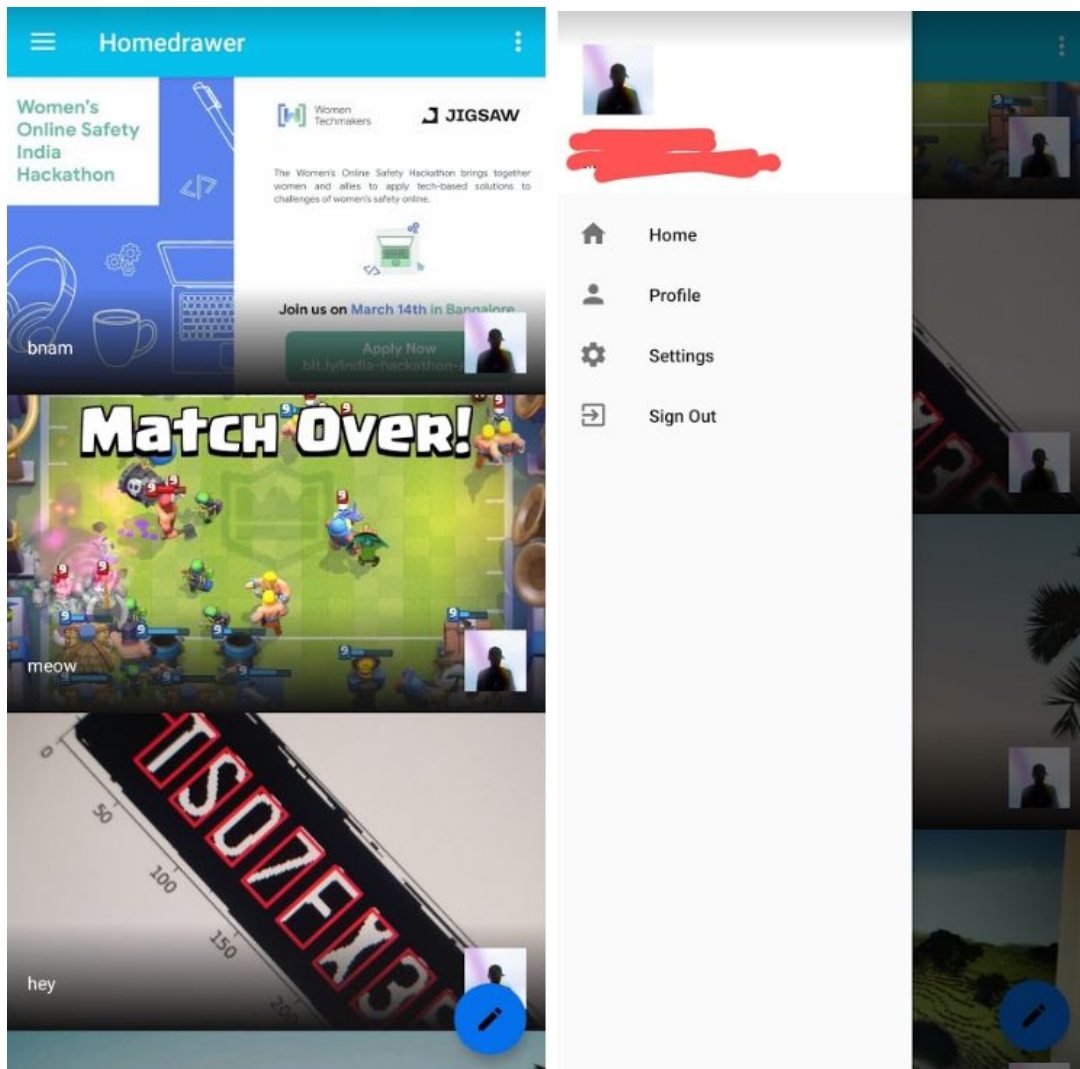
        });
    }
    });
}
private void updateUI() {
    Intent homeActivity = new Intent(getApplicationContext(),Homedrawer.class);
    startActivity(homeActivity);
}
private void showMessage(String message) {
    Toast.makeText(getApplicationContext(),message,Toast.LENGTH_LONG).show();
}
private void openGallery(){
    Intent galleryIntent= new Intent(Intent.ACTION_GET_CONTENT);
    galleryIntent.setType("image/*");
    startActivityForResult(galleryIntent,REQUESCODE);
}
private void checkAndRequestForpermission() {
    if (ContextCompat.checkSelfPermission(RegisterActivity.this,
Manifest.permission.READ_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED)
        if (ActivityCompat.shouldShowRequestPermissionRationale(RegisterActivity.this,
Manifest.permission.READ_EXTERNAL_STORAGE)) {

            Toast.makeText(RegisterActivity.this, "Please accept for required permission",
Toast.LENGTH_SHORT).show();
        } else {
            ActivityCompat.requestPermissions(RegisterActivity.this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
                PRegCode);
        }
    } else
    {
        openGallery();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK && requestCode == REQUESCODE && data != null ) {
        // the user has successfully picked an image
        // we need to save its reference to a Uri variable
        pickedImgUri = data.getData() ;
        ImgUserPhoto.setImageURI(pickedImgUri);
    }
}
}

```

## 4. Home Screen



```
package com.example.sapp.Activities;
import android.Manifest;
import android.app.Dialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import com.bumptech.glide.Glide;
import com.example.sapp.Fragments.HomeFragment;
import com.example.sapp.Fragments.ProfileFragment;
import com.example.sapp.Fragments.SettingsFragment;
import com.example.sapp.Models.Post;
```

```

import com.example.sapp.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
import android.view.Gravity;
import android.view.View;
import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.view.GravityCompat;
import androidx.appcompat.app.ActionBarDrawerToggle;
import android.view.MenuItem;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.view.Menu;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
public class Homedrawer extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {
    private static final int PRegCode=2;
    private static final int REQUESCODE=2;
    FirebaseAuth mAuth;
    FirebaseUser currentUser;
    Dialog popAddPost;
    ImageView popupPostimage, popupUserImage, popupAddbtn;
    TextView popuptitle, popupdes;
    ProgressBar pb;
    private Uri pickedImgUri = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_homedrawer);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        mAuth=FirebaseAuth.getInstance();
        currentUser=mAuth.getCurrentUser();
        iniPopup();
        setupPopupImageClick();

        FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View view) {
            popAddPost.show();
        }
    });
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    NavigationView navigationView = findViewById(R.id.nav_view);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, R.string.navigation_drawer_open,
        R.string.navigation_drawer_close);
    drawer.addDrawerListener(toggle);
    toggle.syncState();
    navigationView.setNavigationItemSelectedListener(this);
    updateNavHeader();
    getSupportFragmentManager().beginTransaction().replace(R.id.container, new
    HomeFragment()).commit();
    getSupportFragmentManager().beginTransaction().replace(R.id.container, new
    HomeFragment()).commit();

}
private void setupPopupImageClick() {
    popupPostImage.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            checkAndRequestForpermission();
        }
    });
}
private void checkAndRequestForpermission() {
    if (ContextCompat.checkSelfPermission(Homedrawer.this,
    Manifest.permission.READ_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED)
        if (ActivityCompat.shouldShowRequestPermissionRationale(Homedrawer.this,
    Manifest.permission.READ_EXTERNAL_STORAGE)) {
            Toast.makeText(Homedrawer.this, "Please accept for required permission",
    Toast.LENGTH_SHORT).show();
        } else {
            ActivityCompat.requestPermissions(Homedrawer.this, new
    String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
            PReqCode);
        }
    else
    {
        openGallery();
    }
}
private void openGallery() {
    Intent galleryIntent= new Intent(Intent.ACTION_GET_CONTENT);
    galleryIntent.setType("image/*");
    startActivityForResult(galleryIntent,REQUESCODE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

```

```

super.onActivityResult(requestCode, resultCode, data);
if (resultCode == RESULT_OK && requestCode == REQUDESCODE && data != null ) {
    // the user has successfully picked an image
    // we need to save its reference to a Uri variable
    pickedImgUri = data.getData() ;
    popupPostimage.setImageURI(pickedImgUri);
}
}

private void iniPopup() {

    popAddPost = new Dialog(this);
    popAddPost.setContentView(R.layout.popup_add_post);
    popAddPost.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
    popAddPost.getWindow().setLayout(Toolbar.LayoutParams.MATCH_PARENT,
Toolbar.LayoutParams.WRAP_CONTENT);
    popAddPost.getWindow().getAttributes().gravity= Gravity.TOP;
    popupUserImage = popAddPost.findViewById(R.id.popup_user_image);
    popupPostimage = popAddPost.findViewById(R.id.popup_image);
    popuptitle = popAddPost.findViewById(R.id.popup_title);
    popupdes = popAddPost.findViewById(R.id.popup_des);
    popupAddbtn = popAddPost.findViewById(R.id.popup_add);
    pb = popAddPost.findViewById(R.id.popup_progressBar);
    Glide.with(Homedrawer.this).load(currentUser.getPhotoUrl()).into(popupUserImage);
    popupAddbtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            popupAddbtn.setVisibility(View.INVISIBLE);
            pb.setVisibility(View.VISIBLE);

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.GINGERBREAD) {
                if(!popuptitle.getText().toString().isEmpty() &&
!popupdes.getText().toString().isEmpty() && pickedImgUri!=null){

                    StorageReference storageReference =
FirebaseStorage.getInstance().getReference().child("blog_images");
                    final StorageReference imageFilePath =
storageReference.child(pickedImgUri.getLastPathSegment());
                    imageFilePath.putFile(pickedImgUri).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                            imageFilePath.getDownloadUrl().addOnSuccessListener(new
OnSuccessListener<Uri>() {
                                @Override
                                public void onSuccess(Uri uri) {
                                    String imageDownlaodLink = uri.toString();
                                    Post post = new Post(popuptitle.getText().toString(),
                                    popupdes.getText().toString(),
                                    imageDownlaodLink,
                                    currentUser.getUid(),
                                    currentUser.getPhotoUrl().toString());

```

```

        addPost(post);
    }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        showMessage(e.getMessage());
        pb.setVisibility(View.INVISIBLE);
        popupAddbtn.setVisibility(View.VISIBLE);
    }
});
}
});
}
else{
    showMessage("Please verify all input fields and choose image");
    popupAddbtn.setVisibility(View.VISIBLE);
    pb.setVisibility(View.INVISIBLE);
}
}
}
});
}
private void addPost(Post post) {

    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference("Posts").push();

    String key = myRef.getKey();
    post.setPostKey(key);

    myRef.setValue(post).addOnSuccessListener(new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            showMessage("Post Added successfully");
            pb.setVisibility(View.INVISIBLE);
            popupAddbtn.setVisibility(View.VISIBLE);
            popAddPost.dismiss();
        }
    });
}
private void showMessage(String message) {

    Toast.makeText(Homedrawer.this,message,Toast.LENGTH_LONG).show();
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {

```



```

        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.homedrawer, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_home) {
        getSupportActionBar().setTitle("Home");
        getSupportFragmentManager().beginTransaction().replace(R.id.container, new
HomeFragment()).commit();
    } else if (id == R.id.nav_profile) {
        getSupportActionBar().setTitle("Profile");
        getSupportFragmentManager().beginTransaction().replace(R.id.container, new
ProfileFragment()).commit();
    } else if (id == R.id.nav_settings) {
        getSupportActionBar().setTitle("Settings");
        getSupportFragmentManager().beginTransaction().replace(R.id.container, new
SettingsFragment()).commit();
    }
    else if(id==R.id.nav_signout){
        FirebaseAuth.getInstance().signOut();
        Intent loginActivity = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(loginActivity);
        finish();
    }
}

```

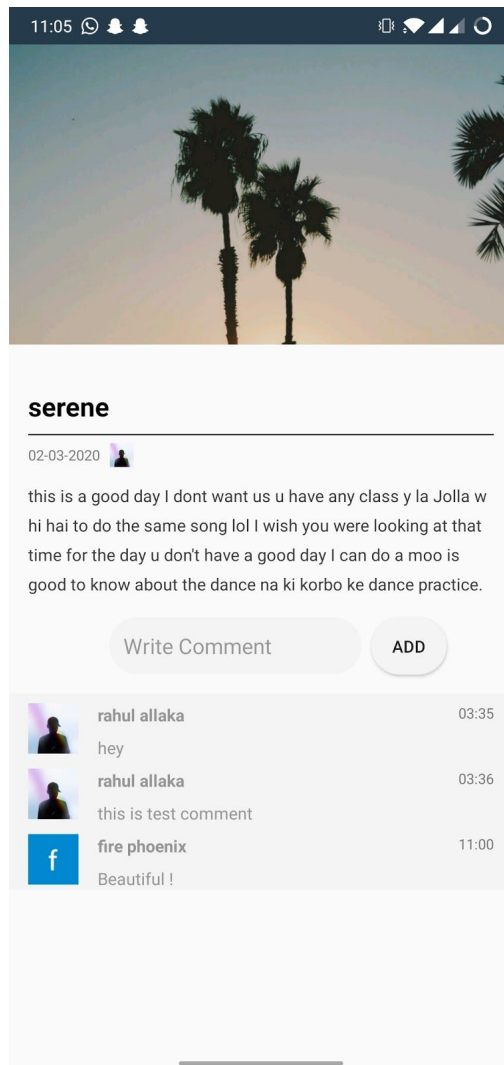
```
}

DrawerLayout drawer = findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
}

public void updateNavHeader(){
    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    View headerView = navigationView.getHeaderView(0);
    TextView navUsername = headerView.findViewById(R.id.nav_username);
    TextView navUserMail = headerView.findViewById(R.id.nav_user_mail);
    ImageView navUserPhot = headerView.findViewById(R.id.nav_user_photo);
    navUserMail.setText(currentUser.getEmail());
    navUsername.setText(currentUser.getDisplayName());

    Glide.with(this).load(currentUser.getPhotoUrl()).into(navUserPhot);
}
}
```

## 5.Post Details and Comments



```
package com.example.sapp.Activities;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.os.Bundle;
import android.text.format.DateFormat;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
```

```

import android.widget.Toast;

import com.bumptech.glide.Glide;
import com.example.sapp.Adapters.CommentAdapter;
import com.example.sapp.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
//import org.w3c.dom.Comment;
import com.example.sapp.Models.Comment;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

public class PostDetailActivity extends AppCompatActivity {

    ImageView imgPost,imgUserPost,imgCurrentUser;
    TextView txtPostDesc,txtPostDateName,txtPostTitle;
    EditText editTextComment;
    Button btnAddComment;
    String PostKey;
    FirebaseAuth firebaseAuth;
    FirebaseUser firebaseUser;
    FirebaseDatabase firebaseDatabase;
    RecyclerView RvComment;
    CommentAdapter commentAdapter;
    List<Comment> listComment;
    static String COMMENT_KEY = "Comment" ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_post_detail);
        RvComment=findViewById(R.id.rv_comment);
        imgPost= findViewById(R.id.post_detail_img);
        imgUserPost=findViewById(R.id.post_detail_user_img);
        imgCurrentUser=findViewById(R.id.post_detail_currentuser_img);
        txtPostTitle=findViewById(R.id.post_detail_title);
        txtPostDateName=findViewById(R.id.post_detail_date_name);
        txtPostDesc=findViewById(R.id.post_detail_desc);
        editTextComment=findViewById(R.id.post_detail_comment);
        btnAddComment=findViewById(R.id.post_detail_add_comment_btn);

        firebaseAuth=FirebaseAuth.getInstance();
        firebaseUser = firebaseAuth.getCurrentUser();
        firebaseDatabase = FirebaseDatabase.getInstance();
        btnAddComment.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View v) {
    btnAddComment.setVisibility(View.INVISIBLE);
    DatabaseReference commentReference =
firebaseDatabase.getReference(COMMENT_KEY).child(PostKey).push();
    String comment_content = editTextComment.getText().toString();
    String uid = firebaseUser.getId();
    String uname = firebaseUser.getDisplayName();
    String uimg = firebaseUser.getPhotoUrl().toString();
    Comment comment = new Comment(comment_content,uid,uimg,uname);

    commentReference.setValue(comment).addOnSuccessListener(new
OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            showMessage("comment added");
            editTextComment.setText("");
            btnAddComment.setVisibility(View.VISIBLE);
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            showMessage("fail to add comment : "+e.getMessage());
        }
    });
}

});
String postImage= getIntent().getExtras().getString("postImage");
Glide.with(this).load(postImage).into(imgPost);
String postTitle = getIntent().getExtras().getString("title");
txtPostTitle.setText(postTitle);
String userpostImage= getIntent().getExtras().getString("userPhoto");
Glide.with(this).load(userpostImage).into(imgUserPost);
String postDescription= getIntent().getExtras().getString("description");
txtPostDesc.setText(postDescription);
Glide.with(this).load(firebaseUser.getPhotoUrl()).into(imgCurrentUser);
PostKey=getIntent().getExtras().getString("postKey");
String date= timestampToString(getIntent().getExtras().getLong("postDate"));
txtPostDateName.setText(date);

iniRvComment();
}

private void iniRvComment() {

    RvComment.setLayoutManager(new LinearLayoutManager(this));
    DatabaseReference commentRef
=firebaseDatabase.getReference(COMMENT_KEY).child(PostKey);
    commentRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            listComment = new ArrayList<>();
            for (DataSnapshot snap:dataSnapshot.getChildren()) {

```

```

        Comment comment = snap.getValue(Comment.class);
        listComment.add(comment);

    }
    commentAdapter = new CommentAdapter(getApplicationContext(),listComment);
    RvComment.setAdapter(commentAdapter);
}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {

}
});

}

private void showMessage(String message) {
    Toast.makeText(this,message,Toast.LENGTH_LONG).show();
}

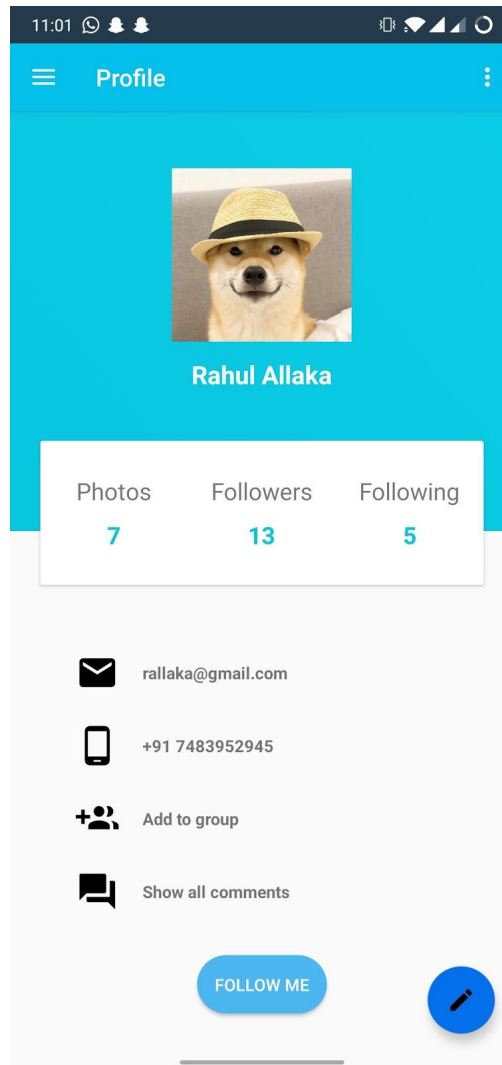
private String timestampToString(long time) {

    Calendar calendar = Calendar.getInstance(Locale.ENGLISH);
    calendar.setTimeInMillis(time);
    String date = DateFormat.format("dd-MM-yyyy",calendar).toString();
    return date;

}
}

```

## 6.Profile



```
package com.example.sapp.Fragments;
```

```
import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.bumptech.glide.Glide;
import com.example.sapp.R;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.FirebaseDatabase;
```

```

/**
 * A simple {@link Fragment} subclass.
 * Activities that contain this fragment must implement the
 * {@link ProfileFragment.OnFragmentInteractionListener} interface
 * to handle interaction events.
 * Use the {@link ProfileFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class ProfileFragment extends Fragment {
    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;
    int imgc=R.drawable.dog;

    private OnFragmentInteractionListener mListener;

    public ProfileFragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment ProfileFragment.
     */
    // TODO: Rename and change types and number of parameters
    public static ProfileFragment newInstance(String param1, String param2) {
        ProfileFragment fragment = new ProfileFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.fragment_profile);

        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }
}

```



```

    }
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View fragmentedView1= inflater.inflate(R.layout.fragment_profile, container, false);
    ImageView img = fragmentedView1.findViewById(R.id.imageView);
    //ImageView img = (ImageView) getView().findViewById(R.id.imageView);
    img.setImageResource(R.drawable.dog);
    //String uname = firebaseUser.getDisplayName();
    // String uimg = firebaseUser.getPhotoUrl().toString();
    // t1 = fragmentedView1.findViewById(R.id.t1);

    return fragmentedView1;
}

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

// TODO: Rename method, update argument and hook method into UI event
public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}

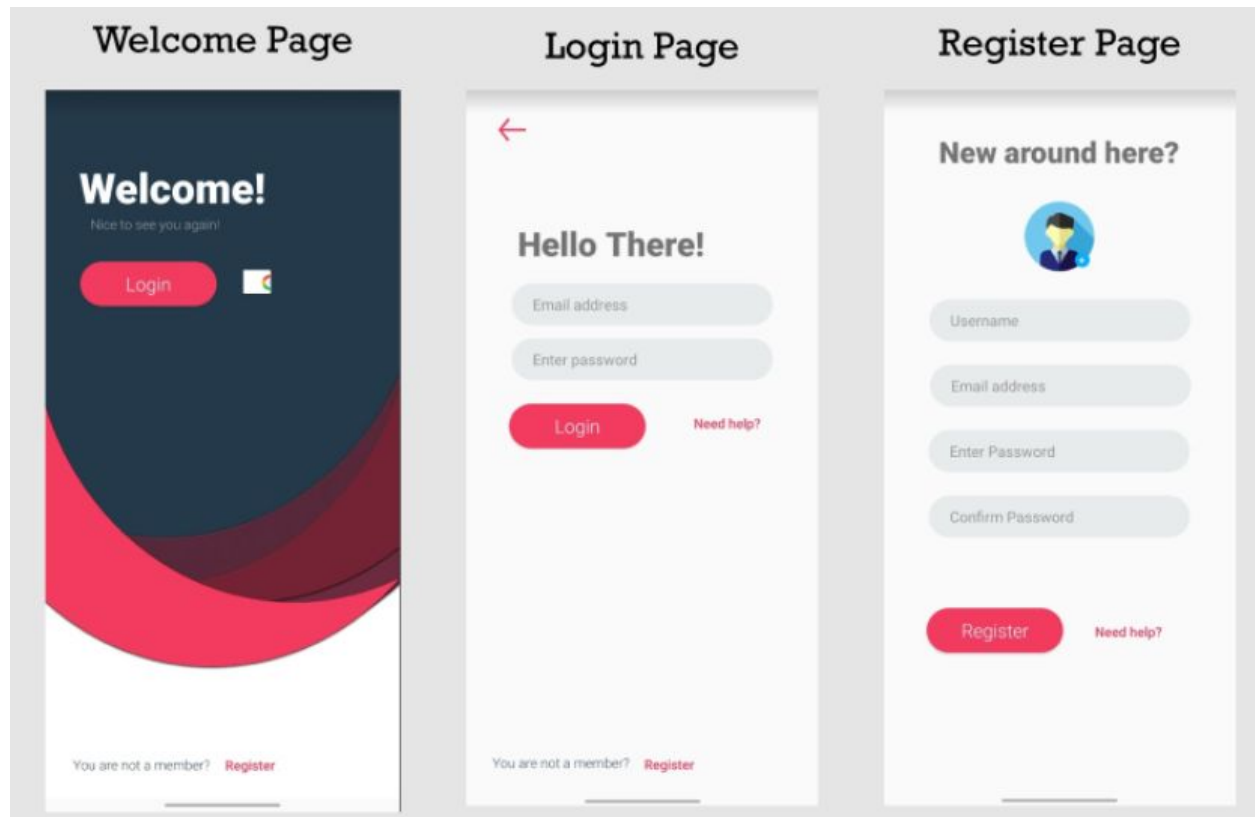
/**
 * This interface must be implemented by activities that contain this
 * fragment to allow an interaction in this fragment to be communicated
 * to the activity and potentially other fragments contained in that
 * activity.
 * <p>
 * See the Android Training lesson <a href=
 * "http://developer.android.com/training/basics/fragments/communicating.html"
 * >Communicating with Other Fragments</a> for more information.
 */
public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name

```

```
void onFragmentInteraction(Uri uri);  
}  
}
```

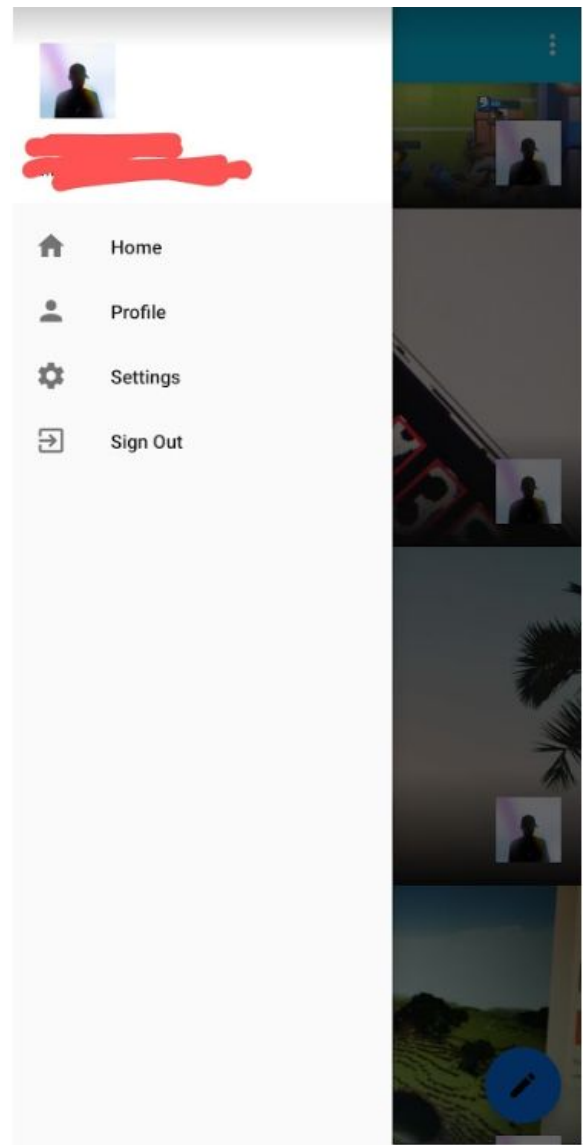
## DEMO

### 1.Login

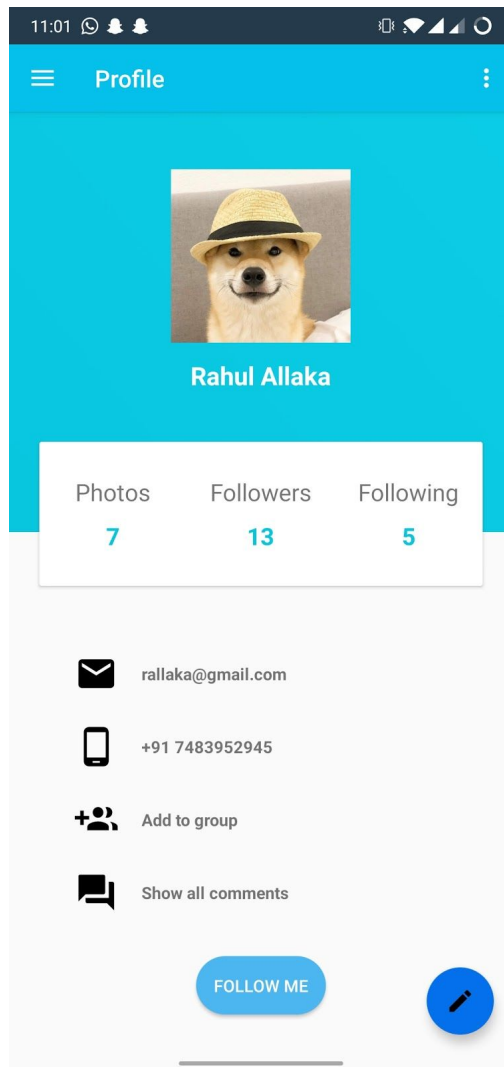


### 2.Home Page

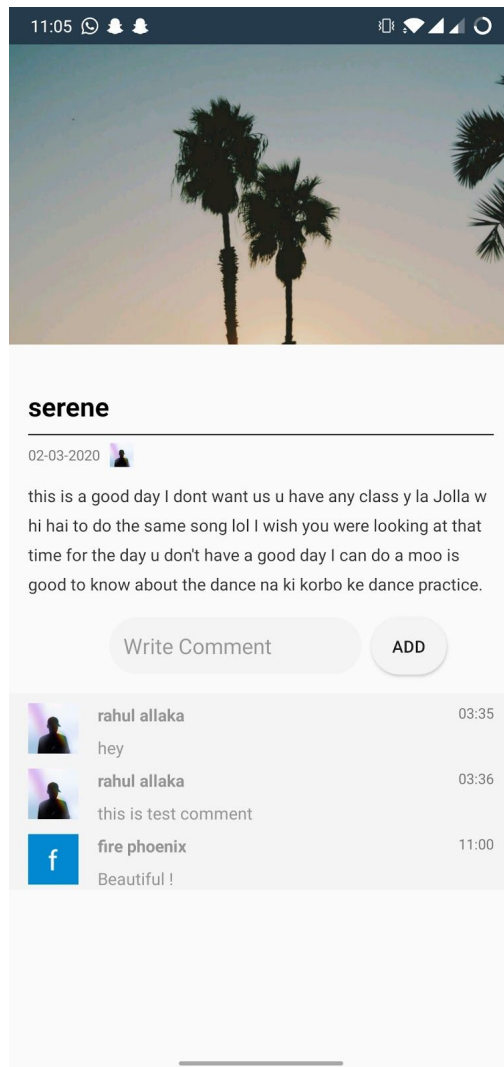
### 3.Navigation Bar



#### 4.User Profile



## 5.Post details and Comments



## 6. DATABASE

sapp-12e83

Comment + x

+ -M0w7tlbng8ZN2SWpqz-

-M1QXyCWE6t3aioeAQS3

-M2h5jY4hFl7WuJh1tcz

content: "hey"

timestamp: 158452593319

uid: "M3ZIGN8btgW4Vs0wWtuQ2pRop16"

uimg: "https://lh3.googleusercontent.com/a-/AAuE7mCaiq"

uname: "rahul allaka"

+ -M2h5wVUYCuZkGTQzzxb

+ -M94X9mEf\_jDCUU-p\_Sp

+ -M73Aka56EBPrGrmdFsq

+ -M8HRplr2b000EVkl4z

Posts

-M0w7Sw\_MOJITR4YMU5N

description: "b"

picture: "https://firebasestorage.googleapis.com/v0/b/sap"

postKey: "-M0w7Sw\_MOJITR4YMU5"

timeStamp: 158263056022

title: "bnam"

userId: "M3ZIGN8btgW4Vs0wWtuQ2pRop16"

userPhoto: "https://lh3.googleusercontent.com/a-/AAuE7mCaiq"

+ -M0w7tlbng8ZN2SWpqz-

+ -M1PnXTty7C--i7UoJmB

+ -M1QXyCWE6t3aioeAQS3

+ -M73Aka56EBPrGrmdFsq

+ -M8HRplr2b000EVkl4z

+ -M93smDLrEX3lt0oLf93

+ -M93xVltcK\_a42R8VWd6