PawPerfectMatch

Members:

Andrade, Billy L.

Fajarda, Stephen Joy A.

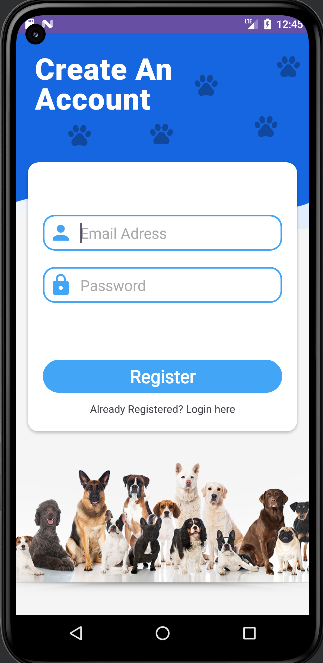
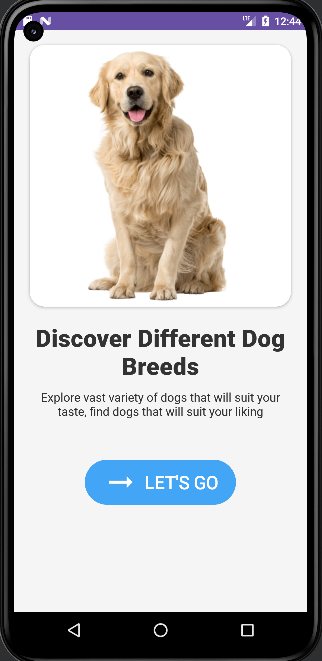
Ignacio, Carl Justine A.

Tuazon, Jaira Midel A.

Carmelo, Mel Vincent A.

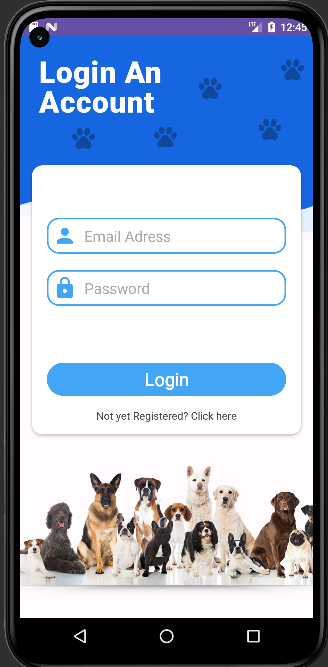
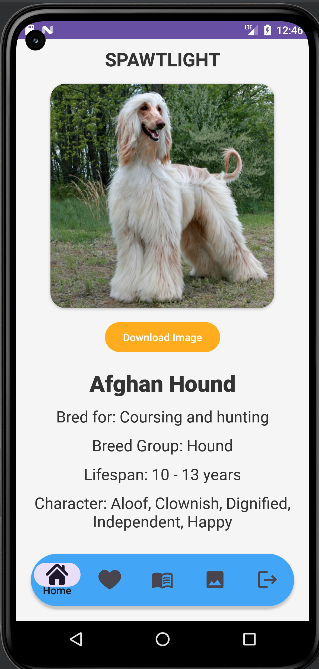
Register Page

Welcome Page



Home Page

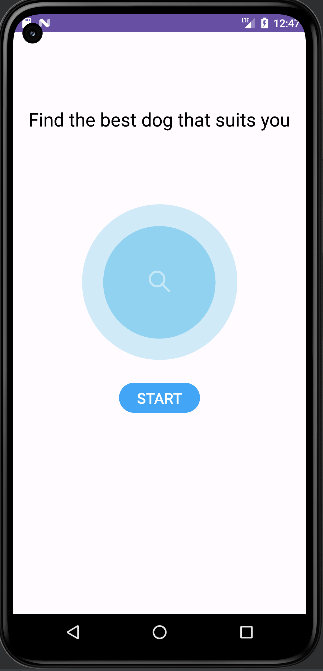
Login Page



Dog Matcher Page 1

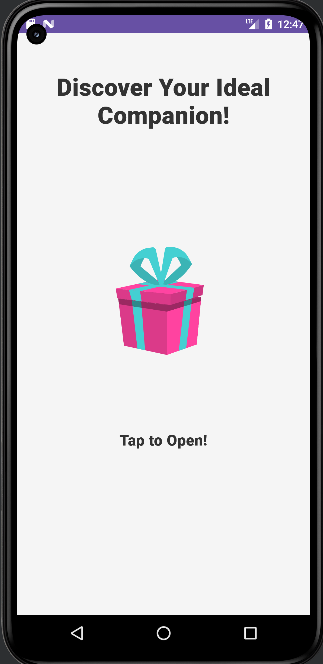
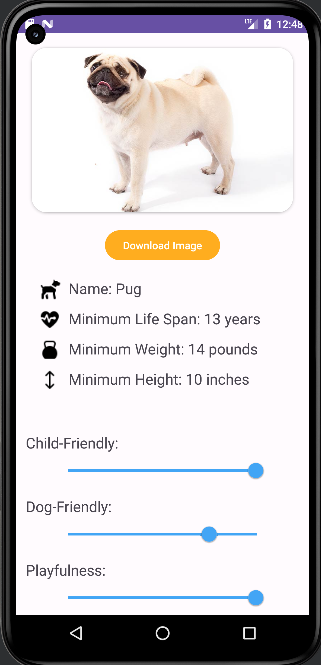
Sample Question





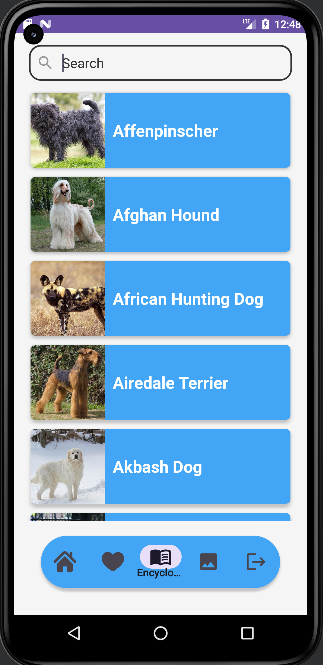
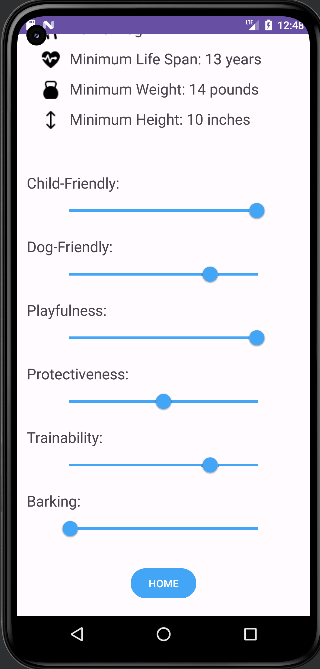
Dog Matcher Page 3

Dog Matcher Page 2



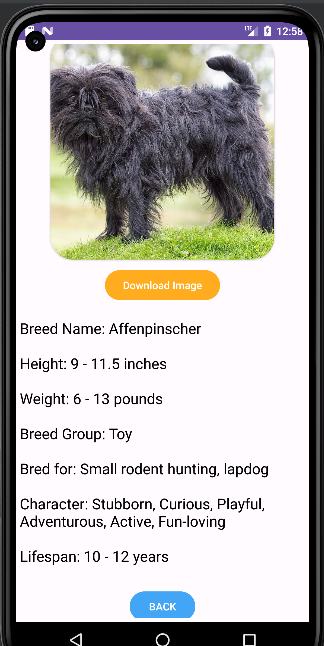
Encyclopedia

Dog Matcher Page 3



Add Dog Page

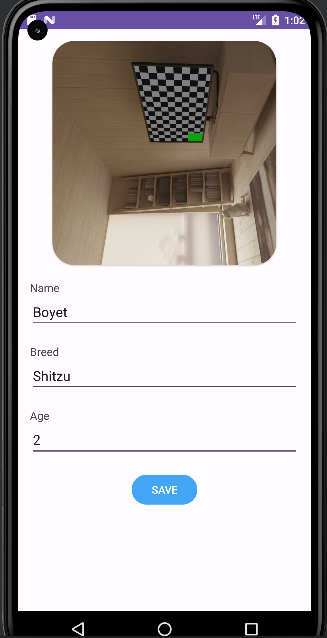
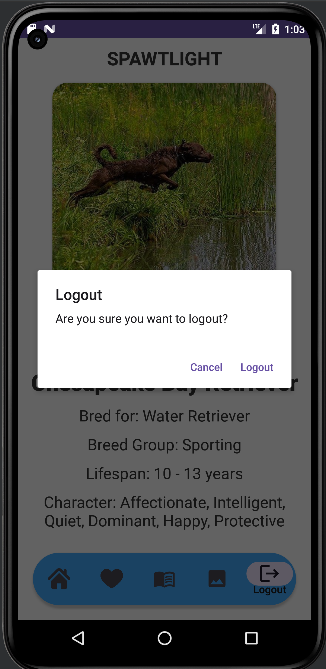
Encyclopedia Details



Logout

\

Add Dog Details



**AddDogActivity.java**

package com.example.newtry;  
  
import android.content.Intent;  
import android.net.Uri;  
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 androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class AddDogActivity extends AppCompatActivity {  
  
 private static final int *PICK\_IMAGE\_REQUEST* = 1;  
  
 private EditText nameEditText;  
 private EditText breedEditText;  
 private EditText ageEditText;  
 private Button saveButton;  
 private ImageView pictureImageView;  
 private Uri pictureUri;  
 private FirebaseFirestore db;  
 private FirebaseAuth auth;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activty\_add\_dog*);  
  
  
  
 nameEditText = findViewById(R.id.*nameEditText*);  
 breedEditText = findViewById(R.id.*breedEditText*);  
 ageEditText = findViewById(R.id.*ageEditText*);  
 saveButton = findViewById(R.id.*saveButton*);  
 pictureImageView = findViewById(R.id.*pictureImageView*);  
  
 db = FirebaseFirestore.*getInstance*();  
 auth = FirebaseAuth.*getInstance*();  
  
 pictureImageView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 openImagePicker();  
 }  
 });  
  
 saveButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 saveDog();  
 }  
 });  
 }  
  
 private void openImagePicker() {  
 Intent intent = new Intent(Intent.*ACTION\_OPEN\_DOCUMENT*);  
 intent.setType("image/\*");  
 intent.addCategory(Intent.*CATEGORY\_OPENABLE*);  
 startActivityForResult(intent, *PICK\_IMAGE\_REQUEST*);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == *PICK\_IMAGE\_REQUEST* && resultCode == *RESULT\_OK* && data != null && data.getData() != null) {  
 pictureUri = data.getData();  
 pictureImageView.setImageURI(pictureUri);  
 }  
 }  
  
 private void saveDog() {  
 String name = nameEditText.getText().toString().trim();  
 String breed = breedEditText.getText().toString().trim();  
 String age = ageEditText.getText().toString().trim();  
  
 // Perform validation for name, breed, and age  
 if (name.isEmpty()) {  
 nameEditText.setError("Please enter a name");  
 return;  
 }  
  
 if (breed.isEmpty()) {  
 breedEditText.setError("Please enter a breed");  
 return;  
 }  
  
 if (age.isEmpty()) {  
 ageEditText.setError("Please enter an age");  
 return;  
 }  
  
 String userId = auth.getCurrentUser().getUid();  
  
 // Create a new Dog object with the entered details  
 String id = db.collection("users").document().getId(); // Generate a unique ID  
 Dog dog = new Dog(id, null, name, breed, age); // Set pictureUrl as null for now  
  
 // Save the dog to Firestore  
 db.collection("users").document(userId).collection("dogs")  
 .add(dog)  
 .addOnSuccessListener(documentReference -> {  
 String dogId = documentReference.getId();  
 String imageFileName = "dogs/" + dogId + ".jpg"; // Set the image file name  
  
 // Upload the image to Firebase Storage  
 StorageReference storageRef = FirebaseStorage.*getInstance*().getReference(imageFileName);  
 storageRef.putFile(pictureUri)  
 .addOnSuccessListener(taskSnapshot -> {  
 // Get the download URL of the uploaded image  
 storageRef.getDownloadUrl()  
 .addOnSuccessListener(uri -> {  
 String pictureUrl = uri.toString();  
  
 // Update the dog document in Firestore with the pictureUrl  
 Map<String, Object> updates = new HashMap<>();  
 updates.put("picture", pictureUrl);  
  
 db.collection("users")  
 .document(userId)  
 .collection("dogs")  
 .document(dogId)  
 .update(updates)  
 .addOnSuccessListener(aVoid -> {  
 Toast.*makeText*(AddDogActivity.this, "Dog saved successfully", Toast.*LENGTH\_SHORT*).show();  
 finish(); // Finish the activity and go back to DogAlbumActivity  
 })  
 .addOnFailureListener(e -> {  
 Toast.*makeText*(AddDogActivity.this, "Error saving dog", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 });  
 })  
 .addOnFailureListener(e -> {  
 Toast.*makeText*(AddDogActivity.this, "Error getting image URL", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 });  
 })  
 .addOnFailureListener(e -> {  
 Toast.*makeText*(AddDogActivity.this, "Error uploading image", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 });  
 })  
 .addOnFailureListener(e -> {  
 Toast.*makeText*(AddDogActivity.this, "Error saving dog", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 });  
 }  
  
  
}

**BestDog.java**

package com.example.newtry;  
  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.drawable.Drawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Environment;  
import android.os.Handler;  
import android.os.Looper;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.FrameLayout;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.airbnb.lottie.LottieAnimationView;  
import com.google.android.material.slider.Slider;  
import com.squareup.picasso.Picasso;  
import com.squareup.picasso.Target;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class BestDog extends AppCompatActivity {  
 ImageView ImDog;  
 TextView tvDogName, tvMinLife, tvMaxLife, tvMinWeight, tvMaxWeight, tvMinHeight, tvMaxHeight;  
 Slider slChildren, slODogs, slPlayful, slProtective, slTrain, slBark;  
 Button btnOk, btnDownload;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*best\_dog*);  
  
 initialize();  
 }  
  
 private void initialize() {  
  
 tvDogName = findViewById(R.id.*tvDogName*);  
 tvMinLife = findViewById(R.id.*tvMinLife*);  
  
 tvMinWeight= findViewById(R.id.*tvMinWeight*);  
  
 tvMinHeight= findViewById(R.id.*tvMinHeight*);  
  
  
 slChildren= findViewById(R.id.*slChildren*);  
 slODogs= findViewById(R.id.*slDog*);  
 slPlayful= findViewById(R.id. *slPlayful*);  
 slProtective= findViewById(R.id.*slProtective*);  
 slTrain= findViewById(R.id. *slTrain*);  
 slBark = findViewById(R.id.*slBark*); ;  
  
 ImDog = findViewById(R.id.*ImDog*);  
 btnOk = findViewById(R.id.*btnOk*);  
  
 //hide contents  
 LinearLayout contents = findViewById(R.id.*contents*);  
 contents.setVisibility(View.*GONE*);  
  
 // Get the intent that started this activity  
 Intent intent1 = getIntent();  
  
 // Extract the data from the intent  
 String name = intent1.getStringExtra("name");  
 String minLife = intent1.getStringExtra("min\_life");  
  
 String minWeight = intent1.getStringExtra("min\_weight");  
  
 String minHeight = intent1.getStringExtra("min\_height");  
  
 String imageUrl = intent1.getStringExtra("image\_url");  
  
 String gChildren = intent1.getStringExtra("gChildren");  
 String gDogs = intent1.getStringExtra("gDogs");  
 String playful = intent1.getStringExtra("playful");  
 String protective = intent1.getStringExtra("protective");  
 String train = intent1.getStringExtra("train");  
 String barking = intent1.getStringExtra("barking");  
  
 slChildren.setValue(Integer.*parseInt*(gChildren));  
 slODogs.setValue(Integer.*parseInt*(gDogs));  
 slPlayful.setValue(Integer.*parseInt*(playful));  
 slProtective.setValue(Integer.*parseInt*(protective));  
 slTrain.setValue(Integer.*parseInt*(train));  
 slBark.setValue(Integer.*parseInt*(barking));  
  
 tvDogName.setText("Name: "+name);  
 tvMinLife.setText("Minimum Life Span: "+minLife.replaceAll("\\.0$", "")+" years");  
  
 tvMinWeight.setText("Minimum Weight: "+minWeight.replaceAll("\\.0$", "")+" pounds");  
  
 tvMinHeight.setText("Minimum Height: "+minHeight.replaceAll("\\.0$", "")+" inches");  
  
 // Load the dog image using Picasso library  
 Picasso.*get*().load(imageUrl)  
 .into(ImDog);  
  
  
 btnDownload = findViewById(R.id.*btnDownload*);  
 btnDownload.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 downloadImage(imageUrl);  
 }  
 });  
  
  
  
 //Show contents  
 FrameLayout animationContainer = findViewById(R.id.*animationContainer*);  
 LottieAnimationView animationView = findViewById(R.id.*animationView*);  
 animationView.setAnimation("cutedog.json");  
 animationView.playAnimation();  
  
 // Stop the animation after 3 seconds  
 new Handler(Looper.*getMainLooper*()).postDelayed(new Runnable() {  
 @Override  
 public void run() {  
  
 animationView.cancelAnimation();  
 animationView.setVisibility(View.*GONE*);  
  
 contents.setVisibility(View.*VISIBLE*);  
  
 }  
 }, 5000);  
  
 btnOk.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // Go back to Home  
 Intent intent = new Intent(BestDog.this, Home.class);  
 startActivity(intent);  
 }  
 });  
  
 }  
  
 private void downloadImage(String imageUrl) {  
 Target imageTarget = new Target() {  
 @Override  
 public void onBitmapLoaded(Bitmap bitmap, Picasso.LoadedFrom from) {  
 saveImageToGallery(bitmap);  
 }  
  
 @Override  
 public void onBitmapFailed(Exception e, Drawable errorDrawable) {  
 Toast.*makeText*(getApplicationContext(), "Failed to download image", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onPrepareLoad(Drawable placeHolderDrawable) {  
 }  
 };  
  
 Picasso.*get*().load(imageUrl).into(imageTarget);  
 }  
  
 private void saveImageToGallery(Bitmap bitmap) {  
 String filename = "image\_" + System.*currentTimeMillis*() + ".jpg";  
 File directory = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*);  
 File file = new File(directory, filename);  
  
 try (FileOutputStream fos = new FileOutputStream(file)) {  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*, 100, fos);  
 fos.flush();  
 // Add the image to the gallery  
 Uri imageUri = Uri.*fromFile*(file);  
 Intent mediaScanIntent = new Intent(Intent.*ACTION\_MEDIA\_SCANNER\_SCAN\_FILE*, imageUri);  
 sendBroadcast(mediaScanIntent);  
  
 Toast.*makeText*(this, "Image downloaded", Toast.*LENGTH\_SHORT*).show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 Toast.*makeText*(this, "Failed to save image", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

**Dog.java**

package com.example.newtry;  
  
import android.os.Parcel;  
import android.os.Parcelable;  
  
public class Dog implements Parcelable {  
 private String id;  
 private String picture;  
 private String name;  
 private String breed;  
 private String age;  
  
  
 public Dog() {  
 // Default constructor required for Firestore deserialization  
 }  
  
 public Dog(String id, String picture, String name, String breed, String age) {  
 this.id = id;  
 this.picture = picture;  
 this.name = name;  
 this.breed = breed;  
 this.age = age;  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getPicture() {  
 return picture;  
 }  
  
 public void setPicture(String picture) {  
 this.picture = picture;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getBreed() {  
 return breed;  
 }  
  
 public void setBreed(String breed) {  
 this.breed = breed;  
 }  
  
 public String getAge() {  
 return age;  
 }  
  
 public void setAge(String age) {  
 this.age = age;  
 }  
  
 // Parcelable implementation  
 @Override  
 public int describeContents() {  
 return 0;  
 }  
  
 @Override  
 public void writeToParcel(Parcel dest, int flags) {  
 dest.writeString(picture);  
 dest.writeString(name);  
 dest.writeString(breed);  
 dest.writeString(age);  
 }  
  
 protected Dog(Parcel in) {  
 picture = in.readString();  
 name = in.readString();  
 breed = in.readString();  
 age = in.readString();  
 }  
  
 public static final Creator<Dog> *CREATOR* = new Creator<Dog>() {  
 @Override  
 public Dog createFromParcel(Parcel in) {  
 return new Dog(in);  
 }  
  
 @Override  
 public Dog[] newArray(int size) {  
 return new Dog[size];  
 }  
 };  
}

**DogAdapter.java**

package com.example.newtry;  
  
import android.content.Context;  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import com.squareup.picasso.Picasso;  
  
import java.util.List;  
  
public class DogAdapter extends ArrayAdapter<Dog> {  
  
 private Context context;  
 private List<Dog> dogList;  
  
 public DogAdapter(Context context, List<Dog> dogList) {  
 super(context, R.layout.*item\_dog*, dogList);  
 this.context = context;  
 this.dogList = dogList;  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 if (convertView == null) {  
 convertView = LayoutInflater.*from*(context).inflate(R.layout.*item\_dog*, parent, false);  
 }  
  
 ImageView imgDog = convertView.findViewById(R.id.*imgDog*);  
 TextView txtName = convertView.findViewById(R.id.*txtName*);  
  
 Dog dog = dogList.get(position);  
  
 if (TextUtils.*isEmpty*(dog.getPicture())) {  
 // Handle empty picture case (e.g., display a placeholder image)  
 imgDog.setImageResource(R.drawable.*ic\_launcher\_background*);  
 } else {  
 // Load the dog's picture using Picasso  
 Picasso.*get*().load(dog.getPicture()).into(imgDog);  
 }  
  
 txtName.setText(dog.getName());  
  
 return convertView;  
 }  
  
}

**DogAlbumActivity.java**

package com.example.newtry;  
  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.firestore.EventListener;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.firestore.QuerySnapshot;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.firestore.DocumentSnapshot;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.firestore.FirebaseFirestoreException;  
import com.google.firebase.firestore.QuerySnapshot;  
  
import java.util.ArrayList;  
  
import java.util.List;  
  
public class DogAlbumActivity extends AppCompatActivity {  
  
 private GridView gridView;  
 private FirebaseFirestore db;  
 private FirebaseAuth auth; // Declare the auth variable  
 private List<Dog> dogList;  
 private DogAdapter adapter;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*dog\_album*);  
  
 // Initialize views and variables  
 gridView = findViewById(R.id.*gridView*);  
 dogList = new ArrayList<>();  
 adapter = new DogAdapter(this, dogList);  
 gridView.setAdapter(adapter);  
  
 db = FirebaseFirestore.*getInstance*();  
 auth = FirebaseAuth.*getInstance*(); // Initialize the auth variable  
 String userId = auth.getCurrentUser().getUid();  
  
 // Retrieve dog data from Firestore  
 // Retrieve dog data from Firestore  
 db.collection("users").document(userId).collection("dogs")  
 .addSnapshotListener(new EventListener<QuerySnapshot>() {  
 @Override  
 public void onEvent(QuerySnapshot snapshot, FirebaseFirestoreException error) {  
 if (error != null) {  
 Log.*e*("DogAlbumActivity", "Error getting dog collection: ", error);  
 return;  
 }  
  
 dogList.clear();  
  
 for (DocumentSnapshot document : snapshot.getDocuments()) {  
 Dog dog = document.toObject(Dog.class);  
 String dogId = document.getId(); // Get the document ID as the dog ID  
 dog.setId(dogId); // Set the dog ID in the Dog object  
 dogList.add(dog);  
 }  
  
 adapter.notifyDataSetChanged();  
 }  
 });  
  
  
 // Set click listener for Add Dog button  
 Button addDogButton = findViewById(R.id.*addDogButton*);  
 addDogButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // Open AddDogActivity to add a new dog  
 Intent intent = new Intent(DogAlbumActivity.this, AddDogActivity.class);  
 startActivity(intent);  
 }  
 });  
  
 // Set item click listener for GridView  
 gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 Dog dog = dogList.get(position);  
 // Open full details activity or fragment for the selected dog  
 Intent intent = new Intent(DogAlbumActivity.this, DogDetailsActivity.class);  
 intent.putExtra("dog", dog);  
 startActivity(intent);  
 }  
 });  
  
 // Set item long click listener for GridView  
 gridView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
 @Override  
 public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {  
 Dog dog = dogList.get(position);  
 showOptionsDialog(dog);  
 return true;  
 }  
 });  
  
  
 BottomNavigationView bottomNavigationView1 = findViewById(R.id.*bottomNavgationView*);  
 bottomNavigationView1.setSelectedItemId(R.id.*bottom\_random*);  
  
  
 bottomNavigationView1.setOnItemSelectedListener(item -> {  
 if (item.getItemId() == R.id.*bottom\_home*) {  
 startActivity(new Intent(getApplicationContext(), Home.class));  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_matcher*) {  
 startActivity(new Intent(getApplicationContext(), startDogMatcher.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_book*) {  
 startActivity(new Intent(getApplicationContext(), Encyclopediaa.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_random*) {  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_logout*) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());  
 builder.setTitle("Logout");  
 builder.setMessage("Are you sure you want to logout?");  
 builder.setPositiveButton("Logout", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 FirebaseAuth.*getInstance*().signOut();  
 Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivity(intent);  
 finish();  
 }  
 });  
 builder.setNegativeButton("Cancel", null);  
 AlertDialog dialog = builder.create();  
 dialog.show();  
 } else {  
 return false;  
 }  
 return true;  
 });  
 }  
  
 private void showOptionsDialog(final Dog dog) {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle("Options")  
 .setItems(new CharSequence[]{"Edit", "Delete"}, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 if (which == 0) {  
 // Open EditDogActivity to edit the selected dog  
  
 // Open EditDogActivity to edit the selected dog  
 Intent intent = new Intent(DogAlbumActivity.this, EditDogActivity.class);  
 intent.putExtra("dog", dog);  
 intent.putExtra("dogId", dog.getId()); // Pass the dog ID  
  
 startActivity(intent);  
 } else if (which == 1) {  
 //Delete option selected, delete the dog  
 deleteDog(dog);  
 }  
 }  
 });  
  
 builder.create().show();  
 }  
  
  
 private void deleteDog(final Dog dog) {  
 // Create a confirmation dialog  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle("Confirmation")  
 .setMessage("Are you sure you want to delete this dog?")  
 .setPositiveButton("Delete", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 // Delete the dog from Firestore  
 deleteDogFromFirestore(dog);  
 }  
 })  
 .setNegativeButton("Cancel", null)  
 .show();  
 }  
  
 private void deleteDogFromFirestore(Dog dog) {  
 // Delete the dog from Firestore  
 db.collection("users")  
 .document(auth.getCurrentUser().getUid())  
 .collection("dogs")  
 .document(dog.getId())  
 .delete()  
 .addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 Toast.*makeText*(DogAlbumActivity.this, "Dog deleted successfully", Toast.*LENGTH\_SHORT*).show();  
 }  
 })  
 .addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(DogAlbumActivity.this, "Error deleting dog", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 }  
 });  
 }  
  
}

**DogApiService.java**

package com.example.newtry;  
  
  
  
import android.os.Handler;  
import android.os.Looper;  
import android.util.Log;  
import android.widget.Toast;  
  
import org.jetbrains.annotations.NotNull;  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
  
import okhttp3.Call;  
import okhttp3.Callback;  
import okhttp3.OkHttpClient;  
import okhttp3.Request;  
import okhttp3.Response;  
  
public class DogApiService {  
 private static final String *DOG\_API\_URL* = "https://api.thedogapi.com/v1/breeds";  
 public void fetchDogBreeds(final DogBreedsCallback callback) {  
 OkHttpClient client = new OkHttpClient();  
  
 Request request = new Request.Builder()  
 .url(*DOG\_API\_URL*)  
 .build();  
  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException {  
 if (response.isSuccessful()) {  
 String responseBody = response.body().string();  
  
 try {  
 JSONArray jsonArray = new JSONArray(responseBody);  
 List<DogBreed> dogBreeds = new ArrayList<>();  
  
 for (int i = 0; i < jsonArray.length(); i++) {  
 JSONObject jsonObject = jsonArray.getJSONObject(i);  
 DogBreed dogBreed = new DogBreed();  
  
 dogBreed.setId(jsonObject.getInt("id"));  
 dogBreed.setName(jsonObject.getString("name"));  
 dogBreed.setBredFor(jsonObject.optString("bred\_for"));  
  
 JSONObject weightObject = jsonObject.getJSONObject("weight");  
 dogBreed.setWeight(weightObject.getString("imperial"));  
  
 JSONObject heightObject = jsonObject.getJSONObject("height");  
 dogBreed.setHeight(heightObject.getString("imperial"));  
  
 dogBreed.setLifeSpan(jsonObject.optString("life\_span"));  
 dogBreed.setBreedGroup(jsonObject.optString("breed\_group"));  
 dogBreed.setTemperament(jsonObject.optString("temperament"));  
  
 // Fetch the image URL  
 JSONObject imageObject = jsonObject.optJSONObject("image");  
 if (imageObject != null) {  
 String imageUrl = imageObject.optString("url");  
 dogBreed.setImageUrl(imageUrl);  
 Log.*d*("ImageUrl", "Value: " + imageUrl);  
 }  
  
 dogBreeds.add(dogBreed);  
 }  
  
 // Callback with the fetched dog breeds  
 callback.onSuccess(dogBreeds);  
 } catch (JSONException e) {  
 e.printStackTrace();  
 onError(new IOException("Failed to parse JSON response"));  
 }  
 } else {  
 onError(new IOException("Failed to fetch dog breeds: " + response.code()));  
 }  
 }  
 @Override  
 public void onFailure(@NotNull Call call, @NotNull IOException e) {  
 e.printStackTrace();  
 onError(e);  
 }  
 private void onError(final IOException e) {  
 // Update UI on the main thread  
 new Handler(Looper.*getMainLooper*()).post(new Runnable() {  
 @Override  
 public void run() {  
 callback.onError(e);  
 }  
 });  
 }  
 });  
 }  
 public interface DogBreedsCallback {  
 void onSuccess(List<DogBreed> dogBreeds);  
 void onError(IOException e);  
 }  
}

**DogBreed.java**

package com.example.newtry;  
  
import java.io.Serializable;  
import java.util.List;  
  
public class DogBreed implements Serializable {  
 private int id;  
 private String name;  
 private String bredFor;  
 private String imageUrl;  
 private String weight;  
 private String height;  
 private String lifeSpan;  
 private String breedGroup;  
 private String temperament;  
 public String getWeight() {  
 return weight;  
 }  
  
 public void setWeight(String weight) {  
 this.weight = weight;  
 }  
  
 public String getHeight() {  
 return height;  
 }  
  
 public void setHeight(String height) {  
 this.height = height;  
 }  
  
 public String getLifeSpan() {  
 return lifeSpan;  
 }  
  
 public void setLifeSpan(String lifeSpan) {  
 this.lifeSpan = lifeSpan;  
 }  
  
 public String getBreedGroup() {  
 return breedGroup;  
 }  
  
 public void setBreedGroup(String breedGroup) {  
 this.breedGroup = breedGroup;  
 }  
  
 public String getTemperament() {  
 return temperament;  
 }  
  
 public void setTemperament(String temperament) {  
 this.temperament = temperament;  
 }  
  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getBredFor() {  
 return bredFor;  
 }  
  
 public void setBredFor(String bredFor) {  
 this.bredFor = bredFor;  
 }  
  
 public String getImageUrl() {  
 return imageUrl;  
 }  
  
 public void setImageUrl(String imageUrl) {  
 this.imageUrl = imageUrl;  
 }  
  
}

**DogBreedAdapter.java**

package com.example.newtry;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.squareup.picasso.Picasso;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class DogBreedAdapter extends RecyclerView.Adapter<DogBreedAdapter.ViewHolder> {  
 private List<DogBreed> dogBreeds;  
 private OnItemClickListener listener;  
 public interface OnItemClickListener {  
 void onItemClick(DogBreed dogBreed);  
 }  
 public DogBreedAdapter(List<DogBreed> dogBreeds, OnItemClickListener listener) {  
 this.dogBreeds = dogBreeds;  
 this.listener = listener;  
 }  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*item\_dog\_breed*, parent, false);  
 return new ViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
 DogBreed dogBreed = dogBreeds.get(position);  
 holder.bind(dogBreed);  
  
 holder.itemView.setOnClickListener(view -> {  
 if (listener != null) {  
 listener.onItemClick(dogBreed);  
 }  
 });  
 }  
  
 @Override  
 public int getItemCount() {  
 return dogBreeds.size();  
 }  
  
 public static class ViewHolder extends RecyclerView.ViewHolder {  
 private TextView breedNameTextView;  
 private ImageView dogImage;  
  
 public ViewHolder(@NonNull View itemView) {  
 super(itemView);  
 breedNameTextView = itemView.findViewById(R.id.*text\_breed\_name*);  
 dogImage = itemView.findViewById(R.id.*dogImage*);  
 }  
 public void bind(DogBreed dogBreed) {  
 breedNameTextView.setText(dogBreed.getName());  
  
 String imageUrl = dogBreed.getImageUrl();  
 if (imageUrl != null && !imageUrl.isEmpty()) {  
 Picasso.*get*().load(imageUrl).into(dogImage);  
 } else {  
 // Set a placeholder image or handle the case when the image URL is not available  
 }  
 }  
 }  
}

**DogDetails.java**

package com.example.newtry;  
  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.drawable.Drawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Environment;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.squareup.picasso.Picasso;  
import com.squareup.picasso.Target;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
public class DogDetails extends AppCompatActivity {  
 private TextView tvDogName;  
 private TextView tvDogBreedFor, tvDogHeight, tvDogWeight, tvDogGroup, tvDogTemperament, tvDogLifeSpan;  
  
 ImageView ImDogImage;  
  
 Button btnDownload, btnBack;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*dog\_details*);  
  
 btnBack = findViewById(R.id.*btnBack*);  
 btnDownload = findViewById(R.id.*btnDownload*);  
 tvDogName = findViewById(R.id.*tvDogName*);  
 tvDogBreedFor= findViewById(R.id.*tvDogBreedFor*);  
 tvDogHeight= findViewById(R.id.*tvDogHeight*);  
 tvDogWeight= findViewById(R.id.*tvDogWeight*);  
 tvDogGroup= findViewById(R.id.*tvDogGroup*);  
 tvDogTemperament= findViewById(R.id.*tvDogTemperament*);  
 tvDogLifeSpan= findViewById(R.id.*tvDogLifeSpan*);  
 ImDogImage= findViewById(R.id.*ImDogImage*);  
  
 // Retrieve the selected dog object from the intent  
 DogBreed selectedDog = (DogBreed) getIntent().getSerializableExtra("selectedDog");  
  
 // Display the dog's data in the appropriate views  
 if (selectedDog != null) {  
 String url\_image = selectedDog.getImageUrl();  
 tvDogName.setText("Breed Name: "+selectedDog.getName());  
 tvDogBreedFor.setText("Bred for: "+selectedDog.getBredFor());  
 tvDogHeight.setText("Height: "+selectedDog.getHeight()+" inches");  
 tvDogWeight.setText("Weight: "+selectedDog.getWeight()+" pounds");  
 tvDogGroup.setText("Breed Group: "+selectedDog.getBreedGroup());  
 tvDogTemperament.setText("Character: "+selectedDog.getTemperament());  
 tvDogLifeSpan.setText("Lifespan: "+selectedDog.getLifeSpan());  
  
 String dogImageUrl = "https://cdn2.thedogapi.com/images/" + url\_image + ".jpg";  
 Picasso.*get*().load(url\_image).into(ImDogImage);  
  
 btnDownload.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 downloadImage(url\_image);  
 }  
 });  
  
 }  
 else{  
 tvDogName.setText("walang laman");  
 }  
  
 btnBack.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
 }  
  
 private void downloadImage(String imageUrl) {  
 Target imageTarget = new Target() {  
 @Override  
 public void onBitmapLoaded(Bitmap bitmap, Picasso.LoadedFrom from) {  
 saveImageToGallery(bitmap);  
 }  
  
 @Override  
 public void onBitmapFailed(Exception e, Drawable errorDrawable) {  
 Toast.*makeText*(getApplicationContext(), "Failed to download image", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onPrepareLoad(Drawable placeHolderDrawable) {  
 }  
 };  
  
 Picasso.*get*().load(imageUrl).into(imageTarget);  
 }  
  
 private void saveImageToGallery(Bitmap bitmap) {  
 String filename = "image\_" + System.*currentTimeMillis*() + ".jpg";  
 File directory = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*);  
 File file = new File(directory, filename);  
  
 try (FileOutputStream fos = new FileOutputStream(file)) {  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*, 100, fos);  
 fos.flush();  
 // Add the image to the gallery  
 Uri imageUri = Uri.*fromFile*(file);  
 Intent mediaScanIntent = new Intent(Intent.*ACTION\_MEDIA\_SCANNER\_SCAN\_FILE*, imageUri);  
 sendBroadcast(mediaScanIntent);  
  
 Toast.*makeText*(this, "Image downloaded", Toast.*LENGTH\_SHORT*).show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 Toast.*makeText*(this, "Failed to save image", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

**DogDetailsActivity.java**

package com.example.newtry;  
  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.squareup.picasso.Picasso;  
  
public class DogDetailsActivity extends AppCompatActivity {  
  
 private TextView txtName;  
 private TextView txtBreed;  
 private TextView txtAge;  
 private ImageView imageView;  
  
 Button btnBack;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activty\_dog\_details*);  
  
 btnBack = findViewById(R.id.*btnBack*);  
 txtName = findViewById(R.id.*txtName*);  
 txtBreed = findViewById(R.id.*txtBreed*);  
 txtAge = findViewById(R.id.*txtAge*);  
 imageView = findViewById(R.id.*imageView*);  
  
 // Retrieve the dog object from the intent  
 Dog dog = getIntent().getParcelableExtra("dog");  
  
 // Set the dog's details in the activity  
 txtName.setText(dog.getName());  
 txtBreed.setText("Breed: " + dog.getBreed());  
 txtAge.setText("Age: "+ String.*valueOf*(dog.getAge()));  
  
 // Load and display the dog's image using Picasso library  
 Picasso.*get*().load(dog.getPicture()).into(imageView);  
  
 btnBack.setOnClickListener(v -> {  
 finish();  
 });  
 }  
}

**DogMatcher.java**

package com.example.newtry;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.CheckBox;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.airbnb.lottie.LottieAnimationView;  
import com.google.android.material.slider.Slider;  
import com.squareup.picasso.Picasso;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.IOException;  
import java.util.Random;  
  
import okhttp3.Call;  
import okhttp3.Callback;  
import okhttp3.HttpUrl;  
import okhttp3.OkHttpClient;  
import okhttp3.Request;  
import okhttp3.Response;  
import okhttp3.ResponseBody;  
  
public class DogMatcher extends AppCompatActivity {  
  
 @Override  
 public void onBackPressed() {  
  
 }  
  
 ImageView ImDog;  
  
 TextView tvDogName, tvMinLife, tvMaxLife, tvMinWeight, tvMaxWeight, tvMinHeight, tvMaxHeight;  
 Slider slChildren, slODogs, slPlayful, slProtective, slTrain, slBark;  
  
 LinearLayout llButton;  
 TextView tvDisplay;  
  
 // Define an array of random dogs  
 String[] randomDogs = {"Shih Tzu", "Dalmatian", "Airedale Terrier",  
 "Alaskan Malamute", "American Foxhound", "American Staffordshire Terrier", "American Water Spaniel",  
 "Anatolian Shepherd Dog", "Australian Cattle Dog", "Akita", "Affenpinscher", "Afghan Hound",  
 "American English Coonhound", "American Eskimo Dog", "Australian Shepherd", "Australian Terrier",  
 "Basenji", "Beagle", "Bulldog", "Basset Hound", "Bearded Collie", "Beauceron", "Bedlington Terrier",  
 "Belgian Malinois", "Belgian Sheepdog", "Belgian Tervuren", "Bergamasco", "Bernese Mountain Dog",  
 "Bichon Frise", "Black and Tan Coonhound", "Berger Picard", "Black Russian Terrier", "Bloodhound",  
 "Boerboel", "Border Collie", "Bluetick Coonhound", "Border Terrier", "Borzoi", "Boston Terrier",  
 "Boxer", "Boykin Spaniel", "Briard", "Brittany", "Brussels Griffon", "Bulldog", "Bull Terrier",  
 "Cairn Terrier", "Canaan Dog", "Bullmastiff", "Cane Corso", "Cardigan Welsh Corgi", "Cavalier King Charles Spaniel",  
 "Cesky Terrier", "Chesapeake Bay Retriever", "Chihuahua", "Chow Chow", "Chinese Crested",  
 "Chinese Shar-Pei", "Chinook", "Cirneco dell’Etna", "Clumber Spaniel", "Cocker Spaniel", "Collie", "Corgi",  
 "Coton de Tulear", "Curly-Coated Retriever", "Dachshund", "Dandie Dinmont Terrier",  
 "Doberman Pinscher", "Dogue de Bordeaux", "English Cocker Spaniel", "English Foxhound", "English Setter",  
 "English Toy Spaniel", "Entlebucher Mountain Dog", "Field Spaniel", "Finnish Lapphund", "German Pinscher",  
 "French Bulldog", "Flat-Coated Retriever", "Finnish Spitz", "German Shepherd Dog", "German Wirehaired Pointer", "Giant Schnauzer",  
 "Glen of Imaal Terrier", "Golden Retriever", "Gordon Setter", "Great Dane",  
 "Great Pyrenees", "Greater Swiss Mountain Dog", "Greyhound", "Harrier", "Havanese", "Ibizan Hound", "Icelandic Sheepdog",  
 "Irish Red and White Setter", "Irish Setter", "Irish Terrier", "Irish Water Spaniel", "Irish Wolfhound",  
 "Italian Greyhound", "Japanese Chin", "Keeshond", "Kerry Blue Terrier", "Komondor", "Kuvasz",  
 "Labrador Retriever", "Lagotto Romagnolo", "Lakeland Terrier", "Leonberger", "Lhasa Apso", "Löwchen",  
 "Maltese", "Manchester Terrier", "Mastiff", "Miniature American Shepherd", "Miniature Bull Terrier", "Miniature Pinscher",  
 "Miniature Schnauzer", "Neapolitan Mastiff", "Newfoundland", "Norfolk Terrier",  
 "Norwegian Buhund", "Norwegian Elkhound", "Norwegian Lundehund", "Norwich Terrier",  
 "Nova Scotia Duck Tolling Retriever", "Old English Sheepdog", "Otterhound", "Papillon", "Parson Russell Terrier",  
 "Pekingese", "Petit Basset Griffon Vendéen", "Pharaoh Houn", "Plott",  
 "Pointer", "Polish Lowland Sheepdog", "Pomeranian", "Poodle", "Portuguese Podengo Pequeno",  
 "Portuguese Water Dog", "Pug", "Puli", "Pyrenean Shepherd", "Rat Terrier", "Redbone Coonhound",  
 "Rhodesian Ridgeback", "Rottweiler", "Russell Terrier", "Saint Bernard", "Saluki", "Samoyed",  
 "Schipperke", "Scottish Deerhound", "Scottish Terrier", "Sealyham Terrier", "Shetland Sheepdog", "Shiba Inu",  
 "Siberian Husky", "Silky Terrier", "Skye Terrier", "Soft Coated Wheaten Terrier", "Spanish Water Dog", "Spinone Italiano",  
 "Staffordshire Bull Terrier", "Standard Schnauzer", "Sussex Spaniel", "Swedish Vallhund", "Tibetan Mastiff",  
 "Tibetan Spaniel", "Tibetan Terrier", "Treeing Walker Coonhound", "Vizsla", "Weimaraner", "Welsh Springer Spaniel",  
 "Welsh Terrier", "West Highland White Terrier", "Whippet", "Wire Fox Terrier", "Wirehaired Pointing Griffon",  
 "Wirehaired Vizsla", "Xoloitzcuintli", "Yorkshire Terrier"  
 };  
  
 private static final String *API\_ENDPOINT* = "https://api.api-ninjas.com/v1/dogs?name=";  
 String APIKey = "DXO7BxMYypdkZ8GORO/eZQ==03RsVosiuTeKMPr6";  
 String userAnswer;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*dog\_matcher*);  
  
 //get userAnswers from the dogmatcherr activity  
 Intent intent = getIntent();  
 userAnswer = intent.getStringExtra("userAnswer");  
  
 initialize();  
 }  
  
 private void initialize() {  
  
 llButton = findViewById(R.id.*llButton*);  
 tvDisplay = findViewById(R.id.*tvDisplay*);  
  
 //Displaying  
 tvDogName = findViewById(R.id.*tvDogName*);  
 tvMinLife = findViewById(R.id.*tvMinLife*);  
  
 tvMinWeight = findViewById(R.id.*tvMinWeight*);  
  
 tvMinHeight = findViewById(R.id.*tvMinHeight*);  
  
  
 slChildren = findViewById(R.id.*slChildren*);  
 slODogs = findViewById(R.id.*slDog*);  
 slPlayful = findViewById(R.id.*slPlayful*);  
 slProtective = findViewById(R.id.*slProtective*);  
 slTrain = findViewById(R.id.*slTrain*);  
 slBark = findViewById(R.id.*slBark*);  
 ;  
  
 ImDog = findViewById(R.id.*ImDog*);  
  
  
 llButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 String userPreference = userAnswer;  
 String selectedDogBreed = getDogBestForYou(userPreference);  
  
 String url = *API\_ENDPOINT* + selectedDogBreed;  
 //tvDisplay.setText(userPreference);  
  
  
 OkHttpClient client = new OkHttpClient();  
 Request request = new Request.Builder()  
 .url(url)  
 .addHeader("X-Api-Key", APIKey)  
 .build();  
  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Call call, IOException e) {  
 e.printStackTrace();  
 }  
  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 if (response.isSuccessful()) {  
 ResponseBody responseBody = response.body();  
 if (responseBody != null) {  
 String json = responseBody.string();  
 try {  
 JSONArray jsonArray = new JSONArray(json);  
  
 if (jsonArray.length() > 0) {  
 for (int i = 0; i < jsonArray.length(); i++) {  
 JSONObject jsonObject = jsonArray.getJSONObject(i);  
 String fetchedBreedName = jsonObject.getString("name");  
  
 if (fetchedBreedName.equalsIgnoreCase(fetchedBreedName)) {  
 String minLife = jsonObject.getString("min\_life\_expectancy");  
  
  
 String minWeight = jsonObject.getString("min\_weight\_female");  
  
 String minHeight = jsonObject.getString("min\_height\_female");  
  
 String imageUrl = jsonObject.getString("image\_link");  
  
 String gChildren = jsonObject.getString("good\_with\_children");  
 String gDogs = jsonObject.getString("good\_with\_other\_dogs");  
 String playful = jsonObject.getString("playfulness");  
 String protective = jsonObject.getString("protectiveness");  
 String train = jsonObject.getString("trainability");  
 String barking = jsonObject.getString("barking");  
  
  
 // Create an intent  
 Intent intent1 = new Intent(DogMatcher.this, BestDog.class);  
  
 // Put extra data in the intent  
 intent1.putExtra("name", fetchedBreedName);  
 intent1.putExtra("min\_life", minLife);  
  
 intent1.putExtra("min\_weight", minWeight);  
  
 intent1.putExtra("min\_height", minHeight);  
  
 intent1.putExtra("image\_url", imageUrl);  
  
 intent1.putExtra("gChildren", gChildren);  
 intent1.putExtra("gDogs", gDogs);  
 intent1.putExtra("playful", playful);  
 intent1.putExtra("protective", protective);  
 intent1.putExtra("train", train);  
 intent1.putExtra("barking", barking);  
  
 //start intent  
 startActivity(intent1);  
  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 }  
 });  
  
 // Break the loop since we found the breed  
 break;  
 }  
 }  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 } else {  
 // Handle the case when the request is not successful  
 }  
 }  
 });  
 }  
 });  
 }  
  
 private String getDogBestForYou(String userAnswer) {  
 //Comparing user answers with the set of possible answers  
  
 if (userAnswer.equals("NO HP LOW BASIC 4HOURS")) {  
 return "Pomeranian";  
 } else if (userAnswer.equals("NO PF LOW ADVANCE HOME")) {  
 return "Labrador Retriever";  
 } else if (userAnswer.equals("NO L HIGH BASIC 8HOURS")) {  
 return "BEAGLE";  
 } else if (userAnswer.equals("YES UNDER 12 HP HIGH ADVANCE 4HOURS")) {  
 return "Doberman Pinscher";  
 } else if (userAnswer.equals("YES UNDER 12 PF MEDIUM BASIC HOME")) {  
 return "Vizsla";  
 } else if (userAnswer.equals("YES UNDER 12 L MEDIUM ADVANCE HOME")) {  
 return "Poodle";  
 } else if (userAnswer.equals("YES ABOVE 12 HP LOW BASIC 4HOURS")) {  
 return "Belgian Malinois";  
 } else if (userAnswer.equals("YES ABOVE 12 PF HIGH ADVANCE HOME")) {  
 return "Collie";  
 } else if (userAnswer.equals("YES ABOVE 12 L MEDIUM BASIC 8HOURS")) {  
 return "Cavalier King";  
 } else if (userAnswer.equals("NO PF HIGH ADVANCE 4HOURS")) {  
 return "Shetland Sheepdog";  
 } else if (userAnswer.equals("NO L MEDIUM BASIC HOME")) {  
 return "Boxer";  
 } else if (userAnswer.equals("NO HP MEDIUM ADVANCE 8HOURS")) {  
 return "Rottweiler";  
 } else if (userAnswer.equals("YES UNDER 12 PF LOW BASIC 4HOURS")) {  
 return "Pug";  
 } else if (userAnswer.equals("YES UNDER 12 L HIGH ADVANCE HOME")) {  
 return "Great Dane";  
 } else if (userAnswer.equals("YES UNDER 12 HP MEDIUM BASIC 8HOURS")) {  
 return "Rhodesian";  
 } else if (userAnswer.equals("YES ABOVE 12 PF LOW ADVANCE 4HOURS")) {  
 return "American Eskimo";  
 } else if (userAnswer.equals("YES ABOVE 12 L HIGH BASIC HOME")) {  
 return "Shih Tzu";  
 } else if (userAnswer.equals("YES ABOVE 12 PF HIGH BASIC 4HOURS")) {  
 return "Siberian Husky";  
 } else if (userAnswer.equals("YES ABOVE 12 L MEDIUM ADVANCE 8HOURS")) {  
 return "Portuguese Water Dog";  
 } else if (userAnswer.equals("NO HP LOW BASIC HOME")) {  
 return "Doberman Pinscher";  
 } else if (userAnswer.equals("NO HP LOW BASIC 8HOURS")) {  
 return "Greyhound";  
 } else if (userAnswer.equals("NO HP LOW ADVANCE HOME")) {  
 return "Bullmastiff";  
 } else if (userAnswer.equals("NO HP LOW ADVANCE 4HOURS")) {  
 return "Akita";  
 } else if (userAnswer.equals("NO HP LOW ADVANCE 8HOURS")) {  
 return "Basenji";  
 } else if (userAnswer.equals("NO HP MEDIUM BASIC HOME")) {  
 return "German Shepherd";  
 } else if (userAnswer.equals("YES ABOVE 12 L HIGH ADVANCE 8HOURS")) {  
 return "Labrador Retriever";  
 } else if (userAnswer.equals("YES ABOVE 12 L HIGH ADVANCE 4HOURS")) {  
 return "Golden Retriever";  
 } else if (userAnswer.equals("YES ABOVE 12 L MEDIUM BASIC 4HOURS")) {  
 return "Beagle";  
 } else if (userAnswer.equals("YES ABOVE 12 L HIGH BASIC 8HOURS")) {  
 return "Bichon Frise";  
 } else if (userAnswer.equals("YES ABOVE 12 PF MEDIUM ADVANCE 4HOURS")) {  
 return "Australian Shepherd";  
 } else if (userAnswer.equals("YES ABOVE 12 HP HIGH ADVANCE HOME")) {  
 return "Rottweiler";  
 } else if (userAnswer.equals("YES ABOVE 12 HP HIGH BASIC 8HOURS")) {  
 return "Newfoundland";  
 } else if (userAnswer.equals("YES ABOVE 12 HP HIGH BASIC 4HOURS")) {  
 return "Boxer";  
 } else if (userAnswer.equals("YES ABOVE 12 HP HIGH BASIC HOME")) {  
 return "German Shepherd";  
 } else if (userAnswer.equals("YES ABOVE 12 HP MEDIUM ADVANCE 8HOURS")) {  
 return "Giant Schnauzer";  
 } else if (userAnswer.equals("YES ABOVE 12 HP MEDIUM ADVANCE 4HOURS")) {  
 return "Belgian Tervuren";  
 } else if (userAnswer.equals("YES ABOVE 12 HP MEDIUM ADVANCE HOME")) {  
 return "Cane Corso";  
 } else if (userAnswer.equals("YES ABOVE 12 HP MEDIUM BASIC 8HOURS")) {  
 return "Chow Chow";  
 } else if (userAnswer.equals("YES ABOVE 12 L HIGH BASIC 4HOURS")) {  
 return "Cavalier King Charles Spaniel";  
 } else if (userAnswer.equals("YES ABOVE 12 L MEDIUM ADVANCE 4HOURS")) {  
 return "Australian Shepherd";  
 } else if (userAnswer.equals("YES ABOVE 12 L MEDIUM ADVANCE HOME")) {  
 return "Bernese Mountain Dog";  
 } else if (userAnswer.equals("YES ABOVE 12 L MEDIUM BASIC HOME")) {  
 return "Cocker Spaniel";  
 } else if (userAnswer.equals("YES ABOVE 12 L LOW ADVANCE 8HOURS")) {  
 return "Chinese Crested";  
 } else if (userAnswer.equals("YES ABOVE 12 L LOW ADVANCE 4HOURS")) {  
 return "Whippet";  
 } else if (userAnswer.equals("YES ABOVE 12 L LOW ADVANCE HOME")) {  
 return "Irish Wolfhound";  
 } else if (userAnswer.equals("NO PF MEDIUM ADVANCE HOME")) {  
 return "Boxer";  
 } else if (userAnswer.equals("NO PF MEDIUM BASIC 8HOURS")) {  
 return "Shiba Inu";  
 } else if (userAnswer.equals("NO PF MEDIUM BASIC 4HOURS")) {  
 return "Australian Shepherd";  
 } else if (userAnswer.equals("NO PF MEDIUM BASIC HOME")) {  
 return "Bernese Mountain Dog";  
 } else if (userAnswer.equals("NO PF LOW BASIC 8HOURS")) {  
 return "Bullmastiff";  
 } else if (userAnswer.equals("YES ABOVE 12 PF MEDIUM ADVANCE 8HOURS")) {  
 return "Giant Schnauzer";  
 } else if (userAnswer.equals("YES ABOVE 12 HP MEDIUM BASIC 4HOURS")) {  
 return "Portuguese Water Dog";  
 } else if (userAnswer.equals("YES ABOVE 12 HP LOW BASIC 8HOURS")) {  
 return "Rhodesian Ridgeback";  
 } else if (userAnswer.equals("YES ABOVE 12 HP LOW BASIC HOME")) {  
 return "Leonberger";  
 } else if (userAnswer.equals("YES UNDER 12 L HIGH ADVANCE 8HOURS")) {  
 return "English Springer Spaniel";  
 } else if (userAnswer.equals("YES UNDER 12 L HIGH ADVANCE 4HOURS")) {  
 return "Border Collie";  
 } else if (userAnswer.equals("YES UNDER 12 L HIGH BASIC 8HOURS")) {  
 return "Beagle";  
 } else if (userAnswer.equals("YES UNDER 12 L HIGH BASIC 4HOURS")) {  
 return "Shih Tzu";  
 } else if (userAnswer.equals("NO HP HIGH BASIC HOME")) {  
 return "Akita";  
 } else if (userAnswer.equals("YES ABOVE 12 PF HIGH BASIC HOME")) {  
 return "English Setter";  
 } else if (userAnswer.equals("YES ABOVE 12 PF HIGH BASIC 8HOURS")) {  
 return "Vizsla";  
 } else {  
  
// Create an instance of the Random class  
 Random random = new Random();  
// Get a random index within the range of the randomDogs array  
 int randomIndex = random.nextInt(randomDogs.length);  
// Return the randomly selected dog  
 return randomDogs[randomIndex];  
 }  
  
 }  
  
}

**dogmatcherr.java**

package com.example.newtry;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageButton;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class dogmatcherr extends AppCompatActivity {  
 TextView tvQuestion;  
 RadioGroup rgChoices;  
 ImageButton btnNext;  
  
 private String[][] answerChoices = {  
 {"YES, I HAVE KIDS UNDER 12 YEARS OLD", "YES, I HAVE KIDS OVER 12 YEARS OLD", "NO, I DON'T HAVE KIDS AT WALA AKONG BALAK"},  
 {"HIGHLY PROTECTIVE, YUNG KAYA AKONG IPAGLABAN", "PROTECTIVE PERO SAKTONG FRIENDLY DIN WITH STRANGERS", "LOVES EVERYONE AT MAHAL NA MAHAL AKO"},  
 {"I PREFER A DOG NA DI MASYADONG NATAHOL", "OKAY NA SAKIN YUNG KONTING BARK BARK", "BARKING IS NOT AN ISSUE FOR ME, KAHIT MAMAOS SYA KAKATAHOL"},  
 {"BASIC OBEDIENCE", "ADVANCED OBEDIENCE", "NOT AN ISSUE FOR ME"},  
 {"I OR SOMEONE ELSE WILL BE HOME MOST OF THE TIME", "ONLY ABOUT 4 HOURS AT A TIME", "MY DOG SHOULD BE FINE BY HIMSELF FOR AT LEAST 8 HOURS"}  
 };  
 private String[] questions = {  
 "Will they have any kids to play with?",  
 "Which best describes your future pet’s personality?",  
 "In terms of barking, how much noise can you tolerate?",  
 "How much training will your new dog receive?",  
 "How much time will your new dog be spending alone?"  
 };  
 private String[] answers = new String[questions.length];  
 private int currentQuestionIndex = 0;  
 private StringBuilder userAnswer = new StringBuilder();  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*dog\_matcherr*);  
  
 tvQuestion = findViewById(R.id.*tvQuestion*);  
 rgChoices = findViewById(R.id.*rgChoices*);  
 btnNext = findViewById(R.id.*btnNext*);  
  
 userAnswer = new StringBuilder();  
  
 btnNext.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 int checkedRadioButtonId = rgChoices.getCheckedRadioButtonId();  
 if (checkedRadioButtonId != -1) {  
 RadioButton selectedRadioButton = findViewById(checkedRadioButtonId);  
  
 answers[currentQuestionIndex] = selectedRadioButton.getText().toString();  
 userAnswer.append(answers[currentQuestionIndex]);  
  
 currentQuestionIndex++;  
  
 if (currentQuestionIndex >= questions.length) {  
 getAnswers();  
 } else {  
 showNextQuestion();  
 }  
 }  
 }  
 });  
  
 showNextQuestion();  
  
  
 BottomNavigationView bottomNavigationView1 = findViewById(R.id.*bottomNavgationView*);  
 bottomNavigationView1.setSelectedItemId(R.id.*bottom\_matcher*);  
  
 bottomNavigationView1.setOnItemSelectedListener(item -> {  
 if (item.getItemId() == R.id.*bottom\_home*) {  
 startActivity(new Intent(getApplicationContext(), Home.class));  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_matcher*) {  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_book*) {  
 startActivity(new Intent(getApplicationContext(), Encyclopediaa.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_random*) {  
 startActivity(new Intent(getApplicationContext(), DogAlbumActivity.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_logout*) {  
 showLogoutConfirmationDialog();  
 return true;  
 } else {  
 return false;  
 }  
 });  
  
  
 }  
  
 private void showNextQuestion() {  
 rgChoices.clearCheck();  
 tvQuestion.setText(questions[currentQuestionIndex]);  
  
 RadioButton radioButton1 = findViewById(R.id.*rb1*);  
 RadioButton radioButton2 = findViewById(R.id.*rb2*);  
 RadioButton radioButton3 = findViewById(R.id.*rb3*);  
  
 radioButton1.setText(answerChoices[currentQuestionIndex][0]);  
 radioButton2.setText(answerChoices[currentQuestionIndex][1]);  
 radioButton3.setText(answerChoices[currentQuestionIndex][2]);  
 }  
  
 private void getAnswers() {  
 //Getting answers  
 StringBuilder userAnswer = new StringBuilder();  
  
 String kidsAnswer = answers[0];  
 String personaAnswer = answers[1];  
 String barkAnswer = answers[2];  
 String trainingAnswer = answers[3];  
 String timeAnswer = answers[4];  
  
 if (kidsAnswer.equals(answerChoices[0][0])) {  
 kidsAnswer = "YES UNDER 12";  
 } else if (kidsAnswer.equals(answerChoices[0][1])) {  
 kidsAnswer = "YES ABOVE 12";  
 } else if (kidsAnswer.equals(answerChoices[0][2])) {  
 kidsAnswer = "NO";  
 }  
 userAnswer.append(kidsAnswer).append(" ");  
  
 if (personaAnswer.equals(answerChoices[1][0])) {  
 personaAnswer = "HP";  
 } else if (personaAnswer.equals(answerChoices[1][1])) {  
 personaAnswer = "PF";  
 } else if (personaAnswer.equals(answerChoices[1][2])) {  
 personaAnswer = "L";  
 }  
 userAnswer.append(personaAnswer).append(" ");  
  
 if (barkAnswer.equals(answerChoices[2][0])) {  
 barkAnswer = "LOW";  
 } else if (barkAnswer.equals(answerChoices[2][1])) {  
 barkAnswer = "MEDIUM";  
 } else if (barkAnswer.equals(answerChoices[2][2])) {  
 barkAnswer = "HIGH";  
 }  
 userAnswer.append(barkAnswer).append(" ");  
  
 if (trainingAnswer.equals(answerChoices[3][0])) {  
 trainingAnswer = "BASIC";  
 } else if (trainingAnswer.equals(answerChoices[3][1])) {  
 trainingAnswer = "ADVANCE";  
 } else if (trainingAnswer.equals(answerChoices[3][2])) {  
 trainingAnswer = "BASIC";  
 }  
 userAnswer.append(trainingAnswer).append(" ");  
  
 if (timeAnswer.equals(answerChoices[4][0])) {  
 timeAnswer = "HOME";  
 } else if (timeAnswer.equals(answerChoices[4][1])) {  
 timeAnswer = "4HOURS";  
 } else if (timeAnswer.equals(answerChoices[4][2])) {  
 timeAnswer = "8HOURS";  
 }  
 userAnswer.append(timeAnswer);  
  
 rgChoices.setVisibility(View.*GONE*);  
 btnNext.setVisibility(View.*GONE*);  
  
 // Pass the userAnswer to the new activity  
 Intent intent = new Intent(dogmatcherr.this, DogMatcher.class);  
 intent.putExtra("userAnswer", userAnswer.toString());  
 startActivity(intent);  
 }  
  
 private void showLogoutConfirmationDialog() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());  
 builder.setTitle("Logout");  
 builder.setMessage("Are you sure you want to logout?");  
 builder.setCancelable(false);  
 builder.setPositiveButton("Logout", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 FirebaseAuth.*getInstance*().signOut();  
 redirectToLogin();  
 }  
 });  
  
 builder.setNegativeButton("Cancel", null);  
  
 AlertDialog dialog = builder.create();  
 dialog.show();  
 }  
  
 private void redirectToLogin() {  
 Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivity(intent);  
 finish();  
 }  
}

**EditDogActivity.java**

package com.example.newtry;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.drawable.BitmapDrawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.util.Base64;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.firestore.DocumentReference;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.UploadTask;  
import com.squareup.picasso.Picasso;  
  
import java.io.ByteArrayOutputStream;  
import java.io.IOException;  
import java.util.HashMap;  
import java.util.Map;  
  
public class EditDogActivity extends AppCompatActivity {  
  
 private EditText nameEditText;  
 private EditText breedEditText;  
 private EditText ageEditText;  
 private Button updateButton;  
 private ImageView pictureImageView;  
 private Dog dog;  
 private String dogId;  
 private FirebaseFirestore db;  
 private FirebaseAuth auth;  
 private boolean imageChanged = false;  
 private static final int *PICK\_IMAGE\_REQUEST* = 1;  
 private Uri pictureUri;  
 private StorageReference storageRef;  
  
 Button btnBack;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*edit\_dog*);  
  
 nameEditText = findViewById(R.id.*nameEditText*);  
 breedEditText = findViewById(R.id.*breedEditText*);  
 ageEditText = findViewById(R.id.*ageEditText*);  
 updateButton = findViewById(R.id.*updateButton*);  
 pictureImageView = findViewById(R.id.*pictureImageView*);  
 btnBack = findViewById(R.id.*btnBack*);  
  
 db = FirebaseFirestore.*getInstance*();  
 auth = FirebaseAuth.*getInstance*();  
 storageRef = FirebaseStorage.*getInstance*().getReference();  
  
 if (getIntent().hasExtra("dog")) {  
 dog = getIntent().getParcelableExtra("dog");  
 dogId = getIntent().getStringExtra("dogId");  
 if (dog != null) {  
 dog.setId(dogId);  
 nameEditText.setText(dog.getName());  
 breedEditText.setText(dog.getBreed());  
 ageEditText.setText(dog.getAge());  
 Picasso.*get*().load(dog.getPicture()).into(pictureImageView);  
 }  
 }  
  
 updateButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 updateDog();  
 }  
 });  
  
 pictureImageView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 openImagePicker();  
 }  
 });  
  
 btnBack.setOnClickListener(v -> {  
 finish();  
 });  
 }  
  
 private void updateDog() {  
 String name = nameEditText.getText().toString().trim();  
 String breed = breedEditText.getText().toString().trim();  
 String age = ageEditText.getText().toString().trim();  
  
 // Perform validation for name, breed, and age  
 if (name.isEmpty()) {  
 nameEditText.setError("Please enter a name");  
 return;  
 }  
  
 if (breed.isEmpty()) {  
 breedEditText.setError("Please enter a breed");  
 return;  
 }  
  
 if (age.isEmpty()) {  
 ageEditText.setError("Please enter an age");  
 return;  
 }  
  
 if (dog == null || dog.getId() == null) {  
 Toast.*makeText*(EditDogActivity.this, "Invalid dog data", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 // Create a HashMap to hold the updated fields  
 Map<String, Object> updates = new HashMap<>();  
  
 // Check if the name has changed and update it  
 if (!name.equals(dog.getName())) {  
 updates.put("name", name);  
 }  
  
 if (!breed.equals(dog.getBreed())) {  
 updates.put("breed", breed);  
 }  
  
 // Check if the age has changed and update it  
 if (!age.equals(dog.getAge())) {  
 updates.put("age", age);  
 }  
  
 // Check if the image has changed and update it  
 if (imageChanged) {  
 uploadImageAndUpdateDog(updates);  
 } else {  
 // Update the dog details in Firestore if there are any changes  
 updateDogDetails(updates, null);  
 }  
 }  
  
 private void uploadImageAndUpdateDog(Map<String, Object> updates) {  
 // Get the bitmap from the ImageView  
 BitmapDrawable drawable = (BitmapDrawable) pictureImageView.getDrawable();  
 Bitmap imageBitmap = drawable.getBitmap();  
  
 // Convert bitmap to byte array  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 imageBitmap.compress(Bitmap.CompressFormat.*JPEG*, 100, baos);  
 byte[] imageData = baos.toByteArray();  
  
 // Define the path and filename in Firebase Storage  
 String imageFileName = "dogs/" + dogId + ".jpg";  
 StorageReference imageRef = storageRef.child(imageFileName);  
  
 // Upload the image to Firebase Storage  
 UploadTask uploadTask = imageRef.putBytes(imageData);  
 uploadTask.addOnSuccessListener(taskSnapshot -> {  
 // Get the download URL of the uploaded image  
 imageRef.getDownloadUrl().addOnSuccessListener(uri -> {  
 String imageUrl = uri.toString();  
 updates.put("picture", imageUrl);  
  
 // Update the dog details in Firestore  
 updateDogDetails(updates, imageUrl);  
 });  
 }).addOnFailureListener(e -> {  
 Toast.*makeText*(EditDogActivity.this, "Error uploading image", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 });  
 }  
  
 private void updateDogDetails(Map<String, Object> updates, String imageUrl) {  
 // Update the dog details in Firestore if there are any changes  
 if (!updates.isEmpty()) {  
 DocumentReference dogRef = db.collection("users")  
 .document(auth.getCurrentUser().getUid())  
 .collection("dogs")  
 .document(dog.getId());  
  
 dogRef.update(updates)  
 .addOnSuccessListener(aVoid -> {  
 Toast.*makeText*(EditDogActivity.this, "Dog updated successfully", Toast.*LENGTH\_SHORT*).show();  
 finish(); // Finish the activity and go back to DogAlbumActivity  
 })  
 .addOnFailureListener(e -> {  
 Toast.*makeText*(EditDogActivity.this, "Error updating dog", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 });  
 } else {  
 // No changes were made  
 Toast.*makeText*(EditDogActivity.this, "No changes made", Toast.*LENGTH\_SHORT*).show();  
 finish(); // Finish the activity and go back to DogAlbumActivity  
 }  
 }  
  
 private void updateDogDetails(Map<String, Object> updates) {  
 updateDogDetails(updates, null);  
 }  
  
 private void openImagePicker() {  
 Intent intent = new Intent(Intent.*ACTION\_OPEN\_DOCUMENT*);  
 intent.setType("image/\*");  
 intent.addCategory(Intent.*CATEGORY\_OPENABLE*);  
 startActivityForResult(intent, *PICK\_IMAGE\_REQUEST*);  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == *PICK\_IMAGE\_REQUEST* && resultCode == *RESULT\_OK* && data != null && data.getData() != null) {  
 pictureUri = data.getData();  
 pictureImageView.setImageURI(pictureUri);  
 imageChanged = true;  
 }  
 }  
}

**Encyclopediaa.java**

package com.example.newtry;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Looper;  
import android.text.Editable;  
import android.text.TextWatcher;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.example.newtry.DogApiService;  
import com.example.newtry.DogBreed;  
import com.example.newtry.DogBreedAdapter;  
import com.example.newtry.DogDetails;  
import com.example.newtry.R;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.auth.FirebaseAuth;  
  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Locale;  
public class Encyclopediaa extends AppCompatActivity implements DogApiService.DogBreedsCallback, DogBreedAdapter.OnItemClickListener {  
  
 private RecyclerView recyclerView;  
 private DogBreedAdapter adapter;  
 private EditText etSearch;  
 private List<DogBreed> dogBreeds;  
 private List<DogBreed> filteredDogBreeds;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*encyclopedia\_recycler*);  
  
 recyclerView = findViewById(R.id.*recycler\_view*);  
 etSearch = findViewById(R.id.*etSearch*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
 DogApiService apiService = new DogApiService();  
 apiService.fetchDogBreeds(this);  
  
 etSearch.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
 }  
  
 @Override  
 public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {  
 String query = charSequence.toString().trim().toLowerCase(Locale.*getDefault*());  
 filterDogBreeds(query); // Call the method to filter the dog breeds based on the search query  
 }  
  
 @Override  
 public void afterTextChanged(Editable editable) {  
 }  
 });  
  
 BottomNavigationView bottomNavigationView1 = findViewById(R.id.*bottomNavgationView*);  
 bottomNavigationView1.setSelectedItemId(R.id.*bottom\_book*);  
  
 bottomNavigationView1.setOnItemSelectedListener(item -> {  
 if (item.getItemId() == R.id.*bottom\_home*) {  
 startActivity(new Intent(getApplicationContext(), Home.class));  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_matcher*) {  
 startActivity(new Intent(getApplicationContext(), startDogMatcher.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_book*) {  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_random*) {  
 startActivity(new Intent(getApplicationContext(), DogAlbumActivity.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_logout*) {  
 showLogoutConfirmationDialog();  
 return true;  
 } else {  
 return false;  
 }  
 });  
 }  
  
 @Override  
 public void onItemClick(DogBreed dogBreed) {  
 if (dogBreed == null) {  
 //Toast.makeText(this, "No dog breed selected", Toast.LENGTH\_SHORT).show();  
 } else {  
 // Toast.makeText(this, "Selected dog breed: " + dogBreed.getName(), Toast.LENGTH\_SHORT).show();  
 Intent intent = new Intent(this, DogDetails.class);  
 intent.putExtra("selectedDog", dogBreed);  
 startActivity(intent);  
 }  
 }  
  
 private void filterDogBreeds(String query) {  
 filteredDogBreeds.clear();  
  
 if (query.isEmpty()) {  
 filteredDogBreeds.addAll(dogBreeds);  
 } else {  
 for (DogBreed dogBreed : dogBreeds) {  
 if (dogBreed.getName().toLowerCase(Locale.*getDefault*()).startsWith(query.toLowerCase(Locale.*getDefault*()))) {  
 filteredDogBreeds.add(dogBreed);  
 }  
 }  
 }  
 if (adapter != null) {  
 adapter.notifyDataSetChanged();  
 }  
 }  
  
  
 @Override  
 public void onSuccess(final List<DogBreed> dogBreeds) {  
 this.dogBreeds = dogBreeds;  
 this.filteredDogBreeds = new ArrayList<>(dogBreeds);  
  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 adapter = new DogBreedAdapter(filteredDogBreeds, Encyclopediaa.this);  
 recyclerView.setAdapter(adapter);  
 }  
 });  
 }  
  
  
 @Override  
 public void onError(final IOException e) {  
 // Update UI on the main thread  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(Encyclopediaa.this, "An error occurred: " + e.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private void showLogoutConfirmationDialog() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());  
 builder.setTitle("Logout");  
 builder.setMessage("Are you sure you want to logout?");  
 builder.setCancelable(false);  
 builder.setPositiveButton("Logout", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 FirebaseAuth.*getInstance*().signOut();  
 redirectToLogin();  
 }  
 });  
  
 builder.setNegativeButton("Cancel", null);  
  
 AlertDialog dialog = builder.create();  
 dialog.show();  
 }  
  
 private void redirectToLogin() {  
 Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
 startActivity(intent);  
 finish();  
 }  
  
}

**Home.java**

package com.example.newtry;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.drawable.Drawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Environment;  
import android.os.Handler;  
import android.os.Looper;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.FrameLayout;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.airbnb.lottie.LottieAnimationView;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.auth.FirebaseAuth;  
import com.squareup.picasso.Picasso;  
import com.squareup.picasso.Target;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.util.Random;  
  
import okhttp3.Call;  
import okhttp3.Callback;  
import okhttp3.OkHttpClient;  
import okhttp3.Request;  
import okhttp3.Response;  
import okhttp3.ResponseBody;  
  
public class Home extends AppCompatActivity {  
 private static final String *API\_ENDPOINT* = "https://api.thedogapi.com/v1/breeds/";  
 private ImageView dogImageView;  
 private TextView breedNameTextView;  
 private TextView bredForTextView;  
 private TextView breedGroupTextView;  
 private TextView lifeSpanTextView;  
 private TextView temperamentTextView;  
 Button btnDownload;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*home*);  
  
 initialize();  
  
 initializeViews();  
  
 fetchRandomDogBreed();  
  
 }  
  
 private void initialize() {  
  
  
 BottomNavigationView bottomNavigationView1 = findViewById(R.id.*bottomNavgationView*);  
 bottomNavigationView1.setSelectedItemId(R.id.*bottom\_home*);  
  
  
 bottomNavigationView1.setOnItemSelectedListener(item -> {  
 if (item.getItemId() == R.id.*bottom\_home*) {  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_matcher*) {  
 startActivity(new Intent(getApplicationContext(), startDogMatcher.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_book*) {  
 startActivity(new Intent(getApplicationContext(), Encyclopediaa.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_random*) {  
 startActivity(new Intent(getApplicationContext(), DogAlbumActivity.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_logout*) {  
 showLogoutConfirmationDialog();  
 return true;  
 } else {  
 return false;  
 }  
 });  
 }  
  
 private void initializeViews() {  
 dogImageView = findViewById(R.id.*dog\_image*);  
 breedNameTextView = findViewById(R.id.*breed\_name\_text*);  
 bredForTextView = findViewById(R.id.*bred\_for\_text*);  
 breedGroupTextView = findViewById(R.id.*breed\_group\_text*);  
 lifeSpanTextView = findViewById(R.id.*life\_span\_text*);  
 temperamentTextView = findViewById(R.id.*temperament\_text*);  
  
  
 //hide contents  
 FrameLayout contents = findViewById(R.id.*contents*);  
 contents.setVisibility(View.*GONE*);  
 BottomNavigationView bottomNavgationView = findViewById(R.id.*bottomNavgationView*);  
 bottomNavgationView.setVisibility(View.*GONE*);  
  
  
 //Show contents  
 //FrameLayout animationContainer = findViewById(R.id.animationContainer);  
 LottieAnimationView animationView = findViewById(R.id.*animationView*);  
 animationView.setAnimation("dogwalkingg.json");  
 animationView.playAnimation();  
  
 // Stop the animation after 3 seconds  
 new Handler(Looper.*getMainLooper*()).postDelayed(new Runnable() {  
 @Override  
 public void run() {  
  
 animationView.cancelAnimation();  
 animationView.setVisibility(View.*GONE*);  
  
 contents.setVisibility(View.*VISIBLE*);  
 bottomNavgationView.setVisibility(View.*VISIBLE*);  
  
 }  
 }, 5000);  
  
  
 }  
  
 private void fetchRandomDogBreed() {  
 OkHttpClient client = new OkHttpClient();  
 Request request = new Request.Builder()  
 .url(*API\_ENDPOINT*)  
 .build();  
  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Call call, IOException e) {  
 e.printStackTrace();  
 }  
  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 if (response.isSuccessful()) {  
 ResponseBody responseBody = response.body();  
 if (responseBody != null) {  
 String json = responseBody.string();  
 try {  
 JSONArray jsonArray = new JSONArray(json);  
  
 int maxId = jsonArray.length(); // Get the maximum ID available  
  
 int randomId = generateRandomId(maxId); // Generate a random ID within the range  
  
 JSONObject jsonObject = jsonArray.getJSONObject(randomId);  
  
 String breedName = jsonObject.getString("name");  
 String bredFor = jsonObject.getString("bred\_for");  
 String breedGroup = jsonObject.getString("breed\_group");  
 String lifeSpan = jsonObject.getString("life\_span");  
 String temperament = jsonObject.getString("temperament");  
 String imageUrl = jsonObject.getString("reference\_image\_id");  
  
 btnDownload = findViewById(R.id.*btnDownload*);  
 btnDownload.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 downloadImage("https://cdn2.thedogapi.com/images/" + imageUrl + ".jpg");  
 }  
 });  
  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 breedNameTextView.setText(breedName);  
 bredForTextView.setText("Bred for: " + bredFor);  
 breedGroupTextView.setText("Breed Group: " + breedGroup);  
 lifeSpanTextView.setText("Lifespan: " + lifeSpan);  
 temperamentTextView.setText("Character: " + temperament);  
  
 Picasso.*get*().load("https://cdn2.thedogapi.com/images/" + imageUrl + ".jpg")  
 .into(dogImageView);  
 }  
  
 });  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
  
 }  
 } else {  
 // Handle the case when the request is not successful  
 }  
 }  
 });  
  
  
 }  
  
 private int generateRandomId(int maxId) {  
 Random random = new Random();  
 return random.nextInt(maxId);  
 }  
  
 private void downloadImage(String imageUrl) {  
 Target imageTarget = new Target() {  
 @Override  
 public void onBitmapLoaded(Bitmap bitmap, Picasso.LoadedFrom from) {  
 saveImageToGallery(bitmap);  
 }  
  
 @Override  
 public void onBitmapFailed(Exception e, Drawable errorDrawable) {  
 Toast.*makeText*(getApplicationContext(), "Failed to download image", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onPrepareLoad(Drawable placeHolderDrawable) {  
 }  
 };  
  
 Picasso.*get*().load(imageUrl).into(imageTarget);  
 }  
  
 private void saveImageToGallery(Bitmap bitmap) {  
 String filename = "image\_" + System.*currentTimeMillis*() + ".jpg";  
 File directory = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*);  
 File file = new File(directory, filename);  
  
 try (FileOutputStream fos = new FileOutputStream(file)) {  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*, 100, fos);  
 fos.flush();  
 // Add the image to the gallery  
 Uri imageUri = Uri.*fromFile*(file);  
 Intent mediaScanIntent = new Intent(Intent.*ACTION\_MEDIA\_SCANNER\_SCAN\_FILE*, imageUri);  
 sendBroadcast(mediaScanIntent);  
  
 Toast.*makeText*(this, "Image downloaded", Toast.*LENGTH\_SHORT*).show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 Toast.*makeText*(this, "Failed to save image", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void showLogoutConfirmationDialog() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(Home.this);  
 builder.setTitle("Logout");  
 builder.setMessage("Are you sure you want to logout?");  
 builder.setCancelable(false);  
 builder.setPositiveButton("Logout", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 FirebaseAuth.*getInstance*().signOut();  
 redirectToLogin();  
 }  
 });  
  
 builder.setNegativeButton("Cancel", null);  
  
 AlertDialog dialog = builder.create();  
 dialog.show();  
 }  
  
 private void redirectToLogin() {  
 Intent intent = new Intent(Home.this, MainActivity.class);  
 startActivity(intent);  
 finish();  
 }  
}

**MainActivity.java**

package com.example.newtry;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Looper;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.FrameLayout;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.airbnb.lottie.LottieAnimationView;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
public class MainActivity extends AppCompatActivity {  
 private FirebaseAuth mAuth;  
 EditText etEmail, etPass;  
 Button btnLogin, btnGo;  
 TextView tvNotreg;  
 private String userId;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 FirebaseApp.*initializeApp*(this);  
 mAuth = FirebaseAuth.*getInstance*();  
  
 etEmail = findViewById(R.id.*etEmail*);  
 etPass = findViewById(R.id.*etPass*);  
 btnLogin = findViewById(R.id.*btnLogin*);  
 tvNotreg = findViewById(R.id.*tvNotreg*);  
 //btnGo = findViewById(R.id.btnGo);  
  
  
 //hide contents  
 LinearLayout contents = findViewById(R.id.*contents*);  
 contents.setVisibility(View.*GONE*);  
  
  
 //Show contents  
 FrameLayout animationContainer = findViewById(R.id.*animationContainer*);  
 LottieAnimationView animationView = findViewById(R.id.*animationView*);  
 animationView.setAnimation("paww.json");  
 animationView.playAnimation();  
  
 // Stop the animation after 3 seconds  
 new Handler(Looper.*getMainLooper*()).postDelayed(new Runnable() {  
 @Override  
 public void run() {  
  
 animationView.cancelAnimation();  
 animationView.setVisibility(View.*GONE*);  
  
 contents.setVisibility(View.*VISIBLE*);  
  
 }  
 }, 3000);  
  
  
  
 btnLogin.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 login();  
  
 }  
 });  
  
 /\*btnGo.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this, Home.class));  
 }  
 });\*/  
  
 tvNotreg.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(MainActivity.this, regActivity.class));  
  
 }  
 });  
 }  
  
 private void login() {  
  
 String user = etEmail.getText().toString().trim();  
 String pass = etPass.getText().toString().trim();  
  
 if(user.isEmpty()){  
  
 etEmail.setError("You must enter Email Address");  
 return;  
 }  
  
 if(pass.isEmpty()){  
  
 etEmail.setError("You must enter Password");  
 return;  
 }  
 else{  
 mAuth.signInWithEmailAndPassword(user, pass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
  
 if(task.isSuccessful()){  
 FirebaseUser user = mAuth.getCurrentUser();  
 if (user != null) {  
 userId = user.getUid();  
 openHomeActivity();  
 }  
 Toast.*makeText*(MainActivity.this, "Login Successful", Toast.*LENGTH\_LONG*).show();  
 }  
 else {  
 Toast.*makeText*(MainActivity.this, "Login Failed"+task.getException().getMessage(), Toast.*LENGTH\_LONG*).show();  
  
 }  
  
 }  
 });  
  
 }  
 }  
  
  
 private void openHomeActivity() {  
 Intent intent = new Intent(MainActivity.this, Home.class);  
 intent.putExtra("userId", userId);  
 startActivity(intent);  
 finish();  
 }  
  
  
}

**RandomImage.java**

package com.example.newtry;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.graphics.Bitmap;  
import android.graphics.drawable.Drawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Environment;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.auth.FirebaseAuth;  
import com.squareup.picasso.Picasso;  
import com.squareup.picasso.Target;  
  
import org.json.JSONArray;  
import org.json.JSONObject;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
import okhttp3.Call;  
import okhttp3.Callback;  
import okhttp3.OkHttpClient;  
import okhttp3.Request;  
import okhttp3.Response;  
import okhttp3.ResponseBody;  
  
public class RandomImage extends AppCompatActivity {  
  
 private ImageView Imrandom;  
 private Button btnDownload, btnGenerate;  
  
 private OkHttpClient client;  
  
 private static final int *PERMISSION\_REQUEST\_CODE* = 100;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*random\_image*);  
  
 Imrandom = findViewById(R.id.*Imrandom*);  
 btnDownload = findViewById(R.id.*btnDownload*);  
 btnGenerate = findViewById(R.id.*btnGenerate*);  
  
 client = new OkHttpClient();  
  
 btnGenerate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 getRandomDogImage();  
 }  
 });  
 BottomNavigationView bottomNavigationView1 = findViewById(R.id.*bottomNavgationView*);  
 bottomNavigationView1.setSelectedItemId(R.id.*bottom\_random*);  
  
 bottomNavigationView1.setOnItemSelectedListener(item -> {  
 if (item.getItemId() == R.id.*bottom\_home*) {  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_matcher*) {  
 startActivity(new Intent(getApplicationContext(), startDogMatcher.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_book*) {  
 startActivity(new Intent(getApplicationContext(), Encyclopediaa.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_random*) {  
 startActivity(new Intent(getApplicationContext(), DogAlbumActivity.class));  
 overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);  
 return true;  
 } else if (item.getItemId() == R.id.*bottom\_logout*) {  
 FirebaseAuth.*getInstance*().signOut();  
 startActivity(new Intent(getApplicationContext(), regActivity.class));  
 return true;  
 } else {  
 return false;  
 }  
 });  
 }  
  
 private void getRandomDogImage() {  
 String url = "https://api.thedogapi.com/v1/images/search?format=json&limit=1";  
 Request request = new Request.Builder()  
 .url(url)  
 .addHeader("Content-Type", "application/json")  
 .addHeader("x-api-key", "live\_xCb7rxFkPpsp3AO5od2Ft8YRmpcklABORmdOdwKtEqREkruJlOYSNZi3dJsPzFb8")  
 .build();  
  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onResponse(@NonNull Call call, @NonNull Response response) throws IOException {  
 if (response.isSuccessful()) {  
 ResponseBody responseBody = response.body();  
 if (responseBody != null) {  
 String json = responseBody.string();  
 try {  
 JSONArray jsonArray = new JSONArray(json);  
 if (jsonArray.length() > 0) {  
 final JSONObject dogObject = jsonArray.getJSONObject(0);  
 final String imageUrl = dogObject.getString("url");  
  
 runOnUiThread(() -> {  
 loadImage(imageUrl);  
 btnDownload.setEnabled(true);  
 // Call the downloadImage() method here passing the imageUrl  
 btnDownload.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 downloadImage(imageUrl);  
 }  
 });  
  
 });  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
  
 @Override  
 public void onFailure(@NonNull Call call, @NonNull IOException e) {  
 e.printStackTrace();  
 }  
 });  
 }  
  
 private void loadImage(String imageUrl) {  
 Picasso.*get*().load(imageUrl).into(Imrandom);  
 }  
  
 public void onDownloadButtonClick(View view) {  
 if (hasWritePermission()) {  
 // The download is now triggered in the getRandomDogImage() method  
 } else {  
 requestWritePermission();  
 }  
 }  
  
 private boolean hasWritePermission() {  
 return ContextCompat.*checkSelfPermission*(this, Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*) == PackageManager.*PERMISSION\_GRANTED*;  
 }  
  
 private void requestWritePermission() {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*}, *PERMISSION\_REQUEST\_CODE*);  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
 if (requestCode == *PERMISSION\_REQUEST\_CODE*) {  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
  
 } else {  
 Toast.*makeText*(this, "Write permission denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
  
 private void downloadImage(String imageUrl) {  
 Target imageTarget = new Target() {  
 @Override  
 public void onBitmapLoaded(Bitmap bitmap, Picasso.LoadedFrom from) {  
 saveImageToGallery(bitmap);  
 }  
  
 @Override  
 public void onBitmapFailed(Exception e, Drawable errorDrawable) {  
 Toast.*makeText*(RandomImage.this, "Failed to download image", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onPrepareLoad(Drawable placeHolderDrawable) {  
 }  
 };  
  
 Picasso.*get*().load(imageUrl).into(imageTarget);  
 }  
 private void saveImageToGallery(Bitmap bitmap) {  
 String filename = "image\_" + System.*currentTimeMillis*() + ".jpg";  
 File directory = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*);  
 File file = new File(directory, filename);  
  
 try (FileOutputStream fos = new FileOutputStream(file)) {  
 bitmap.compress(Bitmap.CompressFormat.*JPEG*, 100, fos);  
 fos.flush();  
 // Add the image to the gallery  
 Uri imageUri = Uri.*fromFile*(file);  
 Intent mediaScanIntent = new Intent(Intent.*ACTION\_MEDIA\_SCANNER\_SCAN\_FILE*, imageUri);  
 sendBroadcast(mediaScanIntent);  
  
 Toast.*makeText*(this, "Image downloaded", Toast.*LENGTH\_SHORT*).show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 Toast.*makeText*(this, "Failed to save image", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

**regActivity.java**

package com.example.newtry;  
  
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.TextView;  
import android.widget.Toast;  
  
import com.example.newtry.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;  
  
public class regActivity extends AppCompatActivity {  
  
 FirebaseAuth mAuth;  
 EditText etRegEmail, etRegPass;  
 Button btnRegister;  
 TextView tvLog;  
  
 @Override  
 public void onBackPressed() {  
 }  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*reg\_activity*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
 btnRegister = findViewById(R.id.*btnRegister*);  
 etRegEmail = findViewById(R.id.*etRegEmail*);  
 etRegPass = findViewById(R.id.*etRegPass*);  
 tvLog = findViewById(R.id.*tvLog*);  
  
 btnRegister.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 register();  
  
 }  
 });  
  
  
 tvLog.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 startActivity(new Intent(getApplicationContext(), MainActivity.class));  
 }  
 });  
  
 }  
  
 private void register() {  
  
 String user = etRegEmail.getText().toString().trim();  
 String pass = etRegPass.getText().toString().trim();  
  
 if (user.isEmpty()){  
  
 etRegEmail.setError("Enter Email");  
  
 }  
 if (pass.isEmpty()){  
  
 etRegPass.setError("Enter Pass");  
  
 }  
 else{  
  
 mAuth.createUserWithEmailAndPassword(user, pass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
  
 if(task.isSuccessful()){  
  
 Toast.*makeText*(regActivity.this, "Registration Successful", Toast.*LENGTH\_LONG*).show();  
 startActivity(new Intent(regActivity.this, MainActivity.class));  
 }  
 Toast.*makeText*(regActivity.this, "Registration Failed\n"+task.getException().getMessage(), Toast.*LENGTH\_LONG*).show();  
  
 }  
 });  
  
 }  
 }  
}

**SlideAdapter.java**

package com.example.newtry;  
  
import android.transition.Slide;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
import androidx.viewpager2.widget.ViewPager2;  
  
import com.makeramen.roundedimageview.RoundedImageView;  
  
import java.util.List;  
  
public class SlideAdapter extends RecyclerView.Adapter<SlideAdapter.slideViewHolder>{  
  
 private List<SlideItem> slideItems;  
 private ViewPager2 viewPager2;  
  
 SlideAdapter(List<SlideItem> slideItems, ViewPager2 viewPager2) {  
 this.slideItems = slideItems;  
 this.viewPager2 = viewPager2;  
 }  
  
 @NonNull  
 @Override  
 public slideViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 return new slideViewHolder(  
 LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*slide\_item\_container*,parent,false)  
 );  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull slideViewHolder holder, int position) {  
 holder.setImage(slideItems.get(position));  
 if(position == slideItems.size()-2){  
 viewPager2.post(runnable);  
 }  
 }  
  
 @Override  
 public int getItemCount() {  
 return slideItems.size();  
 }  
  
 class slideViewHolder extends RecyclerView.ViewHolder{  
 private RoundedImageView imageView;  
  
 slideViewHolder(@NonNull View itemView) {  
 super(itemView);  
 imageView = itemView.findViewById(R.id.*imageSlide*);  
 }  
 void setImage(SlideItem slideItem){  
 imageView.setImageResource(slideItem.getImage());  
 }  
 }  
 private Runnable runnable = new Runnable() {  
 @Override  
 public void run() {  
 slideItems.addAll(slideItems);  
 notifyDataSetChanged();  
 }  
 };  
}

**SlideItem.java**

package com.example.newtry;  
  
public class SlideItem {  
  
 private int image;  
  
 SlideItem(int image){  
 this.image= image;  
 }  
 public int getImage(){  
 return image;  
 }  
}

**StartDogmatcher.java**

package com.example.newtry;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Looper;  
import android.view.View;  
import android.widget.Button;  
import android.widget.FrameLayout;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.airbnb.lottie.LottieAnimationView;  
import com.google.android.material.slider.Slider;  
  
public class startDogMatcher extends AppCompatActivity {  
  
  
 @Override  
 public void onBackPressed() {  
 }  
  
  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*start\_dogmatcher*);  
  
  
 initialize();  
  
  
 }  
  
 private void initialize() {  
  
 Button btnStart;  
 btnStart = findViewById(R.id.*btnStart*);  
  
 //Hide  
 btnStart.setVisibility(View.*GONE*);  
  
 FrameLayout animationContainer = findViewById(R.id.*animationContainer*);  
 LottieAnimationView animationView = findViewById(R.id.*animationFind*);  
 animationView.setAnimation("find.json");  
 animationView.playAnimation();  
  
 // Stop the animation after 3 seconds  
 new Handler(Looper.*getMainLooper*()).postDelayed(new Runnable() {  
 @Override  
 public void run() {  
  
 btnStart.setVisibility(View.*VISIBLE*);  
  
  
 }  
 }, 2000);  
  
  
 btnStart.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 // Start the DogMatcher activity  
 Intent intent = new Intent(startDogMatcher.this, dogmatcherr.class);  
 startActivity(intent);  
  
  
 }  
 });  
 }  
}

**Val.java**

package com.example.newtry;  
  
public class val {  
}

**Welome\_Panel.java**

package com.example.newtry;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.RecyclerView;  
import androidx.viewpager2.widget.CompositePageTransformer;  
import androidx.viewpager2.widget.MarginPageTransformer;  
import androidx.viewpager2.widget.ViewPager2;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
import com.denzcoskun.imageslider.ImageSlider;  
import com.denzcoskun.imageslider.models.SlideModel;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Welcome\_Panel extends AppCompatActivity {  
  
 Button btnClick;  
 ViewPager2 viewPager2;  
 TextView textView;  
 private Handler slideHandler = new Handler();  
  
 @Override  
 public void onBackPressed() {  
 }  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_welcome\_panel*);  
 initialize();  
 viewPager2 = findViewById(R.id.*view\_Pager*);  
  
 List<SlideItem> slideItems = new ArrayList<>();  
 slideItems.add(new SlideItem(R.drawable.*one*));  
 slideItems.add(new SlideItem(R.drawable.*two*));  
 slideItems.add(new SlideItem(R.drawable.*three*));  
 slideItems.add(new SlideItem(R.drawable.*four*));  
 slideItems.add(new SlideItem(R.drawable.*five*));  
  
 viewPager2.setAdapter(new SlideAdapter(slideItems, viewPager2));  
 viewPager2.setClipToPadding(false);  
 viewPager2.setClipChildren(false);  
 viewPager2.setOffscreenPageLimit(5);  
 // viewPager2.getChildAt(0).setOverScrollMode(RecyclerView.OVER\_SCROLL\_ALWAYS);  
 viewPager2.getChildAt(0).setOverScrollMode(RecyclerView.*OVER\_SCROLL\_NEVER*);  
  
 CompositePageTransformer compositePageTransformer = new CompositePageTransformer();  
 compositePageTransformer.addTransformer(new MarginPageTransformer(30));  
 compositePageTransformer.addTransformer(new ViewPager2.PageTransformer() {  
 @Override  
 public void transformPage(@NonNull View page, float position) {  
 float r = 1 - Math.*abs*(position);  
 page.setScaleY(0.85f + r\*0.15f);  
 }  
 });  
 viewPager2.setPageTransformer(compositePageTransformer);  
 viewPager2.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {  
 @Override  
 public void onPageSelected(int position) {  
 super.onPageSelected(position);  
 slideHandler.removeCallbacks(sliderRunnable);  
 slideHandler.postDelayed(sliderRunnable,2000);  
 }  
 });  
  
  
  
 }  
  
 private void initialize() {  
 btnClick = findViewById(R.id.*btnClick*);  
  
 btnClick.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(Welcome\_Panel.this, regActivity.class);  
 startActivity(intent);  
 }  
 });  
 }  
  
 private Runnable sliderRunnable = new Runnable() {  
 @Override  
 public void run() {  
 viewPager2.setCurrentItem(viewPager2.getCurrentItem() + 1);  
 }  
 };  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 slideHandler.removeCallbacks(sliderRunnable);  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 slideHandler.postDelayed(sliderRunnable, 3000);  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:backgroundTint="@color/bg"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <FrameLayout  
 android:id="@+id/animationContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:id="@+id/contents"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="263dp"  
 android:background="@drawable/blue\_bg">  
  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="25dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/black"  
 android:letterSpacing=".02"  
 android:text="Login An"  
 android:textColor="@color/white"  
 android:textSize="40sp" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="25dp"  
 android:layout\_marginTop="60dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/black"  
 android:letterSpacing=".02"  
 android:text="Account"  
 android:textColor="@color/white"  
 android:textSize="40sp" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginLeft="320dp"  
 android:layout\_marginTop="80dp"  
 android:src="@drawable/wahh" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginLeft="180dp"  
 android:layout\_marginTop="90dp"  
 android:src="@drawable/wahh" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginLeft="70dp"  
 android:layout\_marginTop="92dp"  
 android:src="@drawable/wahh" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="60dp"  
 android:layout\_marginLeft="240dp"  
 android:layout\_marginTop="40dp"  
 android:src="@drawable/wahh" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginLeft="350dp"  
 android:layout\_marginTop="0dp"  
 android:src="@drawable/wahh" />  
  
 </FrameLayout>  
  
  
 <androidx.cardview.widget.CardView  
  
 android:id="@+id/linear"  
 android:layout\_width="match\_parent"  
 android:layout\_height="360dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="-90dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="16dp"  
 android:orientation="horizontal"  
 app:cardCornerRadius="14dp"  
 app:cardElevation="3dp"  
 app:cardPreventCornerOverlap="false">  
  
  
 <EditText  
 android:id="@+id/etEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="70dp"  
 android:layout\_marginEnd="20dp"  
 android:background="@drawable/shape"  
 android:drawableLeft="@drawable/www"  
 android:drawablePadding="10dp"  
 android:fontFamily="@font/wow"  
 android:hint="Email Adress"  
 android:padding="8dp"  
 android:textColor="@color/text"  
 android:textSize="20sp" />  
  
 <EditText  
 android:id="@+id/etPass"  
 android:inputType="textPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginEnd="20dp"  
 android:background="@drawable/shape"  
 android:drawableLeft="@drawable/locked"  
 android:drawablePadding="10dp"  
 android:fontFamily="@font/wow"  
 android:hint="Password"  
 android:padding="8dp"  
 android:textColor="@color/text"  
 android:textSize="20sp" />  
  
 <Button  
 android:id="@+id/btnLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="260dp"  
 android:layout\_marginEnd="20dp"  
 android:backgroundTint="@color/secondary"  
 android:fontFamily="@font/wow"  
 android:text="Login"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/tvNotreg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="325dp"  
 android:fontFamily="@font/wow"  
 android:text="Not yet Registered? Click here"  
 android:textAlignment="center" />  
  
  
 </androidx.cardview.widget.CardView>  
  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="273dp">  
  
  
 <ImageView  
 android:layout\_width="500dp"  
 android:layout\_height="230dp"  
 android:layout\_marginStart="-50dp"  
 android:src="@drawable/grop" />  
  
  
 </FrameLayout>  
  
 </LinearLayout>  
  
 <com.airbnb.lottie.LottieAnimationView  
 android:id="@+id/animationView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:visibility="visible"  
 app:lottie\_autoPlay="true"  
 app:lottie\_fileName="paww.json"  
 app:lottie\_loop="true" />  
  
 </FrameLayout>  
  
  
</LinearLayout>

**activity\_welcome\_panel.xml**

<?xml version="1.0" encoding="utf-8"?>  
  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/bg"  
 android:orientation="vertical"  
 tools:context=".Welcome\_Panel">  
  
  
 <androidx.cardview.widget.CardView  
  
 android:id="@+id/linear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_gravity="center"  
 android:orientation="horizontal"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="3dp"  
 app:cardPreventCornerOverlap="false">  
  
  
 <androidx.viewpager2.widget.ViewPager2  
 android:id="@+id/view\_Pager"  
 android:layout\_width="350dp"  
 android:layout\_height="350dp"  
 tools:ignore="MissingConstraints"></androidx.viewpager2.widget.ViewPager2>  
  
  
 </androidx.cardview.widget.CardView>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/black"  
 android:letterSpacing=".01"  
 android:text="Discover Different Dog Breeds"  
 android:textAlignment="center"  
 android:textColor="@color/text"  
 android:textSize="32sp" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_marginStart="30dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="30dp"  
 android:fontFamily="@font/wow"  
 android:letterSpacing=".01"  
 android:text="Explore vast variety of dogs that will suit your taste, find dogs that will suit your liking"  
 android:textAlignment="center"  
 android:textColor="@color/text"  
 android:textSize="16sp" />  
  
  
 <Button  
 android:id="@+id/btnClick"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="50dp"  
 android:backgroundTint="@color/secondary"  
 android:drawableLeft="@drawable/arrow"  
 android:fontFamily="@font/wow"  
 android:text="LET'S GO"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
</LinearLayout>

**activity\_add\_dog.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 app:cardCornerRadius="30dp">  
 <ImageView  
 android:id="@+id/pictureImageView"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:scaleType="centerCrop"  
 android:layout\_gravity="center"  
 android:src="@drawable/add\_photo" />  
 </androidx.cardview.widget.CardView>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textSize="15sp"  
 android:text="Name" />  
  
 <EditText  
 android:id="@+id/nameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textSize="15sp"  
 android:text="Breed" />  
  
 <EditText  
 android:id="@+id/breedEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textSize="15sp"  
 android:text="Age" />  
  
 <EditText  
 android:id="@+id/ageEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="number" />  
  
 <Button  
 android:id="@+id/saveButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_gravity="center"  
 android:backgroundTint="@color/secondary"  
 android:text="SAVE" />  
  
 </LinearLayout>  
  
</RelativeLayout>

**activity\_dog\_details.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 app:cardCornerRadius="30dp">  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:scaleType="centerCrop"  
 android:layout\_gravity="center"  
 android:src="@color/black" />  
 </androidx.cardview.widget.CardView>  
  
 <TextView  
 android:id="@+id/txtName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="25sp"  
 android:layout\_marginTop="20dp"  
 android:gravity="center"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/txtBreed"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:gravity="center"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/txtAge"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:gravity="center"  
 android:textSize="20sp" />  
  
 <Button  
 android:id="@+id/btnBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:backgroundTint="@color/secondary"  
 android:text="BACK"/>  
  
  
  
</LinearLayout>

**best\_dog.xml**

<?xml version="1.0" encoding="utf-8"?>  
  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:backgroundTint="@color/bg">  
  
  
 <FrameLayout  
 android:id="@+id/animationContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
  
 <LinearLayout  
 android:id="@+id/contents"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 tools:context=".BestDog">  
  
 <androidx.cardview.widget.CardView  
  
 android:id="@+id/linear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_gravity="center"  
 android:orientation="horizontal"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="3dp"  
 app:cardPreventCornerOverlap="false">  
  
 <ImageView  
 android:id="@+id/ImDog"  
 android:layout\_width="350dp"  
 android:layout\_height="220dp"  
 android:layout\_gravity="center"  
 android:scaleType="centerCrop"  
 android:src="@color/black" />  
 </androidx.cardview.widget.CardView>  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:backgroundTint="@color/primary"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="Download Image" />  
  
 <TextView  
 android:id="@+id/tvDogName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginTop="20dp"  
 android:drawableLeft="@drawable/names"  
 android:drawablePadding="10dp"  
 android:textSize="20sp" />  
  
  
 <TextView  
 android:id="@+id/tvMinLife"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginTop="10dp"  
 android:drawableLeft="@drawable/life"  
 android:drawablePadding="10dp"  
 android:textSize="20sp" />  
  
  
 <TextView  
 android:id="@+id/tvMinWeight"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginTop="10dp"  
 android:drawableLeft="@drawable/weight"  
 android:drawablePadding="10dp"  
 android:textSize="20sp" />  
  
  
 <TextView  
 android:id="@+id/tvMinHeight"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="30dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginBottom="40dp"  
 android:drawableLeft="@drawable/height"  
 android:drawablePadding="10dp"  
 android:textSize="20sp" />  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="5dp"  
 android:orientation="vertical">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvChildren"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="8dp"  
 android:paddingTop="10dp"  
 android:paddingEnd="8dp"  
 android:text="Child-Friendly: "  
 android:textSize="20sp" />  
  
  
 <com.google.android.material.slider.Slider  
 android:layout\_marginHorizontal="50dp"  
 android:id="@+id/slChildren"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:enabled="false"  
 android:value="1"  
 android:valueFrom="1"  
 android:valueTo="5"  
 app:thumbColor="@color/secondary"  
 app:trackColor="@color/secondary" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvODogs"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="8dp"  
 android:paddingTop="10dp"  
 android:paddingEnd="8dp"  
 android:text="Dog-Friendly: "  
 android:textSize="20sp" />  
  
 <com.google.android.material.slider.Slider  
 android:layout\_marginHorizontal="50dp"  
 android:id="@+id/slDog"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:layout\_marginRight="-5dp"  
 android:enabled="false"  
 android:value="1"  
 android:valueFrom="1"  
 android:valueTo="5"  
 app:thumbColor="@color/secondary"  
 app:trackColor="@color/secondary" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvPlayful"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="8dp"  
 android:paddingTop="10dp"  
 android:paddingEnd="8dp"  
 android:text="Playfulness: "  
 android:textSize="20sp" />  
  
  
 <com.google.android.material.slider.Slider  
 android:layout\_marginHorizontal="50dp"  
 android:id="@+id/slPlayful"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:layout\_marginRight="-5dp"  
 android:enabled="false"  
 android:value="1"  
 android:valueFrom="1"  
 android:valueTo="5"  
 app:thumbColor="@color/secondary"  
 app:trackColor="@color/secondary" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvProtective"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="8dp"  
 android:paddingTop="10dp"  
 android:paddingEnd="8dp"  
 android:text="Protectiveness: "  
 android:textSize="20sp" />  
  
  
 <com.google.android.material.slider.Slider  
 android:id="@+id/slProtective"  
 android:layout\_marginHorizontal="50dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:layout\_marginRight="-5dp"  
 android:enabled="false"  
 android:value="1"  
 android:valueFrom="1"  
 android:valueTo="5"  
 app:thumbColor="@color/secondary"  
 app:trackColor="@color/secondary" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvTrain"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="8dp"  
 android:paddingTop="10dp"  
 android:paddingEnd="8dp"  
 android:text="Trainability: "  
 android:textSize="20sp" />  
  
  
 <com.google.android.material.slider.Slider  
 android:layout\_marginHorizontal="50dp"  
 android:id="@+id/slTrain"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:layout\_marginRight="-5dp"  
 android:enabled="false"  
 android:value="1"  
 android:valueFrom="1"  
 android:valueTo="5"  
 app:thumbColor="@color/secondary"  
 app:trackColor="@color/secondary" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvBark"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="8dp"  
 android:paddingTop="10dp"  
 android:paddingEnd="8dp"  
 android:text="Barking: "  
 android:textSize="20sp" />  
  
  
 <com.google.android.material.slider.Slider  
 android:layout\_marginHorizontal="50dp"  
 android:id="@+id/slBark"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:layout\_marginRight="-5dp"  
 android:enabled="false"  
 android:value="1"  
 android:valueFrom="1"  
 android:valueTo="5"  
 app:thumbColor="@color/secondary"  
 app:trackColor="@color/secondary" />  
 </LinearLayout>  
  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/btnOk"  
 android:layout\_width="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginBottom="20dp"  
 android:paddingHorizontal="10dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:backgroundTint="@color/secondary"  
 android:text="HOME" />  
  
 </LinearLayout>  
  
 <com.airbnb.lottie.LottieAnimationView  
 android:id="@+id/animationView"  
 android:layout\_width="556dp"  
 android:layout\_height="886dp"  
 android:layout\_gravity="center"  
 android:visibility="visible"  
 app:lottie\_autoPlay="true"  
 app:lottie\_fileName="cutedog.json"  
 app:lottie\_loop="true" />  
  
 </FrameLayout>  
  
</ScrollView>

**dog\_album.xml**

<?xml version="1.0" encoding="utf-8"?>  
  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="10dp">  
  
 <GridView  
 android:id="@+id/gridView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:background="#FFFFFF"  
 android:clipToPadding="false"  
 android:horizontalSpacing="8dp"  
 android:numColumns="3"  
 android:padding="8dp"  
 android:scrollbars="vertical"  
 android:verticalSpacing="8dp" />  
  
 <Button  
 android:id="@+id/addDogButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginVertical="10dp"  
 android:backgroundTint="@color/secondary"  
 android:textSize="15sp"  
 android:text="Add a dog" />  
  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bottomNavgationView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:background="@drawable/bottom\_background"  
 android:layout\_gravity="center|bottom"  
 android:elevation="2dp"  
 app:itemIconSize="30dp"  
 app:itemRippleColor="@color/secondary"  
 app:menu="@menu/bottom\_menu" />  
  
</LinearLayout>

**dog\_details.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_margin="5dp"  
 android:orientation="vertical">  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 app:cardCornerRadius="30dp">  
  
 <ImageView  
 android:id="@+id/ImDogImage"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_gravity="center"  
 android:scaleType="centerCrop"  
 android:src="@color/black" />  
 </androidx.cardview.widget.CardView>  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="10dp"  
 android:backgroundTint="@color/primary"  
 android:text="Download Image" />  
  
 <TextView  
 android:id="@+id/tvDogName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Breed Name:"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/tvDogHeight"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Height:"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/tvDogWeight"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Weight:"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/tvDogGroup"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Group:"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
  
 <TextView  
 android:id="@+id/tvDogBreedFor"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Breed For:"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/tvDogTemperament"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Temperament:"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/tvDogLifeSpan"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Life Span:"  
 android:textColor="@color/black"  
 android:textSize="20sp" />  
  
 <Button  
 android:id="@+id/btnBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="30dp"  
 android:backgroundTint="@color/secondary"  
 android:text="BACK" />  
 </LinearLayout>  
  
  
 </ScrollView>  
  
  
</LinearLayout>

**dog\_matcher.xml**

<?xml version="1.0" encoding="utf-8"?>  
  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/bg">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 tools:context=".DogMatcher">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/black"  
 android:letterSpacing=".01"  
 android:text="Discover Your Ideal Companion!"  
 android:textAlignment="center"  
 android:textColor="@color/text"  
 android:textSize="32sp" />  
  
 <LinearLayout  
 android:id="@+id/llButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:layout\_gravity="center">  
  
 <com.airbnb.lottie.LottieAnimationView  
 android:id="@+id/btnDog"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_gravity="center"  
 android:visibility="visible"  
 app:lottie\_autoPlay="true"  
 app:lottie\_fileName="gift\_box.json"  
 app:lottie\_loop="true" />  
 </LinearLayout>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/black"  
 android:letterSpacing=".01"  
 android:text="Tap to Open!"  
 android:textAlignment="center"  
 android:textColor="@color/text"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/tvDisplay"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:text="" />  
  
  
 </LinearLayout>  
  
  
</RelativeLayout>

**dog\_matcherr.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:background="@color/bg"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 >  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 tools:context=".DogMatcher">  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:layout\_gravity="center\_horizontal">  
  
  
  
 <androidx.cardview.widget.CardView  
  
 android:id="@+id/linear"  
 android:layout\_width="match\_parent"  
 android:backgroundTint="@color/primary"  
 android:layout\_height="200dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="10dp"  
 android:orientation="horizontal"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="3dp"  
 app:cardPreventCornerOverlap="false">  
 <TextView  
 android:id="@+id/tvQuestion"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fontFamily="@font/black"  
 android:gravity="center\_horizontal"  
 android:padding="10dp"  
 android:layout\_gravity="center"  
 android:text="Question"  
 android:textAppearance="?android:attr/textAppearanceSmall"  
 android:textColor="@color/black"  
 android:textSize="30sp" />  
  
  
 </androidx.cardview.widget.CardView>  
  
 </FrameLayout>  
  
 <RadioGroup  
 android:id="@+id/rgChoices"  
 android:layout\_width="match\_parent"  
 android:padding="10dp"  
 android:layout\_height="wrap\_content">  
  
 <RadioButton  
 android:id="@+id/rb1"  
 android:textColor="@color/text"  
 android:fontFamily="@font/wow"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:text="Choice 1" />  
  
 <RadioButton  
 android:id="@+id/rb2"  
 android:textColor="@color/text"  
 android:fontFamily="@font/wow"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:text="Choice 2" />  
  
 <RadioButton  
 android:textColor="@color/text"  
 android:fontFamily="@font/wow"  
 android:id="@+id/rb3"  
 android:textSize="18sp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Choice 3" />  
  
 </RadioGroup>  
  
 <ImageButton  
 android:id="@+id/btnNext"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:layout\_marginTop="40dp"  
 android:layout\_gravity="center"  
 android:adjustViewBounds="true"  
 android:background="#0000"  
 android:maxWidth="42dp"  
 android:maxHeight="42dp"  
 android:paddingBottom="0dp"  
 android:scaleType="fitCenter"  
 android:src="@drawable/paw" />  
  
 </LinearLayout>  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:id="@+id/bottomNavgationView"  
 android:layout\_alignParentBottom="true"  
 android:background="@drawable/bottom\_background"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginBottom="20dp"  
 android:elevation="2dp"  
 app:menu="@menu/bottom\_menu"  
 app:itemIconSize="30dp"/>  
</RelativeLayout>

**edit\_dog.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="10dp">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 app:cardCornerRadius="30dp">  
  
 <ImageView  
 android:id="@+id/pictureImageView"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_gravity="center"  
 android:scaleType="centerCrop"  
 android:src="@drawable/ic\_launcher\_background" />  
 </androidx.cardview.widget.CardView>  
  
 <EditText  
 android:id="@+id/nameEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Name" />  
  
 <EditText  
 android:id="@+id/breedEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Breed" />  
  
 <EditText  
 android:id="@+id/ageEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:hint="Age" />  
  
 <Button  
 android:id="@+id/updateButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:backgroundTint="@color/secondary"  
 android:text="Update Dog"  
 android:textSize="20sp" />  
  
 <Button  
 android:id="@+id/btnBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Back"  
 android:textSize="20sp"  
 android:backgroundTint="@color/primary"  
 android:layout\_gravity="center"/>  
  
</LinearLayout>

**encyclopedia\_recycler.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:background="@color/bg"  
 android:padding="16dp">  
  
 <EditText  
 android:id="@+id/etSearch"  
 android:layout\_width="match\_parent"  
 android:layout\_height="48dp"  
 android:layout\_marginStart="4dp"  
 android:layout\_marginEnd="4dp"  
 android:layout\_marginBottom="10dp"  
 android:hint="Search"  
 android:textColor="@color/text"  
 android:textColorHint="@color/text"  
 android:background="@drawable/edt\_backrgound"  
 android:drawableLeft="@drawable/last"  
 android:drawablePadding="10dp"  
 android:paddingLeft="10dp"/>  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recycler\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1" />  
  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:id="@+id/bottomNavgationView"  
 android:layout\_alignParentBottom="true"  
 android:background="@drawable/bottom\_background"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginBottom="20dp"  
 android:elevation="2dp"  
 app:menu="@menu/bottom\_menu"  
 app:itemIconSize="30dp"  
 app:itemRippleColor="@color/secondary"/>  
  
  
</LinearLayout>

**home.xml**

<?xml version="1.0" encoding="utf-8"?>  
  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/bg"  
 android:orientation="vertical"  
 tools:context=".Home">  
  
 <com.airbnb.lottie.LottieAnimationView  
 android:id="@+id/animationView"  
 android:layout\_width="556dp"  
 android:layout\_height="886dp"  
 android:layout\_gravity="center"  
 android:visibility="visible"  
 app:lottie\_autoPlay="true"  
 app:lottie\_fileName="cutedog.json"  
 app:lottie\_loop="true" />  
  
 <FrameLayout  
 android:id="@+id/contents"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:text="SPAWTLIGHT"  
 android:textAlignment="center"  
 android:textColor="@color/text"  
 android:textSize="25sp"  
 android:textStyle="bold" />  
  
  
 <androidx.cardview.widget.CardView  
  
 android:id="@+id/linear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center|top"  
 android:layout\_marginTop="60dp"  
 android:orientation="horizontal"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="3dp"  
 app:cardPreventCornerOverlap="false">  
  
  
 <ImageView  
 android:id="@+id/dog\_image"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:layout\_gravity="center"  
 android:scaleType="centerCrop"  
 android:src="@color/black" />  
  
 </androidx.cardview.widget.CardView>  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center|top"  
 android:layout\_marginTop="375dp"  
 android:backgroundTint="@color/primary"  
 android:text="Download Image" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:paddingBottom="20dp">  
  
 <TextView  
  
 android:id="@+id/breed\_name\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="440dp"  
 android:fontFamily="@font/black"  
 android:textAlignment="center"  
 android:textColor="@color/text"  
 android:textSize="30sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/bred\_for\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/wow"  
 android:textAlignment="center"  
 android:textAppearance="?android:textAppearanceMedium"  
 android:textColor="@color/text"  
 android:textSize="21sp" />  
  
 <TextView  
 android:id="@+id/breed\_group\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/wow"  
 android:textAlignment="center"  
 android:textAppearance="?android:textAppearanceMedium"  
 android:textColor="@color/text"  
 android:textSize="21sp" />  
  
 <TextView  
 android:id="@+id/life\_span\_text"  
 android:layout\_width="match\_parent"  
  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/wow"  
 android:textAlignment="center"  
 android:textAppearance="?android:textAppearanceMedium"  
 android:textColor="@color/text"  
 android:textSize="21sp" />  
  
 <TextView  
 android:id="@+id/temperament\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/wow"  
 android:textAlignment="center"  
 android:textAppearance="?android:textAppearanceMedium"  
 android:textColor="@color/text"  
 android:textSize="21sp" />  
 </LinearLayout>  
 </ScrollView>  
  
 </FrameLayout>  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bottomNavgationView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:background="@drawable/bottom\_background"  
 android:elevation="2dp"  
 app:itemIconSize="30dp"  
 app:itemRippleColor="@color/secondary"  
 app:menu="@menu/bottom\_menu" />  
  
</RelativeLayout>

**item\_dog.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <ImageView  
 android:id="@+id/imgDog"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:scaleType="centerCrop"  
 android:src="@color/black" />  
  
 <TextView  
 android:id="@+id/txtName"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/imgDog"  
 android:gravity="center"  
 android:paddingTop="8dp"  
 android:textColor="#000000"  
 android:textSize="14sp" />  
  
</RelativeLayout>

**item\_dog\_breed.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_margin="6dp"  
 app:cardCornerRadius="6dp"  
 app:cardElevation="4dp"  
 app:cardBackgroundColor="@color/secondary"  
 android:layout\_height="wrap\_content">  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/idRLCurrency">  
  
  
  
 <ImageView  
 android:id="@+id/dogImage"  
 android:src="@mipmap/ic\_launcher"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:text="Rate"  
 android:scaleType="centerCrop"  
 android:textColor="@color/white" />  
  
 <TextView  
 android:id="@+id/text\_breed\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Symbol"  
 android:textColor="@color/white"  
 android:textSize="22sp"  
 android:paddingLeft="10dp"  
 android:textStyle="bold" />  
  
  
 </LinearLayout>  
  
</androidx.cardview.widget.CardView>

**random\_images.xml**

<?xml version="1.0" encoding="utf-8"?>  
  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
  
  
 <androidx.cardview.widget.CardView  
  
 android:id="@+id/linear"  
 android:layout\_width="match\_parent"  
 android:layout\_height="358dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="16dp"  
 android:orientation="horizontal"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="3dp"  
 app:cardPreventCornerOverlap="false">  
  
  
 <ImageView  
 android:id="@+id/Imrandom"  
 android:layout\_width="325dp"  
 android:layout\_height="298dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="28dp"  
 android:scaleType="centerCrop"  
 android:src="@color/black" />  
  
 </androidx.cardview.widget.CardView>  
  
  
  
  
 <Button  
 android:id="@+id/btnGenerate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="85dp"  
 android:backgroundTint="@color/secondary"  
 android:layout\_marginTop="420dp"  
 android:text="Generate Random Image" />  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:backgroundTint="@color/primary"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginLeft="112dp"  
 android:layout\_marginTop="480dp"  
 android:text="Download Image" />  
  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:id="@+id/bottomNavgationView"  
 android:layout\_alignParentBottom="true"  
 android:background="@drawable/bottom\_background"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginBottom="20dp"  
 android:elevation="2dp"  
 app:menu="@menu/bottom\_menu"  
 app:itemIconSize="30dp"  
 app:itemRippleColor="@color/secondary" />  
  
  
</RelativeLayout>

**reg\_activity.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/bg"  
 android:orientation="vertical"  
 tools:context=".regActivity">  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="263dp"  
 android:background="@drawable/blue\_bg">  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginLeft="320dp"  
 android:layout\_marginTop="80dp"  
 android:src="@drawable/wahh" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginLeft="180dp"  
 android:layout\_marginTop="90dp"  
 android:src="@drawable/wahh" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginLeft="70dp"  
 android:layout\_marginTop="92dp"  
 android:src="@drawable/wahh" />  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="40dp"  
 android:layout\_marginLeft="240dp"  
 android:src="@drawable/wahh" />  
  
 <ImageView  
 android:layout\_width="30dp"  
 android:layout\_height="90dp"  
 android:layout\_marginTop="0dp"  
 android:layout\_marginLeft="350dp"  
 android:src="@drawable/wahh" />  
  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/black"  
 android:letterSpacing=".02"  
 android:layout\_marginStart="25dp"  
 android:layout\_marginEnd="10dp"  
 android:text="Create An"  
 android:layout\_marginTop="20dp"  
 android:textColor="@color/white"  
 android:textSize="40sp"/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="25dp"  
 android:layout\_marginTop="60dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/black"  
 android:letterSpacing=".02"  
 android:text="Account"  
 android:textColor="@color/white"  
 android:textSize="40sp" />  
  
 </FrameLayout>  
  
  
  
 <androidx.cardview.widget.CardView  
  
 android:id="@+id/linear"  
 android:layout\_width="match\_parent"  
 android:layout\_height="360dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="-90dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="16dp"  
 android:orientation="horizontal"  
 app:cardCornerRadius="14dp"  
 app:cardElevation="3dp"  
 app:cardPreventCornerOverlap="false">  
  
  
 <EditText  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:drawableLeft="@drawable/www"  
 android:background="@drawable/shape"  
 android:padding="8dp"  
 android:drawablePadding="10dp"  
 android:layout\_marginTop="70dp"  
 android:id="@+id/etRegEmail"  
 android:textColor="@color/text"  
 android:layout\_width="match\_parent"  
 android:fontFamily="@font/wow"  
 android:textSize="20sp"  
 android:layout\_height="wrap\_content"  
 android:hint="Email Adress"/>  
  
 <EditText  
 android:drawableLeft="@drawable/locked"  
 android:inputType="textPassword"  
 android:drawablePadding="10dp"  
 android:textSize="20sp"  
 android:padding="8dp"  
 android:textColor="@color/text"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:background="@drawable/shape"  
 android:fontFamily="@font/wow"  
 android:id="@+id/etRegPass"  
 android:layout\_marginTop="140dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Password"/>  
  
 <Button  
 android:fontFamily="@font/wow"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginTop="260dp"  
 android:id="@+id/btnRegister"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Register"  
 android:backgroundTint="@color/secondary"/>  
  
 <TextView  
 android:id="@+id/tvLog"  
 android:layout\_width="match\_parent"  
 android:textAlignment="center"  
 android:fontFamily="@font/wow"  
 android:layout\_marginTop="320dp"  
 android:layout\_height="wrap\_content"  
 android:text="Already Registered? Login here"/>  
  
  
  
  
  
 </androidx.cardview.widget.CardView>  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="273dp">  
  
  
  
  
 <ImageView  
 android:layout\_width="500dp"  
 android:layout\_height="230dp"  
 android:layout\_marginLeft="-60dp"  
 android:src="@drawable/grop" />  
  
  
 </FrameLayout>  
  
  
  
  
  
</LinearLayout>

**slide\_item\_container.xml**

<?xml version="1.0" encoding="utf-8"?>  
<com.makeramen.roundedimageview.RoundedImageView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/imageSlide"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:riv\_corner\_radius="12dp">  
  
</com.makeramen.roundedimageview.RoundedImageView>

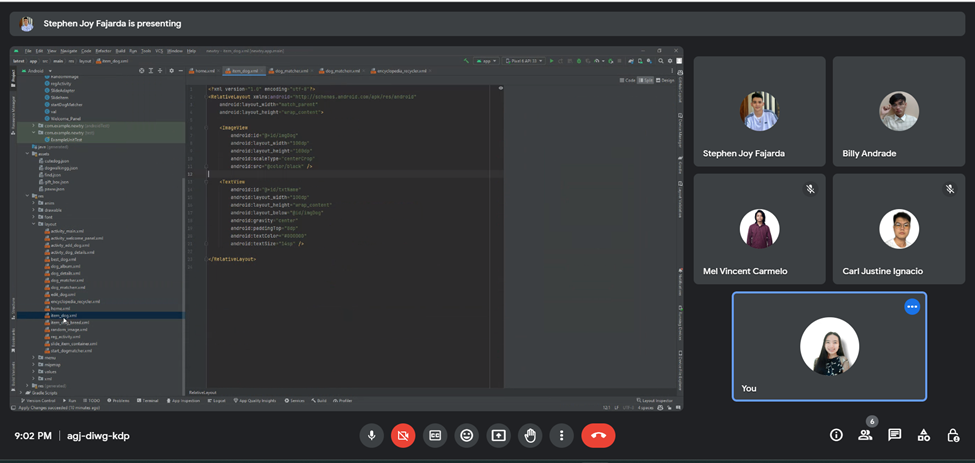
**start\_dogmatcher.xml**

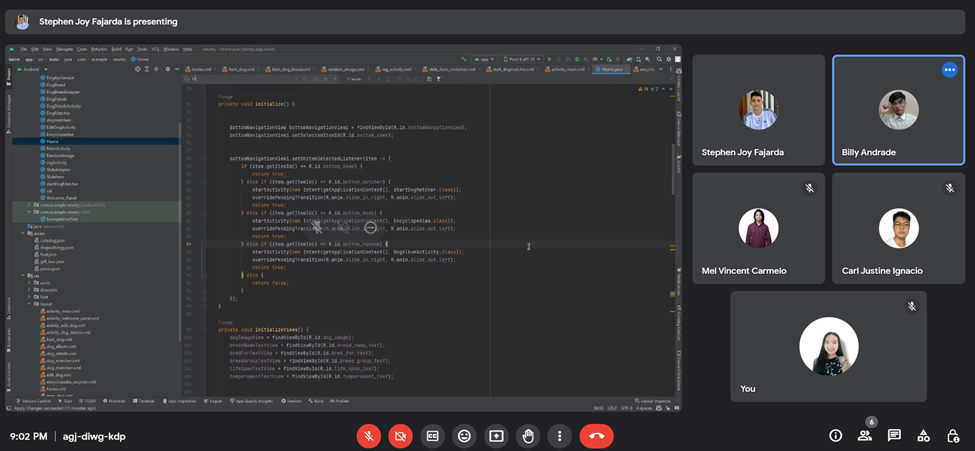
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
  
  
 <FrameLayout  
 android:id="@+id/animationContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Find the best dog that suits you"  
 android:textSize="25sp"  
 android:textColor="@color/black"  
 android:layout\_marginTop="100dp"  
 android:textAlignment="center"  
 android:layout\_gravity="center"/>  
  
 <com.airbnb.lottie.LottieAnimationView  
 android:id="@+id/animationFind"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="50dp"  
 android:visibility="visible"  
 app:lottie\_autoPlay="true"  
 app:lottie\_fileName="find.json"  
 app:lottie\_loop="true" />  
  
 </LinearLayout>  
  
  
 <Button  
 android:id="@+id/btnStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="20sp"  
 android:layout\_marginTop="100dp"  
 android:layout\_gravity="center"  
 android:backgroundTint="@color/secondary"  
 android:text="START" />  
  
 </FrameLayout>  
  
  
</RelativeLayout>

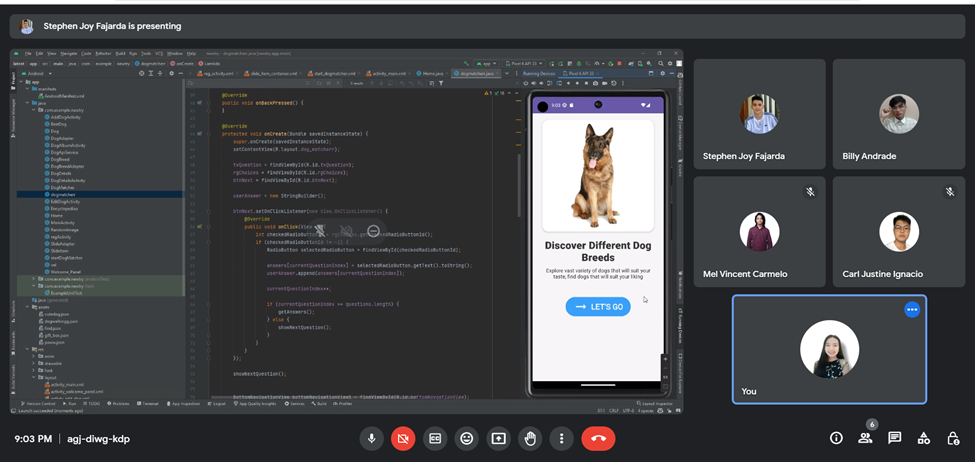
**Bottom\_menu.xml**

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:id="@+id/bottom\_home"  
 android:icon="@drawable/home"  
 android:title="Home"  
 />  
 <item android:id="@+id/bottom\_matcher"  
 android:icon="@drawable/heart"  
 android:title="Matcher"/>  
  
 <item android:id="@+id/bottom\_book"  
 android:icon="@drawable/baseline\_menu\_book\_24"  
 android:title="Encyclopedia"/>  
  
 <item android:id="@+id/bottom\_random"  
 android:icon="@drawable/baseline\_image\_24"  
 android:title="Album"/>  
  
 <item android:id="@+id/bottom\_logout"  
 android:icon="@drawable/logout"  
 android:title="Logout"/>  
  
  
</menu>

**Screenshot of meetings**

****

****

****