Slot C || Expense Management App

Guided By

Prof. Mudigonda Krishna Siva Prasad

**Group Members:**

1. Abhay Chaudhary 19BCE7290
2. Aryan Subrahmonyaru 19BCE7347
3. Parakram Kumar 19BCE7312
4. Joshi Ritvij Kumar 19BCN7223

**Section and slot:** C-CSE 4002-TH-ONL00169

The project can be found in the link given below on Github

https://github.com/Abhayindia/Expense-Management-App

**Index**

|  |  |  |
| --- | --- | --- |
| S.No. | Topic | Page No. |
| 1 | Abstract | 4 |
| 2 | Introduction | 5 |
| 3 | Source Code | 6-104 |
| 4 | Screenshot | 105 |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |
| 16 |  |  |

**List of Images**

|  |  |  |
| --- | --- | --- |
| S.No. | Topic | Page No. |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

**Abstract**

In today’s busy and expensive life we are in a great rush to make money. But at the end of the month we broke off. As we are unknowingly spending money on little and unwanted things. So, we have come over with the idea to track our earnings. Small Change aims to help everyone who are planning to know their expenses and save from it. Small Change is an app which users can execute in their mobile phones and update their daily expenses so that they are well known to their expenses. Here user can define their own categories for expense type like food, clothing, rent and bills where they have to enter the money that has been spent and also can add some information in additional information to specify the expense. User can also define expense categories. User will be able to see pie chart of expense. Although this app is focused on new job holders, interns and teenagers, everyone who wants to track their expense can use this app.

**Introduction**

Small Change is designed to keep a track of Income-Expense of a person on a day-to-day basis. This System divides the Income based on daily expenses. If you exceed day’s expense, system will cut it from your income and will provide new daily expense allowed amount. If that day’s expense is less, system will add it in savings. Daily expense tracking System will generate report at the end of month to show Income-Expense Curve. It will let you add the savings amount, which you had saved for some particular Festivals or days like Birthday or Anniversary. Daily Expense Tracker System is a system which will keep a track of Income-Expense of a person on a day to day basics. Small Change aims to help everyone who are planning to know their expenses and save from it. Small Change is an android app which users can execute in their mobile phones and update their daily expenses so that they are well known to their expenses. Here user can define their own categories for expense type like food, clothing, rent and bills where they have to enter the money that has been spent and also can add some information in additional information to specify the expense. User can also define expense categories. User will be able to see pie chart of expense.

**Source Code**

**Data.java**

public class Data {  
 private int amount;  
 private String type;  
 private String note;  
 private String id;  
 private String date;  
 public Data(int amount, String type, String note, String id, String date) {  
 this.amount = amount;  
 this.type = type;  
 this.note = note;  
 this.id = id;  
 this.date = date;  
 }  
 public int getAmount() {  
 return amount;  
 }  
 public void setAmount(int amount) {  
 this.amount = amount;  
 }  
 public String getType() {  
 return type;  
 }  
 public void setType(String type) {  
 this.type = type;  
 }  
 public String getNote() {  
 return note;  
 }  
 public void setNote(String note) {  
 this.note = note;  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getDate() {  
 return date;  
 }  
  
 public void setDate(String date) {  
 this.date = date;  
 }  
  
 public Data(){  
  
 }  
}

**User.java**

public class User {  
 public String id;  
 public String username;  
 /\*public int userImg;\*/  
  
 public User(String id, String username) {  
 this.id = id;  
 this.username = username;  
 }  
  
 public User() {  
  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getUsername() {  
 return username;  
 }  
  
 public void setUsername(String username) {  
 this.username = username;  
 }  
}

**Dashboard.java**

import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.cardview.widget.CardView;  
import androidx.constraintlayout.widget.ConstraintLayout;  
import androidx.fragment.app.Fragment;  
  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.ProgressBar;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.Query;  
import com.google.firebase.database.ValueEventListener;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
import com.squareup.picasso.Picasso;  
  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.List;  
  
import id.ac.umn.leleair.kelompok.smallchange.Model.Data;  
import id.ac.umn.leleair.kelompok.smallchange.Model.User;  
  
public class Dashboard extends Fragment {  
 private ImageView upperBox, upperBox2, user\_photo, backgroundBox;  
 private TextView welocomeText, username, incomeText, outcomeText, currentBalance, tvViewHistory;  
 private ConstraintLayout userInfo;  
 private ProgressBar incomeProgress, outcomeProgress;  
 private CardView photoContainer;  
 private int sumOutcome, sumIncome;  
 private float totIncome, totOutcome;  
 private String stTotalValue, stUsername;  
 private RadioGroup radioGroup;  
 private RadioButton checkedRadioButton;  
  
 //Firebase  
 private FirebaseAuth mAuth;  
 private DatabaseReference mIncomeDatabase;  
 private DatabaseReference mOutcomeDatabase;  
 private DatabaseReference mUsername;  
 private DatabaseReference mImage;  
 private StorageReference storageProfileImg;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view = inflater.inflate(R.layout.*fragment\_dashboard*, container, false);  
  
 //Firebase Initialization  
 mAuth = FirebaseAuth.*getInstance*();  
 FirebaseUser mUser = mAuth.getCurrentUser();  
 String uid = mUser.getUid();  
  
 mUsername = FirebaseDatabase.*getInstance*().getReference().child("Username").child(uid);  
 mIncomeDatabase = FirebaseDatabase.*getInstance*().getReference().child("IncomeData").child(uid);  
 mOutcomeDatabase = FirebaseDatabase.*getInstance*().getReference().child("OutcomeData").child(uid);  
 mImage = FirebaseDatabase.*getInstance*().getReference().child("User");  
 storageProfileImg = FirebaseStorage.*getInstance*().getReference().child("Profile Img");  
  
 upperBox = view.findViewById(R.id.*upperBox*);  
 upperBox2 = view.findViewById(R.id.*upperBox2*);  
 welocomeText = view.findViewById(R.id.*tvHello*);  
 username = view.findViewById(R.id.*usernameDashboard*);  
 userInfo = view.findViewById(R.id.*userInfoContainer*);  
 incomeText = view.findViewById(R.id.*income\_progressText*);  
 incomeProgress = view.findViewById(R.id.*income\_progressBar*);  
 outcomeText = view.findViewById(R.id.*outcome\_progressText*);  
 outcomeProgress = view.findViewById(R.id.*outcome\_progressBar*);  
 user\_photo = view.findViewById(R.id.*photoProfile*);  
 backgroundBox = view.findViewById(R.id.*backgroundBox*);  
 photoContainer = view.findViewById(R.id.*photoProfileContainer*);  
 currentBalance = view.findViewById(R.id.*tvCurrentBalance*);  
 tvViewHistory = view.findViewById(R.id.*tvViewHistory*);  
 radioGroup = view.findViewById(R.id.*RadioFilter*);  
 checkedRadioButton = (RadioButton)radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());  
  
 checkDatabaseUpdate();  
  
 statisticFilter();  
 radioGroup.check(R.id.*RadioToday*);  
  
 upperBox.animate().translationX(0).alpha(1).setDuration(800);  
 upperBox2.animate().translationX(0).alpha(1).setDuration(800).setStartDelay(1000);  
 userInfo.animate().translationX(0).alpha(1).setDuration(800).setStartDelay(600);  
 backgroundBox.animate().translationY(0).alpha(1).setDuration(800).setStartDelay(600);  
 photoContainer.animate().alpha(1).setDuration(600).setStartDelay(1400);  
  
 tvViewHistory.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 HomeActivity.*viewPager*.setCurrentItem(3);  
 HomeActivity.*tabLayoutHome*.getTabAt(3).select();  
 }  
 });  
  
 return view;  
 }  
  
 private void statisticFilter() {  
 // This overrides the radiogroup onCheckListener  
 radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener()  
 {  
 public void onCheckedChanged(RadioGroup group, int checkedId)  
 {  
 Query incomeQuery, outcomeQuery;  
 String dateTarget;  
  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Calendar cal = Calendar.*getInstance*();  
 Date date = cal.getTime();  
 String mDate = df.format(date);  
  
 // This will get the radiobutton that has changed in its check state  
 RadioButton checkedRadioButton = (RadioButton)group.findViewById(checkedId);  
 // This puts the value (true/false) into the variable  
 boolean isChecked = checkedRadioButton.isChecked();  
 // If the radiobutton that has changed in check state is now checked...  
 if (isChecked)  
 {  
 if(checkedId == R.id.RadioToday) {  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(mDate).endAt(mDate);  
 outcomeQuery = mOutcomeDatabase.orderByChild("date").startAt(mDate).endAt(mDate);  
  
 incomeQuery.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 totIncome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 totIncome += data.getAmount();  
 }  
  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, 0, calculatePercentage(totIncome, totOutcome));  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, 0, calculatePercentage(totOutcome, totIncome));  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {}  
 });  
  
 outcomeQuery.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 totOutcome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 totOutcome += data.getAmount();  
 }  
  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, 0, calculatePercentage(totIncome, totOutcome));  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, 0, calculatePercentage(totOutcome, totIncome));  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {}  
 });  
 }  
 else if(checkedId == R.id.Radio7D) {  
 cal.add(Calendar.DATE,-7);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 outcomeQuery = mOutcomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
  
 incomeQuery.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 totIncome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 totIncome += data.getAmount();  
 }  
  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, 0, calculatePercentage(totIncome, totOutcome));  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, 0, calculatePercentage(totOutcome, totIncome));  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {}  
 });  
  
 outcomeQuery.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 totOutcome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 totOutcome += data.getAmount();  
 }  
  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, 0, calculatePercentage(totIncome, totOutcome));  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, 0, calculatePercentage(totOutcome, totIncome));  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {}  
 });  
 }  
 else if(checkedId == R.id.Radio31D) {  
 cal.add(Calendar.DATE,-31);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 outcomeQuery = mOutcomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
  
 incomeQuery.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 totIncome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 totIncome += data.getAmount();  
 }  
  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, 0, calculatePercentage(totIncome, totOutcome));  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, 0, calculatePercentage(totOutcome, totIncome));  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {}  
 });  
  
 outcomeQuery.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 totOutcome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 totOutcome += data.getAmount();  
 }  
  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, 0, calculatePercentage(totIncome, totOutcome));  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, 0, calculatePercentage(totOutcome, totIncome));  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {}  
 });  
 }  
 }  
 }  
 });  
 }  
  
 public void checkDatabaseUpdate() {  
 mUsername.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 stUsername = "";  
  
 for(DataSnapshot mSnapshot:snapshot.getChildren()){  
 User user = mSnapshot.getValue(User.class);  
 assert user != null;  
 stUsername = user.getUsername();  
 }  
 username.setText(stUsername);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 mIncomeDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 sumIncome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 sumIncome += data.getAmount();  
 }  
  
 stTotalValue = String.valueOf(sumIncome-sumOutcome);  
 currentBalance.setText(stTotalValue);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) { }  
 });  
  
 mOutcomeDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 sumOutcome = 0;  
  
 for(DataSnapshot dataSnapshot:snapshot.getChildren()){  
 Data data = dataSnapshot.getValue(Data.class);  
 assert data != null;  
 sumOutcome += data.getAmount();  
 }  
  
 //Debug Only  
 Log.i("Income",String.valueOf(sumIncome));  
 Log.i("Outcome",String.valueOf(sumOutcome));  
  
 stTotalValue = String.valueOf(sumIncome-sumOutcome);  
 currentBalance.setText(stTotalValue);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) { }  
 });  
  
 mImage.child(mAuth.getCurrentUser().getUid()).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.exists() && dataSnapshot.getChildrenCount() > 0){  
 if (dataSnapshot.hasChild("image"))  
 {  
 String image = dataSnapshot.child("image").getValue().toString();  
 Picasso.get().load(image).into(user\_photo);  
 }  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 }  
  
 public float calculatePercentage(float x, float y){  
 float percentage;  
 Log.i("xValue", String.valueOf(x));  
 Log.i("yValue", String.valueOf(y));  
 percentage = Math.round ((x / (x+y)) \* 100);  
 Log.i("xValue", String.valueOf(x));  
 Log.i("yValue", String.valueOf(y));  
 Log.i("Percentage", String.valueOf(percentage));  
 return percentage;  
 }  
  
  
 public void playAnimIn(){  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, 0, calculatePercentage(totIncome, totOutcome));  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, 0, calculatePercentage(totOutcome, totIncome));  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
  
 backgroundBox.animate().translationY(0).alpha(1).setDuration(800).setStartDelay(600);  
 }  
  
 public void playAnimOut(){  
 ProgressBarAnimation anim1 = new ProgressBarAnimation(incomeProgress, incomeText, calculatePercentage(totIncome, totOutcome), 0);  
 ProgressBarAnimation anim2 = new ProgressBarAnimation(outcomeProgress, outcomeText, calculatePercentage(totOutcome, totIncome), 0);  
 anim1.setDuration(1400);  
 anim2.setDuration(1400);  
 incomeProgress.startAnimation(anim1);  
 outcomeProgress.startAnimation(anim2);  
  
 backgroundBox.animate().translationY(400).alpha(0).setDuration(800).setStartDelay(600);  
 }  
}

**History.java**

import android.content.Context;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.LinearSnapHelper;  
import androidx.recyclerview.widget.RecyclerView;  
import androidx.recyclerview.widget.SnapHelper;  
  
import android.view.ContextThemeWrapper;  
import android.view.LayoutInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.PopupMenu;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.ChildEventListener;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.Query;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
  
import id.ac.umn.leleair.kelompok.smallchange.Model.Data;  
import www.sanju.zoomrecyclerlayout.ZoomRecyclerLayout;  
  
public class History extends Fragment {  
 private ImageView backgroundBox;  
 private TextView tvIncome, tvOutcome;  
 private Button showFilterBtn;  
  
 //Firebase  
 private FirebaseAuth mAuth;  
 private Query mIncomeDatabase;  
 private Query mOutcomeDatabase;  
  
 //Recycler View  
 private RecyclerView mRecyclerIncome;  
 private RecyclerView mRecyclerOutcome;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view = inflater.inflate(R.layout.*fragment\_history*, container, false);  
  
 //Firebase Initialization  
 mAuth = FirebaseAuth.*getInstance*();  
 FirebaseUser mUser = mAuth.getCurrentUser();  
 String uid = mUser.getUid();  
  
 mIncomeDatabase = FirebaseDatabase.*getInstance*().getReference().child("IncomeData").child(uid);  
 mOutcomeDatabase = FirebaseDatabase.*getInstance*().getReference().child("OutcomeData").child(uid);  
  
 backgroundBox = view.findViewById(R.id.*backgroundBoxHistory*);  
 mRecyclerIncome = view.findViewById(R.id.*recyclerIncome*);  
 mRecyclerOutcome = view.findViewById(R.id.*recyclerOutcome*);  
 tvIncome = view.findViewById(R.id.*tvIncome*);  
 tvOutcome = view.findViewById(R.id.*tvOutcome*);  
 showFilterBtn = view.findViewById(R.id.*btnFilter*);  
  
 //Setup RecyclerView  
 LinearLayoutManager layoutManagerIncome = new ZoomRecyclerLayout(getActivity(), LinearLayoutManager.*HORIZONTAL*, false);  
 SnapHelper snapHelperIncome = new LinearSnapHelper();  
 layoutManagerIncome.setStackFromEnd(true);  
 layoutManagerIncome.setReverseLayout(true);  
 snapHelperIncome.attachToRecyclerView(mRecyclerIncome);  
 mRecyclerIncome.setHasFixedSize(true);  
 mRecyclerIncome.setNestedScrollingEnabled(false);  
 mRecyclerIncome.setLayoutManager(layoutManagerIncome);  
  
 LinearLayoutManager layoutManagerOutcome = new ZoomRecyclerLayout(getActivity(), LinearLayoutManager.*HORIZONTAL*, false);  
 SnapHelper snapHelperOutcome = new LinearSnapHelper();  
 layoutManagerOutcome.setStackFromEnd(true);  
 layoutManagerOutcome.setReverseLayout(true);  
 snapHelperOutcome.attachToRecyclerView(mRecyclerOutcome);  
 mRecyclerOutcome.setHasFixedSize(true);  
 mRecyclerOutcome.setNestedScrollingEnabled(false);  
 mRecyclerOutcome.setLayoutManager(layoutManagerOutcome);  
  
 //init popup menu  
 Context wrapper = new ContextThemeWrapper(getActivity().getApplicationContext(), R.style.*popupMenuStyle*);  
 PopupMenu popupMenu = new PopupMenu(wrapper,showFilterBtn);  
 popupMenu.getMenuInflater().inflate(R.menu.*filter\_popup*, popupMenu.getMenu());  
  
 //handle popup menu item clicks  
 popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem item) {  
 int id = item.getItemId();  
 Query incomeQuery, outcomeQuery;  
 String dateTarget;  
  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Calendar cal = Calendar.*getInstance*();  
 Date date = cal.getTime();  
 String mDate = df.format(date);  
 FirebaseRecyclerAdapter<Data, IncomeViewHolder> incomeAdapter;  
 FirebaseRecyclerAdapter<Data, OutcomeViewHolder> outcomeAdapter;  
  
 if (id == R.id.*showAll*){  
 Toast.*makeText*(getActivity().getApplicationContext(), "Show all transaction history", Toast.*LENGTH\_SHORT*).show();  
  
 incomeAdapter = new FirebaseRecyclerAdapter<Data, IncomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_income\_item*,  
 History.IncomeViewHolder.class,  
 mIncomeDatabase  
 ) {  
 @Override  
 protected void populateViewHolder(IncomeViewHolder incomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 incomeViewHolder.setIncomeType(model.getType());  
 incomeViewHolder.setIncomeAmount(model.getAmount());  
 incomeViewHolder.setIncomeDate(formattedDate);  
 }  
 };  
  
  
  
 outcomeAdapter = new FirebaseRecyclerAdapter<Data, OutcomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_outcome\_item*,  
 History.OutcomeViewHolder.class,  
 mOutcomeDatabase  
 ) {  
 @Override  
 protected void populateViewHolder(OutcomeViewHolder outcomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 outcomeViewHolder.setOutcomeType(model.getType());  
 outcomeViewHolder.setOutcomeAmount(model.getAmount());  
 outcomeViewHolder.setOutcomeDate(formattedDate);  
 }  
 };  
 mRecyclerIncome.setAdapter(incomeAdapter);  
 mRecyclerOutcome.setAdapter(outcomeAdapter);  
 return true;  
 }  
  
 else if (id == R.id.*today*) {  
 Toast.*makeText*(getActivity().getApplicationContext(), "Today transaction history", Toast.*LENGTH\_SHORT*).show();  
  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(mDate).endAt(mDate);  
 outcomeQuery = mOutcomeDatabase.orderByChild("date").startAt(mDate).endAt(mDate);  
  
 incomeAdapter = new FirebaseRecyclerAdapter<Data, IncomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_income\_item*,  
 History.IncomeViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(IncomeViewHolder incomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 incomeViewHolder.setIncomeType(model.getType());  
 incomeViewHolder.setIncomeAmount(model.getAmount());  
 incomeViewHolder.setIncomeDate(formattedDate);  
 }  
 };  
  
 outcomeAdapter = new FirebaseRecyclerAdapter<Data, OutcomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_outcome\_item*,  
 History.OutcomeViewHolder.class,  
 outcomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(OutcomeViewHolder outcomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 outcomeViewHolder.setOutcomeType(model.getType());  
 outcomeViewHolder.setOutcomeAmount(model.getAmount());  
 outcomeViewHolder.setOutcomeDate(formattedDate);  
 }  
 };  
 mRecyclerIncome.setAdapter(incomeAdapter);  
 mRecyclerOutcome.setAdapter(outcomeAdapter);  
 return true;  
 }  
  
 else if (id == R.id.*sevenDays*) {  
 cal.add(Calendar.*DATE*,-7);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 Toast.*makeText*(getActivity().getApplicationContext(), "7 days transaction history", Toast.*LENGTH\_SHORT*).show();  
  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 outcomeQuery = mOutcomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
  
 incomeAdapter = new FirebaseRecyclerAdapter<Data, IncomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_income\_item*,  
 History.IncomeViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(IncomeViewHolder incomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 incomeViewHolder.setIncomeType(model.getType());  
 incomeViewHolder.setIncomeAmount(model.getAmount());  
 incomeViewHolder.setIncomeDate(formattedDate);  
 }  
 };  
  
 outcomeAdapter = new FirebaseRecyclerAdapter<Data, OutcomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_outcome\_item*,  
 History.OutcomeViewHolder.class,  
 outcomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(OutcomeViewHolder outcomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 outcomeViewHolder.setOutcomeType(model.getType());  
 outcomeViewHolder.setOutcomeAmount(model.getAmount());  
 outcomeViewHolder.setOutcomeDate(formattedDate);  
 }  
 };  
 mRecyclerIncome.setAdapter(incomeAdapter);  
 mRecyclerOutcome.setAdapter(outcomeAdapter);  
 return true;  
 }  
  
 else if (id == R.id.*thirtyOneDays*) {  
 cal.add(Calendar.*DATE*,-31);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 Toast.*makeText*(getActivity().getApplicationContext(), "7 days transaction history", Toast.*LENGTH\_SHORT*).show();  
  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 outcomeQuery = mOutcomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
  
 incomeAdapter = new FirebaseRecyclerAdapter<Data, IncomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_income\_item*,  
 History.IncomeViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(IncomeViewHolder incomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 incomeViewHolder.setIncomeType(model.getType());  
 incomeViewHolder.setIncomeAmount(model.getAmount());  
 incomeViewHolder.setIncomeDate(formattedDate);  
 }  
 };  
  
 outcomeAdapter = new FirebaseRecyclerAdapter<Data, OutcomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_outcome\_item*,  
 History.OutcomeViewHolder.class,  
 outcomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(OutcomeViewHolder outcomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 outcomeViewHolder.setOutcomeType(model.getType());  
 outcomeViewHolder.setOutcomeAmount(model.getAmount());  
 outcomeViewHolder.setOutcomeDate(formattedDate);  
 }  
 };  
 mRecyclerIncome.setAdapter(incomeAdapter);  
 mRecyclerOutcome.setAdapter(outcomeAdapter);  
 return true;  
 }  
 return false;  
 }  
 });  
  
 showFilterBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 popupMenu.show();  
 }  
 });  
  
 return view;  
 }  
  
 @Override  
 public void onStart() {  
 super.onStart();  
  
 FirebaseRecyclerAdapter<Data, IncomeViewHolder> incomeAdapter = new FirebaseRecyclerAdapter<Data, IncomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_income\_item*,  
 History.IncomeViewHolder.class,  
 mIncomeDatabase  
 ) {  
 @Override  
 protected void populateViewHolder(IncomeViewHolder incomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 incomeViewHolder.setIncomeType(model.getType());  
 incomeViewHolder.setIncomeAmount(model.getAmount());  
 incomeViewHolder.setIncomeDate(formattedDate);  
 }  
 };  
  
 FirebaseRecyclerAdapter<Data, OutcomeViewHolder> outcomeAdapter = new FirebaseRecyclerAdapter<Data, OutcomeViewHolder>  
 (  
 Data.class,  
 R.layout.*history\_outcome\_item*,  
 History.OutcomeViewHolder.class,  
 mOutcomeDatabase  
 ) {  
 @Override  
 protected void populateViewHolder(OutcomeViewHolder outcomeViewHolder, Data model, int i) {  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 outcomeViewHolder.setOutcomeType(model.getType());  
 outcomeViewHolder.setOutcomeAmount(model.getAmount());  
 outcomeViewHolder.setOutcomeDate(formattedDate);  
 }  
 };  
  
 mRecyclerIncome.setAdapter(incomeAdapter);  
 mRecyclerOutcome.setAdapter(outcomeAdapter);  
 }  
  
 // Income Data  
 public static class IncomeViewHolder extends RecyclerView.ViewHolder {  
 View mIncomeView;  
  
 public IncomeViewHolder(View itemView) {  
 super(itemView);  
 mIncomeView = itemView;  
 }  
  
 public void setIncomeType(String type) {  
 TextView mtype = mIncomeView.findViewById(R.id.*IncomeTitle*);  
 mtype.setText(type);  
 }  
  
 public void setIncomeAmount(Integer amount) {  
 TextView mAmount = mIncomeView.findViewById(R.id.*IncomeAmount*);  
 String stAmount = String.*valueOf*(amount);  
 mAmount.setText(stAmount);  
 }  
  
 public void setIncomeDate(String date) {  
 TextView mDate = mIncomeView.findViewById(R.id.*IncomeDate*);  
 mDate.setText(date);  
 }  
 }  
  
  
 // Outcome Data  
 public static class OutcomeViewHolder extends RecyclerView.ViewHolder {  
 View mOutcomeView;  
  
 public OutcomeViewHolder(View itemView) {  
 super(itemView);  
 mOutcomeView = itemView;  
 }  
  
 public void setOutcomeType(String type) {  
 TextView mtype = mOutcomeView.findViewById(R.id.*OutcomeTitle*);  
 mtype.setText(type);  
 }  
  
 public void setOutcomeAmount(Integer amount) {  
 TextView mAmount = mOutcomeView.findViewById(R.id.*OutcomeAmount*);  
 String stAmount = String.*valueOf*(amount);  
 mAmount.setText(stAmount);  
 }  
  
 public void setOutcomeDate(String date) {  
 TextView mDate = mOutcomeView.findViewById(R.id.*OutcomeDate*);  
 mDate.setText(date);  
 }  
 }  
  
 public void playAnimOut(){  
 backgroundBox.animate().translationY(300).alpha(0).setDuration(200);  
 tvIncome.animate().scaleX(0.5f).scaleY(0.5f).alpha(0).setDuration(200);  
 mRecyclerIncome.animate().scaleX(0.5f).scaleY(0.5f).alpha(0).setDuration(200);  
 tvOutcome.animate().scaleX(0.5f).scaleY(0.5f).alpha(0).setDuration(200);  
 mRecyclerOutcome.animate().scaleX(0.5f).scaleY(0.5f).alpha(0).setDuration(200);  
 }  
  
 public void playAnimIn(){  
 backgroundBox.animate().translationY(0).alpha(1).setDuration(600);  
 tvIncome.animate().scaleX(1).scaleY(1).alpha(1).setDuration(400);  
 mRecyclerIncome.animate().scaleX(1).scaleY(1).alpha(1).setDuration(400);  
 tvOutcome.animate().scaleX(1).scaleY(1).alpha(1).setDuration(400).setStartDelay(400);  
 mRecyclerOutcome.animate().scaleX(1).scaleY(1).alpha(1).setDuration(500).setStartDelay(400);  
 }  
}

**HomeActivity.java**

mport androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.Fragment;  
import androidx.viewpager.widget.ViewPager;  
  
import android.os.Bundle;  
import android.view.View;  
  
import com.google.android.material.tabs.TabLayout;  
  
import java.util.ArrayList;  
  
public class HomeActivity extends AppCompatActivity {  
  
 //Initialize Variable  
 public static TabLayout *tabLayoutHome*;  
 public static ViewPager *viewPager*;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_home*);  
  
 //Assign Variable  
 *tabLayoutHome* = findViewById(R.id.*homeTabLayout*);  
 *viewPager* = findViewById(R.id.*viewpagerHome*);  
  
 //Initialize View Pager  
 ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());  
 adapter.addFrag(new Income());  
 adapter.addFrag(new UserPanel());  
 adapter.addFrag(new Dashboard());  
 adapter.addFrag(new History());  
 adapter.addFrag(new Outcome());  
 *viewPager*.setAdapter(adapter);  
  
 *tabLayoutHome*.setupWithViewPager(*viewPager*);  
 setCustomTabs();  
 *viewPager*.setCurrentItem(2);  
 *tabLayoutHome*.getTabAt(2).select();  
  
 *tabLayoutHome*.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {  
 @Override  
 public void onTabSelected(TabLayout.Tab tab) {  
 *viewPager*.setCurrentItem(tab.getPosition());  
 if(tab.getPosition() == 0) {  
 Fragment income = adapter.getItem(0);  
 ((Income) income).playAnimIn();  
 }  
 if(tab.getPosition() == 1) {  
 Fragment userpanel = adapter.getItem(1);  
 ((UserPanel) userpanel).playAnimIn();  
 }  
 if(tab.getPosition() == 2) {  
 Fragment dashboard = adapter.getItem(2);  
 ((Dashboard) dashboard).playAnimIn();  
 }  
 if(tab.getPosition() == 3) {  
 Fragment history = adapter.getItem(3);  
 ((History) history).playAnimIn();  
 }  
 if(tab.getPosition() == 4) {  
 Fragment outcome = adapter.getItem(4);  
 ((Outcome) outcome).playAnimIn();  
 }  
 }  
 @Override  
 public void onTabUnselected(TabLayout.Tab tab) {  
 if(tab.getPosition() == 0) {  
 Fragment income = adapter.getItem(0);  
 ((Income) income).playAnimOut();  
 }  
 if(tab.getPosition() == 1){  
 Fragment userpanel = adapter.getItem(1);  
 ((UserPanel) userpanel).playAnimOut();  
 }  
 if(tab.getPosition() == 2) {  
 Fragment dashboard = adapter.getItem(2);  
 ((Dashboard) dashboard).playAnimOut();  
 }  
 if(tab.getPosition() == 3) {  
 Fragment history = adapter.getItem(3);  
 ((History) history).playAnimOut();  
 }  
 if(tab.getPosition() == 4) {  
 Fragment outcome = adapter.getItem(4);  
 ((Outcome) outcome).playAnimOut();  
 }  
 }  
 @Override  
 public void onTabReselected(TabLayout.Tab tab) {  
 }  
 });  
 }  
  
  
 private void setCustomTabs() {  
  
 View view1 = getLayoutInflater().inflate(R.layout.*tab\_icon\_custom*, null);  
 view1.findViewById(R.id.*icon*).setBackgroundResource(R.drawable.*ic\_income\_13*);  
 *tabLayoutHome*.getTabAt(0).setCustomView(view1);  
  
 View view2 = getLayoutInflater().inflate(R.layout.*tab\_icon\_custom*, null);  
 view2.findViewById(R.id.*icon*).setBackgroundResource(R.drawable.*ic\_userpanel\_13*);  
 *tabLayoutHome*.getTabAt(1).setCustomView(view2);  
  
 View view3 = getLayoutInflater().inflate(R.layout.*tab\_icon\_custom*, null);  
 view3.findViewById(R.id.*icon*).setBackgroundResource(R.drawable.*ic\_dashboardicon\_13*);  
 *tabLayoutHome*.getTabAt(2).setCustomView(view3);  
  
 View view4 = getLayoutInflater().inflate(R.layout.*tab\_icon\_custom*, null);  
 view4.findViewById(R.id.*icon*).setBackgroundResource(R.drawable.*ic\_history\_13*);  
 *tabLayoutHome*.getTabAt(3).setCustomView(view4);  
  
 View view5 = getLayoutInflater().inflate(R.layout.*tab\_icon\_custom*, null);  
 view5.findViewById(R.id.*icon*).setBackgroundResource(R.drawable.*ic\_outcome\_13*);  
 *tabLayoutHome*.getTabAt(4).setCustomView(view5);  
 }  
}

**Income.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import android.app.Dialog;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.constraintlayout.widget.ConstraintLayout;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Spinner;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.ChildEventListener;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.Query;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Calendar;  
import java.util.Date;  
  
import id.ac.umn.leleair.kelompok.smallchange.Model.Data;  
  
public class Income extends Fragment {  
 private Spinner filter;  
 private ConstraintLayout PageTitle;  
 private ImageView backgroundBox;  
 private FloatingActionButton fabAddIncome;  
  
 //Firebase  
 private FirebaseAuth mAuth;  
 private DatabaseReference mIncomeDatabase;  
  
 //RecyclerView  
 private RecyclerView recyclerView;  
  
 //Initialize for Update Item  
 private EditText editAmount;  
 private EditText editType;  
 private EditText editNote;  
 private Button btnUpdate;  
 private Button btnDelete;  
  
 //Data item value  
 private String type;  
 private String note;  
 private int amount;  
 private String post\_key;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view = inflater.inflate(R.layout.*fragment\_income*, container, false);  
  
 //Firebase Initialization  
 mAuth = FirebaseAuth.*getInstance*();  
 FirebaseUser mUser = mAuth.getCurrentUser();  
 String uid = mUser.getUid();  
  
 mIncomeDatabase = FirebaseDatabase.*getInstance*().getReference().child("IncomeData").child(uid);  
  
 filter = view.findViewById(R.id.*filterIncome*);  
 PageTitle = view.findViewById(R.id.*PageTitleIncome*);  
 backgroundBox = view.findViewById(R.id.backgroundBoxIncome);  
 fabAddIncome = view.findViewById((R.id.fabAddIncome));  
 recyclerView = view.findViewById((R.id.recyclerViewIncome));  
  
 LinearLayoutManager layoutManager = new LinearLayoutManager(getActivity());  
  
 layoutManager.setReverseLayout(true);  
 layoutManager.setStackFromEnd(true);  
 recyclerView.setHasFixedSize(true);  
 recyclerView.setLayoutManager(layoutManager);  
  
 initializeFilter();  
  
 fabAddIncome.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 insertIncomeData();  
 }  
 });  
  
 return view;  
 }  
  
  
 @Override  
 public void onStart(){  
 super.onStart();  
  
 filter.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parentView, View selectedItemView, int position, long id) {  
 Query incomeQuery;  
 String dateTarget;  
  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Calendar cal = Calendar.getInstance();  
 Date date = cal.getTime();  
 String mDate = df.format(date);  
  
  
 if (position == 0) {  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 mIncomeDatabase  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateIncomeData();  
 }  
 });  
 }  
 };  
  
 mIncomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
 }  
  
 else if (position == 1){  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(mDate).endAt(mDate);  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateIncomeData();  
 }  
 });  
 }  
 };  
  
 mIncomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
 }  
 else if (position == 2){  
 cal.add(Calendar.DATE,-7);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateIncomeData();  
 }  
 });  
 }  
 };  
  
 mIncomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
  
 }  
 else if (position == 3){  
 cal.add(Calendar.DATE,-31);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 incomeQuery = mIncomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateIncomeData();  
 }  
 });  
 }  
 };  
  
 mIncomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
  
 }  
  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parentView) {  
 // your code here  
 }  
  
 });  
 }  
  
 public void insertIncomeData(){  
 //New Transaction Form  
 Dialog mdialog = new Dialog(getActivity());  
 LayoutInflater inflater = LayoutInflater.from(getActivity());  
 View myviewm = inflater.inflate(R.layout.insert\_data\_form, null);  
 mdialog.setContentView(myviewm);  
 mdialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));  
  
 EditText editAmount = myviewm.findViewById(R.id.editTransactionAmount);  
 EditText editType = myviewm.findViewById(R.id.editTransactionName);  
 EditText editNote = myviewm.findViewById(R.id.editTransactionNote);  
  
 Button btnSave = myviewm.findViewById(R.id.btnSaveTransaction);  
  
 btnSave.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String type = editType.getText().toString().trim();  
 String amount = editAmount.getText().toString().trim();  
 String note = editNote.getText().toString().trim();  
  
 // Validate input from new transaction form  
 if (TextUtils.isEmpty(amount)) {  
 editAmount.setError("Required Field");  
 return;  
 }  
 if (TextUtils.isEmpty(type)) {  
 editType.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateTitle(type)){  
 editType.setError("Max 16 characters");  
 return;  
 }  
 }  
 if (TextUtils.isEmpty(note)) {  
 editNote.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateNote(note)){  
 editNote.setError("Max 30 characters");  
 return;  
 }  
 }  
 int ouramountint = Integer.parseInt((amount));  
  
 // Save transaction to database  
 String id = mIncomeDatabase.push().getKey();  
 if(id != null){  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = Calendar.getInstance().getTime();  
 String mDate = df.format(date);  
 Data data = new Data(ouramountint, type, note, id, mDate);  
 mIncomeDatabase.child(id).setValue(data);  
 Toast.makeText(getActivity(), "Transaction added successfully", Toast.LENGTH\_SHORT).show();  
  
 mdialog.dismiss();  
 }  
 }  
 });  
 mdialog.show();  
 }  
  
  
 public void playAnimIn(){  
 backgroundBox.animate().translationY(0).alpha(1).setDuration(600);  
 PageTitle.animate().translationY(0).alpha(1).setDuration(400);  
 filter.animate().alpha(1).setDuration(400).setStartDelay(600);  
 fabAddIncome.animate().translationY(0).alpha(1).setDuration(400);  
 recyclerView.animate().translationY(0).alpha(1).setDuration(400).setStartDelay(600);  
 }  
  
 public void playAnimOut(){  
 backgroundBox.animate().translationY(300).alpha(0).setDuration(200);  
 PageTitle.animate().translationY(-130).alpha(0).setDuration(200);  
 filter.animate().alpha(0).setDuration(200);  
 fabAddIncome.animate().translationY(100).alpha(0).setDuration(200);  
 recyclerView.animate().translationY(100).alpha(0).setDuration(200);  
 }  
  
 private void initializeFilter() {  
 String[] value = {"Show All", "Today", "7 days ago", "31 days ago"};  
 ArrayList<String> arrayList = new ArrayList<>(Arrays.asList(value));  
 ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(getActivity(),R.layout.customview\_spinner,arrayList){  
  
 @Override  
 public View getDropDownView(int position, View convertView,ViewGroup parent) {  
 // TODO Auto-generated method stub  
  
 View view = super.getView(position, convertView, parent);  
  
 TextView text = (TextView)view.findViewById(R.id.tvFilter);  
 text.setTextColor(getResources().getColor(R.color.biru));  
  
 return view;  
  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 // TODO Auto-generated method stub  
  
 View view = super.getView(position, convertView, parent);  
  
 TextView text = (TextView)view.findViewById(R.id.tvFilter);  
 text.setTextColor(getResources().getColor(R.color.white));  
  
 return view;  
  
 }  
 };  
 filter.setAdapter(arrayAdapter);  
 }  
  
 private void updateIncomeData(){  
 //Update Transaction Form  
 Dialog mdialog = new Dialog(getActivity());  
 LayoutInflater inflater = LayoutInflater.from(getActivity());  
 View myviewm = inflater.inflate(R.layout.update\_data\_form,null);  
 mdialog.setContentView(myviewm);  
 mdialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));  
  
 editAmount = myviewm.findViewById(R.id.editTransactionAmount);  
 editType = myviewm.findViewById(R.id.editTransactionName);  
 editNote = myviewm.findViewById(R.id.editTransactionNote);  
  
 //Set Income data to edit text  
 editType.setText(type);  
 editType.setSelection(type.length());  
  
 editNote.setText(note);  
 editNote.setSelection(note.length());  
  
 editAmount.setText(String.valueOf(amount));  
 editAmount.setSelection(String.valueOf(amount).length());  
  
 //Define button  
 btnUpdate = myviewm.findViewById(R.id.btnUpdateTransaction);  
 btnDelete = myviewm.findViewById(R.id.btnDeleteTransaction);  
  
 btnUpdate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 type = editType.getText().toString().trim();  
 note = editNote.getText().toString().trim();  
  
 String amountValue = String.valueOf(amount);  
 amountValue = editAmount.getText().toString().trim();  
  
  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = Calendar.getInstance().getTime();  
 String mDate = df.format(date);  
  
 // Validate input from update transaction form  
 if (TextUtils.isEmpty(amountValue)) {  
 editAmount.setError("Required Field");  
 return;  
 }  
 if (TextUtils.isEmpty(type)) {  
 editType.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateTitle(type)){  
 editType.setError("Max 16 characters");  
 return;  
 }  
 }  
 if (TextUtils.isEmpty(note)) {  
 editNote.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateNote(note)){  
 editNote.setError("Max 30 characters");  
 return;  
 }  
 }  
 int myAmount = Integer.parseInt(amountValue);  
  
 // Update transaction to database  
 if(post\_key != null){  
 Data data = new Data(myAmount, type, note, post\_key, mDate);  
 mIncomeDatabase.child(post\_key).setValue(data);  
 Toast.makeText(getActivity(), "Transaction has been updated", Toast.LENGTH\_SHORT).show();  
  
 mdialog.dismiss();  
 }  
 }  
 });  
  
 btnDelete.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 mIncomeDatabase.child(post\_key).removeValue();  
 mdialog.dismiss();  
 }  
 });  
 mdialog.show();  
 }  
  
 public boolean validateTitle(String title) {  
 return title.length() <= 16;  
 }  
  
 public boolean validateNote(String note) {  
 return note.length() <= 16;  
 }  
}

**Login.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.fragment.app.FragmentManager;  
import androidx.viewpager2.widget.ViewPager2;  
  
import android.content.Context;  
import android.content.Intent;  
import android.net.ConnectivityManager;  
import android.os.Bundle;  
import android.provider.Settings;  
import android.widget.Toast;  
  
import com.google.android.material.tabs.TabLayout;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
import java.lang.reflect.Method;  
  
public class Login extends AppCompatActivity {  
 TabLayout tabLayoutLogin;  
 ViewPager2 pager2Login;  
 LoginFragmentAdapter loginAdapter;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 tabLayoutLogin = findViewById(R.id.*loginTabLayout*);  
 pager2Login = findViewById(R.id.*viewpagerLogin*);  
  
 checkUserSession();  
  
 FragmentManager fm = getSupportFragmentManager();  
 loginAdapter = new LoginFragmentAdapter(fm, getLifecycle());  
 pager2Login.setAdapter(loginAdapter);  
  
 tabLayoutLogin.addTab(tabLayoutLogin.newTab().setText("Sign In"));  
 tabLayoutLogin.addTab(tabLayoutLogin.newTab().setText("Sign Up"));  
  
 tabLayoutLogin.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {  
 @Override  
 public void onTabSelected(TabLayout.Tab tab) {  
 pager2Login.setCurrentItem(tab.getPosition());  
 }  
  
 @Override  
 public void onTabUnselected(TabLayout.Tab tab) {  
  
 }  
  
 @Override  
 public void onTabReselected(TabLayout.Tab tab) {  
  
 }  
 });  
  
 pager2Login.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {  
 @Override  
 public void onPageSelected(int position) {  
 tabLayoutLogin.selectTab(tabLayoutLogin.getTabAt(position));  
 }  
 });  
 }  
  
 private void checkUserSession(){  
 //Initialize Firebase Auth & User  
 FirebaseAuth firebaseAuth = FirebaseAuth.*getInstance*();  
 FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();  
  
 //check if user already signed in  
 if(firebaseUser != null){  
 Intent intent = new Intent(this, HomeActivity.class);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*|Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
 }  
  
  
}

**LoginFragmentAdapter.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.lifecycle.Lifecycle;  
import androidx.viewpager2.adapter.FragmentStateAdapter;  
  
public class LoginFragmentAdapter extends FragmentStateAdapter {  
 public LoginFragmentAdapter(@NonNull FragmentManager fragmentManager, @NonNull Lifecycle lifecycle) {  
 super(fragmentManager, lifecycle);  
 }  
  
 @NonNull  
 @Override  
 public Fragment createFragment(int position) {  
 if (position == 1){  
 return new SignUp();  
 }  
 return new SignIn();  
 }  
  
 @Override  
 public int getItemCount() {  
 return 2;  
 }  
}

**MainActivity.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
}

**MyViewHolder.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import android.view.View;  
import android.widget.TextView;  
  
import androidx.recyclerview.widget.RecyclerView;  
  
public class MyViewHolder extends RecyclerView.ViewHolder {  
 View mView;  
  
 public MyViewHolder(View itemView){  
 super(itemView);  
 mView = itemView;  
 }  
  
 public void setType(String type){  
 TextView mType = mView.findViewById(R.id.*tvType*);  
 mType.setText(type);  
 }  
  
 public void setNote(String note){  
 TextView mNote = mView.findViewById(R.id.*tvNote*);  
 mNote.setText(note);  
 }  
  
 public void setDate(String date){  
 TextView mDate = mView.findViewById(R.id.*tvDate*);  
 mDate.setText(date);  
 }  
  
 public void setAmount(int amount){  
 TextView mAmount = mView.findViewById(R.id.*tvAmount*);  
 String stamount = String.*valueOf*(amount);  
 mAmount.setText(stamount);  
 }  
}

**Outcome.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import android.app.Dialog;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.constraintlayout.widget.ConstraintLayout;  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Spinner;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.ChildEventListener;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.Query;  
  
import java.text.DateFormat;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Calendar;  
import java.util.Date;  
  
import id.ac.umn.leleair.kelompok.smallchange.Model.Data;  
  
public class Outcome extends Fragment {  
 private Spinner filter;  
 private ConstraintLayout PageTitle;  
 private ImageView backgroundBox;  
 private FloatingActionButton fabAddOutcome;  
  
 //Firebase  
 private FirebaseAuth mAuth;  
 private DatabaseReference mOutcomeDatabase;  
  
 //RecyclerView  
 private RecyclerView recyclerView;  
  
 //Initialize for Update Item  
 private EditText editAmount;  
 private EditText editType;  
 private EditText editNote;  
 private Button btnUpdate;  
 private Button btnDelete;  
  
 //Data item value  
 private String type;  
 private String note;  
 private int amount;  
 private String post\_key;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view = inflater.inflate(R.layout.*fragment\_outcome*, container, false);  
  
 //Firebase Initialization  
 mAuth = FirebaseAuth.*getInstance*();  
 FirebaseUser mUser = mAuth.getCurrentUser();  
 String uid = mUser.getUid();  
  
 mOutcomeDatabase = FirebaseDatabase.*getInstance*().getReference().child("OutcomeData").child(uid);  
  
 filter = view.findViewById(R.id.*filterOutcome*);  
 PageTitle = view.findViewById(R.id.*PageTitleOutcome*);  
 backgroundBox = view.findViewById(R.id.*backgroundBoxOutcome*);  
 fabAddOutcome = view.findViewById((R.id.*fabAddOutcome*));  
 recyclerView = view.findViewById((R.id.*recyclerViewOutcome*));  
  
 LinearLayoutManager layoutManager = new LinearLayoutManager(getActivity());  
  
 layoutManager.setReverseLayout(true);  
 layoutManager.setStackFromEnd(true);  
 recyclerView.setHasFixedSize(true);  
 recyclerView.setLayoutManager(layoutManager);  
  
 initializeFilter();  
  
 fabAddOutcome.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 insertOutcomeData();  
 }  
 });  
  
 return view;  
 }  
  
  
 @Override  
 public void onStart(){  
 super.onStart();  
  
 filter.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parentView, View selectedItemView, int position, long id) {  
 Query incomeQuery;  
 String dateTarget;  
  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Calendar cal = Calendar.getInstance();  
 Date date = cal.getTime();  
 String mDate = df.format(date);  
  
  
 if (position == 0) {  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 mOutcomeDatabase  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateOutcomeData();  
 }  
 });  
 }  
 };  
  
 mOutcomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
 }  
  
 else if (position == 1){  
 incomeQuery = mOutcomeDatabase.orderByChild("date").startAt(mDate).endAt(mDate);  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateOutcomeData();  
 }  
 });  
 }  
 };  
  
 mOutcomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
 }  
 else if (position == 2){  
 cal.add(Calendar.DATE,-7);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 incomeQuery = mOutcomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateOutcomeData();  
 }  
 });  
 }  
 };  
  
 mOutcomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
  
 }  
 else if (position == 3){  
 cal.add(Calendar.DATE,-31);  
 date = cal.getTime();  
 dateTarget = df.format(date);  
  
 incomeQuery = mOutcomeDatabase.orderByChild("date").startAt(dateTarget).endAt(mDate);  
 FirebaseRecyclerAdapter<Data, MyViewHolder>adapter = new FirebaseRecyclerAdapter<Data, MyViewHolder>  
 (  
 Data.class,  
 R.layout.income\_item,  
 MyViewHolder.class,  
 incomeQuery  
 ) {  
 @Override  
 protected void populateViewHolder(MyViewHolder viewHolder, Data model, int position) {  
 viewHolder.setType(model.getType());  
 viewHolder.setNote(model.getNote());  
  
 //parse date format  
 String unFormattedDate = model.getDate();  
 SimpleDateFormat inFormat = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = null;  
 try {  
 date = inFormat.parse(unFormattedDate);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 SimpleDateFormat outFormat = new SimpleDateFormat("MMM dd, yyyy");  
 String formattedDate = outFormat.format(date);  
  
 //set formatted date  
 viewHolder.setDate(formattedDate);  
 viewHolder.setAmount(model.getAmount());  
  
 viewHolder.mView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 post\_key = getRef(position).getKey();  
  
 type = model.getType();  
 note = model.getNote();  
 amount = model.getAmount();  
  
 updateOutcomeData();  
 }  
 });  
 }  
 };  
  
 mOutcomeDatabase.addChildEventListener(new ChildEventListener() {  
 @Override  
 public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
  
 //Set Recycler view adapter  
 recyclerView.setAdapter(adapter);  
  
 }  
  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parentView) {  
 // your code here  
 }  
  
 });  
 }  
  
 public void insertOutcomeData(){  
 //New Transaction Form  
 Dialog mdialog = new Dialog(getActivity());  
 LayoutInflater inflater = LayoutInflater.from(getActivity());  
 View myviewm = inflater.inflate(R.layout.insert\_data\_form, null);  
 mdialog.setContentView(myviewm);  
 mdialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));  
  
 EditText editAmount = myviewm.findViewById(R.id.editTransactionAmount);  
 EditText editType = myviewm.findViewById(R.id.editTransactionName);  
 EditText editNote = myviewm.findViewById(R.id.editTransactionNote);  
  
 Button btnSave = myviewm.findViewById(R.id.btnSaveTransaction);  
  
 btnSave.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String type = editType.getText().toString().trim();  
 String amount = editAmount.getText().toString().trim();  
 String note = editNote.getText().toString().trim();  
  
 // Validate input from new transaction form  
 if (TextUtils.isEmpty(amount)) {  
 editAmount.setError("Required Field");  
 return;  
 }  
 if (TextUtils.isEmpty(type)) {  
 editType.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateTitle(type)){  
 editType.setError("Max 16 characters");  
 return;  
 }  
 }  
 if (TextUtils.isEmpty(note)) {  
 editNote.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateNote(note)){  
 editNote.setError("Max 30 characters");  
 return;  
 }  
 }  
 int ouramountint = Integer.parseInt((amount));  
  
 // Save transaction to database  
 String id = mOutcomeDatabase.push().getKey();  
 if(id != null){  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = Calendar.getInstance().getTime();  
 String mDate = df.format(date);  
 Data data = new Data(ouramountint, type, note, id, mDate);  
 mOutcomeDatabase.child(id).setValue(data);  
 Toast.makeText(getActivity(), "Transaction added successfully", Toast.LENGTH\_SHORT).show();  
  
 mdialog.dismiss();  
 }  
 }  
 });  
 mdialog.show();  
 }  
  
 public void playAnimIn(){  
 backgroundBox.animate().translationY(0).alpha(1).setDuration(600);  
 PageTitle.animate().translationY(0).alpha(1).setDuration(400);  
 filter.animate().alpha(1).setDuration(400).setStartDelay(600);  
 fabAddOutcome.animate().translationY(0).alpha(1).setDuration(400);  
 recyclerView.animate().translationY(0).alpha(1).setDuration(400).setStartDelay(600);  
 }  
  
 public void playAnimOut(){  
 backgroundBox.animate().translationY(300).alpha(0).setDuration(200);  
 PageTitle.animate().translationY(-130).alpha(0).setDuration(200);  
 filter.animate().alpha(0).setDuration(200);  
 fabAddOutcome.animate().translationY(100).alpha(0).setDuration(200);  
 recyclerView.animate().translationY(100).alpha(0).setDuration(200);  
 }  
  
 private void initializeFilter() {  
 String[] value = {"Show All", "Today", "7 days ago", "31 days ago"};  
 ArrayList<String> arrayList = new ArrayList<>(Arrays.asList(value));  
 ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(getActivity(),R.layout.customview\_spinner,arrayList){  
  
 @Override  
 public View getDropDownView(int position, View convertView,ViewGroup parent) {  
 // TODO Auto-generated method stub  
  
 View view = super.getView(position, convertView, parent);  
  
 TextView text = (TextView)view.findViewById(R.id.tvFilter);  
 text.setTextColor(getResources().getColor(R.color.biru));  
  
 return view;  
  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 // TODO Auto-generated method stub  
  
 View view = super.getView(position, convertView, parent);  
  
 TextView text = (TextView)view.findViewById(R.id.tvFilter);  
 text.setTextColor(getResources().getColor(R.color.white));  
  
 return view;  
  
 }  
 };  
 filter.setAdapter(arrayAdapter);  
 }  
  
 private void updateOutcomeData(){  
 //Update Transaction Form  
 Dialog mdialog = new Dialog(getActivity());  
 LayoutInflater inflater = LayoutInflater.from(getActivity());  
 View myviewm = inflater.inflate(R.layout.update\_data\_form, null);  
 mdialog.setContentView(myviewm);  
 mdialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));  
  
 editAmount = myviewm.findViewById(R.id.editTransactionAmount);  
 editType = myviewm.findViewById(R.id.editTransactionName);  
 editNote = myviewm.findViewById(R.id.editTransactionNote);  
  
 //Set Income data to edit text  
 editType.setText(type);  
 editType.setSelection(type.length());  
  
 editNote.setText(note);  
 editNote.setSelection(note.length());  
  
 editAmount.setText(String.valueOf(amount));  
 editAmount.setSelection(String.valueOf(amount).length());  
  
 //Define button  
 btnUpdate = myviewm.findViewById(R.id.btnUpdateTransaction);  
 btnDelete = myviewm.findViewById(R.id.btnDeleteTransaction);  
  
 btnUpdate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 type = editType.getText().toString().trim();  
 note = editNote.getText().toString().trim();  
  
 String amountValue = String.valueOf(amount);  
 amountValue = editAmount.getText().toString().trim();  
  
  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");  
 Date date = Calendar.getInstance().getTime();  
 String mDate = df.format(date);  
  
 // Validate input from update transaction form  
 if (TextUtils.isEmpty(amountValue)) {  
 editAmount.setError("Required Field");  
 return;  
 }  
 if (TextUtils.isEmpty(type)) {  
 editType.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateTitle(type)){  
 editType.setError("Max 16 characters");  
 return;  
 }  
 }  
 if (TextUtils.isEmpty(note)) {  
 editNote.setError("Required Field");  
 return;  
 }  
 else {  
 if (!validateNote(note)){  
 editNote.setError("Max 30 characters");  
 return;  
 }  
 }  
 int myAmount = Integer.parseInt(amountValue);  
  
 // Update transaction to database  
 if(post\_key != null){  
 Data data = new Data(myAmount, type, note, post\_key, mDate);  
 mOutcomeDatabase.child(post\_key).setValue(data);  
 Toast.makeText(getActivity(), "Transaction has been updated", Toast.LENGTH\_SHORT).show();  
  
 mdialog.dismiss();  
 }  
 }  
 });  
  
 btnDelete.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 mOutcomeDatabase.child(post\_key).removeValue();  
 mdialog.dismiss();  
 }  
 });  
 mdialog.show();  
 }  
  
 public boolean validateTitle(String title) {return title.length() <= 16;}  
  
 public boolean validateNote(String note) {return note.length() <= 16;}  
}

**ProgressBarAnimation.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import android.view.animation.Animation;  
import android.view.animation.Transformation;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
  
public class ProgressBarAnimation extends Animation {  
 private ProgressBar progressBar;  
 private TextView textView;  
 private float from;  
 private float to;  
  
 public ProgressBarAnimation(ProgressBar progressBar, TextView textView, float from, float to) {  
 super();  
 this.progressBar = progressBar;  
 this.textView = textView;  
 this.from = from;  
 this.to = to;  
 }  
  
 @Override  
 protected void applyTransformation(float interpolatedTime, Transformation t) {  
 super.applyTransformation(interpolatedTime, t);  
 float value = from + (to - from) \* interpolatedTime;  
 progressBar.setProgress((int) value);  
 textView.setText((int)value + "%");  
 }  
  
}

**SignIn.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import android.app.Dialog;  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthCredential;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.EmailAuthProvider;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
import java.util.Objects;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class SignIn extends Fragment {  
 private TextInputLayout mEmailSignInWrapper, mPassSignInWrapper;  
 private Button btnLogin;  
 private TextView mForgetPassword;  
  
 private static final String EMAIL\_PATTERN = "^[a-zA-Z0-9#\_~!$&'()\*+,;=:.\"(),:;<>@\\[\\]\\\\]+@[a-zA-Z0-9-]+(\\.[a-zA-Z0-9-]+)\*$";  
 private Pattern EmailPattern = Pattern.compile(EMAIL\_PATTERN);  
 private static final String PASSWORD\_PATTERN = "^(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[@#$%^&+=])(?=\\S+$).{4,}$";  
 private Pattern PassPattern = Pattern.compile(PASSWORD\_PATTERN);  
 private Matcher matcherEmail, matcherPass;  
  
 private ProgressDialog mDialog;  
  
 //Firebase Auth  
 private FirebaseAuth mAuth;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 //Set Auth instance  
 mAuth = FirebaseAuth.getInstance();  
 mDialog = new ProgressDialog(getActivity());  
  
 //Inflate view  
 View view = inflater.inflate(R.layout.fragment\_sign\_in, container, false);  
  
 mEmailSignInWrapper = (TextInputLayout) view.findViewById(R.id.editEmailSignInWrapper);  
 mPassSignInWrapper = (TextInputLayout) view.findViewById(R.id.editPasswordSignInWrapper);  
 btnLogin = (Button) view.findViewById(R.id.btnSignIn);  
 mForgetPassword = (TextView) view.findViewById(R.id.ForgetPassword);  
  
 mEmailSignInWrapper.setHint("E-mail");  
 mPassSignInWrapper.setHint("Password");  
  
  
 btnLogin.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String emailSignIn = mEmailSignInWrapper.getEditText().getText().toString().trim();  
 String passSignIn = mPassSignInWrapper.getEditText().getText().toString().trim();  
  
 if(!TextUtils.isEmpty(emailSignIn) && !TextUtils.isEmpty(passSignIn)){  
 if (!validateEmail(emailSignIn)) {  
 mEmailSignInWrapper.setError("Not a valid email address!");  
 }  
 //The password must be at least 8 characters long and include a number, lowercase letter, uppercase letter and special character  
 else if (!validatePassword(passSignIn)) {  
 mPassSignInWrapper.setError("Not a valid password!\n" +  
 "Must be 8 characters (number, lowercase, uppercase and special character)");  
 }  
 else {  
 mEmailSignInWrapper.setErrorEnabled(false);  
 mPassSignInWrapper.setErrorEnabled(false);  
  
 mDialog.setMessage("Processing...");  
 mDialog.show();  
 mAuth.signInWithEmailAndPassword(emailSignIn,passSignIn).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 mDialog.dismiss();  
 moveToHome();  
 Toast.makeText(getActivity().getApplicationContext(), "Sign In Succesfull", Toast.LENGTH\_SHORT).show();  
 }  
 else {  
 mDialog.dismiss();  
 Toast.makeText(getActivity().getApplicationContext(), "Incorect E-mail or Password", Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });  
 }  
 }  
 else {  
 if(TextUtils.isEmpty(emailSignIn)){  
 mEmailSignInWrapper.setError("Email Required...");  
 return;  
 }  
 if(TextUtils.isEmpty(passSignIn)){  
 mPassSignInWrapper.setError("Password Required...");  
 return;  
 }  
 }  
 }  
 });  
  
 mForgetPassword.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 forgotPassword();  
 }  
 });  
  
 return view;  
 }  
  
  
 public void forgotPassword() {  
 //Change Password Form  
 Dialog forgotPassDialog = new Dialog(getActivity());  
 LayoutInflater inflater = LayoutInflater.from(getActivity());  
 View myviewm = inflater.inflate(R.layout.forgot\_password, null);  
 forgotPassDialog.setContentView(myviewm);  
 forgotPassDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));  
 int width = (int)(getResources().getDisplayMetrics().widthPixels\*0.90);  
 int height = (int)(getResources().getDisplayMetrics().heightPixels\*0.41);  
  
 forgotPassDialog.getWindow().setLayout(width, height);  
  
 EditText editConfirmEmail = myviewm.findViewById(R.id.tvEmailConfirm);  
  
 Button btnSend = myviewm.findViewById(R.id.btnSend);  
  
 btnSend.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String ConfirmEmail = editConfirmEmail.getText().toString().trim();  
  
 if(TextUtils.isEmpty(ConfirmEmail)){  
 editConfirmEmail.setError("Required Field");  
 return;  
 }  
 else if (!validateEmail(ConfirmEmail)){  
 editConfirmEmail.setError("Not a valid email address!");  
 }  
  
 FirebaseAuth.getInstance().sendPasswordResetEmail(ConfirmEmail)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()) {  
 Toast.makeText(getActivity().getApplicationContext(), "Email Sent", Toast.LENGTH\_SHORT).show();  
 forgotPassDialog.dismiss();  
 Log.d("Status", "Email sent.");  
 }  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override public void onFailure(@NonNull Exception e) {  
 Toast.makeText(getActivity().getApplicationContext(), "Email not registered", Toast.LENGTH\_SHORT).show();  
 forgotPassDialog.dismiss();  
 Log.d("Status", "Failed to sent.");  
 }  
 });  
 }  
 });  
 forgotPassDialog.show();  
 }  
  
  
 public boolean validateEmail(String email) {  
 matcherEmail = EmailPattern.matcher(email);  
 return matcherEmail.matches();  
 }  
  
 public boolean validatePassword(String password) {  
 matcherPass = PassPattern.matcher(password);  
 return matcherPass.matches();  
 }  
 private void moveToHome() {  
 Intent intent = new Intent(getActivity(), HomeActivity.class);  
 intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK|Intent.FLAG\_ACTIVITY\_NEW\_TASK);  
 startActivity(intent);  
 }  
}

**SignUp.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputLayout;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
import id.ac.umn.leleair.kelompok.smallchange.Model.User;  
  
  
public class SignUp extends Fragment {  
 private TextInputLayout mEmailSignUpWrapper, mUsernameSignUpWrapper, mPassSignUpWrapper;  
 private Button btnReg;  
  
 private static final String *EMAIL\_PATTERN* = "^[a**-**zA**-**Z0**-**9#\_~!$&'()\*+,;=:.**\"**(),:;<>@\\[\\]\\\\]+@[a**-**zA**-**Z0**-**9-]+(\\.[a**-**zA**-**Z0**-**9-]+)\*$";  
 private Pattern EmailPattern = Pattern.*compile*(*EMAIL\_PATTERN*);  
 private static final String *PASSWORD\_PATTERN* = "^(?=.\*[0**-**9])(?=.\*[a**-**z])(?=.\*[A**-**Z])(?=.\*[@#$%^&+=])(?=\\S+$).{4,}$";  
 private Pattern PassPattern = Pattern.*compile*(*PASSWORD\_PATTERN*);  
 private Matcher matcherEmail, matcherPass;  
  
 private ProgressDialog mDialog;  
 private DatabaseReference mUsername;  
  
 //Firebase Authentification  
 private FirebaseAuth mAuth;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 mAuth = FirebaseAuth.*getInstance*();  
 mDialog = new ProgressDialog(getActivity());  
 //Inflate View  
 View view = inflater.inflate(R.layout.*fragment\_sign\_up*, container, false);  
  
 mEmailSignUpWrapper = (TextInputLayout) view.findViewById(R.id.*editEmailSignUpWrapper*);  
 mPassSignUpWrapper = (TextInputLayout) view.findViewById(R.id.*editPasswordSignUpWrapper*);  
 mUsernameSignUpWrapper = (TextInputLayout) view.findViewById(R.id.*editUsernameSignUpWrapper*);  
 btnReg = (Button) view.findViewById(R.id.*btnSignUp*);  
  
 mEmailSignUpWrapper.setHint("E-mail");  
 mPassSignUpWrapper.setHint("Password");  
 mUsernameSignUpWrapper.setHint("Username");  
  
 btnReg.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String emailSignUp = mEmailSignUpWrapper.getEditText().getText().toString().trim();  
 String usernameSignUp = mUsernameSignUpWrapper.getEditText().getText().toString().trim();  
 String passSignUp = mPassSignUpWrapper.getEditText().getText().toString().trim();  
  
 if(!TextUtils.*isEmpty*(emailSignUp) && !TextUtils.*isEmpty*(usernameSignUp) && !TextUtils.*isEmpty*(passSignUp)){  
 if (!validateEmail(emailSignUp)) {  
 mEmailSignUpWrapper.setError("Not a valid email address!");  
 }  
 else if (!validateUsername(usernameSignUp)) {  
 mUsernameSignUpWrapper.setError("Username too long!");  
 }  
 //The password must be at least 8 characters long and include a number, lowercase letter, uppercase letter and special character  
 else if (!validatePassword(passSignUp)) {  
 mPassSignUpWrapper.setError("Not a valid password!\n" +  
 "Must be 8 characters (number, lowercase, uppercase and special character)");  
 }  
 else {  
 mEmailSignUpWrapper.setErrorEnabled(false);  
  
 mUsernameSignUpWrapper.setErrorEnabled(false);  
  
 mDialog.setMessage("Processing...");  
 mDialog.show();  
 mAuth.createUserWithEmailAndPassword(emailSignUp,passSignUp).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 //get User id  
 FirebaseUser mUser = mAuth.getCurrentUser();  
 String uid = mUser.getUid();  
  
 //Create database child for username  
 mUsername = FirebaseDatabase.*getInstance*().getReference().child("Username").child(uid);  
 String id = mUsername.push().getKey();  
 if(id != null){  
 User user = new User(id, usernameSignUp);  
 mUsername.child(id).setValue(user);  
 }  
 mDialog.dismiss();  
 Toast.*makeText*(getActivity().getApplicationContext(), "Registration Complete", Toast.*LENGTH\_SHORT*).show();  
 moveToHome();  
 }  
 else {  
 mDialog.dismiss();  
 mPassSignUpWrapper.setErrorEnabled(true);  
 mPassSignUpWrapper.setError("Incorrect e-mail or password");  
 Toast.*makeText*(getActivity().getApplicationContext(), "Registration Failed", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 }  
 else {  
 if(TextUtils.*isEmpty*(emailSignUp)){  
 mEmailSignUpWrapper.setError("Email Required...");  
 return;  
 }  
 if(TextUtils.*isEmpty*(usernameSignUp)){  
 mUsernameSignUpWrapper.setError("Username Required...");  
 return;  
 }  
 if(TextUtils.*isEmpty*(passSignUp)){  
 mPassSignUpWrapper.setError("Password Required...");  
 }  
 }  
  
 }  
 });  
  
 return view;  
 }  
  
 public boolean validateEmail(String email) {  
 matcherEmail = EmailPattern.matcher(email);  
 return matcherEmail.matches();  
 }  
  
 public boolean validateUsername(String username) {  
 return username.length() <= 15;  
 }  
  
 public boolean validatePassword(String password) {  
 matcherPass = PassPattern.matcher(password);  
 return matcherPass.matches();  
 }  
  
 private void moveToHome() {  
 Intent intent = new Intent(getActivity(), HomeActivity.class);  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*|Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(intent);  
 }  
}

**SplashScreen.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.app.AppCompatDelegate;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
public class SplashScreen extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_splash\_screen);  
  
 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO);  
  
 final Handler handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 startActivity(new Intent(getApplicationContext(),Login.class));  
 finish();  
 }  
 }, 3000L);  
 }  
}

**UserPanel.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.app.AppCompatDelegate;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
public class SplashScreen extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_splash\_screen*);  
  
 AppCompatDelegate.*setDefaultNightMode*(AppCompatDelegate.*MODE\_NIGHT\_NO*);  
  
 final Handler handler = new Handler();  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 startActivity(new Intent(getApplicationContext(),Login.class));  
 finish();  
 }  
 }, 3000L);  
 }  
}

**ViewPagerAdapter.java**

package id.ac.umn.leleair.kelompok.smallchange;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentPagerAdapter;  
import androidx.lifecycle.Lifecycle;  
import androidx.viewpager2.adapter.FragmentStateAdapter;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class ViewPagerAdapter extends FragmentPagerAdapter {  
 private final List<Fragment> mFragmentList = new ArrayList<>();  
  
 public ViewPagerAdapter(FragmentManager manager) {  
 super(manager);  
 }  
  
 @Override  
 public Fragment getItem(int position) {  
 return mFragmentList.get(position);  
 }  
  
 @Override  
 public int getCount() {  
 return mFragmentList.size();  
 }  
  
 public void addFrag(Fragment fragment) {  
 mFragmentList.add(fragment);  
 }  
  
}

**tab\_color\_selector.xml**

<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:color="@color/merah" android:state\_selected="true"/>  
 <item android:color="@color/biru"/>  
</selector>

**Background\_icon.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid  
 android:color="@color/biru" />  
</shape>

**Bordered\_box.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid  
 android:color="#FFFFFF" />  
  
 <corners  
 android:topLeftRadius="40sp"  
 android:topRightRadius="40sp" />  
  
  
</shape>

**Bordered\_box2.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid  
 android:color="#f2f3fd" />  
  
 <corners  
 android:topLeftRadius="40sp"  
 android:topRightRadius="40sp" />  
  
  
</shape>

**Bordered\_box3.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid  
 android:color="#f2f3fd" />  
  
 <corners  
 android:topRightRadius="40sp" />  
  
  
</shape>

**Bordered\_box4.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid  
 android:color="#f2f3fd" />  
  
 <corners  
 android:topLeftRadius="40sp" />  
  
  
</shape>

**Bordered\_box\_transparent.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
</shape>

**Bordered\_edit\_text.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <stroke  
 android:width="3sp"  
 android:color="#e0bedd"/>  
  
 <corners  
 android:topLeftRadius="16sp"  
 android:topRightRadius="16sp"  
 android:bottomRightRadius="16sp"  
 android:bottomLeftRadius="16sp"/>  
  
  
</shape>

**Circle.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="oval">  
  
 <solid  
 android:color="@color/biru" />  
  
 <size  
 android:width="2000dp"  
 android:height="2000dp"/>  
  
  
</shape>

**Circle\_progress\_income.xml**

<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">  
 <item>  
 <shape  
 android:shape="ring"  
 android:thicknessRatio="6"  
 android:useLevel="false">  
 <solid android:color="@color/transparentLightBlue"/>  
 </shape>  
 </item>  
 <item>  
 <rotate  
 android:fromDegrees="270"  
 android:toDegrees="270">  
 <shape  
 android:shape="ring"  
 android:thicknessRatio="6"  
 android:useLevel="true">  
 <solid android:color="@color/hijau"/>  
 </shape>  
 </rotate>  
 </item>  
</layer-list>

**Circle\_progress\_outcome.xml**

<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">  
 <item>  
 <shape  
 android:shape="ring"  
 android:thicknessRatio="6"  
 android:useLevel="false">  
 <solid android:color="@color/transparentLightBlue"/>  
 </shape>  
 </item>  
 <item>  
 <rotate  
 android:fromDegrees="270"  
 android:toDegrees="270">  
 <shape  
 android:shape="ring"  
 android:thicknessRatio="6"  
 android:useLevel="true">  
 <solid android:color="@color/merah"/>  
 </shape>  
 </rotate>  
 </item>  
</layer-list>

**Circle\_white.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="oval">  
  
 <solid  
 android:color="@color/white" />  
  
 <size  
 android:width="2000dp"  
 android:height="2000dp"/>  
  
  
</shape>

**Ic\_baseline\_add\_24.xml**

<vector android:autoMirrored="true" android:height="24dp"  
 android:tint="#FFFFFF" android:viewportHeight="24"  
 android:viewportWidth="24" android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M19,13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/>  
</vector>

**Ic\_baseline\_arrow\_drop\_down\_24.xml**

<vector android:autoMirrored="true" android:height="24dp"  
 android:tint="#FFFFFF" android:viewportHeight="24"  
 android:viewportWidth="24" android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M7,10l5,5 5,-5z"/>  
</vector>

**Ic\_changephotobtn\_18.xml**

<vector android:autoMirrored="true" android:height="20dp"  
 android:viewportHeight="87.95" android:viewportWidth="87.95"  
 android:width="20dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="#fff" android:pathData="M43.97,43.97m-43.97,0a43.97,43.97 0,1 1,87.94 0a43.97,43.97 0,1 1,-87.94 0"/>  
 <path android:fillColor="#284098" android:pathData="M43.97,43.97m-33.18,0a33.18,33.18 0,1 1,66.36 0a33.18,33.18 0,1 1,-66.36 0"/>  
 <path android:fillColor="#fff" android:pathData="M49.8,46.41A5.83,5.83 0,1 1,44 40.58a5.83,5.83 0,0 1,5.83 5.83ZM62,38.13L62,54.69a4,4 0,0 1,-4 4L29.9,58.69a4,4 0,0 1,-4 -4L25.9,38.13a4,4 0,0 1,4 -4h4.92L34.82,32.76a3.5,3.5 0,0 1,3.49 -3.5L49.64,29.26a3.5,3.5 0,0 1,3.49 3.5v1.38h4.92a4,4 0,0 1,4 4ZM52.8,46.41A8.83,8.83 0,1 0,44 55.24a8.83,8.83 0,0 0,8.83 -8.83ZM52.8,46.41"/>  
</vector>

**Ic\_dashboardicon\_13.xml**

<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 android:width="54.64dp"  
 android:height="46.38dp"  
 android:viewportWidth="54.64"  
 android:viewportHeight="46.38">  
 <path  
 android:pathData="M4.41,0L50.23,0A4.41,4.41 0,0 1,54.64 4.41L54.64,4.41A4.41,4.41 0,0 1,50.23 8.82L4.41,8.82A4.41,4.41 0,0 1,0 4.41L0,4.41A4.41,4.41 0,0 1,4.41 0z"  
 android:fillColor="#284098"/>  
 <path  
 android:pathData="M4.41,18.17L50.23,18.17A4.41,4.41 0,0 1,54.64 22.58L54.64,22.58A4.41,4.41 0,0 1,50.23 26.99L4.41,26.99A4.41,4.41 0,0 1,0 22.58L0,22.58A4.41,4.41 0,0 1,4.41 18.17z"  
 android:fillColor="#284098"/>  
 <path  
 android:pathData="M4.41,37.56L50.23,37.56A4.41,4.41 0,0 1,54.64 41.97L54.64,41.97A4.41,4.41 0,0 1,50.23 46.38L4.41,46.38A4.41,4.41 0,0 1,0 41.97L0,41.97A4.41,4.41 0,0 1,4.41 37.56z"  
 android:fillColor="#284098"/>  
</vector>

**Ic\_default\_img\_white\_01.xml**

<vector android:autoMirrored="true" android:height="441.5dp"  
 android:viewportHeight="441.5" android:viewportWidth="441.5"  
 android:width="441.5dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillAlpha="0" android:fillColor="#fff"  
 android:pathData="M0,0h441.5v441.5h-441.5z" android:strokeAlpha="0"/>  
 <path android:fillColor="#fff" android:pathData="M220.75,221.58A56.25,56.25 0,1 1,277 165.33,56.31 56.31,0 0,1 220.75,221.58Z"/>  
 <path android:fillColor="#fff" android:pathData="M308.25,322.42h-175a6.25,6.25 0,0 1,-6.25 -6.25,82 82,0 0,1 23.76,-57.84 80.19,80.19 0,0 1,57.49 -24.25h25a80.19,80.19 0,0 1,57.49 24.25,82 82,0 0,1 23.76,57.84A6.25,6.25 0,0 1,308.25 322.42Z"/>  
</vector>

**Ic\_default\_user\_photo\_01.xml**

<vector android:autoMirrored="true" android:height="441.5dp"  
 android:viewportHeight="441.5" android:viewportWidth="441.5"  
 android:width="441.5dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillAlpha="0" android:fillColor="@color/biru"  
 android:pathData="M0,0h441.5v441.5h-441.5z" android:strokeAlpha="0"/>  
 <path android:fillColor="@color/biru" android:pathData="M220.75,221.58A56.25,56.25 0,1 1,277 165.33,56.31 56.31,0 0,1 220.75,221.58Z"/>  
 <path android:fillColor="@color/biru" android:pathData="M308.25,322.42h-175a6.25,6.25 0,0 1,-6.25 -6.25,82 82,0 0,1 23.76,-57.84 80.19,80.19 0,0 1,57.49 -24.25h25a80.19,80.19 0,0 1,57.49 24.25,82 82,0 0,1 23.76,57.84A6.25,6.25 0,0 1,308.25 322.42Z"/>  
</vector>

**Activity\_home.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".HomeActivity">  
  
 <androidx.viewpager.widget.ViewPager  
 android:id="@+id/viewpagerHome"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"/>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp"  
 android:paddingBottom="5dp">  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/cardView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="5dp"  
 app:cardPreventCornerOverlap="true"  
 app:cardUseCompatPadding="true"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent">  
  
 <com.google.android.material.tabs.TabLayout  
 android:id="@+id/homeTabLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="64sp"  
 android:background="@drawable/bordered\_box"  
 app:tabIndicatorFullWidth="false"  
 app:tabIndicatorHeight="4sp"  
 app:tabIconTint="@color/tab\_color\_selector"/>  
 </androidx.cardview.widget.CardView>  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Activity\_login.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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/biru"  
 tools:context=".Login">  
  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/constraintLayout1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50sp"  
 android:foregroundGravity="center\_horizontal"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="63sp"  
 android:layout\_height="96sp"  
 android:background="@drawable/ic\_logosplash\_11"  
 android:scaleType="center"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8sp"  
 android:fontFamily="@font/bold"  
 android:text="mall\nChange"  
 android:textColor="@color/white"  
 android:textSize="32sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/constraintLayout2"  
 android:layout\_marginTop="50sp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="700sp"  
 android:background="@drawable/bordered\_box"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout1">  
  
 <androidx.viewpager2.widget.ViewPager2  
 android:id="@+id/viewpagerLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/bordered\_box" />  
  
 <com.google.android.material.tabs.TabLayout  
 android:id="@+id/loginTabLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="60sp"  
 android:background="@drawable/bordered\_box2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/viewpagerLogin"  
 app:tabIndicatorFullWidth="false"  
 app:tabIndicatorHeight="4sp"  
 app:tabTextAppearance="@style/tab\_text"  
 app:tabTextColor="@color/biru" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Activity\_splash\_screen.xml**

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <attr format="reference" name="animate\_relativeTo"/>  
 <attr format="enum" name="arcMode">  
 <enum name="startVertical" value="0"/>  
 <enum name="startHorizontal" value="1"/>  
 <enum name="flip" value="2"/>  
 </attr>  
 <attr format="boolean" name="barrierAllowsGoneWidgets"/>  
 <attr format="enum" name="barrierDirection">  
 <enum name="left" value="0"/>  
 <enum name="right" value="1"/>  
 <enum name="top" value="2"/>  
 <enum name="bottom" value="3"/>  
 <enum name="start" value="5"/>  
 <enum name="end" value="6"/>  
 </attr>  
 <attr format="dimension" name="barrierMargin"/>  
 <attr format="boolean" name="chainUseRtl"/>  
 <attr format="dimension" name="circleRadius"/>  
 <attr format="reference" name="constraintSet"/>  
 <attr format="string" name="constraint\_referenced\_ids"/>  
 <attr format="string" name="constraint\_referenced\_tags"/>  
 <attr format="reference" name="content"/>  
 <attr format="enum" name="curveFit">  
 <enum name="spline" value="0"/>  
 <enum name="linear" value="1"/>  
 </attr>  
 <attr format="integer" name="defaultDuration"/>  
 <attr format="float" name="deltaPolarAngle"/>  
 <attr format="float" name="deltaPolarRadius"/>  
 <attr format="enum" name="dragDirection">  
 <enum name="dragUp" value="0"/>  
 <enum name="dragDown" value="1"/>  
 <enum name="dragLeft" value="2"/>  
 <enum name="dragRight" value="3"/>  
 <enum name="dragStart" value="4"/>  
 <enum name="dragEnd" value="5"/>  
 </attr>  
 <attr format="enum" name="drawPath">  
 <enum name="none" value="0"/>  
 <enum name="path" value="1"/>  
 <enum name="pathRelative" value="2"/>  
 <enum name="deltaRelative" value="3"/>  
 <enum name="asConfigured" value="4"/>  
 <enum name="rectangles" value="5"/>  
 </attr>  
 <attr format="integer" name="duration"/>  
 <attr format="float" name="flow\_firstHorizontalBias"/>  
 <attr format="enum" name="flow\_firstHorizontalStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="float" name="flow\_firstVerticalBias"/>  
 <attr format="enum" name="flow\_firstVerticalStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="enum" name="flow\_horizontalAlign">  
 <enum name="start" value="0"/>  
 <enum name="end" value="1"/>  
 <enum name="center" value="2"/>  
 </attr>  
 <attr format="float" name="flow\_horizontalBias"/>  
 <attr format="dimension" name="flow\_horizontalGap"/>  
 <attr format="enum" name="flow\_horizontalStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="float" name="flow\_lastHorizontalBias"/>  
 <attr format="enum" name="flow\_lastHorizontalStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="float" name="flow\_lastVerticalBias"/>  
 <attr format="enum" name="flow\_lastVerticalStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="integer" name="flow\_maxElementsWrap"/>  
 <attr format="dimension" name="flow\_padding"/>  
 <attr format="enum" name="flow\_verticalAlign">  
 <enum name="top" value="0"/>  
 <enum name="bottom" value="1"/>  
 <enum name="center" value="2"/>  
 <enum name="baseline" value="3"/>  
 </attr>  
 <attr format="float" name="flow\_verticalBias"/>  
 <attr format="dimension" name="flow\_verticalGap"/>  
 <attr format="enum" name="flow\_verticalStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="enum" name="flow\_wrapMode">  
 <enum name="none" value="0"/>  
 <enum name="chain" value="1"/>  
 <enum name="aligned" value="2"/>  
 </attr>  
 <attr format="integer" name="framePosition"/>  
 <attr format="reference" name="layoutDescription"/>  
 <attr format="enum" name="layoutDuringTransition">  
 <enum name="ignoreRequest" value="0"/>  
 <enum name="honorRequest" value="1"/>  
 </attr>  
 <attr format="boolean" name="layout\_constrainedHeight"/>  
 <attr format="boolean" name="layout\_constrainedWidth"/>  
 <attr format="integer" name="layout\_constraintBaseline\_creator"/>  
 <attr format="reference|enum" name="layout\_constraintBaseline\_toBaselineOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="integer" name="layout\_constraintBottom\_creator"/>  
 <attr format="reference|enum" name="layout\_constraintBottom\_toBottomOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference|enum" name="layout\_constraintBottom\_toTopOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference" name="layout\_constraintCircle"/>  
 <attr format="integer" name="layout\_constraintCircleAngle"/>  
 <attr format="dimension" name="layout\_constraintCircleRadius"/>  
 <attr format="string" name="layout\_constraintDimensionRatio"/>  
 <attr format="reference|enum" name="layout\_constraintEnd\_toEndOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference|enum" name="layout\_constraintEnd\_toStartOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="dimension" name="layout\_constraintGuide\_begin"/>  
 <attr format="dimension" name="layout\_constraintGuide\_end"/>  
 <attr format="float" name="layout\_constraintGuide\_percent"/>  
 <attr name="layout\_constraintHeight\_default">  
 <enum name="spread" value="0"/>  
 <enum name="wrap" value="1"/>  
 <enum name="percent" value="2"/>  
 </attr>  
 <attr format="dimension|enum" name="layout\_constraintHeight\_max">  
 <enum name="wrap" value="-2"/>  
 </attr>  
 <attr format="dimension|enum" name="layout\_constraintHeight\_min">  
 <enum name="wrap" value="-2"/>  
 </attr>  
 <attr format="float" name="layout\_constraintHeight\_percent"/>  
 <attr format="float" name="layout\_constraintHorizontal\_bias"/>  
 <attr format="enum" name="layout\_constraintHorizontal\_chainStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="float" name="layout\_constraintHorizontal\_weight"/>  
 <attr format="integer" name="layout\_constraintLeft\_creator"/>  
 <attr format="reference|enum" name="layout\_constraintLeft\_toLeftOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference|enum" name="layout\_constraintLeft\_toRightOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="integer" name="layout\_constraintRight\_creator"/>  
 <attr format="reference|enum" name="layout\_constraintRight\_toLeftOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference|enum" name="layout\_constraintRight\_toRightOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference|enum" name="layout\_constraintStart\_toEndOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference|enum" name="layout\_constraintStart\_toStartOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="string" name="layout\_constraintTag"/>  
 <attr format="integer" name="layout\_constraintTop\_creator"/>  
 <attr format="reference|enum" name="layout\_constraintTop\_toBottomOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="reference|enum" name="layout\_constraintTop\_toTopOf">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr format="float" name="layout\_constraintVertical\_bias"/>  
 <attr format="enum" name="layout\_constraintVertical\_chainStyle">  
 <enum name="spread" value="0"/>  
 <enum name="spread\_inside" value="1"/>  
 <enum name="packed" value="2"/>  
 </attr>  
 <attr format="float" name="layout\_constraintVertical\_weight"/>  
 <attr name="layout\_constraintWidth\_default">  
 <enum name="spread" value="0"/>  
 <enum name="wrap" value="1"/>  
 <enum name="percent" value="2"/>  
 </attr>  
 <attr format="dimension|enum" name="layout\_constraintWidth\_max">  
 <enum name="wrap" value="-2"/>  
 </attr>  
 <attr format="dimension|enum" name="layout\_constraintWidth\_min">  
 <enum name="wrap" value="-2"/>  
 </attr>  
 <attr format="float" name="layout\_constraintWidth\_percent"/>  
 <attr format="dimension" name="layout\_editor\_absoluteX"/>  
 <attr format="dimension" name="layout\_editor\_absoluteY"/>  
 <attr format="dimension" name="layout\_goneMarginBottom"/>  
 <attr format="dimension" name="layout\_goneMarginEnd"/>  
 <attr format="dimension" name="layout\_goneMarginLeft"/>  
 <attr format="dimension" name="layout\_goneMarginRight"/>  
 <attr format="dimension" name="layout\_goneMarginStart"/>  
 <attr format="dimension" name="layout\_goneMarginTop"/>  
 <attr name="layout\_optimizationLevel">  
 <flag name="none" value="0"/>  
 <flag name="standard" value="257"/> <!-- direct, measure caches -->  
 <flag name="direct" value="1"/>  
 <flag name="barrier" value="2"/>  
 <flag name="chains" value="4"/>  
 <flag name="dimensions" value="8"/>  
 <flag name="ratio" value="16"/>  
 <flag name="groups" value="32"/>  
 <flag name="graph" value="64"/>  
 <flag name="graph\_wrap" value="128"/>  
 <flag name="cache\_measures" value="256"/>  
 <flag name="dependency\_ordering" value="512"/>  
 <flag name="grouping" value="1024"/>  
 </attr>  
 <attr format="float" name="motionPathRotate"/>  
 <attr format="float" name="motionProgress"/>  
 <attr format="float" name="motionStagger"/>  
 <attr format="reference|string" name="motionTarget"/>  
 <attr format="enum" name="pathMotionArc">  
 <enum name="none" value="0"/>  
 <enum name="startVertical" value="1"/>  
 <enum name="startHorizontal" value="2"/>  
 <enum name="flip" value="3"/>  
 </attr>  
 <attr format="float" name="path\_percent"/>  
 <attr format="float" name="perpendicularPath\_percent"/>  
 <attr format="reference|enum" name="pivotAnchor">  
 <enum name="parent" value="0"/>  
 </attr>  
 <attr name="placeholder\_emptyVisibility">  
 <enum name="visible" value="0"/>  
 <enum name="invisible" value="4"/>  
 <enum name="gone" value="8"/>  
 </attr>  
 <attr format="float" name="sizePercent"/>  
 <attr format="reference" name="targetId"/>  
 <attr format="reference" name="touchAnchorId"/>  
 <attr format="enum" name="touchAnchorSide">  
 <enum name="top" value="0"/>  
 <enum name="left" value="1"/>  
 <enum name="right" value="2"/>  
 <enum name="bottom" value="3"/>  
 <enum name="middle" value="4"/>  
 <enum name="start" value="5"/>  
 <enum name="end" value="6"/>  
 </attr>  
 <attr format="string|enum" name="transitionEasing">  
 <enum name="standard" value="0"/>  
 <enum name="accelerate" value="1"/>  
 <enum name="decelerate" value="2"/>  
 <enum name="linear" value="3"/>  
 </attr>  
 <attr format="float" name="transitionPathRotate"/>  
 <attr format="enum" name="visibilityMode">  
 <enum name="normal" value="0"/>  
 <enum name="ignore" value="1"/>  
 </attr>  
 <attr format="integer" name="waveDecay"/>  
 <attr format="float|dimension" name="waveOffset"/>  
 <attr format="float" name="wavePeriod"/>  
 <attr format="enum" name="waveShape">  
 <enum name="sin" value="0"/>  
 <enum name="square" value="1"/>  
 <enum name="triangle" value="2"/>  
 <enum name="sawtooth" value="3"/>  
 <enum name="reverseSawtooth" value="4"/>  
 <enum name="cos" value="5"/>  
 <enum name="bounce" value="6"/>  
 </attr>  
 <attr format="enum" name="waveVariesBy">  
 <enum name="position" value="0"/>  
 <enum name="path" value="1"/>  
 </attr>  
 <declare-styleable name="Constraint"><attr name="android:orientation"/><attr name="android:id"/><attr name="android:visibility"/><attr name="visibilityMode"/><attr name="android:alpha"/><attr name="android:elevation"/><attr name="android:rotation"/><attr name="android:rotationX"/><attr name="android:rotationY"/><attr name="android:scaleX"/><attr name="android:scaleY"/><attr name="pivotAnchor"/><attr name="android:transformPivotX"/><attr name="android:transformPivotY"/><attr name="android:translationX"/><attr name="android:translationY"/><attr name="android:translationZ"/><attr name="android:layout\_width"/><attr name="android:layout\_height"/><attr name="android:layout\_marginStart"/><attr name="android:layout\_marginBottom"/><attr name="android:layout\_marginTop"/><attr name="android:layout\_marginEnd"/><attr name="android:layout\_marginLeft"/><attr name="android:layout\_marginRight"/><attr name="layout\_constraintCircle"/><attr name="layout\_constraintCircleRadius"/><attr name="layout\_constraintCircleAngle"/><attr name="layout\_constraintGuide\_begin"/><attr name="layout\_constraintGuide\_end"/><attr name="layout\_constraintGuide\_percent"/><attr name="layout\_constraintLeft\_toLeftOf"/><attr name="layout\_constraintLeft\_toRightOf"/><attr name="layout\_constraintRight\_toLeftOf"/><attr name="layout\_constraintRight\_toRightOf"/><attr name="layout\_constraintTop\_toTopOf"/><attr name="layout\_constraintTop\_toBottomOf"/><attr name="layout\_constraintBottom\_toTopOf"/><attr name="layout\_constraintBottom\_toBottomOf"/><attr name="layout\_constraintBaseline\_toBaselineOf"/><attr name="layout\_constraintStart\_toEndOf"/><attr name="layout\_constraintStart\_toStartOf"/><attr name="layout\_constraintEnd\_toStartOf"/><attr name="layout\_constraintEnd\_toEndOf"/><attr name="layout\_goneMarginLeft"/><attr name="layout\_goneMarginTop"/><attr name="layout\_goneMarginRight"/><attr name="layout\_goneMarginBottom"/><attr name="layout\_goneMarginStart"/><attr name="layout\_goneMarginEnd"/><attr name="layout\_constrainedWidth"/><attr name="layout\_constrainedHeight"/><attr name="layout\_constraintHorizontal\_bias"/><attr name="layout\_constraintVertical\_bias"/><attr name="layout\_constraintWidth\_default"/><attr name="layout\_constraintHeight\_default"/><attr name="layout\_constraintWidth\_min"/><attr name="layout\_constraintWidth\_max"/><attr name="layout\_constraintWidth\_percent"/><attr name="layout\_constraintHeight\_min"/><attr name="layout\_constraintHeight\_max"/><attr name="layout\_constraintHeight\_percent"/><attr name="layout\_constraintLeft\_creator"/><attr name="layout\_constraintTop\_creator"/><attr name="layout\_constraintRight\_creator"/><attr name="layout\_constraintBottom\_creator"/><attr name="layout\_constraintBaseline\_creator"/><attr name="layout\_constraintDimensionRatio"/><attr name="layout\_constraintHorizontal\_weight"/><attr name="layout\_constraintVertical\_weight"/><attr name="layout\_constraintHorizontal\_chainStyle"/><attr name="layout\_constraintVertical\_chainStyle"/><attr name="layout\_editor\_absoluteX"/><attr name="layout\_editor\_absoluteY"/><attr name="barrierDirection"/><attr name="barrierMargin"/><attr name="constraint\_referenced\_ids"/><attr name="constraint\_referenced\_tags"/><attr name="android:maxHeight"/><attr name="android:maxWidth"/><attr name="android:minHeight"/><attr name="android:minWidth"/><attr name="barrierAllowsGoneWidgets"/><attr name="chainUseRtl"/><attr name="flow\_horizontalStyle"/><attr name="flow\_verticalStyle"/><attr name="flow\_verticalAlign"/><attr name="flow\_horizontalAlign"/><attr name="flow\_verticalBias"/><attr name="flow\_horizontalBias"/><attr name="flow\_wrapMode"/><attr name="flow\_maxElementsWrap"/><attr name="flow\_horizontalGap"/><attr name="flow\_verticalGap"/><attr name="flow\_firstHorizontalStyle"/><attr name="flow\_firstVerticalStyle"/><attr name="flow\_firstHorizontalBias"/><attr name="flow\_firstVerticalBias"/><attr name="flow\_lastHorizontalStyle"/><attr name="flow\_lastVerticalStyle"/><attr name="flow\_lastHorizontalBias"/><attr name="flow\_lastVerticalBias"/><attr name="animate\_relativeTo"/><attr name="transitionEasing"/><attr name="pathMotionArc"/><attr name="transitionPathRotate"/><attr name="drawPath"/><attr name="motionProgress"/><attr name="layout\_constraintTag"/><attr name="motionStagger"/></declare-styleable>  
 <declare-styleable name="ConstraintLayout\_Layout"><attr name="android:orientation"/><attr name="android:minWidth"/><attr name="android:minHeight"/><attr name="android:maxWidth"/><attr name="android:maxHeight"/><attr name="android:visibility"/><attr name="android:elevation"/><attr name="layout\_optimizationLevel"/><attr name="layoutDescription"/><attr name="constraintSet"/><attr name="barrierDirection"/><attr name="barrierAllowsGoneWidgets"/><attr name="barrierMargin"/><attr name="constraint\_referenced\_ids"/><attr name="constraint\_referenced\_tags"/><attr name="chainUseRtl"/><attr name="flow\_horizontalStyle"/><attr name="flow\_verticalStyle"/><attr name="flow\_wrapMode"/><attr name="flow\_maxElementsWrap"/><attr name="flow\_horizontalGap"/><attr name="flow\_verticalGap"/><attr name="android:padding"/><attr name="android:paddingTop"/><attr name="android:paddingBottom"/><attr name="android:paddingLeft"/><attr name="android:paddingRight"/><attr name="android:paddingStart"/><attr name="android:paddingEnd"/><attr name="flow\_verticalAlign"/><attr name="flow\_horizontalAlign"/><attr name="flow\_verticalBias"/><attr name="flow\_horizontalBias"/><attr name="flow\_firstHorizontalStyle"/><attr name="flow\_firstVerticalStyle"/><attr name="flow\_firstHorizontalBias"/><attr name="flow\_firstVerticalBias"/><attr name="flow\_lastHorizontalStyle"/><attr name="flow\_lastVerticalStyle"/><attr name="flow\_lastHorizontalBias"/><attr name="flow\_lastVerticalBias"/><attr name="layout\_constraintCircle"/><attr name="layout\_constraintCircleRadius"/><attr name="layout\_constraintCircleAngle"/><attr name="layout\_constraintGuide\_begin"/><attr name="layout\_constraintGuide\_end"/><attr name="layout\_constraintGuide\_percent"/><attr name="layout\_constraintLeft\_toLeftOf"/><attr name="layout\_constraintLeft\_toRightOf"/><attr name="layout\_constraintRight\_toLeftOf"/><attr name="layout\_constraintRight\_toRightOf"/><attr name="layout\_constraintTop\_toTopOf"/><attr name="layout\_constraintTop\_toBottomOf"/><attr name="layout\_constraintBottom\_toTopOf"/><attr name="layout\_constraintBottom\_toBottomOf"/><attr name="layout\_constraintBaseline\_toBaselineOf"/><attr name="layout\_constraintStart\_toEndOf"/><attr name="layout\_constraintStart\_toStartOf"/><attr name="layout\_constraintEnd\_toStartOf"/><attr name="layout\_constraintEnd\_toEndOf"/><attr name="layout\_goneMarginLeft"/><attr name="layout\_goneMarginTop"/><attr name="layout\_goneMarginRight"/><attr name="layout\_goneMarginBottom"/><attr name="layout\_goneMarginStart"/><attr name="layout\_goneMarginEnd"/><attr name="layout\_constrainedWidth"/><attr name="layout\_constrainedHeight"/><attr name="layout\_constraintHorizontal\_bias"/><attr name="layout\_constraintVertical\_bias"/><attr name="layout\_constraintWidth\_default"/><attr name="layout\_constraintHeight\_default"/><attr name="layout\_constraintWidth\_min"/><attr name="layout\_constraintWidth\_max"/><attr name="layout\_constraintWidth\_percent"/><attr name="layout\_constraintHeight\_min"/><attr name="layout\_constraintHeight\_max"/><attr name="layout\_constraintHeight\_percent"/><attr name="layout\_constraintLeft\_creator"/><attr name="layout\_constraintTop\_creator"/><attr name="layout\_constraintRight\_creator"/><attr name="layout\_constraintBottom\_creator"/><attr name="layout\_constraintBaseline\_creator"/><attr name="layout\_constraintDimensionRatio"/><attr name="layout\_constraintHorizontal\_weight"/><attr name="layout\_constraintVertical\_weight"/><attr name="layout\_constraintHorizontal\_chainStyle"/><attr name="layout\_constraintVertical\_chainStyle"/><attr name="layout\_editor\_absoluteX"/><attr name="layout\_editor\_absoluteY"/><attr name="layout\_constraintTag"/></declare-styleable>  
 <declare-styleable name="ConstraintLayout\_placeholder"><attr name="placeholder\_emptyVisibility"/><attr name="content"/></declare-styleable>  
 <declare-styleable name="ConstraintSet"><attr format="reference" name="deriveConstraintsFrom"/><attr name="android:orientation"/><attr name="android:id"/><attr name="android:visibility"/><attr name="android:alpha"/><attr name="android:elevation"/><attr name="android:rotation"/><attr name="android:rotationX"/><attr name="android:rotationY"/><attr name="android:scaleX"/><attr name="android:scaleY"/><attr name="pivotAnchor"/><attr name="android:pivotX"/><attr name="android:pivotY"/><attr name="android:transformPivotX"/><attr name="android:transformPivotY"/><attr name="android:translationX"/><attr name="android:translationY"/><attr name="android:translationZ"/><attr name="android:layout\_width"/><attr name="android:layout\_height"/><attr name="android:layout\_marginStart"/><attr name="android:layout\_marginBottom"/><attr name="android:layout\_marginTop"/><attr name="android:layout\_marginEnd"/><attr name="android:layout\_marginLeft"/><attr name="android:layout\_marginRight"/><attr name="layout\_constraintCircle"/><attr name="layout\_constraintCircleRadius"/><attr name="layout\_constraintCircleAngle"/><attr name="layout\_constraintGuide\_begin"/><attr name="layout\_constraintGuide\_end"/><attr name="layout\_constraintGuide\_percent"/><attr name="layout\_constraintLeft\_toLeftOf"/><attr name="layout\_constraintLeft\_toRightOf"/><attr name="layout\_constraintRight\_toLeftOf"/><attr name="layout\_constraintRight\_toRightOf"/><attr name="layout\_constraintTop\_toTopOf"/><attr name="layout\_constraintTop\_toBottomOf"/><attr name="layout\_constraintBottom\_toTopOf"/><attr name="layout\_constraintBottom\_toBottomOf"/><attr name="layout\_constraintBaseline\_toBaselineOf"/><attr name="layout\_constraintStart\_toEndOf"/><attr name="layout\_constraintStart\_toStartOf"/><attr name="layout\_constraintEnd\_toStartOf"/><attr name="layout\_constraintEnd\_toEndOf"/><attr name="layout\_goneMarginLeft"/><attr name="layout\_goneMarginTop"/><attr name="layout\_goneMarginRight"/><attr name="layout\_goneMarginBottom"/><attr name="layout\_goneMarginStart"/><attr name="layout\_goneMarginEnd"/><attr name="layout\_constrainedWidth"/><attr name="layout\_constrainedHeight"/><attr name="layout\_constraintHorizontal\_bias"/><attr name="layout\_constraintVertical\_bias"/><attr name="layout\_constraintWidth\_default"/><attr name="layout\_constraintHeight\_default"/><attr name="layout\_constraintWidth\_min"/><attr name="layout\_constraintWidth\_max"/><attr name="layout\_constraintWidth\_percent"/><attr name="layout\_constraintHeight\_min"/><attr name="layout\_constraintHeight\_max"/><attr name="layout\_constraintHeight\_percent"/><attr name="layout\_constraintLeft\_creator"/><attr name="layout\_constraintTop\_creator"/><attr name="layout\_constraintRight\_creator"/><attr name="layout\_constraintBottom\_creator"/><attr name="layout\_constraintBaseline\_creator"/><attr name="layout\_constraintDimensionRatio"/><attr name="layout\_constraintHorizontal\_weight"/><attr name="layout\_constraintVertical\_weight"/><attr name="layout\_constraintHorizontal\_chainStyle"/><attr name="layout\_constraintVertical\_chainStyle"/><attr name="layout\_editor\_absoluteX"/><attr name="layout\_editor\_absoluteY"/><attr name="barrierDirection"/><attr name="constraint\_referenced\_ids"/><attr name="constraint\_referenced\_tags"/><attr name="barrierMargin"/><attr name="android:maxHeight"/><attr name="android:maxWidth"/><attr name="android:minHeight"/><attr name="android:minWidth"/><attr name="barrierAllowsGoneWidgets"/><attr name="chainUseRtl"/><attr name="flow\_horizontalStyle"/><attr name="flow\_verticalStyle"/><attr name="flow\_verticalAlign"/><attr name="flow\_horizontalAlign"/><attr name="flow\_verticalBias"/><attr name="flow\_horizontalBias"/><attr name="flow\_wrapMode"/><attr name="flow\_maxElementsWrap"/><attr name="flow\_horizontalGap"/><attr name="flow\_verticalGap"/><attr name="flow\_firstHorizontalStyle"/><attr name="flow\_firstVerticalStyle"/><attr name="flow\_firstHorizontalBias"/><attr name="flow\_firstVerticalBias"/><attr name="flow\_lastHorizontalStyle"/><attr name="flow\_lastVerticalStyle"/><attr name="flow\_lastHorizontalBias"/><attr name="flow\_lastVerticalBias"/><attr name="animate\_relativeTo"/><attr name="transitionEasing"/><attr name="pathMotionArc"/><attr name="transitionPathRotate"/><attr name="motionStagger"/><attr name="drawPath"/><attr name="motionProgress"/><attr name="layout\_constraintTag"/></declare-styleable>  
 <declare-styleable name="CustomAttribute"><attr format="string" name="attributeName"/><attr format="color" name="customColorValue"/><attr format="color" name="customColorDrawableValue"/><attr format="integer" name="customIntegerValue"/><attr format="float" name="customFloatValue"/><attr format="string" name="customStringValue"/><attr format="dimension" name="customDimension"/><attr format="dimension" name="customPixelDimension"/><attr format="boolean" name="customBoolean"/></declare-styleable>  
 <declare-styleable name="ImageFilterView"><attr format="reference" name="altSrc"/><attr format="float" name="saturation"/><attr format="float" name="brightness"/><attr format="float" name="warmth"/><attr format="float" name="contrast"/><attr format="float" name="crossfade"/><attr format="dimension" name="round"/><attr format="boolean" name="overlay"/><attr format="float" name="roundPercent"/></declare-styleable>  
 <declare-styleable name="KeyAttribute"><attr name="framePosition"/><attr name="motionTarget"/><attr name="transitionEasing"/><attr name="curveFit"/><attr name="motionProgress"/><attr name="android:alpha"/><attr name="android:elevation"/><attr name="android:rotation"/><attr name="android:rotationX"/><attr name="android:rotationY"/><attr name="android:transformPivotX"/><attr name="android:transformPivotY"/><attr name="transitionPathRotate"/><attr name="android:scaleX"/><attr name="android:scaleY"/><attr name="android:translationX"/><attr name="android:translationY"/><attr name="android:translationZ"/></declare-styleable>  
 <declare-styleable name="KeyCycle"><attr name="motionTarget"/><attr name="curveFit"/><attr name="framePosition"/><attr name="transitionEasing"/><attr name="motionProgress"/><attr name="waveShape"/><attr name="wavePeriod"/><attr name="waveOffset"/><attr name="waveVariesBy"/><attr name="transitionPathRotate"/><attr name="android:alpha"/><attr name="android:elevation"/><attr name="android:rotation"/><attr name="android:rotationX"/><attr name="android:rotationY"/><attr name="android:scaleX"/><attr name="android:scaleY"/><attr name="android:translationX"/><attr name="android:translationY"/><attr name="android:translationZ"/></declare-styleable>  
 <declare-styleable name="KeyFrame"/>  
 <declare-styleable name="KeyFramesAcceleration"/>  
 <declare-styleable name="KeyFramesVelocity"/>  
 <declare-styleable name="KeyPosition"><attr format="enum" name="keyPositionType">  
 <enum name="deltaRelative" value="0"/>  
 <enum name="pathRelative" value="1"/>  
 <enum name="parentRelative" value="2"/>  
 </attr><attr format="float" name="percentX"/><attr format="float" name="percentY"/><attr format="float" name="percentWidth"/><attr format="float" name="percentHeight"/><attr name="framePosition"/><attr name="motionTarget"/><attr name="transitionEasing"/><attr name="pathMotionArc"/><attr name="curveFit"/><attr name="drawPath"/><attr name="sizePercent"/></declare-styleable>  
 <declare-styleable name="KeyTimeCycle"><attr name="framePosition"/><attr name="motionTarget"/><attr name="transitionEasing"/><attr name="curveFit"/><attr name="waveShape"/><attr name="wavePeriod"/><attr name="motionProgress"/><attr name="waveOffset"/><attr name="waveDecay"/><attr name="android:alpha"/><attr name="android:elevation"/><attr name="android:rotation"/><attr name="android:rotationX"/><attr name="android:rotationY"/><attr name="transitionPathRotate"/><attr name="android:scaleX"/><attr name="android:scaleY"/><attr name="android:translationX"/><attr name="android:translationY"/><attr name="android:translationZ"/></declare-styleable>  
 <declare-styleable name="KeyTrigger"><attr name="framePosition"/><attr name="motionTarget"/><attr format="reference" name="triggerReceiver"/><attr format="string" name="onNegativeCross"/><attr format="string" name="onPositiveCross"/><attr format="string" name="onCross"/><attr format="float" name="triggerSlack"/><attr format="reference" name="triggerId"/><attr format="boolean" name="motion\_postLayoutCollision"/><attr format="reference" name="motion\_triggerOnCollision"/></declare-styleable>  
 <declare-styleable name="Layout"><attr name="android:layout\_width"/><attr name="android:layout\_height"/><attr name="android:layout\_marginStart"/><attr name="android:layout\_marginBottom"/><attr name="android:layout\_marginTop"/><attr name="android:layout\_marginEnd"/><attr name="android:layout\_marginLeft"/><attr name="android:layout\_marginRight"/><attr name="layout\_constraintCircle"/><attr name="layout\_constraintCircleRadius"/><attr name="layout\_constraintCircleAngle"/><attr name="layout\_constraintGuide\_begin"/><attr name="layout\_constraintGuide\_end"/><attr name="layout\_constraintGuide\_percent"/><attr name="layout\_constraintLeft\_toLeftOf"/><attr name="layout\_constraintLeft\_toRightOf"/><attr name="layout\_constraintRight\_toLeftOf"/><attr name="layout\_constraintRight\_toRightOf"/><attr name="layout\_constraintTop\_toTopOf"/><attr name="layout\_constraintTop\_toBottomOf"/><attr name="layout\_constraintBottom\_toTopOf"/><attr name="layout\_constraintBottom\_toBottomOf"/><attr name="layout\_constraintBaseline\_toBaselineOf"/><attr name="layout\_constraintStart\_toEndOf"/><attr name="layout\_constraintStart\_toStartOf"/><attr name="layout\_constraintEnd\_toStartOf"/><attr name="layout\_constraintEnd\_toEndOf"/><attr name="layout\_goneMarginLeft"/><attr name="layout\_goneMarginTop"/><attr name="layout\_goneMarginRight"/><attr name="layout\_goneMarginBottom"/><attr name="layout\_goneMarginStart"/><attr name="layout\_goneMarginEnd"/><attr name="layout\_constrainedWidth"/><attr name="layout\_constrainedHeight"/><attr name="layout\_constraintHorizontal\_bias"/><attr name="layout\_constraintVertical\_bias"/><attr name="layout\_constraintWidth\_default"/><attr name="layout\_constraintHeight\_default"/><attr name="layout\_constraintWidth\_min"/><attr name="layout\_constraintWidth\_max"/><attr name="android:orientation"/><attr name="layout\_constraintWidth\_percent"/><attr name="layout\_constraintHeight\_min"/><attr name="layout\_constraintHeight\_max"/><attr name="layout\_constraintHeight\_percent"/><attr name="layout\_constraintLeft\_creator"/><attr name="layout\_constraintTop\_creator"/><attr name="layout\_constraintRight\_creator"/><attr name="layout\_constraintBottom\_creator"/><attr name="layout\_constraintBaseline\_creator"/><attr name="layout\_constraintDimensionRatio"/><attr name="layout\_constraintHorizontal\_weight"/><attr name="layout\_constraintVertical\_weight"/><attr name="layout\_constraintHorizontal\_chainStyle"/><attr name="layout\_constraintVertical\_chainStyle"/><attr name="layout\_editor\_absoluteX"/><attr name="layout\_editor\_absoluteY"/><attr name="barrierDirection"/><attr name="barrierMargin"/><attr name="constraint\_referenced\_ids"/><attr name="constraint\_referenced\_tags"/><attr format="dimension" name="maxHeight"/><attr format="dimension" name="maxWidth"/><attr format="dimension" name="minHeight"/><attr format="dimension" name="minWidth"/><attr name="barrierAllowsGoneWidgets"/><attr name="chainUseRtl"/></declare-styleable>  
 <declare-styleable name="MockView"><attr format="string" name="mock\_label"/><attr format="color" name="mock\_labelColor"/><attr format="color" name="mock\_labelBackgroundColor"/><attr format="color" name="mock\_diagonalsColor"/><attr format="boolean" name="mock\_showDiagonals"/><attr format="boolean" name="mock\_showLabel"/></declare-styleable>  
 <declare-styleable name="Motion"><attr name="animate\_relativeTo"/><attr name="transitionEasing"/><attr name="pathMotionArc"/><attr name="motionPathRotate"/><attr name="motionStagger"/><attr name="drawPath"/></declare-styleable>  
 <declare-styleable name="MotionHelper"><attr format="boolean" name="onShow"/><attr format="boolean" name="onHide"/></declare-styleable>  
 <declare-styleable name="MotionLayout"><attr name="layoutDescription"/><attr format="reference" name="currentState"/><attr name="motionProgress"/><attr format="boolean" name="applyMotionScene"/><attr format="boolean" name="showPaths"/><attr format="enum" name="motionDebug">  
 <enum name="NO\_DEBUG" value="0"/>  
 <enum name="SHOW\_PROGRESS" value="1"/>  
 <enum name="SHOW\_PATH" value="2"/>  
 <enum name="SHOW\_ALL" value="3"/>  
 </attr></declare-styleable>  
 <declare-styleable name="MotionScene"><attr name="defaultDuration"/><attr name="layoutDuringTransition"/></declare-styleable>  
 <declare-styleable name="MotionTelltales"><attr format="enum" name="telltales\_velocityMode">  
 <enum name="layout" value="0"/>  
 <enum name="postLayout" value="1"/>  
 <enum name="staticPostLayout" value="2"/>  
 <enum name="staticLayout" value="3"/>  
 </attr><attr format="color" name="telltales\_tailColor"/><attr format="float" name="telltales\_tailScale"/></declare-styleable>  
 <declare-styleable name="OnClick"><attr name="targetId"/><attr name="clickAction">  
 <flag name="toggle" value="0x0011"/>  
 <flag name="transitionToEnd" value="0x0001"/>  
 <flag name="transitionToStart" value="0x0010"/>  
 <flag name="jumpToEnd" value="0x100"/>  
 <flag name="jumpToStart" value="0x1000"/>  
 </attr></declare-styleable>  
 <declare-styleable name="OnSwipe"><attr format="float" name="dragScale"/><attr format="float" name="dragThreshold"/><attr format="float" name="maxVelocity"/><attr format="float" name="maxAcceleration"/><attr name="dragDirection"/><attr name="touchAnchorId"/><attr name="touchAnchorSide"/><attr format="reference" name="touchRegionId"/><attr format="reference" name="limitBoundsTo"/><attr format="flags" name="nestedScrollFlags">  
 <flag name="none" value="0"/>  
 <flag name="disablePostScroll" value="1"/>  
 <flag name="disableScroll" value="2"/>  
 </attr><attr format="boolean" name="moveWhenScrollAtTop"/><attr format="enum" name="onTouchUp">  
 <enum name="autoComplete" value="0"/>  
 <enum name="autoCompleteToStart" value="1"/>  
 <enum name="autoCompleteToEnd" value="2"/>  
 <enum name="stop" value="3"/>  
 <enum name="decelerate" value="4"/>  
 <enum name="decelerateAndComplete" value="5"/>  
 </attr></declare-styleable>  
 <declare-styleable name="PropertySet"><attr name="android:visibility"/><attr name="visibilityMode"/><attr format="float" name="android:alpha"/><attr name="motionProgress"/><attr name="layout\_constraintTag"/></declare-styleable>  
 <declare-styleable name="State"><attr name="android:id"/><attr format="reference" name="constraints"/></declare-styleable>  
 <declare-styleable name="StateSet"><attr format="reference" name="defaultState"/></declare-styleable>  
 <declare-styleable name="Transform"><attr name="android:elevation"/><attr name="android:rotation"/><attr name="android:rotationX"/><attr name="android:rotationY"/><attr name="android:scaleX"/><attr name="android:scaleY"/><attr name="android:transformPivotX"/><attr name="android:transformPivotY"/><attr format="dimension" name="android:translationX"/><attr format="dimension" name="android:translationY"/><attr format="dimension" name="android:translationZ"/></declare-styleable>  
 <declare-styleable name="Transition"><attr name="android:id"/><attr format="reference" name="constraintSetStart"/><attr format="reference" name="constraintSetEnd"/><attr format="boolean" name="transitionDisable"/><attr name="layoutDuringTransition"/><attr name="pathMotionArc"/><attr format="enum" name="autoTransition">  
 <enum name="none" value="0"/>  
 <enum name="jumpToStart" value="1"/>  
 <enum name="jumpToEnd" value="2"/>  
 <enum name="animateToStart" value="3"/>  
 <enum name="animateToEnd" value="4"/>  
 </attr><attr format="string|reference|enum" name="motionInterpolator">  
 <enum name="easeInOut" value="0"/>  
 <enum name="easeIn" value="1"/>  
 <enum name="easeOut" value="2"/>  
 <enum name="linear" value="3"/>  
 <enum name="bounce" value="5"/>  
 </attr><attr name="duration"/><attr format="float" name="staggered"/><attr name="transitionFlags">  
 <flag name="none" value="0"/>  
 <flag name="beginOnFirstDraw" value="1"/>  
 />  
 </attr></declare-styleable>  
 <declare-styleable name="Variant"><attr format="dimension" name="region\_widthLessThan"/><attr format="dimension" name="region\_widthMoreThan"/><attr format="dimension" name="region\_heightLessThan"/><attr format="dimension" name="region\_heightMoreThan"/><attr name="constraints"/></declare-styleable>  
 <item name="motion\_base" type="id"/>  
</resources>

**Change\_pass\_form.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/changePassPopUp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="5dp"  
 app:cardPreventCornerOverlap="true"  
 app:cardUseCompatPadding="true">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="30dp">  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="210dp"  
 android:fontFamily="@font/bold"  
 android:text="Change Password"  
 android:textColor="@color/biru"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <EditText  
 android:id="@+id/tvOldPass"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="20dp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:fontFamily="@font/medium"  
 android:hint="Old Password"  
 android:inputType="textPassword"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp"  
 android:textAlignment="center"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3" />  
  
 <EditText  
 android:id="@+id/tvNewPass"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:fontFamily="@font/medium"  
 android:hint="New Password"  
 android:inputType="textPassword"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp"  
 android:textAlignment="center"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvOldPass" />  
  
 <com.google.android.material.button.MaterialButton  
 android:id="@+id/btnChangePass"  
 android:layout\_width="match\_parent"  
 android:layout\_height="60sp"  
 android:layout\_marginTop="20dp"  
 android:fontFamily="@font/bold"  
 android:text="Change"  
 android:textAppearance="@style/button\_text"  
 app:cornerRadius="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvNewPass" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</androidx.cardview.widget.CardView>

**Customview\_spinner.xml**

<?xml version="1.0" encoding="utf-8"?>  
<TextView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/tvFilter"  
 android:textAlignment="center"  
 android:fontFamily="@font/semibold"  
 android:textSize="16dp"  
 android:textColor="@color/white"  
 android:padding="10dp">  
  
</TextView>

**Forgot\_password.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/forgotPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 app:cardCornerRadius="20dp"  
 app:cardElevation="5dp"  
 app:cardPreventCornerOverlap="true"  
 app:cardUseCompatPadding="true">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="30dp">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
 android:fontFamily="@font/bold"  
 android:text="I forgot my Email or Password"  
 android:textColor="@color/biru"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:fontFamily="@font/medium"  
 android:text="Enter your email address"  
 android:textColor="@color/biru"  
 android:textSize="16sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView1" />  
  
 <EditText  
 android:id="@+id/tvEmailConfirm"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:fontFamily="@font/medium"  
 android:hint="Your Email"  
 android:inputType="textEmailAddress"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp"  
 android:textAlignment="center"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3" />  
  
 <com.google.android.material.button.MaterialButton  
 android:id="@+id/btnSend"  
 android:layout\_width="match\_parent"  
 android:layout\_height="60sp"  
 android:layout\_marginTop="20dp"  
 android:fontFamily="@font/bold"  
 android:text="Send"  
 android:textAppearance="@style/button\_text"  
 app:cornerRadius="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvEmailConfirm" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</androidx.cardview.widget.CardView>

**Fragment\_dashboard.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".Dashboard">  
  
 <ImageView  
 android:id="@+id/backgroundBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="400dp"  
 android:translationY="400dp"  
 android:alpha="0"  
 android:background="@drawable/bordered\_box2"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="110dp"  
 android:background="@drawable/ic\_wave\_03"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="110dp"  
 android:scrollbarThumbHorizontal="@null"  
 android:scrollbarThumbVertical="@null"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingLeft="20dp"  
 android:paddingRight="20dp">  
  
 <RadioGroup  
 android:id="@+id/RadioFilter"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/cardView3">  
  
 <RadioButton  
 android:id="@+id/RadioToday"  
 android:layout\_width="92dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:background="@drawable/radio\_selector"  
 android:button="@android:color/transparent"  
 android:elevation="4dp"  
 android:fontFamily="@font/bold"  
 android:padding="16dp"  
 android:text="Today"  
 android:textAlignment="center"  
 android:textColor="@drawable/radio\_button\_text" />  
  
 <RadioButton  
 android:id="@+id/Radio7D"  
 android:layout\_width="92dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:background="@drawable/radio\_selector"  
 android:button="@android:color/transparent"  
 android:elevation="4dp"  
 android:fontFamily="@font/bold"  
 android:padding="16dp"  
 android:text="7 days"  
 android:textAlignment="center"  
 android:textColor="@drawable/radio\_button\_text" />  
  
 <RadioButton  
 android:id="@+id/Radio31D"  
 android:layout\_width="92dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:background="@drawable/radio\_selector"  
 android:button="@android:color/transparent"  
 android:elevation="4dp"  
 android:fontFamily="@font/bold"  
 android:padding="16dp"  
 android:text="31 days"  
 android:textAlignment="center"  
 android:textColor="@drawable/radio\_button\_text" />  
  
 </RadioGroup>  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/cardView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 app:cardCornerRadius="26dp"  
 app:cardElevation="4dp"  
 app:cardPreventCornerOverlap="true"  
 app:cardUseCompatPadding="true"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="20dp">  
  
 <TextView  
 android:id="@+id/tvTextBalance"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:text="Current Balance"  
 android:textColor="@color/black"  
 android:textSize="16sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/tvCurrentBalance"  
 android:layout\_width="260dp"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:fontFamily="@font/bold"  
 android:text="0"  
 android:textColor="@color/black"  
 android:textSize="34sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvTextBalance" />  
  
 <TextView  
 android:id="@+id/tvViewHistory"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:fontFamily="@font/medium"  
 android:text="View History"  
 android:textColor="@color/biru"  
 android:textSize="14sp"  
 android:clickable="true"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvCurrentBalance" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </androidx.cardview.widget.CardView>  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/cardView3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginRight="10dp"  
 app:cardCornerRadius="26dp"  
 app:cardElevation="4dp"  
 app:cardPreventCornerOverlap="true"  
 app:cardUseCompatPadding="true"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="10dp"  
 android:padding="20dp">  
  
 <TextView  
 android:id="@+id/tvTextPercentage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:text="Financial Percentages"  
 android:textColor="@color/black"  
 android:textSize="16sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/constraintLayout4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="26dp"  
 android:layout\_marginBottom="50dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvTextPercentage">  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="10dp"  
 android:layout\_height="120dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ProgressBar  
 android:id="@+id/income\_progressBar"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:indeterminateOnly="false"  
 android:progressDrawable="@drawable/circle\_progress\_income"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/imageView2"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/outcome\_progressText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:text="0%"  
 app:layout\_constraintBottom\_toBottomOf="@+id/outcome\_progressBar"  
 app:layout\_constraintEnd\_toEndOf="@+id/outcome\_progressBar"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView2"  
 app:layout\_constraintTop\_toTopOf="@+id/outcome\_progressBar" />  
  
 <ProgressBar  
 android:id="@+id/outcome\_progressBar"  
 android:layout\_width="120dp"  
 android:layout\_height="120dp"  
 android:indeterminateOnly="false"  
 android:progressDrawable="@drawable/circle\_progress\_outcome"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView2"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/income\_progressText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:text="0%"  
 android:textColor="@color/hijau"  
 app:layout\_constraintBottom\_toBottomOf="@+id/income\_progressBar"  
 app:layout\_constraintEnd\_toEndOf="@+id/income\_progressBar"  
 app:layout\_constraintStart\_toStartOf="@+id/income\_progressBar"  
 app:layout\_constraintTop\_toTopOf="@+id/income\_progressBar" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout4">  
  
 <ImageView  
 android:id="@+id/imageView5"  
 android:layout\_width="20dp"  
 android:layout\_height="20dp"  
 android:background="@color/hijau"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/semibold"  
 android:text="Income"  
 android:textColor="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/imageView6"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView5"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ImageView  
 android:id="@+id/imageView6"  
 android:layout\_width="20dp"  
 android:layout\_height="20dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ImageView  
 android:id="@+id/imageView7"  
 android:layout\_width="20dp"  
 android:layout\_height="20dp"  
 android:background="@color/merah"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/textVie8"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginEnd="12dp"  
 android:fontFamily="@font/semibold"  
 android:text="Outcome"  
 android:textColor="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/imageView7"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView6"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </androidx.cardview.widget.CardView>  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </ScrollView>  
  
  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/constraintLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="100dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <ImageView  
 android:id="@+id/upperBox"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:alpha="0"  
 android:background="@drawable/upper\_box"  
 android:translationX="300dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/constraintLayout2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <ImageView  
 android:id="@+id/upperBox2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:alpha="0"  
 android:background="@drawable/upper\_box2"  
 android:translationX="200dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/photoProfileContainer"  
 android:layout\_width="76dp"  
 android:layout\_height="76dp"  
 app:cardCornerRadius="20dp"  
 android:alpha="0"  
 app:cardBackgroundColor="@color/white"  
 app:layout\_constraintBottom\_toBottomOf="@+id/constraintLayout2"  
 app:layout\_constraintEnd\_toStartOf="@+id/constraintLayout"  
 app:layout\_constraintStart\_toStartOf="@+id/constraintLayout2"  
 app:layout\_constraintTop\_toTopOf="@+id/constraintLayout2">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="65dp"  
 android:layout\_height="65dp"  
 android:alpha="1"  
 app:cardBackgroundColor="@color/white"  
 app:cardCornerRadius="15dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
 <ImageView  
 android:id="@+id/photoProfile"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/ic\_default\_user\_photos\_01" />  
  
 </androidx.cardview.widget.CardView>  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
  
 </androidx.cardview.widget.CardView>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/userInfoContainer"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginEnd="42dp"  
 android:alpha="0"  
 android:translationX="300dp"  
 app:layout\_constraintEnd\_toEndOf="@+id/constraintLayout"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:id="@+id/tvHello"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/medium"  
 android:text="Hello,"  
 android:textColor="@color/white"  
 android:textSize="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/usernameDashboard"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:text="Andrio Effendi"  
 android:textColor="@color/white"  
 android:textSize="22dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvHello" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Fragment\_history.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 tools:context=".History">  
  
 <ImageView  
 android:id="@+id/backgroundBoxHistory"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:background="@drawable/bordered\_box2"  
 android:alpha="0"  
 android:translationY="300dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
  
 <ImageView  
 android:id="@+id/imageView13"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="110dp"  
 android:background="@drawable/ic\_wave\_03"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/constraintLayout3"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <ImageView  
 android:id="@+id/upperBox3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:alpha="1"  
 android:background="@drawable/upper\_box3"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ImageView  
 android:id="@+id/imageView8"  
 android:layout\_width="46dp"  
 android:layout\_height="46dp"  
 android:layout\_marginStart="40dp"  
 android:src="@drawable/ic\_history\_13"  
 app:layout\_constraintBottom\_toBottomOf="@+id/upperBox3"  
 app:layout\_constraintStart\_toStartOf="@+id/upperBox3"  
 app:layout\_constraintTop\_toTopOf="@+id/upperBox3"  
 app:tint="@color/white" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="26dp"  
 android:fontFamily="@font/bold"  
 android:text="Transaction\nHistory"  
 android:textColor="@color/white"  
 android:textSize="22dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView8"  
 app:layout\_constraintTop\_toTopOf="@+id/upperBox3" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:cardCornerRadius="10dp"  
 app:cardElevation="5dp"  
 app:cardPreventCornerOverlap="true"  
 app:cardUseCompatPadding="true"  
 app:layout\_constraintBottom\_toBottomOf="@+id/constraintLayout3"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/constraintLayout3"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp">  
  
 <com.google.android.material.button.MaterialButton  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:background="@drawable/ic\_dashboardicon\_13"  
 app:elevation="10dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:strokeColor="@color/merah" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </androidx.cardview.widget.CardView>  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout3">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:id="@+id/tvIncome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:fontFamily="@font/bold"  
 android:text="Income"  
 android:scaleX="0.5"  
 android:scaleY="0.5"  
 android:alpha="0"  
 android:textColor="@color/black"  
 android:textSize="22sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerIncome"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:scaleX="0.5"  
 android:scaleY="0.5"  
 android:alpha="0"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvIncome" />  
  
 <TextView  
 android:id="@+id/tvOutcome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:fontFamily="@font/bold"  
 android:text="Outcome"  
 android:scaleX="0.5"  
 android:scaleY="0.5"  
 android:alpha="0"  
 android:textColor="@color/black"  
 android:textSize="22sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/recyclerIncome" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerOutcome"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:scaleX="0.5"  
 android:scaleY="0.5"  
 android:alpha="0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvOutcome" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </ScrollView>  
</androidx.constraintlayout.widget.ConstraintLayout>

**Fragment\_sign\_in.xml**

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout 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"  
 tools:context=".SignIn">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="50sp">  
  
 <TextView  
 android:layout\_marginTop="40sp"  
 android:id="@+id/tvEmailSignIn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:gravity="center"  
 android:text="E-mail"  
 android:textColor="@color/biru"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/editEmailSignInWrapper"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 app:boxStrokeWidth="0dp"  
 app:boxStrokeWidthFocused="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvEmailSignIn">  
  
 <EditText  
 android:id="@+id/editEmailSignIn"  
 android:inputType="textEmailAddress"  
 android:textSize="18sp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10sp"  
 android:paddingStart="18sp"  
 android:paddingEnd="18sp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:textColor="@color/biru"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvEmailSignIn" />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <TextView  
 android:id="@+id/tvPasswordSignIn"  
 android:layout\_marginTop="20dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:gravity="center"  
 android:text="Password"  
 android:textColor="@color/biru"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editEmailSignInWrapper" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/editPasswordSignInWrapper"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 app:boxStrokeWidth="0dp"  
 app:boxStrokeWidthFocused="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvPasswordSignIn">  
  
 <EditText  
 android:id="@+id/editPasswordSignIn"  
 android:textSize="18sp"  
 android:inputType="textPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10sp"  
 android:paddingStart="18sp"  
 android:paddingEnd="18sp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:textColor="@color/biru"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvPasswordSignIn" />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <com.google.android.material.button.MaterialButton  
 android:layout\_marginTop="30dp"  
 android:id="@+id/btnSignIn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="60sp"  
 android:fontFamily="@font/bold"  
 android:text="Sign In"  
 android:textSize="18sp"  
 app:cornerRadius="18sp"  
 app:layout\_constraintTop\_toBottomOf="@+id/editPasswordSignInWrapper"  
 tools:layout\_editor\_absoluteX="50sp" />  
  
 <TextView  
 android:id="@+id/ForgetPassword"  
 android:layout\_marginTop="10dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/semibold"  
 android:gravity="center"  
 android:text="Forgot Password ?"  
 android:textColor="@color/biru"  
 android:textSize="18sp"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnSignIn"  
 tools:layout\_editor\_absoluteX="50dp" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</FrameLayout>

**Fragment\_sign\_up.xml**

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 tools:context=".SignUp">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="50dp">  
  
 <TextView  
 android:id="@+id/tvEmailSignUp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:fontFamily="@font/bold"  
 android:gravity="center"  
 android:text="E-mail"  
 android:textColor="@color/biru"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/editEmailSignUpWrapper"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 app:boxStrokeWidth="0dp"  
 app:boxStrokeWidthFocused="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvEmailSignUp">  
  
 <EditText  
 android:id="@+id/editEmailSignUp"  
 android:inputType="textEmailAddress"  
 android:textSize="18sp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10sp"  
 android:paddingStart="18sp"  
 android:paddingEnd="18sp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:textColor="@color/biru"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvEmailSignUp" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <TextView  
 android:id="@+id/tvUsernameSignUp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:gravity="center"  
 android:layout\_marginTop="10dp"  
 android:text="Username"  
 android:textColor="@color/biru"  
 android:textSize="20sp"  
 app:layout\_constraintTop\_toBottomOf="@+id/editEmailSignUpWrapper" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/editUsernameSignUpWrapper"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 app:boxStrokeWidth="0dp"  
 app:boxStrokeWidthFocused="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvUsernameSignUp">  
  
 <EditText  
 android:id="@+id/editUsernameSignUp"  
 android:inputType="textPersonName"  
 android:textSize="18sp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10sp"  
 android:paddingStart="18sp"  
 android:paddingEnd="18sp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:textColor="@color/biru"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvUsernameSignUp" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <TextView  
 android:id="@+id/tvPasswordSignUp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/bold"  
 android:gravity="center"  
 android:layout\_marginTop="10dp"  
 android:text="Password"  
 android:textColor="@color/biru"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editUsernameSignUpWrapper" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/editPasswordSignUpWrapper"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 app:boxStrokeWidth="0dp"  
 app:boxStrokeWidthFocused="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvPasswordSignUp">  
  
 <EditText  
 android:id="@+id/editPasswordSignUp"  
 android:inputType="textPassword"  
 android:textSize="18sp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="10sp"  
 android:paddingStart="18sp"  
 android:paddingEnd="18sp"  
 android:background="@drawable/bordered\_edit\_text"  
 android:textColor="@color/biru"  
 app:layout\_constraintTop\_toBottomOf="@+id/tvPasswordSignUp" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <com.google.android.material.button.MaterialButton  
 android:layout\_width="match\_parent"  
 android:id="@+id/btnSignUp"  
 android:layout\_height="60dp"  
 android:fontFamily="@font/bold"  
 android:text="Sign Up"  
 android:textColor="@color/white"  
 android:textSize="18sp"  
 app:cornerRadius="18dp"  
 android:layout\_marginTop="30dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editPasswordSignUpWrapper" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
</FrameLayout>

**Google\_services.json**

{

  "project\_info": {

    "project\_number": "436512616830",

    "project\_id": "small-change-ac678",

    "storage\_bucket": "small-change-ac678.appspot.com"

  },

  "client": [

    {

      "client\_info": {

        "mobilesdk\_app\_id": "1:436512616830:android:c0630a6c1b0a51664d032f",

        "android\_client\_info": {

          "package\_name": "id.ac.umn.effendi.andrio.smallchange"

        }

      },

      "oauth\_client": [

        {

          "client\_id": "436512616830-dlodqmm1j6bf72ng5r00s03cbp4k83n3.apps.googleusercontent.com",

          "client\_type": 3

        }

      ],

      "api\_key": [

        {

          "current\_key": "AIzaSyChprqRBcaXwOxTw6eOMPA6PsJBb3kj4q4"

        }

      ],

      "services": {

        "appinvite\_service": {

          "other\_platform\_oauth\_client": [

            {

              "client\_id": "436512616830-dlodqmm1j6bf72ng5r00s03cbp4k83n3.apps.googleusercontent.com",

              "client\_type": 3

            }

          ]

        }

      }

    }

  ],

  "configuration\_version": "1"

}

**Screenshot**