package com.example.myapplication.AdminSection;  
  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.Toast;  
  
import com.example.myapplication.R;  
import com.example.myapplication.Model.User;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.\*;  
  
import java.util.ArrayList;  
  
public class ManageUsersActivity extends AppCompatActivity {  
 private ListView userListView;  
 private FirebaseDatabase database;  
 private DatabaseReference usersRef;  
 private DatabaseReference adminsRef; // Добавляем ссылку на администраторов  
 private ArrayAdapter<String> adapter;  
 private ArrayList<String> userNames;  
 private ArrayList<String> userIds;  
 private FirebaseAuth auth;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 setContentView(R.layout.*activity\_manage\_users*);  
  
 androidx.appcompat.widget.Toolbar toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
 setTitle("Управление пользователями");  
 getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
  
 userListView = findViewById(R.id.*user\_list\_view*);  
  
 auth = FirebaseAuth.*getInstance*();  
 database = FirebaseDatabase.*getInstance*();  
 usersRef = database.getReference("users");  
 adminsRef = database.getReference("admins"); // Инициализируем ссылку  
  
 // Проверяем, является ли пользователь администратором  
 checkForAdmin();  
 }  
  
 private void checkForAdmin() {  
 adminsRef.child(auth.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 Boolean isAdminValue = snapshot.getValue(Boolean.class);  
 if (isAdminValue != null && isAdminValue) {  
 loadUsers();  
 } else {  
 Toast.*makeText*(ManageUsersActivity.this, "Вы не имеете доступа к этой странице.", Toast.*LENGTH\_SHORT*).show();  
 finish(); // Закрываем активность  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
 Toast.*makeText*(ManageUsersActivity.this, "Ошибка базы данных.", Toast.*LENGTH\_SHORT*).show();  
 finish(); // Закрываем активность  
 }  
 });  
 }  
  
 private void loadUsers() {  
 // Получаем список администраторов  
 adminsRef.addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot adminsSnapshot) {  
 ArrayList<String> adminIds = new ArrayList<>();  
 for (DataSnapshot adminSnapshot : adminsSnapshot.getChildren()) {  
 if (adminSnapshot.getValue(Boolean.class)) {  
 adminIds.add(adminSnapshot.getKey());  
 }  
 }  
  
 // Теперь загружаем пользователей, исключая администраторов  
 usersRef.addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot usersSnapshot) {  
 userNames = new ArrayList<>();  
 userIds = new ArrayList<>();  
 for (DataSnapshot userSnapshot : usersSnapshot.getChildren()) {  
 String userId = userSnapshot.getKey();  
 // Пропускаем администратора  
 if (adminIds.contains(userId)) {  
 continue;  
 }  
 User user = userSnapshot.getValue(User.class);  
 if (user != null) {  
 userIds.add(userId);  
 userNames.add(user.getName() + " (" + userId + ")");  
 }  
 }  
  
 adapter = new ArrayAdapter<>(ManageUsersActivity.this, android.R.layout.*simple\_list\_item\_1*, userNames);  
 userListView.setAdapter(adapter);  
  
 userListView.setOnItemClickListener((parent, view, position, id) -> showDeleteUserDialog(position));  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 Toast.*makeText*(ManageUsersActivity.this, "Ошибка загрузки пользователей.", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 Toast.*makeText*(ManageUsersActivity.this, "Ошибка загрузки администраторов.", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private void showDeleteUserDialog(int position) {  
 String userName = userNames.get(position);  
 String userId = userIds.get(position);  
  
 new AlertDialog.Builder(this)  
 .setTitle("Удалить пользователя")  
 .setMessage("Вы уверены, что хотите удалить пользователя: " + userName + "?")  
 .setPositiveButton("Удалить", (dialog, which) -> deleteUser(userId, position))  
 .setNegativeButton(android.R.string.*cancel*, null)  
 .show();  
 }  
  
 private void deleteUser(String userId, int position) {  
 // Удаляем пользователя из узла 'users'  
 usersRef.child(userId).removeValue();  
  
 // Удаляем результаты пользователя  
 DatabaseReference resultsRef = FirebaseDatabase.*getInstance*().getReference("Results");  
 resultsRef.addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot resultsSnapshot) {  
 for (DataSnapshot testSnapshot : resultsSnapshot.getChildren()) {  
 testSnapshot.child(userId).getRef().removeValue();  
 }  
 }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 // Игнорируем для данного примера  
 }  
 });  
  
 // Удаляем пользователя из списка  
 userNames.remove(position);  
 userIds.remove(position);  
 adapter.notifyDataSetChanged();  
  
 Toast.*makeText*(this, "Пользователь удален", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public boolean onSupportNavigateUp() {  
 finish();  
 return true;  
 }  
}

package com.example.myapplication.Attempt\_Quiz\_Section;  
  
import android.app.Activity;  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.graphics.Point;  
import android.os.Bundle;  
import android.os.CountDownTimer;  
import androidx.annotation.ColorInt;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.recyclerview.widget.RecyclerView;  
import androidx.viewpager2.widget.ViewPager2;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.animation.TranslateAnimation;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.BaseAdapter;  
import android.widget.Button;  
import android.widget.GridView;  
import android.widget.LinearLayout;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
  
import com.example.myapplication.R;  
import com.example.myapplication.Model.Question;  
import com.example.myapplication.Model.Test;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import java.util.ArrayList;  
import java.util.concurrent.TimeUnit;  
import es.dmoral.toasty.Toasty;  
  
public class AttemptTest extends AppCompatActivity {  
 ArrayList<Question> questions;  
 String []answers;  
 Toolbar toolbar;  
 ViewPager2 scrollView;  
 LinearLayout indexLayout;  
 GridView quesGrid;  
 ArrayList<String> list;  
 ArrayList<String> arrayList;  
 int flag\_controller = 1;  
 long timer;  
 popGridAdapter popGrid;  
 Button next, prev;  
 TextView textView;  
 private DatabaseReference mDatabase;  
 private FirebaseAuth auth;  
 private String TESTNAME;  
 private RadioGroup group;  
 private int countPaused = 0;  
 private boolean hasTimeLimit = false; // Новая переменная для проверки наличия ограничения по времени  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 mDatabase = FirebaseDatabase.*getInstance*().getReference();  
 auth = FirebaseAuth.*getInstance*();  
 setContentView(R.layout.*activity\_attempt*);  
 questions = ((Test) getIntent().getExtras().get("Questions")).getQuestions();  
 TESTNAME = (String) getIntent().getExtras().get("TESTNAME");  
 toolbar = findViewById(R.id.*toolbar*);  
 toolbar.setTitleTextColor(getResources().getColor(android.R.color.*black*));  
 answers = new String[questions.size()];  
 setSupportActionBar(toolbar);  
 toolbar.setTitleTextColor(getResources().getColor(android.R.color.*black*));  
  
 // Изменено на ViewPager2  
 scrollView = findViewById(R.id.*discrete*);  
 scrollView.setOrientation(ViewPager2.*ORIENTATION\_HORIZONTAL*);  
  
 final QuestionAdapter questionAdapter = new QuestionAdapter(questions);  
 scrollView.setAdapter(questionAdapter);  
  
 next = findViewById(R.id.*next*);  
 next.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Изменено для ViewPager2  
 if (scrollView.getCurrentItem() == questions.size() - 1) {  
 showPopUp();  
 } else {  
 scrollView.setCurrentItem(scrollView.getCurrentItem() + 1);  
 }  
 }  
 });  
  
 prev = findViewById(R.id.*prev*);  
 prev.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Изменено для ViewPager2  
 if (scrollView.getCurrentItem() != 0) {  
 scrollView.setCurrentItem(scrollView.getCurrentItem() - 1);  
 }  
 }  
 });  
  
 // Изменено для ViewPager2  
 setNextPrevButton(scrollView.getCurrentItem());  
 indexLayout = findViewById(R.id.*index\_layout*);  
 indexLayout.setAlpha(.5f);  
 quesGrid = findViewById(R.id.*pop\_grid*);  
 popGrid = new popGridAdapter(AttemptTest.this);  
 quesGrid.setAdapter(popGrid);  
 quesGrid.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 // Изменено для ViewPager2  
 scrollView.setCurrentItem(i);  
 slideUp(indexLayout);  
 }  
 });  
  
 // Добавлен слушатель страниц для ViewPager2  
 scrollView.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {  
 @Override  
 public void onPageSelected(int position) {  
 super.onPageSelected(position);  
 setNextPrevButton(position);  
 }  
 });  
  
 // Получаем время теста  
 Long testTime = ((Test) getIntent().getExtras().get("Questions")).getTime();  
 if (testTime != null && testTime > 0) {  
 timer = testTime \* 60 \* 1000; // переводим минуты в миллисекунды  
 hasTimeLimit = true;  
 } else {  
 timer = -1; // Отрицательное значение означает отсутствие ограничения по времени  
 hasTimeLimit = false;  
 }  
  
 // Если есть ограничение по времени, то выведем предупреждение при выходе из приложения  
 if (hasTimeLimit) {  
 Toasty.*info*(AttemptTest.this, "У Вас ограничение по времени! Завершите или отправьте свой тест.", Toasty.*LENGTH\_LONG*).show();  
 }  
 }  
  
 void showPopUp() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(AttemptTest.this);  
 builder.setMessage("Вы хотите отправить результаты?");  
 builder.setPositiveButton("Да", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 submit();  
 dialogStart();  
 }  
 });  
  
 builder.setNegativeButton("Нет", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 dialogInterface.cancel();  
 }  
 });  
 builder.show();  
 }  
  
 /\* Отправка результатов в базу данных \*/  
 void submit() {  
 flag\_controller = 0;  
 int score = 0;  
 list = new ArrayList<>();  
 arrayList = new ArrayList<>();  
 for (int i = 0; i < answers.length; i++) {  
 if (answers[i] != null && answers[i].equals(questions.get(i).getAnswer())) {  
 score++;  
 }  
 String temp = (answers[i] != null) ? answers[i] + ")" : "null) ";  
  
 list.add("Ваш выбор (" +  
 temp +  
 " Правильный ответ (" + questions.get(i).getAnswer() + ")");  
 arrayList.add(questions.get(i).getQuestion());  
 }  
  
 try {  
 mDatabase.child("Results").child(((Test) getIntent().getExtras().get("Questions")).getName())  
 .child(auth.getUid()).setValue(score);  
 } catch (Exception e) {  
 Log.*e*("Result Update Failed ", e.getMessage());  
 }  
 }  
  
 void dialogStart() {  
  
 final AlertDialog.Builder builderSingle = new AlertDialog.Builder(AttemptTest.this);  
 builderSingle.setIcon(R.mipmap.*ic\_launcher\_round*);  
 builderSingle.setTitle("Ответы на " + TESTNAME);  
 final ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>  
 (AttemptTest.this, android.R.layout.*select\_dialog\_singlechoice*);  
 final ArrayAdapter<String> arrayAdapter1 = new ArrayAdapter<>  
 (AttemptTest.this, android.R.layout.*select\_dialog\_singlechoice*);  
  
 for (String y : arrayList) {  
 arrayAdapter1.add(y);  
 }  
 for (String x : list) {  
 arrayAdapter.add(x);  
 }  
  
 builderSingle.setCancelable(false);  
 builderSingle.setNegativeButton("Готово!", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 finish();  
 dialog.dismiss();  
 }  
 });  
  
 builderSingle.setAdapter(arrayAdapter1, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 String strName = arrayAdapter.getItem(which);  
 AlertDialog.Builder builderInner = new AlertDialog.Builder(AttemptTest.this);  
 builderInner.setMessage(strName);  
 builderInner.setCancelable(false);  
 builderInner.setTitle("Ваш Ответ:");  
 builderInner.setPositiveButton("Ок", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 builderSingle.show();  
 }  
 });  
 builderInner.show();  
 }  
 });  
 builderSingle.show();  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 if (hasTimeLimit && countPaused <= 2 && countPaused >= 0 && flag\_controller == 1) {  
 Toasty.*info*(AttemptTest.this, "Завершите или отправьте свой тест! У вас ограничение по времени.", Toasty.*LENGTH\_LONG*).show();  
 }  
 countPaused++;  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 if (hasTimeLimit && countPaused > 2) {  
 Toasty.*success*(AttemptTest.this, "Спасибо! Ваши ответы были отправлены.", Toasty.*LENGTH\_SHORT*).show();  
 countPaused = -1000;  
 submit();  
 dialogStart();  
 }  
 }  
  
 @Override  
 protected void onRestart() {  
 super.onRestart();  
 if (hasTimeLimit) {  
 Toasty.*info*(AttemptTest.this, "Завершите или отправьте свой тест! У вас ограничение по времени.", Toasty.*LENGTH\_LONG*).show();  
 }  
 }  
  
 void setNextPrevButton(int pos) {  
 if (pos == 0) {  
 prev.setText("");  
 } else {  
 prev.setText("Предыдущий");  
 }  
 if (pos == questions.size() - 1) {  
 next.setText("Отправить");  
 } else {  
 next.setText("Следующий");  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 // Убираем вызов super.onBackPressed(), чтобы предотвратить случайный выход  
 super.onBackPressed();  
 showPopUp();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
  
 super.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.*attempt\_menu*, menu);  
  
 if (hasTimeLimit) {  
 final MenuItem counter = menu.findItem(R.id.*counter*);  
  
 new CountDownTimer(timer, 1000) {  
 public void onTick(long millisUntilFinished) {  
 long millis = millisUntilFinished;  
 long hr = TimeUnit.*MILLISECONDS*.toHours(millis),  
 mn = (TimeUnit.*MILLISECONDS*.toMinutes(millis) -  
 TimeUnit.*HOURS*.toMinutes(TimeUnit.*MILLISECONDS*.toHours(millis))),  
 sc = TimeUnit.*MILLISECONDS*.toSeconds(millis) -  
 TimeUnit.*MINUTES*.toSeconds(TimeUnit.*MILLISECONDS*.toMinutes(millis));  
  
 String hms = format(hr) + ":" + format(mn) + ":" + format(sc);  
 counter.setTitle(hms);  
 timer = millis;  
 }  
  
 String format(long n) {  
 if (n < 10)  
 return "0" + n;  
 else return "" + n;  
 }  
  
 public void onFinish() {  
 submit();  
 dialogStart();  
 }  
 }.start();  
 } else {  
 // Если нет ограничения по времени, скрываем элемент меню с таймером  
 MenuItem counter = menu.findItem(R.id.*counter*);  
 counter.setVisible(false);  
 }  
  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
 if (id == R.id.*submit*) {  
 showPopUp();  
  
 return true;  
 } else if (id == R.id.*info*) {  
 togglePopUp();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 void togglePopUp() {  
 if (indexLayout.getVisibility() == View.*GONE*) {  
 slideDown(indexLayout);  
 } else slideUp(indexLayout);  
 }  
  
 class QuestionAdapter extends RecyclerView.Adapter<QuestionAdapter.ViewHolder> {  
  
 private ArrayList<Question> data;  
  
 QuestionAdapter(ArrayList<Question> data) {  
 this.data = data;  
 }  
  
 @Override  
 public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 LayoutInflater inflater = LayoutInflater.*from*(parent.getContext());  
 View v = inflater.inflate(R.layout.*frag\_test*, parent, false);  
 // Убедитесь, что элементы занимают всю доступную высоту  
 ViewGroup.LayoutParams params = new ViewGroup.LayoutParams(  
 ViewGroup.LayoutParams.*MATCH\_PARENT*,  
 ViewGroup.LayoutParams.*MATCH\_PARENT*);  
 v.setLayoutParams(params);  
 return new ViewHolder(v);  
 }  
  
 @Override  
 public void onBindViewHolder(final ViewHolder holder, final int position) {  
 Question currentQuestion = data.get(position);  
 holder.questionText.setText(currentQuestion.getQuestion());  
 holder.r1.setText(currentQuestion.getOpt\_A());  
 holder.r2.setText(currentQuestion.getOpt\_B());  
 holder.r5.setText("Снять выбор");  
  
 // Управляем видимостью вариантов в зависимости от количества опций  
 int optionCount = currentQuestion.getOptionCount();  
 if (optionCount >= 3) {  
 holder.r3.setVisibility(View.*VISIBLE*);  
 holder.r3.setText(currentQuestion.getOpt\_C());  
 } else {  
 holder.r3.setVisibility(View.*GONE*);  
 }  
 if (optionCount == 4) {  
 holder.r4.setVisibility(View.*VISIBLE*);  
 holder.r4.setText(currentQuestion.getOpt\_D());  
 } else {  
 holder.r4.setVisibility(View.*GONE*);  
 }  
  
 holder.radioGroup.setOnCheckedChangeListener(null); // Отключаем слушатель при обновлении  
  
 if (answers[position] == null) {  
 holder.radioGroup.clearCheck();  
 } else if (answers[position].equals("A")) {  
 holder.radioGroup.check(R.id.*radioButton*);  
 } else if (answers[position].equals("B")) {  
 holder.radioGroup.check(R.id.*radioButton2*);  
 } else if (answers[position].equals("C")) {  
 holder.radioGroup.check(R.id.*radioButton3*);  
 } else if (answers[position].equals("D")) {  
 holder.radioGroup.check(R.id.*radioButton4*);  
 }  
  
 holder.radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(RadioGroup radioGroup, int i) {  
 int adapterPosition = holder.getAdapterPosition();  
 if (adapterPosition != RecyclerView.*NO\_POSITION*) {  
 if (i == R.id.*radioButton*) {  
 answers[adapterPosition] = "A";  
 } else if (i == R.id.*radioButton2*) {  
 answers[adapterPosition] = "B";  
 } else if (i == R.id.*radioButton3*) {  
 answers[adapterPosition] = "C";  
 } else if (i == R.id.*radioButton4*) {  
 answers[adapterPosition] = "D";  
 } else if (i == R.id.*radioButton5*) {  
 holder.radioGroup.clearCheck();  
 answers[adapterPosition] = null;  
 }  
 popGrid.notifyDataSetChanged();  
 }  
 }  
 });  
  
 }  
  
 @Override  
 public int getItemCount() {  
 return data.size();  
 }  
  
 class ViewHolder extends RecyclerView.ViewHolder {  
  
 private TextView questionText;  
 private RadioGroup radioGroup;  
 private RadioButton r1, r2, r3, r4, r5;  
  
 ViewHolder(View itemView) {  
 super(itemView);  
 questionText = itemView.findViewById(R.id.*questionTextView*);  
 radioGroup = itemView.findViewById(R.id.*radioGroup*);  
 r1 = itemView.findViewById(R.id.*radioButton*);  
 r2 = itemView.findViewById(R.id.*radioButton2*);  
 r3 = itemView.findViewById(R.id.*radioButton3*);  
 r4 = itemView.findViewById(R.id.*radioButton4*);  
 r5 = itemView.findViewById(R.id.*radioButton5*);  
 }  
 }  
 }  
  
  
 class popGridAdapter extends BaseAdapter {  
 Context mContext;  
  
 popGridAdapter(Context context) {  
 mContext = context;  
 }  
  
 @Override  
 public Object getItem(int i) {  
 return null;  
 }  
  
 @Override  
 public long getItemId(int i) {  
 return i;  
 }  
  
 @Override  
 public int getCount() {  
 return questions.size();  
 }  
  
 @Override  
 public View getView(final int i, View view, ViewGroup viewGroup) {  
 View convertView;  
 if (view == null) {  
 convertView = new Button(mContext);  
 } else convertView = view;  
 if (answers[i] == null)  
 (convertView).setBackgroundColor(getResources().getColor(android.R.color.*darker\_gray*));  
 else  
 (convertView).setBackgroundColor(getResources().getColor(android.R.color.*holo\_green\_dark*));  
  
 ((Button) convertView).setText("" + (i + 1));  
  
 (convertView).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 scrollView.setCurrentItem(i);  
 }  
 });  
 return convertView;  
 }  
 }  
  
 public void slideUp(View view) {  
 TranslateAnimation animate = new TranslateAnimation(  
 0, // fromXDelta  
 0, // toXDelta  
 0, // fromYDelta  
 -view.getHeight()); // toYDelta  
 animate.setDuration(500);  
 view.startAnimation(animate);  
 view.setVisibility(View.*GONE*);  
 }  
  
 // slide the view from its current position to below itself  
 public void slideDown(View view) {  
 view.setVisibility(View.*VISIBLE*);  
 TranslateAnimation animate = new TranslateAnimation(  
 0, // fromXDelta  
 0, // toXDelta  
 -view.getHeight(), // fromYDelta  
 0); // toYDelta  
 animate.setDuration(500);  
 view.startAnimation(animate);  
 }  
  
 @Override  
 protected void onDestroy() {  
 // Убираем автоматическую отправку при закрытии, чтобы пользователь мог решать, когда отправить  
 super.onDestroy();  
 }  
}

package com.example.myapplication.Attempt\_Quiz\_Section;  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.core.content.ContextCompat;  
  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.Filterable;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.myapplication.R;  
import com.example.myapplication.Model.Question;  
import com.example.myapplication.Model.Test;  
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.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.Objects;  
  
public class Tests extends AppCompatActivity {  
 private FirebaseDatabase database;  
 private DatabaseReference myRef;  
 private ProgressBar progressBar;  
 private ListView listView;  
 private TestAdapter testAdapter;  
 private int lastPos = -1;  
  
 ArrayList<Test> tests = new ArrayList<>();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_tests*);  
  
 // Update layout file to use ProgressBar instead of AVLoadingIndicatorView  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 toolbar.setTitleTextColor(getResources().getColor(android.R.color.*black*));  
 setSupportActionBar(toolbar);  
  
 progressBar = findViewById(R.id.*loader1*);  
 progressBar.setVisibility(View.*VISIBLE*);  
  
 setSupportActionBar(toolbar);  
 Objects.*requireNonNull*(getSupportActionBar())  
 .setDisplayHomeAsUpEnabled(true);  
  
 database = FirebaseDatabase.*getInstance*();  
 myRef = database.getReference();  
  
 listView = findViewById(R.id.*test\_listview*);  
 testAdapter = new TestAdapter(Tests.this, tests);  
 listView.setAdapter(testAdapter);  
  
 getQues();  
 }  
  
 @Override  
 protected void onRestart() {  
 super.onRestart();  
 Toast.*makeText*(Tests.this, "Finish or submit your test! Before time runs out.", Toast.*LENGTH\_LONG*).show();  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
 if (id == android.R.id.*home*) {  
 finish();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 public void getQues() {  
 myRef.child("tests").addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 tests.clear();  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 Test t = new Test();  
 t.setName(snapshot.getKey());  
 t.setTime(Long.*parseLong*(snapshot.child("Time").getValue().toString()));  
  
 ArrayList<Question> ques = new ArrayList<>();  
 for (DataSnapshot qSnap : snapshot.child("Questions").getChildren()) {  
 ques.add(qSnap.getValue(Question.class));  
 }  
 t.setQuestions(ques);  
 tests.add(t);  
 }  
  
 testAdapter.dataList = tests;  
 testAdapter.notifyDataSetChanged();  
 progressBar.setVisibility(View.*GONE*);  
  
 Log.*e*("The read success: ", "su" + tests.size());  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 progressBar.setVisibility(View.*GONE*);  
 Log.*e*("The read failed: ", databaseError.getMessage());  
 }  
 });  
 }  
  
 class TestAdapter extends ArrayAdapter<Test> implements Filterable {  
 private Context mContext;  
 ArrayList<Test> dataList;  
  
 public TestAdapter(Context context, ArrayList<Test> list) {  
 super(context, 0, list);  
 mContext = context;  
 dataList = list;  
 }  
  
 @SuppressLint("SetTextI18n")  
 @NonNull  
 @Override  
 public View getView(final int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 View listItem = convertView;  
 if (listItem == null)  
 listItem = LayoutInflater.*from*(mContext).inflate(R.layout.*test\_item*, parent, false);  
  
 ((ImageView) listItem.findViewById(R.id.*item\_imageView*))  
 .setImageDrawable(ContextCompat.*getDrawable*(mContext, R.drawable.*ic\_appicon*));  
  
 ((TextView) listItem.findViewById(R.id.*item\_textView*))  
 .setText(dataList.get(position).getName() + " : " + dataList.get(position).getTime() + "Min");  
  
 ((Button) listItem.findViewById(R.id.*item\_button*)).setText("Attempt");  
  
 listItem.findViewById(R.id.*item\_button*).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(mContext, AttemptTest.class);  
 intent.putExtra("Questions", dataList.get(position));  
 intent.putExtra("TESTNAME", dataList.get(position).getName());  
 startActivity(intent);  
 }  
 });  
  
 Animation animation = AnimationUtils.*loadAnimation*(getContext(),  
 (position > lastPos) ? R.anim.*up\_from\_bottom* : R.anim.*down\_from\_top*);  
 listItem.startAnimation(animation);  
 lastPos = position;  
  
 return listItem;  
 }  
 }  
}

package com.example.myapplication.Auth\_Controller;  
  
import android.content.Intent;  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
import com.example.myapplication.View.MainActivity;  
import com.example.myapplication.R;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class LoginActivity extends AppCompatActivity {  
 private EditText inputEmail, inputPassword;  
 private FirebaseAuth auth;  
 private ProgressBar progressBar;  
 private Button btnSignup, btnLogin, btnReset;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 auth = FirebaseAuth.*getInstance*();  
 if (auth.getCurrentUser() != null) {  
 Intent intent = new Intent(LoginActivity.this, MainActivity.class);  
 startActivity(intent);  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 finish();  
 }  
 setContentView(R.layout.*activity\_login*);  
  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
 inputEmail = findViewById(R.id.*email*);  
 inputPassword = findViewById(R.id.*password*);  
 progressBar = findViewById(R.id.*loader1*);  
 btnSignup = findViewById(R.id.*btn\_signup*);  
 btnLogin = findViewById(R.id.*btn\_login*);  
 btnReset = findViewById(R.id.*btn\_reset\_password*);  
  
 auth = FirebaseAuth.*getInstance*();  
  
 btnSignup.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(LoginActivity.this, SignUp.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 }  
 });  
  
 btnReset.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(LoginActivity.this, ResetPasswordActivity.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 }  
 });  
  
 btnLogin.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String email = inputEmail.getText().toString();  
 final String password = inputPassword.getText().toString();  
  
 if (TextUtils.*isEmpty*(email)) {  
 Toast.*makeText*(getApplicationContext(), "Enter email address!", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 if (TextUtils.*isEmpty*(password)) {  
 Toast.*makeText*(getApplicationContext(), "Enter password!", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 progressBar.setVisibility(View.*VISIBLE*);  
  
 auth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 progressBar.setVisibility(View.*GONE*);  
 if (!task.isSuccessful()) {  
 if (password.length() < 6) {  
 inputPassword.setError(getString(R.string.minimum\_password));  
 } else {  
 Toast.makeText(LoginActivity.this, getString(R.string.auth\_failed), Toast.LENGTH\_LONG).show();  
 }  
 } else {  
 startActivity(new Intent(LoginActivity.this, MainActivity.class));  
 overridePendingTransition(android.R.anim.fade\_in, android.R.anim.fade\_out);  
 finish();  
 }  
 }  
 });  
 }  
 });  
 }  
}

package com.example.myapplication.Auth\_Controller;  
  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
import com.example.myapplication.R;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
  
public class ResetPasswordActivity extends AppCompatActivity {  
 private EditText inputEmail;  
 private Button btnReset, btnBack;  
 private FirebaseAuth auth;  
 private ProgressBar progressBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_reset\_password*);  
 inputEmail = findViewById(R.id.*email*);  
 btnReset = findViewById(R.id.*btn\_reset\_password*);  
 btnBack = findViewById(R.id.*btn\_back*);  
 progressBar = findViewById(R.id.*progressBar*);  
 auth = FirebaseAuth.*getInstance*();  
  
 btnBack.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 btnReset.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String email = inputEmail.getText().toString().trim();  
 if (TextUtils.*isEmpty*(email)) {  
 Toast.*makeText*(getApplicationContext(), "Enter your registered email id", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 progressBar.setVisibility(View.*VISIBLE*);  
  
 auth.sendPasswordResetEmail(email)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 progressBar.setVisibility(View.*GONE*);  
 if (task.isSuccessful()) {  
 Toast.*makeText*(ResetPasswordActivity.this, "We have sent you instructions to reset your password!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(ResetPasswordActivity.this, "Failed to send reset email!", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 });  
 }  
 });  
 }  
}

package com.example.myapplication.Auth\_Controller;  
  
import android.content.Intent;  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
import com.example.myapplication.R;  
import com.example.myapplication.View.MainActivity;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.example.myapplication.Model.User;  
  
public class SignUp extends AppCompatActivity {  
 private EditText inputEmail, inputPassword, inputName;  
 private Button btnSignIn, btnSignUp, btnResetPassword;  
 private ProgressBar progressBar;  
 private FirebaseAuth auth;  
 private DatabaseReference databaseReference;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_signup*);  
  
 auth = FirebaseAuth.*getInstance*();  
 databaseReference = FirebaseDatabase.*getInstance*().getReference();  
  
 btnSignIn = findViewById(R.id.*sign\_in\_button*);  
 btnSignUp = findViewById(R.id.*sign\_up\_button*);  
 inputEmail = findViewById(R.id.*email*);  
 inputPassword = findViewById(R.id.*password*);  
 inputName = findViewById(R.id.*name*);  
 progressBar = findViewById(R.id.*loader1*);  
 btnResetPassword = findViewById(R.id.*btn\_reset\_password*);  
  
 btnResetPassword.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(SignUp.this, ResetPasswordActivity.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 }  
 });  
  
 btnSignIn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 overridePendingTransition(android.R.anim.fade\_in, android.R.anim.fade\_out);  
 finish();  
 }  
 });  
  
 btnSignUp.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String email = inputEmail.getText().toString().trim();  
 String password = inputPassword.getText().toString().trim();  
 String name = inputName.getText().toString().trim();  
  
 if (TextUtils.isEmpty(email)) {  
 inputEmail.setError("Enter valid Email");  
 return;  
 }  
  
 if (TextUtils.isEmpty(password)) {  
 inputPassword.setError("Password should be at least 6 characters!");  
 return;  
 }  
  
 if (password.length() < 6) {  
 Toast.makeText(getApplicationContext(), "Password too short, enter minimum 6 characters!", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 progressBar.setVisibility(View.VISIBLE);  
  
 auth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(SignUp.this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 progressBar.setVisibility(View.GONE);  
 if (!task.isSuccessful()) {  
 Toast.makeText(SignUp.this, "Authentication failed." + task.getException(), Toast.LENGTH\_SHORT).show();  
 } else {  
 // Создание пользователя в базе данных  
 String userId = auth.getCurrentUser().getUid();  
 User user = new User(name);  
 databaseReference.child("users").child(userId).setValue(user)  
 .addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task) {  
 if (task.isSuccessful()) {  
 Toast.makeText(SignUp.this, "Registration successful!", Toast.LENGTH\_SHORT).show();  
 startActivity(new Intent(SignUp.this, MainActivity.class));  
 overridePendingTransition(android.R.anim.fade\_in, android.R.anim.fade\_out);  
 finish();  
 } else {  
 Toast.makeText(SignUp.this, "Failed to register user details!", Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
 });  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 progressBar.setVisibility(View.GONE);  
 }  
}

package com.example.myapplication.Create\_Quiz;  
  
import android.content.Intent;  
import android.os.Bundle;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import java.util.Objects;  
import com.example.myapplication.R;  
  
public class create\_quiz\_main extends AppCompatActivity {  
 private Button addButton;  
 private Toolbar toolbar;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*content\_main\_1*);  
  
 toolbar = findViewById(R.id.*toolbartst*);  
 toolbar.setTitleTextColor(getResources().getColor(android.R.color.*black*));  
 setSupportActionBar(toolbar);  
 Objects.*requireNonNull*(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);  
  
 // Инициализируем кнопку "Add"  
 addButton = findViewById(R.id.*add\_button*);  
  
 // Устанавливаем обработчик нажатия на кнопку  
 addButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Запускаем активность создания теста  
 Intent intent = new Intent(create\_quiz\_main.this, Custom\_quiz.class);  
 startActivity(intent);  
 }  
 });  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 if (item.getItemId() == android.R.id.*home*) {  
 finish();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
}

package com.example.myapplication.Create\_Quiz;  
  
import android.app.AlertDialog;  
import android.app.ProgressDialog;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.cardview.widget.CardView;  
  
import android.text.TextUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.bumptech.glide.Glide;  
import com.example.myapplication.R;  
import com.example.myapplication.Model.Question;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.storage.StorageReference;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.gson.Gson;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
import java.util.ArrayList;  
import java.util.Map;  
  
public class Custom\_quiz extends AppCompatActivity {  
  
 EditText question;  
 EditText aText;  
 EditText bText;  
 EditText cText;  
 EditText dText;  
 RadioButton aRadio;  
 RadioButton bRadio;  
 RadioButton cRadio;  
 RadioButton dRadio;  
  
 RadioGroup optionsGroup;  
  
 int currentQuestion = 1;  
 int previousQuestion = 1;  
 TextView questionNumber;  
  
 ArrayList<Question> ques;  
 JSONArray jsonArray;  
 String selectedOption = "";  
  
 String fileName = "file";  
 private FirebaseDatabase database;  
 private FirebaseAuth auth;  
 private DatabaseReference myRef;  
 CardView fab, f2, fl;  
  
 Button addOptionButton;  
 Button removeOptionButton;  
  
 int optionCount = 2;  
  
 private LinearLayout optionCLayout, optionDLayout;  
  
 // Новые переменные для выбора изображения  
 private Button selectImageButton;  
 private ImageView questionImageView;  
 private Uri selectedImageUri = null;  
 private StorageReference storageReference;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
  
 super.onCreate(savedInstanceState);  
 jsonArray = new JSONArray();  
 setContentView(R.layout.*activity\_create\_quiz*);  
  
 question = findViewById(R.id.*questionView*);  
 aText = findViewById(R.id.*aText*);  
 bText = findViewById(R.id.*bText*);  
 cText = findViewById(R.id.*cText*);  
 dText = findViewById(R.id.*dText*);  
 questionNumber = findViewById(R.id.*questionNumber*);  
 aRadio = findViewById(R.id.*aRadio*);  
 bRadio = findViewById(R.id.*bRadio*);  
 cRadio = findViewById(R.id.*cRadio*);  
 dRadio = findViewById(R.id.*dRadio*);  
 optionsGroup = findViewById(R.id.*optionsGroup*);  
  
 addOptionButton = findViewById(R.id.*addOptionButton*);  
 removeOptionButton = findViewById(R.id.*removeOptionButton*);  
  
 optionCLayout = findViewById(R.id.*optionCLayout*);  
 optionDLayout = findViewById(R.id.*optionDLayout*);  
  
 optionCLayout.setVisibility(View.*GONE*);  
 optionDLayout.setVisibility(View.*GONE*);  
  
 auth = FirebaseAuth.*getInstance*();  
 database = FirebaseDatabase.*getInstance*();  
 myRef = database.getReference();  
 storageReference = FirebaseStorage.*getInstance*().getReference();  
  
 selectedOption = "";  
 currentQuestion = 1;  
 setListeners();  
  
 ques = new ArrayList<>();  
  
 fab = findViewById(R.id.*nextfab*);  
 fl = findViewById(R.id.*fab2*); // кнопка сохранения  
 f2 = findViewById(R.id.*pre\_card*);  
  
 f2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 if (previousQuestion > 1) {  
 previousQuestion--;  
 setAllData(previousQuestion);  
 }  
 if (previousQuestion == 1)  
 f2.setVisibility(View.*INVISIBLE*);  
 Toast.*makeText*(Custom\_quiz.this, String.*valueOf*(previousQuestion), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 fab.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 // Сохраняем текущий вопрос  
 boolean cont = getEnteredQuestionsValue();  
 if (cont) {  
 previousQuestion++;  
 currentQuestion++;  
 Toast.*makeText*(Custom\_quiz.this, "Вопрос " + currentQuestion, Toast.*LENGTH\_SHORT*).show();  
 questionNumber.setText(String.*valueOf*(currentQuestion));  
 clearAllData();  
 f2.setVisibility(View.*VISIBLE*);  
 resetOptionCount();  
 }  
 }  
 });  
  
 fl.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 // Сохраняем текущий вопрос, если он ещё не сохранён  
 boolean cont = getEnteredQuestionsValue();  
  
 if (cont || jsonArray.length() > 0) {  
 final JSONObject tempObject = new JSONObject();  
 // Получаем вид из dialog\_custom.xml  
 LayoutInflater li = LayoutInflater.*from*(Custom\_quiz.this);  
 View promptsView = li.inflate(R.layout.*dialog\_custom*, null);  
 AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(Custom\_quiz.this);  
  
 // Устанавливаем dialog\_custom.xml для AlertDialog Builder  
 alertDialogBuilder.setView(promptsView);  
 final EditText userInput = promptsView.findViewById(R.id.*editTextDialogUserInput*);  
 final EditText userTime = promptsView.findViewById(R.id.*editTextDialogUserInput1*);  
  
 // Опционально устанавливаем ограничение по времени  
 userTime.setHint("Введите время в минутах (опционально)");  
  
 // Настраиваем диалоговое окно  
 alertDialogBuilder  
 .setCancelable(false)  
 .setPositiveButton("OK", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
  
 String timeStr = userTime.getText().toString().trim();  
 try {  
 tempObject.put("Questions", jsonArray);  
 if (!TextUtils.*isEmpty*(timeStr)) {  
 int time = Integer.*parseInt*(timeStr);  
 tempObject.put("Time", time);  
 } else {  
 tempObject.put("Time", 0); // Без ограничения по времени  
 }  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 final String jsonStr = tempObject.toString();  
 // Получаем имя теста  
 if (!TextUtils.*isEmpty*(userInput.getText().toString())) {  
 String testName = userInput.getText().toString().trim();  
 Map<String, Object> result = new Gson().fromJson(jsonStr, Map.class);  
 myRef.child("tests").child(testName).setValue(result);  
 Toast.makeText(getApplicationContext(), "Тест успешно сохранён", Toast.LENGTH\_SHORT).show();  
 } else {  
 Toast.makeText(getApplicationContext(), "Имя теста не может быть пустым", Toast.LENGTH\_SHORT).show();  
 }  
 }  
 })  
 .setNegativeButton("Отмена", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int id) {  
 dialog.cancel();  
 }  
 });  
 // Создаём AlertDialog  
 AlertDialog alertDialog = alertDialogBuilder.create();  
 // Отображаем его  
 alertDialog.show();  
 } else {  
 Toast.makeText(getApplicationContext(),  
 "Некорректный формат вопроса", Toast.LENGTH\_SHORT).show();  
 }  
 }  
 });  
  
 addOptionButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 increaseOptionCount();  
 }  
 });  
  
 removeOptionButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 decreaseOptionCount();  
 }  
 });  
  
 // Устанавливаем начальное количество вариантов  
 resetOptionCount();  
  
 // Инициализируем элементы для выбора изображения  
 selectImageButton = findViewById(R.id.selectImageButton);  
 questionImageView = findViewById(R.id.questionImageView);  
  
 selectImageButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent();  
 intent.setType("image/\*");  
 intent.setAction(Intent.ACTION\_GET\_CONTENT);  
 startActivityForResult(Intent.createChooser(intent, "Выберите изображение"), 1);  
 }  
 });  
 }  
  
 private void resetOptionCount() {  
 optionCount = 2;  
 optionCLayout.setVisibility(View.GONE);  
 optionDLayout.setVisibility(View.GONE);  
  
 cText.setText("");  
 dText.setText("");  
 cRadio.setChecked(false);  
 dRadio.setChecked(false);  
 checkOptionButtons();  
 }  
  
 private void increaseOptionCount() {  
 if (optionCount < 4) {  
 optionCount++;  
 if (optionCount == 3) {  
 optionCLayout.setVisibility(View.VISIBLE);  
 cText.setVisibility(View.VISIBLE);  
 cRadio.setVisibility(View.VISIBLE);  
 } else if (optionCount == 4) {  
 optionDLayout.setVisibility(View.VISIBLE);  
 dText.setVisibility(View.VISIBLE);  
 dRadio.setVisibility(View.VISIBLE);  
 }  
 }  
 checkOptionButtons();  
 }  
  
 private void decreaseOptionCount() {  
 if (optionCount > 2) {  
 if (optionCount == 4) {  
 optionDLayout.setVisibility(View.GONE);  
 dText.setText("");  
 dText.setVisibility(View.GONE);  
 dRadio.setChecked(false);  
 dRadio.setVisibility(View.GONE);  
 if (dRadio.isChecked()) {  
 dRadio.setChecked(false);  
 selectedOption = "";  
 }  
 } else if (optionCount == 3) {  
 optionCLayout.setVisibility(View.GONE);  
 cText.setText("");  
 cText.setVisibility(View.GONE);  
 cRadio.setChecked(false);  
 cRadio.setVisibility(View.GONE);  
 if (cRadio.isChecked()) {  
 cRadio.setChecked(false);  
 selectedOption = "";  
 }  
 }  
 optionCount--;  
 }  
 checkOptionButtons();  
 }  
  
 private void checkOptionButtons() {  
 if (optionCount == 2) {  
 removeOptionButton.setEnabled(false);  
 } else {  
 removeOptionButton.setEnabled(true);  
 }  
 if (optionCount == 4) {  
 addOptionButton.setEnabled(false);  
 } else {  
 addOptionButton.setEnabled(true);  
 }  
 }  
  
 @Override  
 public void onBackPressed() {  
 super.onBackPressed();  
 final AlertDialog.Builder builder = new AlertDialog.Builder(Custom\_quiz.this);  
 builder.setMessage("Выйти без сохранения?");  
 builder.setPositiveButton("Ок", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
  
 finish();  
 }  
 }).setNegativeButton("Отмена", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
  
 dialog.cancel();  
  
 }  
 });  
 builder.show();  
 }  
  
 public void setAllData(int position) {  
 clearAllData();  
 Question question1 = ques.get(position - 1);  
 questionNumber.setText(String.*valueOf*(question1.getId()));  
 question.setText(question1.getQuestion());  
 aText.setText(question1.getOpt\_A());  
 bText.setText(question1.getOpt\_B());  
 switch (question1.getOptionCount()) {  
 case 2:  
 resetOptionCount();  
 break;  
 case 3:  
 increaseOptionCount(); // до 3  
 cText.setText(question1.getOpt\_C());  
 break;  
 case 4:  
 increaseOptionCount(); // до 3  
 increaseOptionCount(); // до 4  
 cText.setText(question1.getOpt\_C());  
 dText.setText(question1.getOpt\_D());  
 break;  
 }  
 switch (question1.getAnswer()) {  
 case "A":  
 aRadio.setChecked(true);  
 break;  
 case "B":  
 bRadio.setChecked(true);  
 break;  
 case "C":  
 cRadio.setChecked(true);  
 break;  
 case "D":  
 dRadio.setChecked(true);  
 break;  
 }  
  
 // Загружаем изображение вопроса, если оно есть  
 if (question1.getImageUrl() != null) {  
 questionImageView.setVisibility(View.*VISIBLE*);  
 Glide.*with*(this).load(question1.getImageUrl()).into(questionImageView);  
 } else {  
 questionImageView.setVisibility(View.*GONE*);  
 }  
 }  
  
 private void clearAllData() {  
 aRadio.setChecked(false);  
 bRadio.setChecked(false);  
 cRadio.setChecked(false);  
 dRadio.setChecked(false);  
 aText.setText(null);  
 bText.setText(null);  
 cText.setText(null);  
 dText.setText(null);  
 question.setText(null);  
 selectedOption = "";  
  
 selectedImageUri = null;  
 questionImageView.setImageDrawable(null);  
 questionImageView.setVisibility(View.GONE);  
  
 resetOptionCount();  
 }  
  
 private boolean getEnteredQuestionsValue() {  
  
 boolean cont = false;  
 if (TextUtils.isEmpty(question.getText().toString().trim())) {  
 question.setError("Пожалуйста, заполните вопрос");  
 } else if (TextUtils.isEmpty(aText.getText().toString().trim())) {  
 aText.setError("Пожалуйста, заполните вариант A");  
 } else if (TextUtils.isEmpty(bText.getText().toString().trim())) {  
 bText.setError("Пожалуйста, заполните вариант B");  
 } else if (optionCount >= 3 && TextUtils.isEmpty(cText.getText().toString().trim())) {  
 cText.setError("Пожалуйста, заполните вариант C");  
 } else if (optionCount == 4 && TextUtils.isEmpty(dText.getText().toString().trim())) {  
 dText.setError("Пожалуйста, заполните вариант D");  
 } else if (selectedOption.equals("")) {  
 Toast.makeText(this, "Пожалуйста, выберите правильный ответ", Toast.LENGTH\_SHORT).show();  
 } else {  
 // Создаем объект Question  
 final Question quest = new Question();  
 quest.setId(currentQuestion);  
 quest.setQuestion(question.getText().toString());  
 quest.setOpt\_A(aText.getText().toString());  
 quest.setOpt\_B(bText.getText().toString());  
 if (optionCount >= 3)  
 quest.setOpt\_C(cText.getText().toString());  
 if (optionCount == 4)  
 quest.setOpt\_D(dText.getText().toString());  
 quest.setOptionCount(optionCount);  
 quest.setAnswer(selectedOption);  
  
 if (selectedImageUri != null) {  
 // Если изображение выбрано, загружаем его в Firebase Storage  
 final ProgressDialog progressDialog = new ProgressDialog(this);  
 progressDialog.setTitle("Загрузка изображения...");  
 progressDialog.show();  
  
 StorageReference ref = storageReference.child("question\_images/" + System.currentTimeMillis());  
 ref.putFile(selectedImageUri)  
 .addOnSuccessListener(taskSnapshot -> {  
 progressDialog.dismiss();  
 // Получаем URL загруженного изображения  
 ref.getDownloadUrl().addOnSuccessListener(uri -> {  
 String imageUrl = uri.toString();  
 quest.setImageUrl(imageUrl);  
 // Добавляем вопрос в список и JSON  
 addQuestion(quest);  
 });  
 })  
 .addOnFailureListener(e -> {  
 progressDialog.dismiss();  
 Toast.makeText(Custom\_quiz.this, "Ошибка загрузки изображения", Toast.LENGTH\_SHORT).show();  
 })  
 .addOnProgressListener(taskSnapshot -> {  
 double progress = (100.0 \* taskSnapshot.getBytesTransferred() / taskSnapshot.getTotalByteCount());  
 progressDialog.setMessage("Загружено " + (int) progress + "%");  
 });  
 } else {  
 // Если изображение не выбрано, добавляем вопрос сразу  
 addQuestion(quest);  
 }  
 cont = true;  
 }  
 return cont;  
 }  
  
 private void addQuestion(Question quest) {  
 ques.add(quest);  
  
 JSONObject jsonObject = new JSONObject();  
 try {  
 jsonObject.put("answer", quest.getAnswer());  
 jsonObject.put("opt\_A", quest.getOpt\_A());  
 jsonObject.put("opt\_B", quest.getOpt\_B());  
 if (optionCount >= 3)  
 jsonObject.put("opt\_C", quest.getOpt\_C());  
 if (optionCount == 4)  
 jsonObject.put("opt\_D", quest.getOpt\_D());  
 jsonObject.put("question", quest.getQuestion());  
 jsonObject.put("optionCount", quest.getOptionCount());  
 if (quest.getImageUrl() != null)  
 jsonObject.put("imageUrl", quest.getImageUrl());  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
  
 jsonArray.put(jsonObject);  
 }  
  
 private void setListeners() {  
 aRadio.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 selectedOption = "A";  
 bRadio.setChecked(false);  
 cRadio.setChecked(false);  
 dRadio.setChecked(false);  
 }  
 });  
 bRadio.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 selectedOption = "B";  
 aRadio.setChecked(false);  
 cRadio.setChecked(false);  
 dRadio.setChecked(false);  
 }  
 });  
 cRadio.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 selectedOption = "C";  
 aRadio.setChecked(false);  
 bRadio.setChecked(false);  
 dRadio.setChecked(false);  
 }  
 });  
 dRadio.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 selectedOption = "D";  
 aRadio.setChecked(false);  
 bRadio.setChecked(false);  
 cRadio.setChecked(false);  
 }  
 });  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 // Обрабатываем результат выбора изображения  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == 1 && resultCode == RESULT\_OK && data != null && data.getData() != null) {  
 selectedImageUri = data.getData();  
 questionImageView.setImageURI(selectedImageUri);  
 questionImageView.setVisibility(View.VISIBLE);  
 }  
 }  
}

package com.example.myapplication.Misc\_Section;  
  
import android.content.Context;  
import android.graphics.Bitmap;  
import android.graphics.drawable.Drawable;  
import android.widget.ImageView;  
  
import androidx.core.graphics.drawable.RoundedBitmapDrawable;  
import androidx.core.graphics.drawable.RoundedBitmapDrawableFactory;  
  
import com.bumptech.glide.Glide;  
import com.bumptech.glide.load.engine.DiskCacheStrategy;  
import com.bumptech.glide.request.RequestListener;  
import com.bumptech.glide.request.RequestOptions;  
import com.bumptech.glide.request.target.BitmapImageViewTarget;  
  
  
public class ImageUtils {  
  
 // Prevent instantiation  
 private ImageUtils() {  
  
 }  
  
 */\*\*  
 \* Crops image into a circle that fits within the ImageView.  
 \*/* public static void displayRoundImageFromUrl(final Context context, final String url, final ImageView imageView) {  
 RequestOptions myOptions = new RequestOptions()  
 .centerCrop()  
 .dontAnimate();  
  
 Glide.*with*(context)  
 .asBitmap()  
 .apply(myOptions)  
 .load(url)  
 .diskCacheStrategy(DiskCacheStrategy.*DATA*)  
 .into(new BitmapImageViewTarget(imageView) {  
 @Override  
 protected void setResource(Bitmap resource) {  
 RoundedBitmapDrawable circularBitmapDrawable =  
 RoundedBitmapDrawableFactory.*create*(context.getResources(), resource);  
 circularBitmapDrawable.setCircular(true);  
 imageView.setImageDrawable(circularBitmapDrawable);  
 }  
 });  
 }  
  
 public static void displayImageFromUrl(final Context context, final String url,  
 final ImageView imageView) {  
 *displayImageFromUrl*(context, url, imageView, null, null);  
 }  
  
 public static void displayImageFromUrl(final Context context, final String url,  
 final ImageView imageView, Drawable placeholderDrawable) {  
 *displayImageFromUrl*(context, url, imageView, placeholderDrawable, null);  
 }  
  
 */\*\*  
 \* Displays an image from a URL in an ImageView.  
 \*/* public static void displayImageFromUrl(final Context context, final String url,  
 final ImageView imageView, Drawable placeholderDrawable, RequestListener listener) {  
 RequestOptions myOptions = new RequestOptions()  
 .dontAnimate()  
 .diskCacheStrategy(DiskCacheStrategy.AUTOMATIC)  
 .placeholder(placeholderDrawable);  
  
 if (listener != null) {  
 Glide.with(context)  
 .load(url)  
 .apply(myOptions)  
 .listener(listener)  
 .into(imageView);  
 } else {  
 Glide.with(context)  
 .load(url)  
 .apply(myOptions)  
 .listener(listener)  
 .into(imageView);  
 }  
 }  
  
 public static void displayRoundImageFromUrlWithoutCache(final Context context, final String url,  
 final ImageView imageView) {  
 displayRoundImageFromUrlWithoutCache(context, url, imageView, null);  
 }  
  
 public static void displayRoundImageFromUrlWithoutCache(final Context context, final String url,  
 final ImageView imageView, RequestListener listener) {  
 RequestOptions myOptions = new RequestOptions()  
 .centerCrop()  
 .dontAnimate()  
 .diskCacheStrategy(DiskCacheStrategy.NONE)  
 .skipMemoryCache(true);  
  
 if (listener != null) {  
 Glide.with(context)  
 .asBitmap()  
 .load(url)  
 .apply(myOptions)  
 .listener(listener)  
 .into(new BitmapImageViewTarget(imageView) {  
 @Override  
 protected void setResource(Bitmap resource) {  
 RoundedBitmapDrawable circularBitmapDrawable = RoundedBitmapDrawableFactory.create(context.getResources(), resource);  
 circularBitmapDrawable.setCircular(true);  
 imageView.setImageDrawable(circularBitmapDrawable);  
 }  
 });  
 } else {  
 Glide.with(context)  
 .asBitmap()  
 .load(url)  
 .apply(myOptions)  
 .into(new BitmapImageViewTarget(imageView) {  
 @Override  
 protected void setResource(Bitmap resource) {  
 RoundedBitmapDrawable circularBitmapDrawable = RoundedBitmapDrawableFactory.create(context.getResources(), resource);  
 circularBitmapDrawable.setCircular(true);  
 imageView.setImageDrawable(circularBitmapDrawable);  
 }  
 });  
 }  
 }  
  
 */\*\*  
 \* Displays an image from a URL in an ImageView.  
 \* If the image is loading or nonexistent, displays the specified placeholder image instead.  
 \*/* public static void displayImageFromUrlWithPlaceHolder(final Context context, final String url,  
 final ImageView imageView,  
 int placeholderResId) {  
 RequestOptions myOptions = new RequestOptions()  
 .dontAnimate()  
 .diskCacheStrategy(DiskCacheStrategy.AUTOMATIC)  
 .placeholder(placeholderResId);  
  
 Glide.with(context)  
 .load(url)  
 .apply(myOptions)  
 .into(imageView);  
 }  
  
 */\*\*  
 \* Displays an image from a URL in an ImageView.  
 \*/* public static void displayGifImageFromUrl(Context context, String url, ImageView imageView, Drawable placeholderDrawable, RequestListener listener) {  
 RequestOptions myOptions = new RequestOptions()  
 .dontAnimate()  
 .diskCacheStrategy(DiskCacheStrategy.AUTOMATIC)  
 .placeholder(placeholderDrawable);  
  
 if (listener != null) {  
 Glide.with(context)  
 .asGif()  
 .load(url)  
 .apply(myOptions)  
 .listener(listener)  
 .into(imageView);  
 } else {  
 Glide.with(context)  
 .asGif()  
 .load(url)  
 .apply(myOptions)  
 .into(imageView);  
 }  
 }  
  
 public static void displayGifImageFromUrl(Context context, String url, ImageView imageView, String thumbnailUrl) {  
 displayGifImageFromUrl(context, url, imageView, thumbnailUrl, null);  
 }  
  
 */\*\*  
 \* Displays an GIF image from a URL in an ImageView.  
 \*/* public static void displayGifImageFromUrl(Context context, String url, ImageView imageView, String thumbnailUrl, Drawable placeholderDrawable) {  
 RequestOptions myOptions = new RequestOptions()  
 .dontAnimate()  
 .diskCacheStrategy(DiskCacheStrategy.AUTOMATIC)  
 .placeholder(placeholderDrawable);  
  
 if (thumbnailUrl != null) {  
 Glide.with(context)  
 .asGif()  
 .load(url)  
 .apply(myOptions)  
 .thumbnail(Glide.with(context).asGif().load(thumbnailUrl))  
 .into(imageView);  
 } else {  
 Glide.with(context)  
 .asGif()  
 .load(url)  
 .apply(myOptions)  
 .into(imageView);  
 }  
 }  
}

package com.example.myapplication.Model;  
  
import java.io.Serializable;  
  
public class Question implements Serializable {  
  
 private int id;  
 private String question;  
 private String opt\_A;  
 private String opt\_B;  
 private String opt\_C;  
 private String opt\_D;  
 private String answer;  
 private int optionCount;  
 private String imageUrl; // Новое поле для URL изображения  
  
 public Question() {  
  
 }  
  
 public Question(int id, String question, String opt\_A, String opt\_B, String opt\_C, String opt\_D, String answer, int optionCount, String imageUrl) {  
 this.id = id;  
 this.question = question;  
 this.opt\_A = opt\_A;  
 this.opt\_B = opt\_B;  
 this.opt\_C = opt\_C;  
 this.opt\_D = opt\_D;  
 this.answer = answer;  
 this.optionCount = optionCount;  
 this.imageUrl = imageUrl;  
 }  
  
 // Геттеры и сеттеры для всех полей, включая новое поле imageUrl  
 public int getId() {  
 return id;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public String getOpt\_A() {  
 return opt\_A;  
 }  
  
 public String getOpt\_B() {  
 return opt\_B;  
 }  
  
 public String getOpt\_C() {  
 return opt\_C;  
 }  
  
 public String getOpt\_D() {  
 return opt\_D;  
 }  
  
 public String getAnswer() {  
 return answer;  
 }  
  
 public int getOptionCount() {  
 return optionCount;  
 }  
  
 public String getImageUrl() {  
 return imageUrl;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public void setQuestion(String question) {  
 this.question = question;  
 }  
  
 public void setOpt\_A(String opt\_A) {  
 this.opt\_A = opt\_A;  
 }  
  
 public void setOpt\_B(String opt\_B) {  
 this.opt\_B = opt\_B;  
 }  
  
 public void setOpt\_C(String opt\_C) {  
 this.opt\_C = opt\_C;  
 }  
  
 public void setOpt\_D(String opt\_D) {  
 this.opt\_D = opt\_D;  
 }  
  
 public void setAnswer(String answer) {  
 this.answer = answer;  
 }  
  
 public void setOptionCount(int optionCount) {  
 this.optionCount = optionCount;  
 }  
  
 public void setImageUrl(String imageUrl) {  
 this.imageUrl = imageUrl;  
 }  
}

package com.example.myapplication.Model;  
  
public class Result {  
 private String test, score;  
  
 public String getTest() {  
 return test;  
 }  
  
 public void setTest(String test) {  
 this.test = test;  
 }  
  
 public String getScore() {  
 return score;  
 }  
  
 public void setScore(String score) {  
 this.score = score;  
 }  
}

package com.example.myapplication.Model;  
  
import java.io.Serializable;  
import java.util.ArrayList;  
  
public class Test implements Serializable {  
 String Name;  
 ArrayList<Question> Questions;  
 Long time;  
  
 public Test() {  
 }  
  
 public Long getTime() {  
 return time;  
 }  
  
 public void setTime(Long time) {  
 this.time = time; // time can be null  
 }  
  
 public String getName() {  
 return Name;  
 }  
  
 public void setName(String name) {  
 Name = name;  
 }  
  
 public ArrayList<Question> getQuestions() {  
 return Questions;  
 }  
  
 public void setQuestions(ArrayList<Question> questions) {  
 Questions = questions;  
 }  
}

package com.example.myapplication.Model;  
  
public class User {  
 public String name;  
  
 public User() {  
 // Default constructor required for calls to DataSnapshot.getValue(User.class)  
 }  
  
 public User(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

package com.example.myapplication.Results\_section;  
  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.Bundle;  
import android.util.DisplayMetrics;  
import android.view.MenuItem;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import com.example.myapplication.R;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.Objects;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class GetDetailReport extends AppCompatActivity {  
  
 CircleImageView circleImageView;  
 TextView textView1;  
 TextView textView2;  
 TextView textView3;  
 TextView textView4;  
 TextView textView5;  
 private FirebaseDatabase database;  
 private FirebaseAuth auth;  
 private DatabaseReference myRef;  
 final long ONE\_MEGABYTE = 1024 \* 1024;  
  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.complete\_student\_details);  
 Toolbar toolbar = findViewById(R.id.toolbartst);  
 setSupportActionBar(toolbar);  
 toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));  
 Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);  
 circleImageView = findViewById(R.id.USERIMAGE);  
 textView1 = findViewById(R.id.StudentID);  
 textView2 = findViewById(R.id.SemID);  
 textView3 = findViewById(R.id.Branchid);  
 textView4 = findViewById(R.id.SectionID);  
 textView5 = findViewById(R.id.this\_Section);  
 String temp = getIntent().getStringExtra("USERID");  
  
  
  
 textView1.setText(getIntent().getStringExtra("DetailID"));  
 textView2.setText(getIntent().getStringExtra("DetailBranch"));  
 textView3.setText(getIntent().getStringExtra("DetailSem"));  
 textView4.setText(getIntent().getStringExtra("DetailSec"));  
 String Temp = getIntent().getStringExtra("TestNAME") + " results is: " +  
 getIntent().getStringExtra("Marks");  
 textView5.setText(Temp);  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
  
 if(item.getItemId() == android.R.id.home) {  
 finish();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
}

package com.example.myapplication.Results\_section;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import com.example.myapplication.R;  
import com.google.firebase.auth.FirebaseAuth;  
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.ValueEventListener;  
import java.util.ArrayList;  
import java.util.Objects;  
  
public class ResultsAdmin extends AppCompatActivity {  
  
 private FirebaseDatabase database;  
 private FirebaseAuth auth;  
 private DatabaseReference myRef;  
 private ProgressBar progressBar;  
 private ListView listView;  
 private TestAdapter testAdapter;  
 ArrayList<String> result = new ArrayList<>();  
 private int lastPos = -1;  
 private boolean isAdmin = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 auth = FirebaseAuth.getInstance();  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_tests);  
 isAdmin = getIntent().getBooleanExtra("ISADMIN", false);  
 if (!isAdmin)  
 setTitle("Results");  
 Toolbar toolbar = findViewById(R.id.toolbar);  
 toolbar.setTitleTextColor(getResources().getColor(android.R.color.black));  
 setSupportActionBar(toolbar);  
 Objects.requireNonNull(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);  
 progressBar = findViewById(R.id.loader1);  
 progressBar.setVisibility(View.VISIBLE);  
 database = FirebaseDatabase.getInstance();  
 myRef = database.getReference();  
 listView = findViewById(R.id.test\_listview);  
 testAdapter = new TestAdapter(ResultsAdmin.this, result);  
 listView.setAdapter(testAdapter);  
 getResults();  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 if (item.getItemId() == android.R.id.home) {  
 finish();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 public void getResults() {  
 if (isAdmin) {  
 myRef.child("Results").addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 result.clear();  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 result.add(snapshot.getKey());  
 }  
 testAdapter.dataList = result;  
 testAdapter.notifyDataSetChanged();  
 progressBar.setVisibility(View.GONE);  
 Log.e("The read success: ", "su" + result.size());  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 progressBar.setVisibility(View.GONE);  
 Log.e("The read failed: ", databaseError.getMessage());  
 }  
 });  
 } else {  
 myRef.child("Results").addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 result.clear();  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 if (snapshot.hasChild(Objects.requireNonNull(auth.getUid())))  
 result.add(snapshot.getKey());  
 }  
 testAdapter.dataList = result;  
 testAdapter.notifyDataSetChanged();  
 progressBar.setVisibility(View.GONE);  
 Log.e("The read success: ", "su" + result.size());  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 progressBar.setVisibility(View.GONE);  
 Log.e("The read failed: ", databaseError.getMessage());  
 }  
 });  
 }  
 }  
  
 class TestAdapter extends ArrayAdapter<String> {  
  
 private Context mContext;  
 ArrayList<String> dataList;  
  
 public TestAdapter(Context context, ArrayList<String> list) {  
 super(context, 0, list);  
 mContext = context;  
 dataList = list;  
 }  
  
 @NonNull  
 @Override  
 public View getView(final int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 View listItem = convertView;  
 if (listItem == null)  
 listItem = LayoutInflater.from(mContext).inflate(R.layout.test\_item, parent, false);  
 ((ImageView) listItem.findViewById(R.id.item\_imageView))  
 .setImageDrawable(mContext.getDrawable(R.drawable.ranking));  
 ((ImageView) listItem.findViewById(R.id.item\_imageView)).setPadding(10, 0, 0, 0);  
 ((TextView) listItem.findViewById(R.id.item\_textView)).setText(dataList.get(position));  
 ((Button) listItem.findViewById(R.id.item\_button)).setText("View");  
  
 Animation animation = AnimationUtils.loadAnimation(getContext(),  
 (position > lastPos) ? R.anim.up\_from\_bottom : R.anim.down\_from\_top);  
 listItem.startAnimation(animation);  
 lastPos = position;  
 ((Button) listItem.findViewById(R.id.item\_button)).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(ResultsAdmin.this, ResultsAdminDetailed.class);  
 intent.putExtra("test", dataList.get(position));  
 intent.putExtra("ISAdmin", isAdmin);  
 startActivity(intent);  
 }  
 });  
 return listItem;  
 }  
 }  
}

package com.example.myapplication.Results\_section;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
import androidx.cardview.widget.CardView;  
import androidx.core.content.ContextCompat;  
  
import android.os.Environment;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.myapplication.Model.User;  
import com.example.myapplication.R;  
import com.google.firebase.auth.FirebaseAuth;  
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.ValueEventListener;  
  
import org.apache.poi.hssf.usermodel.HSSFWorkbook;  
import org.apache.poi.ss.usermodel.Row;  
import org.apache.poi.ss.usermodel.Sheet;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.Objects;  
  
public class ResultsAdminDetailed extends AppCompatActivity {  
  
 private DatabaseReference myRef;  
 private ProgressBar progressBar;  
 private CardView button;  
 private ResultsAdminDetailed.TestAdapter testAdapter;  
 ArrayList<TestResults> result = new ArrayList<>();  
 private String testName;  
 private int lastPos = -1;  
 public boolean isAdmin = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 FirebaseAuth auth = FirebaseAuth.*getInstance*();  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_tests*);  
 button = findViewById(R.id.*download\_card*);  
 isAdmin = getIntent().getBooleanExtra("ISAdmin", false);  
 testName = getIntent().getStringExtra("test");  
  
 if (!isAdmin) {  
 setTitle("Result");  
 }  
  
 if (isAdmin) {  
 button.setVisibility(View.*VISIBLE*);  
 }  
  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 toolbar.setTitleTextColor(getResources().getColor(android.R.color.*black*));  
 setSupportActionBar(toolbar);  
 Objects.*requireNonNull*(getSupportActionBar()).setDisplayHomeAsUpEnabled(true);  
  
 progressBar = findViewById(R.id.*loader1*);  
 progressBar.setVisibility(View.*VISIBLE*);  
  
 FirebaseDatabase database = FirebaseDatabase.*getInstance*();  
 myRef = database.getReference();  
  
 ListView listView = findViewById(R.id.*test\_listview*);  
 testAdapter = new ResultsAdminDetailed.TestAdapter(ResultsAdminDetailed.this, result);  
 listView.setAdapter(testAdapter);  
 getSupportActionBar().setTitle(testName);  
  
 getResults();  
  
 button.setOnClickListener(v -> {  
 // check if available and not read only  
 if (!isExternalStorageAvailable() || isExternalStorageReadOnly()) {  
 Log.*e*("TAG", "Storage not available or read only");  
 return;  
 }  
 saveToExcel(testName.concat(".xls"));  
 });  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 if (item.getItemId() == android.R.id.*home*) {  
 finish();  
 }  
 return super.onOptionsItemSelected(item);  
 }  
  
 public void saveToExcel(String fileName) {  
 getResults();  
 //New Workbook  
 HSSFWorkbook hSSFWorkbook = new HSSFWorkbook();  
 Sheet createSheet = hSSFWorkbook.createSheet(this.testName);  
 //create row  
 Row createRow = createSheet.createRow(0);  
 createRow.createCell(0).setCellValue("ID");  
 createRow.createCell(1).setCellValue("Semester");  
 createRow.createCell(2).setCellValue("Branch");  
 createRow.createCell(3).setCellValue("Section");  
 createRow.createCell(4).setCellValue("Score");  
  
 int i = 0;  
 while (i < this.testAdapter.dataList.size()) {  
 int i2 = i + 1;  
 Row createRow2 = createSheet.createRow(i2);  
 createRow2.createCell(0).setCellValue(this.testAdapter.dataList.get(i).user.name);  
 createRow2.createCell(1).setCellValue(this.testAdapter.dataList.get(i).score);  
 i = i2;  
 }  
  
 //save the file under download folder of android  
 File file = new File(Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_DOWNLOADS*),fileName);  
  
 try {  
 FileOutputStream fileOutputStream = new FileOutputStream(file);  
 hSSFWorkbook.write(fileOutputStream);  
 StringBuilder sb = new StringBuilder();  
 sb.append("Writing file");  
 sb.append(fileName);  
 Log.*w*("FileSaver",sb.toString());  
 fileOutputStream.close();  
 Toast.*makeText*(ResultsAdminDetailed.this, "File is saved under Download folder", Toast.*LENGTH\_SHORT*).show();  
 } catch (Exception e) {  
 Toast.*makeText*(ResultsAdminDetailed.this, "Can't be saved", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 }  
 }  
  
 public boolean isExternalStorageReadOnly() {  
 return "mounted\_ro".equals(Environment.*getExternalStorageState*());  
 }  
  
 public boolean isExternalStorageAvailable() {  
 return "mounted".equals(Environment.*getExternalStorageState*());  
 }  
  
 public void getResults() {  
 myRef.child("Results").child(testName).addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 result.clear();  
  
 for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 TestResults t = new TestResults();  
 t.userID = snapshot.getKey();  
 t.score = Objects.*requireNonNull*(snapshot.getValue()).toString();  
 result.add(t);  
 }  
  
 Collections.*sort*(result, (o1, o2) -> o2.score.compareTo(o1.score));  
  
 getDetails();  
 Log.*e*("The read success: ", "su" + result.size());  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 progressBar.setVisibility(View.*GONE*);  
 Log.*e*("The read failed: ", databaseError.getMessage());  
 }  
 });  
 }  
  
 private void getDetails() {  
 myRef.child("users").addListenerForSingleValueEvent(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 for (int i = 0; i < result.size(); i++) {  
 if (dataSnapshot.child(result.get(i).userID).exists())  
 result.get(i).user = dataSnapshot.child(result.get(i).userID).getValue(User.class);  
 else {  
 User user1 = new User();  
 user1.setName("Unknown");  
 result.get(i).user = user1;  
 }  
 }  
  
 testAdapter.dataList = result;  
 testAdapter.notifyDataSetChanged();  
 progressBar.setVisibility(View.*GONE*);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 progressBar.setVisibility(View.*GONE*);  
 }  
 });  
 }  
  
 class TestResults {  
 public String userID, score;  
 public User user;  
 }  
  
 class TestAdapter extends ArrayAdapter<TestResults> {  
 private Context mContext;  
 ArrayList<TestResults> dataList;  
  
 public TestAdapter(Context context, ArrayList<TestResults> list) {  
 super(context, 0, list);  
 mContext = context;  
 dataList = list;  
 }  
  
 @NonNull  
 @Override  
 public View getView(final int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 View listItem = convertView;  
 if (listItem == null)  
 listItem = LayoutInflater.*from*(mContext).inflate(R.layout.*test\_item*, parent, false);  
  
 ((ImageView) listItem.findViewById(R.id.*item\_imageView*))  
 .setImageDrawable(ContextCompat.*getDrawable*(mContext, R.drawable.*ranking*));  
  
 if (dataList.get(position).user != null)  
 ((TextView) listItem.findViewById(R.id.*item\_textView*)).setText(dataList.get(position).user.name);  
 else {  
 ((TextView) listItem.findViewById(R.id.*item\_textView*)).setText("Details not added yet");  
 }  
  
 Animation animation = AnimationUtils.*loadAnimation*(getContext(), (position > lastPos) ? R.anim.*up\_from\_bottom* : R.anim.*down\_from\_top*);  
 (listItem).startAnimation(animation);  
 lastPos = position;  
  
 if (isAdmin) {  
 ((TextView) listItem.findViewById(R.id.*item\_textView*)).setOnClickListener(v -> {  
 if (dataList.get(position).user.name != null) {  
 Intent intent = new Intent(ResultsAdminDetailed.this, GetDetailReport.class);  
 intent.putExtra("USERID", dataList.get(position).userID);  
 intent.putExtra("DetailID", dataList.get(position).user.name);  
 intent.putExtra("TestNAME", testName);  
 intent.putExtra("Marks", dataList.get(position).score);  
 startActivity(intent);  
 }  
 });  
 }  
  
 ((Button) listItem.findViewById(R.id.*item\_button*)).setText(dataList.get(position).score);  
 return listItem;  
 }  
 }  
}

package com.example.myapplication.View;  
  
import android.app.Dialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.graphics.drawable.ColorDrawable;  
import android.net.ConnectivityManager;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.ActionBarDrawerToggle;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import androidx.core.view.GravityCompat;  
import androidx.drawerlayout.widget.DrawerLayout;  
  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.Window;  
  
import android.widget.ImageView;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.myapplication.Attempt\_Quiz\_Section.Tests;  
import com.example.myapplication.Create\_Quiz.create\_quiz\_main;  
import com.example.myapplication.Model.User;  
import com.example.myapplication.R;  
import com.example.myapplication.Auth\_Controller.ResetPasswordActivity;  
import com.example.myapplication.Auth\_Controller.LoginActivity;  
import com.example.myapplication.Results\_section.ResultsAdmin;  
import com.example.myapplication.AdminSection.ManageUsersActivity; // Import the ManageUsersActivity class  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.google.android.material.navigation.NavigationView;  
  
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.ValueEventListener;  
  
import java.util.Objects;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {  
  
 private FirebaseDatabase database;  
 private FirebaseAuth auth;  
 private DatabaseReference myRef;  
 private TextView USer\_email;  
 private boolean isAdmin = false;  
 private TextView userID;  
 private CircleImageView imageView;  
 private CircleImageView imageView1;  
 private FloatingActionButton floatingActionButton;  
 private NavigationView navigationView; // Declare navigationView as a class member variable  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*); // Make sure you have the correct layout  
  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 imageView = findViewById(R.id.*card1*);  
 floatingActionButton = findViewById(R.id.*chatHead*);  
 auth = FirebaseAuth.*getInstance*();  
  
 database = FirebaseDatabase.*getInstance*();  
 myRef = database.getReference();  
  
 // Initialize navigationView before calling checkForAdmin()  
 navigationView = findViewById(R.id.*nav\_view*);  
  
 checkForAdmin(); // Check for admin privileges after initializing navigationView  
  
 DrawerLayout drawer = findViewById(R.id.*drawer\_layout*);  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(  
 this, drawer, toolbar,  
 R.string.*navigation\_drawer\_open*,  
 R.string.*navigation\_drawer\_close*);  
 drawer.addDrawerListener(toggle);  
 toggle.syncState();  
  
 // Set up navigation header  
 View header = navigationView.getHeaderView(0);  
 imageView1 = header.findViewById(R.id.*imageView*);  
 USer\_email = header.findViewById(R.id.*text\_user\_name*);  
 setTextOnUser();  
 navigationView.setNavigationItemSelectedListener(this);  
 userID = findViewById(R.id.*text\_user\_card*);  
 setUserEmail();  
  
 floatingActionButton.setOnClickListener(v -> {  
 if (isAdmin) {  
 startActivity(new Intent(MainActivity.this, create\_quiz\_main.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 } else {  
 Toast.*makeText*(MainActivity.this, "You are not an admin", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 public void checkForAdmin() {  
 myRef.child("admins").addListenerForSingleValueEvent(  
 new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.child(Objects.*requireNonNull*(auth.getUid()))  
 .exists() &&  
 Objects.*requireNonNull*(dataSnapshot.child(auth.getUid())  
 .getValue()).toString().equals("true")) {  
 isAdmin = true;  
 Toast.*makeText*(getApplicationContext(),  
 "Hello Admin",  
 Toast.*LENGTH\_LONG*).show();  
 }  
  
 // Обновляем видимость пункта меню "Управление пользователями"  
 Menu nav\_Menu = navigationView.getMenu();  
 nav\_Menu.findItem(R.id.*nav\_manage\_users*).setVisible(isAdmin);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 // Handle possible errors.  
 }  
 });  
 }  
  
 @Override  
 public void onBackPressed() {  
 DrawerLayout drawer = findViewById(  
 R.id.*drawer\_layout*);  
 if (drawer.isDrawerOpen(  
 GravityCompat.*START*)) {  
 drawer.closeDrawer(  
 GravityCompat.*START*);  
 } else {  
 super.onBackPressed();  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // Inflate the menu if you want to add items to the action bar  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 // Handle action bar item clicks here  
 return super.onOptionsItemSelected(item);  
 }  
  
 @SuppressWarnings("StatementWithEmptyBody")  
 @Override  
 public boolean onNavigationItemSelected(MenuItem item) {  
 // Handle navigation view item clicks here  
  
 int id = item.getItemId();  
  
 if (id == R.id.*nav\_test*) {  
 if (isNetworkAvailable(MainActivity.this)) {  
 startActivity(new Intent(MainActivity.this, Tests.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 } else {  
 alertNoConnection();  
 }  
 } else if (id == R.id.*nav\_result*) {  
 if (isNetworkAvailable(MainActivity.this)) {  
 Intent intent = new Intent(MainActivity.this, ResultsAdmin.class);  
 intent.putExtra("ISADMIN", isAdmin);  
 startActivity(intent);  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 } else {  
 alertNoConnection();  
 }  
 } else if (id == R.id.*nav\_manage\_users*) { // Handle the new menu item  
 if (isAdmin && isNetworkAvailable(MainActivity.this)) {  
 startActivity(new Intent(MainActivity.this, ManageUsersActivity.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 } else if (isNetworkAvailable(MainActivity.this)) {  
 Toast.*makeText*(getApplicationContext(), "You are not an Admin!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 alertNoConnection();  
 }  
 } else if (id == R.id.*create\_test*) {  
 if (isAdmin && isNetworkAvailable(MainActivity.this)) {  
 startActivity(new Intent(MainActivity.this, create\_quiz\_main.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 } else if (isNetworkAvailable(MainActivity.this)) {  
 Toast.*makeText*(getApplicationContext(), "You are not an Admin!", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 alertNoConnection();  
 }  
 } else if (id == R.id.*nav\_respass*) {  
 if (isNetworkAvailable(MainActivity.this)) {  
 startActivity(new Intent(MainActivity.this, ResetPasswordActivity.class));  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 } else {  
 alertNoConnection();  
 }  
 } else if (id == R.id.*nav\_signout*) {  
 auth.signOut();  
 startActivity(new Intent(MainActivity.this, LoginActivity.class));  
 finish();  
 overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);  
 }  
  
 DrawerLayout drawer = findViewById(  
 R.id.*drawer\_layout*);  
 drawer.closeDrawer(  
 GravityCompat.*START*);  
 return true;  
 }  
  
 public void setUserEmail() {  
 DatabaseReference mDatabase;  
 FirebaseAuth auth;  
 mDatabase = FirebaseDatabase.*getInstance*().getReference();  
 auth = FirebaseAuth.*getInstance*();  
 mDatabase.child("users").child(Objects.*requireNonNull*(auth.getUid()))  
 .addValueEventListener(  
 new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 if (dataSnapshot.exists()) {  
 User user = dataSnapshot.getValue(User.class);  
 String temp = "Hey " + user.name + " what's up!";  
 userID.setText(temp);  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 // Handle possible errors.  
 }  
 });  
 }  
  
 public void setTextOnUser() {  
 FirebaseUser usero = FirebaseAuth.*getInstance*().getCurrentUser();  
 USer\_email.setText(Objects.*requireNonNull*(usero).getEmail());  
 }  
  
 public boolean isNetworkAvailable(Context context) {  
 ConnectivityManager connectivityManager = ((ConnectivityManager) context  
 .getSystemService(Context.*CONNECTIVITY\_SERVICE*));  
  
 return connectivityManager.getActiveNetworkInfo() != null &&  
 connectivityManager.getActiveNetworkInfo().isConnected();  
 }  
  
 public void alertNoConnection() {  
 Dialog builder = new Dialog(this);  
 builder.requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 if (builder.getWindow() != null) {  
 builder.getWindow().setBackgroundDrawable(  
 new ColorDrawable(android.graphics.Color.*TRANSPARENT*));  
 }  
  
 builder.setOnDismissListener(dialogInterface -> {  
 // Do nothing  
 });  
  
 ImageView imageView = new ImageView(this);  
 imageView.setImageResource(R.drawable.*nowifi*);  
 builder.addContentView(imageView, new RelativeLayout.LayoutParams(  
 400,  
 400));  
 builder.show();  
 }  
  
}

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/Theme.MyApplication.AppBarOverlay">  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?android:attr/actionBarSize"  
 android:background="?android:attr/colorPrimary"  
 app:popupTheme="@style/Theme.MyApplication.PopupOverlay" />  
 </com.google.android.material.appbar.AppBarLayout>  
  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <androidx.viewpager2.widget.ViewPager2  
 android:id="@+id/discrete"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_above="@id/linearLayout"/>  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:padding="5dp"  
 android:orientation="horizontal">  
 <Button  
 style="?android:attr/buttonBarButtonStyle"  
 android:layout\_marginEnd="5dp"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:textColor="#ffff"  
 android:layout\_weight="1"  
 android:id="@+id/prev"  
 android:background="@drawable/circular\_coloured\_button"/>  
  
 <Button  
 style="?android:attr/buttonBarButtonStyle"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:textColor="#ffff"  
 android:layout\_weight="1"  
 android:id="@+id/next"  
 android:background="@drawable/circular\_coloured\_button"/>  
 </LinearLayout>  
 </RelativeLayout>  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="top|center\_horizontal"  
 android:gravity="center\_horizontal"  
 android:id="@+id/index\_layout"  
 android:layout\_marginTop="?android:attr/actionBarSize"  
 android:orientation="vertical"  
 android:visibility="gone">  
 <GridView  
 android:id="@+id/pop\_grid"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:numColumns="4">  
 </GridView>  
 </LinearLayout>  
 </FrameLayout>  
  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<androidx.coordinatorlayout.widget.CoordinatorLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/coordinatorLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <!-- Включаем основной контент -->  
 <include  
 android:id="@+id/content\_create\_quiz"  
 layout="@layout/content\_create\_quiz" />  
  
 <!-- Кнопка "Далее" -->  
 <RelativeLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="end|bottom">  
 <androidx.cardview.widget.CardView  
 android:id="@+id/nextfab"  
 android:layout\_width="120dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"  
 app:cardCornerRadius="5dp"  
 app:cardElevation="10dp"  
 app:cardBackgroundColor="@color/colorPrimaryDark">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:text="Далее"  
 android:textColor="@android:color/black"/>  
 </androidx.cardview.widget.CardView>  
 </RelativeLayout>  
  
 <!-- Кнопка "Предыдущий" -->  
 <RelativeLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="start|bottom">  
 <androidx.cardview.widget.CardView  
 android:id="@+id/pre\_card"  
 android:layout\_width="120dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"  
 android:visibility="invisible"  
 app:cardCornerRadius="5dp"  
 app:cardElevation="10dp"  
 app:cardBackgroundColor="@color/colorPrimaryDark">  
 <TextView  
 android:id="@+id/prevTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:text="Предыдущий"  
 android:textColor="@android:color/black"/>  
 </androidx.cardview.widget.CardView>  
 </RelativeLayout>  
  
 <!-- Кнопка "Сохранить и загрузить" -->  
 <RelativeLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="end">  
 <androidx.cardview.widget.CardView  
 android:id="@+id/fab2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"  
 app:cardCornerRadius="5dp"  
 app:cardElevation="10dp"  
 app:cardBackgroundColor="@color/colorPrimaryDark">  
 <TextView  
 android:id="@+id/saveTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:text="Сохранить"  
 android:textColor="@android:color/black"/>  
 </androidx.cardview.widget.CardView>  
 </RelativeLayout>  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView  
 android:layout\_height="match\_parent"  
 android:layout\_width="match\_parent"  
 xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:theme="@style/Theme.AppCompat.Light.NoActionBar"  
 android:layout\_height="match\_parent">  
  
 <!-- Add Toolbar -->  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />  
  
 <androidx.appcompat.widget.AppCompatTextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/Sign\_in\_login"  
 android:id="@+id/textview1"  
 android:fontFamily="@font/lato\_black"  
 android:textColor="#366ce0"  
 android:textSize="25sp"  
 android:layout\_marginStart="40dp"  
 android:gravity="center"  
 android:layout\_marginTop="180dp">  
 </androidx.appcompat.widget.AppCompatTextView>  
  
 <View  
 android:id="@+id/View1"  
 android:layout\_width="80dp"  
 android:layout\_height="7dp"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="10dp"  
 android:background="#366ce0"  
 android:elevation="10dp"  
 android:layout\_alignStart="@id/textview1"  
 android:layout\_below="@+id/textview1">  
 </View>  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/card2"  
 android:visibility="visible"  
 android:layout\_marginTop="30dp"  
 android:layout\_below="@id/View1"  
 android:padding="20dp"  
 app:cardUseCompatPadding="true"  
 app:cardElevation="7dp"  
 android:background="@drawable/circular\_card"  
 app:cardCornerRadius="7dp">  
  
 <androidx.appcompat.widget.LinearLayoutCompat  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <androidx.appcompat.widget.AppCompatTextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/e\_mail\_text\_view\_login"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_marginStart="20dp"  
 android:textSize="20sp"  
 android:textStyle="bold">  
 </androidx.appcompat.widget.AppCompatTextView>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <EditText  
 android:id="@+id/email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:fontFamily="@font/lato\_bold"  
 android:maxLines="1"  
 android:singleLine="true"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:hint="@string/hint\_email"  
 android:inputType="textEmailAddress"  
 android:textColor="@android:color/black"  
 android:textColorHint="@android:color/white" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <androidx.appcompat.widget.AppCompatTextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/e\_mail\_text\_view\_pass"  
 android:layout\_marginStart="20dp"  
 android:fontFamily="@font/lato\_black"  
 android:textSize="20sp"  
 android:textStyle="bold">  
 </androidx.appcompat.widget.AppCompatTextView>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <EditText  
 android:id="@+id/password"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:hint="@string/hint\_password"  
 android:fontFamily="@font/lato\_bold"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:inputType="textPassword"  
 android:textColor="@android:color/black"  
 android:textColorHint="@android:color/black" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <androidx.appcompat.widget.AppCompatCheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/radio1"  
 android:text="@string/remember\_me"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_marginStart="20dp"  
 android:textStyle="normal"  
 android:textSize="15sp"  
 android:textColor="#366ce0">  
 </androidx.appcompat.widget.AppCompatCheckBox>  
  
 <Button  
 android:id="@+id/btn\_reset\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="0dp"  
 android:layout\_marginEnd="10dp"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_gravity="end"  
 android:background="@null"  
 android:text="@string/btn\_forgot\_password"  
 android:textAllCaps="false"  
 android:textColor="@color/colorAccent" />  
  
 <Button  
 android:id="@+id/btn\_login"  
 android:layout\_width="150dp"  
 android:layout\_gravity="center"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:fontFamily="@font/lato\_black"  
 android:background="@drawable/button\_circular"  
 android:text="@string/btn\_login"  
 android:gravity="center"  
 android:elevation="2dp"  
 android:textColor="#ffff" />  
  
 <Button  
 android:id="@+id/btn\_signup"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dip"  
 android:background="@drawable/circular\_coloured\_button"  
 android:text="@string/btn\_link\_to\_register"  
 android:fontFamily="@font/lato\_black"  
 android:textAllCaps="false"  
 android:textColor="@color/fui\_linkColor"  
 android:textSize="15sp" />  
  
 </androidx.appcompat.widget.LinearLayoutCompat>  
  
 </androidx.cardview.widget.CardView>  
  
 <!-- Add ProgressBar -->  
 <ProgressBar  
 android:id="@+id/loader1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:visibility="gone" />  
  
 </RelativeLayout>  
</ScrollView>

<?xml version="1.0" encoding="utf-8"?>  
<androidx.drawerlayout.widget.DrawerLayout  
 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:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context=".View.MainActivity">  
  
 <include  
 layout="@layout/app\_bar\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <com.google.android.material.navigation.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_main"  
 app:menu="@menu/activity\_main\_drawer" />  
  
</androidx.drawerlayout.widget.DrawerLayout>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:theme="@style/Theme.AppCompat.Light.NoActionBar"  
 android:layout\_gravity="center"  
 android:background="#ffff"  
 android:fitsSystemWindows="true">  
  
 <LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:padding="@dimen/activity\_horizontal\_margin">  
  
 <ImageView  
 android:contentDescription="USER\_IMAGE"  
 android:layout\_width="@dimen/logo\_w\_h"  
 android:layout\_height="@dimen/logo\_w\_h"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginBottom="10dp"  
 android:src="@drawable/forget\_pass" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:fontFamily="@font/lato\_bolditalic"  
 android:padding="10dp"  
 android:text="@string/lbl\_forgot\_password"  
 android:textColor="@android:color/holo\_blue\_light"  
 android:textSize="20sp" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:gravity="center\_horizontal"  
 android:fontFamily="@font/lato\_black"  
 android:padding="@dimen/activity\_horizontal\_margin"  
 android:text="@string/forgot\_password\_msg"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <EditText  
 android:id="@+id/email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:layout\_marginTop="20dp"  
 android:focusable="true"  
 android:fontFamily="@font/lato\_light"  
 android:hint="@string/hint\_email"  
 android:inputType="textEmailAddress"  
 android:textColor="@android:color/black"  
 android:textColorHint="@android:color/white" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <!-- Login Button -->  
  
 <Button  
 android:id="@+id/btn\_reset\_password"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dip"  
 android:background="@drawable/button\_circular"  
 android:fontFamily="@font/lato\_black"  
 android:text="@string/btn\_reset\_password"  
 android:textColor="#ffff" />  
  
 <Button  
 android:id="@+id/btn\_back"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/button\_circular"  
 android:fontFamily="@font/lato\_black"  
 android:text="@string/btn\_back"  
 android:textColor="#ffff" />  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:layout\_gravity="center|bottom"  
 android:layout\_marginBottom="20dp"  
 android:visibility="gone">  
 </ProgressBar>  
  
 </LinearLayout>  
  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView  
 android:layout\_height="match\_parent"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent">  
  
 <RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:theme="@style/Theme.AppCompat.Light.NoActionBar"  
 android:orientation="vertical"  
 android:background="#ededed"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <androidx.appcompat.widget.AppCompatTextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/Sign\_in\_signup"  
 android:id="@+id/textview3"  
 android:fontFamily="@font/lato\_black"  
 android:textColor="#366ce0"  
 android:textSize="25sp"  
 android:layout\_marginStart="40dp"  
 android:gravity="center"  
 android:layout\_marginTop="180dp">  
 </androidx.appcompat.widget.AppCompatTextView>  
  
 <View  
 android:id="@+id/View2"  
 android:layout\_width="80dp"  
 android:layout\_height="7dp"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="10dp"  
 android:background="#366ce0"  
 android:elevation="10dp"  
 android:layout\_alignStart="@id/textview3"  
 android:layout\_below="@+id/textview3">  
 </View>  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/card3"  
 android:visibility="visible"  
 android:layout\_marginTop="30dp"  
 android:layout\_below="@id/View2"  
 android:padding="20dp"  
 app:cardUseCompatPadding="true"  
 app:cardElevation="7dp"  
 android:background="@drawable/circular\_card"  
 app:cardCornerRadius="7dp">  
  
 <androidx.appcompat.widget.LinearLayoutCompat  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <androidx.appcompat.widget.AppCompatTextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/e\_mail\_text\_view\_login"  
 android:layout\_marginStart="20dp"  
 android:fontFamily="@font/lato\_black"  
 android:textSize="20sp"  
 android:textStyle="bold">  
 </androidx.appcompat.widget.AppCompatTextView>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <EditText  
 android:id="@+id/email"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="@string/email"  
 android:inputType="textEmailAddress"  
 android:fontFamily="@font/lato\_bold"  
 android:maxLines="1"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:singleLine="true"  
 android:textColor="@android:color/black" />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <androidx.appcompat.widget.AppCompatTextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/e\_mail\_text\_view\_pass"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_marginStart="20dp"  
 android:textSize="20sp"  
 android:textStyle="bold">  
 </androidx.appcompat.widget.AppCompatTextView>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <EditText  
 android:id="@+id/password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/lato\_bold"  
 android:focusableInTouchMode="true"  
 android:hint="@string/hint\_password"  
 android:inputType="textPassword"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:maxLines="1"  
 android:singleLine="true"  
 android:textColor="@android:color/black" />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <androidx.appcompat.widget.AppCompatTextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Name"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_marginStart="20dp"  
 android:textSize="20sp"  
 android:textStyle="bold">  
 </androidx.appcompat.widget.AppCompatTextView>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <EditText  
 android:id="@+id/name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Name"  
 android:fontFamily="@font/lato\_bold"  
 android:maxLines="1"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:singleLine="true"  
 android:textColor="@android:color/black" />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/sign\_up\_button"  
 style="?android:textAppearanceSmall"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_marginTop="15dp"  
 android:background="@drawable/button\_circular"  
 android:text="Sign Up"  
 android:textColor="@android:color/black"  
 android:textStyle="bold" />  
  
 <Button  
 android:id="@+id/btn\_reset\_password"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dip"  
 android:fontFamily="@font/lato\_black"  
 android:background="@null"  
 android:text="Forgot Password?"  
 android:textAllCaps="false"  
 android:textColor="#f2fe71"  
 android:visibility="gone"/>  
  
 <Button  
 android:id="@+id/sign\_in\_button"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_gravity="center"  
 android:background="@drawable/circular\_coloured\_button"  
 android:text="Already have an account? Sign In"  
 android:textAllCaps="false"  
 android:padding="2dp"  
 android:textColor="@color/white"  
 android:textSize="12sp" />  
  
 </androidx.appcompat.widget.LinearLayoutCompat>  
  
 </androidx.cardview.widget.CardView>  
  
 <ProgressBar  
 android:id="@+id/loader1"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:layout\_gravity="center|bottom"  
 android:layout\_marginBottom="20dp"  
 android:visibility="gone">  
 </ProgressBar>  
  
 </RelativeLayout>  
  
</ScrollView>

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:id="@+id/appbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/Theme.MyApplication.AppBarOverlay">  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?android:attr/actionBarSize"  
 android:background="?android:attr/colorPrimary"  
 app:popupTheme="@style/Theme.MyApplication.PopupOverlay" />  
 </com.google.android.material.appbar.AppBarLayout>  
  
 <ProgressBar  
 android:id="@+id/loader1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:visibility="gone" />  
  
 <ListView  
 android:id="@+id/test\_listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@id/appbar"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginEnd="16dp"  
 android:divider="@android:color/darker\_gray"  
 android:dividerHeight="1dp" />  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/download\_card"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/test\_listview"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginEnd="16dp"  
 app:cardCornerRadius="8dp"  
 app:cardElevation="4dp">  
  
  
 </androidx.cardview.widget.CardView>  
  
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>  
<androidx.coordinatorlayout.widget.CoordinatorLayout  
 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=".View.MainActivity">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/Theme.MyApplication.AppBarOverlay">  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/Theme.MyApplication.PopupOverlay" />  
  
 </com.google.android.material.appbar.AppBarLayout>  
  
 <include layout="@layout/content\_main" />  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/chatHead"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom|end"  
 android:layout\_margin="@dimen/fab\_margin"  
 app:srcCompat="@android:drawable/ic\_dialog\_info" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:background="@android:color/white"  
 android:layout\_gravity="center"  
 android:orientation="vertical">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/Theme.MyApplication.AppBarOverlay"  
 android:id="@+id/tooltst">  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbartst"  
 app:navigationIcon="@drawable/ic\_keyboard\_arrow\_left\_black\_24dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?android:attr/actionBarSize"  
 android:background="?android:attr/colorPrimary"  
 app:popupTheme="@style/Theme.MyApplication.PopupOverlay">  
 </androidx.appcompat.widget.Toolbar>  
  
 </com.google.android.material.appbar.AppBarLayout>  
  
 <TableLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TableRow  
 android:id="@+id/tableRow1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:gravity="center" >  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="Student Details"  
 android:fontFamily="@font/lato\_black"  
 android:textColor="@color/bg\_main"  
 android:textSize="14sp"  
 android:textStyle="bold" />  
 </TableRow>  
 </TableLayout>  
  
 <TableLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TableRow  
 android:layout\_height="match\_parent"  
 android:layout\_width="wrap\_content"  
 android:paddingBottom="15dp"  
 android:paddingTop="15dp"  
 android:gravity="center">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/USERIMAGE"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:src="@drawable/user">  
  
 </de.hdodenhof.circleimageview.CircleImageView>  
 </TableRow>  
 </TableLayout>  
  
  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="20dp"  
 android:paddingEnd="20dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="@color/bg\_main"  
 android:text="StudentID"  
 android:fontFamily="@font/lato\_black"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/StudentID"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="@color/colorAccent"  
 android:fontFamily="@font/lato\_bold"  
 android:text="Details not added yet"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="20dp"  
 android:paddingEnd="20dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="@color/bg\_main"  
 android:fontFamily="@font/lato\_black"  
 android:text="Semester"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/SemID"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="@color/colorAccent"  
 android:fontFamily="@font/lato\_bold"  
 android:text="Details not added yet"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="20dp"  
 android:paddingEnd="20dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:fontFamily="@font/lato\_black"  
 android:background="@color/bg\_main"  
 android:text="Branch"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/Branchid"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="@color/colorAccent"  
 android:fontFamily="@font/lato\_bold"  
 android:text="Details not added yet"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
 </TableRow>  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingStart="20dp"  
 android:paddingEnd="20dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="@color/bg\_main"  
 android:fontFamily="@font/lato\_black"  
 android:text="Section"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/SectionID"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="@color/colorAccent"  
 android:fontFamily="@font/lato\_bold"  
 android:text="Details not added yet"  
 android:padding="5dp"  
 android:textSize="14sp"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
 </TableRow>  
  
 <TableLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" >  
  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:gravity="center" >  
  
 <TextView  
 android:id="@+id/this\_Section"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="This test marks is: "  
 android:fontFamily="@font/lato\_black"  
 android:textColor="@color/bg\_main"  
 android:textSize="14sp"  
 android:textStyle="bold" />  
 </TableRow>  
  
 </TableLayout>  
  
</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView  
 android:layout\_height="match\_parent"  
 android:layout\_width="match\_parent"  
 xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <RelativeLayout  
 android:id="@+id/mainLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <!-- Заголовок вопроса и номер -->  
 <LinearLayout  
 android:id="@+id/questionHeaderLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:orientation="horizontal"  
 android:paddingLeft="20dp"  
 android:paddingRight="20dp">  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Вопрос"  
 android:textSize="24sp"  
 android:textStyle="bold"/>  
 <TextView  
 android:id="@+id/questionNumber"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:text="1"  
 android:textSize="24sp"  
 android:textStyle="bold"/>  
 </LinearLayout>  
  
 <EditText  
 android:id="@+id/questionView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="150dp"  
 android:layout\_below="@id/questionHeaderLayout"  
 android:layout\_margin="20dp"  
 android:gravity="top|start"  
 android:hint="Введите ваш вопрос здесь"  
 android:textSize="16sp"/>  
  
 <!-- Кнопка для выбора изображения -->  
 <Button  
 android:id="@+id/selectImageButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Выбрать изображение"  
 android:layout\_below="@id/questionView"  
 android:layout\_marginTop="10dp"  
 android:layout\_centerHorizontal="true"/>  
  
 <!-- ImageView для отображения выбранного изображения -->  
 <ImageView  
 android:id="@+id/questionImageView"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:layout\_below="@id/selectImageButton"  
 android:layout\_marginTop="10dp"  
 android:layout\_centerHorizontal="true"  
 android:visibility="gone"/>  
  
 <!-- Группа вариантов (RadioGroup) -->  
 <RadioGroup  
 android:id="@+id/optionsGroup"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/questionImageView"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginRight="20dp"  
 android:orientation="vertical">  
  
 <!-- Вариант A -->  
 <LinearLayout  
 android:id="@+id/optionALayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <RadioButton  
 android:id="@+id/aRadio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="A"/>  
 <EditText  
 android:id="@+id/aText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:hint="Вариант A"/>  
 </LinearLayout>  
  
 <!-- Вариант B -->  
 <LinearLayout  
 android:id="@+id/optionBLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <RadioButton  
 android:id="@+id/bRadio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="B"/>  
 <EditText  
 android:id="@+id/bText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:hint="Вариант B"/>  
 </LinearLayout>  
  
 <!-- Вариант C -->  
 <LinearLayout  
 android:id="@+id/optionCLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:visibility="gone">  
 <RadioButton  
 android:id="@+id/cRadio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="C"/>  
 <EditText  
 android:id="@+id/cText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:hint="Вариант C"/>  
 </LinearLayout>  
  
 <!-- Вариант D -->  
 <LinearLayout  
 android:id="@+id/optionDLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:visibility="gone">  
 <RadioButton  
 android:id="@+id/dRadio"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="D"/>  
 <EditText  
 android:id="@+id/dText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
 android:hint="Вариант D"/>  
 </LinearLayout>  
 </RadioGroup>  
  
 <!-- Кнопки добавления/удаления вариантов -->  
 <LinearLayout  
 android:id="@+id/option\_buttons\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/optionsGroup"  
 android:layout\_marginTop="10dp"  
 android:orientation="horizontal"  
 android:gravity="center\_horizontal">  
 <Button  
 android:id="@+id/addOptionButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Добавить вариант" />  
 <Button  
 android:id="@+id/removeOptionButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:text="Удалить вариант" />  
 </LinearLayout>  
  
 </RelativeLayout>  
</ScrollView>

<?xml version="1.0" encoding="utf-8"?>  
<androidx.coordinatorlayout.widget.CoordinatorLayout  
 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"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 android:layout\_gravity="center"  
 android:orientation="vertical"  
 tools:context=".View.MainActivity"  
 tools:showIn="@layout/app\_bar\_main"  
 android:layout\_marginTop="?android:attr/actionBarSize">  
  
  
 <androidx.appcompat.widget.LinearLayoutCompat  
 android:id="@+id/li1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="288dp"  
 android:orientation="vertical">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="230dp"  
 android:layout\_gravity="center\_horizontal|center"  
 android:background="@drawable/circular\_card"  
 android:backgroundTint="@android:color/holo\_blue\_light">  
  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/card1"  
 android:layout\_width="180dp"  
 android:layout\_height="180dp"  
 android:layout\_gravity="center\_horizontal"  
 android:paddingTop="25dp"  
 android:scaleType="centerCrop"  
 android:src="@drawable/icon1">  
  
 </de.hdodenhof.circleimageview.CircleImageView>  
  
 </androidx.cardview.widget.CardView>  
  
 <TextView  
 android:id="@+id/text\_user\_card"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:fontFamily="@font/lato\_black"  
 android:padding="7dp"  
 android:gravity="center"  
 android:text="@string/Welcome"  
 android:textAllCaps="true"  
 android:textColor="@color/fui\_bgGitHub"  
 android:textSize="15sp" />  
  
  
 </androidx.appcompat.widget.LinearLayoutCompat>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_marginTop="50dp"  
 app:layout\_anchorGravity="bottom"  
 android:layout\_gravity="bottom"  
 android:visibility="gone"  
 app:layout\_anchor="@id/li1">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Recent Activities"  
 android:textColor="@android:color/holo\_red\_light"  
 android:fontFamily="@font/lato\_black"  
 android:textSize="20sp">  
  
 </TextView>  
  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="10"  
 android:orientation="vertical"  
 android:id="@+id/recent\_activity\_list"  
 android:divider="@null"  
 android:dividerHeight="0dp">  
  
 </androidx.recyclerview.widget.RecyclerView>  
 </LinearLayout>  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="40dp"  
 android:id="@+id/chatHead"  
 android:layout\_marginEnd="17dp"  
 android:backgroundTint="#ffff"  
 android:src="@drawable/chat"  
 app:icon="@drawable/chat"  
 android:background="@drawable/circular\_coloured\_button"  
 android:tint="@android:color/black"  
 android:scaleType="center"  
 app:fabSize="normal"  
 app:maxImageSize="56dp"  
 android:elevation="10dp"  
 android:layout\_gravity="bottom|end">  
  
 </com.google.android.material.floatingactionbutton.FloatingActionButton>  
  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

<?xml version="1.0" encoding="utf-8"?>  
<androidx.coordinatorlayout.widget.CoordinatorLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:fitsSystemWindows="true">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/Theme.MyApplication.AppBarOverlay"  
 android:id="@+id/tooltst">  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbartst"  
 app:navigationIcon="@drawable/ic\_keyboard\_arrow\_left\_black\_24dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?android:attr/actionBarSize"  
 android:background="?android:attr/colorPrimary"  
 app:popupTheme="@style/Theme.MyApplication.PopupOverlay">  
 </androidx.appcompat.widget.Toolbar>  
  
 </com.google.android.material.appbar.AppBarLayout>  
  
  
 <RelativeLayout  
 android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:layout\_gravity="end|bottom"  
 android:layout\_margin="20dp">  
  
 <Button  
 android:id="@+id/add\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Add"  
 android:textColor="@android:color/white"  
 android:background="@drawable/button\_circular"  
 android:elevation="4dp"  
 android:paddingHorizontal="20dp"  
 android:paddingVertical="10dp" />  
 </RelativeLayout>  
  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>

<?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"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="?android:attr/actionBarSize">  
  
  
 <ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/test\_listview"  
 android:divider="@android:color/transparent"  
 android:dividerHeight="0dp">  
  
 </ListView>  
  
 <RelativeLayout  
 android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:layout\_gravity="end|bottom">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:id="@+id/download\_card"  
 android:visibility="invisible"  
 app:cardElevation="10dp"  
 app:cardCornerRadius="5dp"  
 app:cardBackgroundColor="@color/colorPrimaryDark"  
 android:layout\_margin="20dp">  
  
 <TextView  
 android:layout\_height="wrap\_content"  
 android:layout\_width="wrap\_content"  
 android:layout\_margin="10dp"  
 android:text="Download"  
 android:textColor="@android:color/white"/>  
 </androidx.cardview.widget.CardView>  
 </RelativeLayout>  
  
  
  
</FrameLayout>

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/layout\_root"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical"  
 android:padding="10dp" >  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/lato\_black"  
 android:text="Save test "  
 android:textAppearance="?android:attr/textAppearanceLarge" />  
  
 <EditText  
 android:id="@+id/editTextDialogUserInput"  
 android:layout\_width="match\_parent"  
 android:fontFamily="@font/lato\_bold"  
 android:layout\_height="wrap\_content" >  
  
 <requestFocus />  
  
 </EditText>  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:hint="Test Name"  
 android:fontFamily="@font/lato\_black"  
 android:text="Specify Time"  
 android:textAppearance="?android:attr/textAppearanceLarge" />  
  
 <EditText  
 android:id="@+id/editTextDialogUserInput1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/lato\_bold"  
 android:inputType="number"  
 android:hint="Enter time in digit">  
  
 </EditText>  
  
</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingTop="@dimen/activity\_horizontal\_margin"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin">  
  
 <TextView  
 android:text="questions"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/questionTextView"  
 android:fontFamily="@font/lato\_black"  
 android:textStyle="bold"  
 android:textSize="28sp"  
 android:textColor="#eb346e"  
 android:paddingStart="10dp"  
 android:layout\_gravity="start"  
 android:paddingBottom="10dp"/>  
  
 <ImageView  
 android:id="@+id/questionImageView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:visibility="gone"  
 android:layout\_marginBottom="10dp"/>  
  
 <RadioGroup  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="start|center\_vertical"  
 android:layout\_marginStart="20dp"  
 android:scrollbarSize="20dp"  
 android:gravity="center"  
 android:id="@+id/radioGroup">  
  
 <RadioButton  
 android:text="Rb1"  
 android:textSize="20sp"  
 android:layout\_width="match\_parent"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/radioButton"  
 android:paddingBottom="10dp">  
 </RadioButton>  
 <RadioButton  
 android:textSize="20sp"  
 android:text="Rb2"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/radioButton2" />  
  
 <RadioButton  
 android:textSize="20sp"  
 android:text="Rb3"  
 android:layout\_width="match\_parent"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/radioButton3" />  
 <RadioButton  
 android:textSize="20sp"  
 android:text="Rb4"  
 android:layout\_width="match\_parent"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/radioButton4" />  
 <RadioButton  
 android:textSize="20sp"  
 android:text="Clear Selection"  
 android:layout\_width="match\_parent"  
 android:fontFamily="@font/lato\_black"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/radioButton5" />  
  
 </RadioGroup>  
</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/nav\_header\_height"  
 android:background="@drawable/side\_nav\_bar"  
 android:gravity="bottom"  
 android:orientation="vertical"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/imageView"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:contentDescription="@string/nav\_header\_desc"  
 android:paddingTop="@dimen/nav\_header\_vertical\_spacing"  
 android:src="@drawable/user" />  
  
 <TextView  
 android:id="@+id/text\_user\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="@dimen/nav\_header\_vertical\_spacing"  
 android:fontFamily="@font/lato\_regular"  
 android:text="UserName"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:fontFamily="@font/lato\_bold"  
 android:text="Email-Id" />  
  
</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <androidx.cardview.widget.CardView  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:cardCornerRadius="8dp"  
 app:cardElevation="6dp"  
 app:cardPreventCornerOverlap="false"  
 app:cardUseCompatPadding="true">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="88dp"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:background="@color/colorPrimary">  
 <ImageView  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:layout\_height="40dp"  
 android:src="@drawable/ic\_appicon"  
 android:id="@+id/item\_imageView"  
 android:gravity="center"/>  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_weight="4"  
 android:layout\_height="wrap\_content"  
 android:text="Quiz1"  
 style="@style/CardStyle"  
 android:id="@+id/item\_textView" />  
 <Button  
 android:layout\_margin="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:fontFamily="@font/lato\_black"  
 android:id="@+id/item\_button"  
 android:background="@drawable/button\_circular"  
 android:layout\_alignParentEnd="true"/>  
  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
  
</LinearLayout>