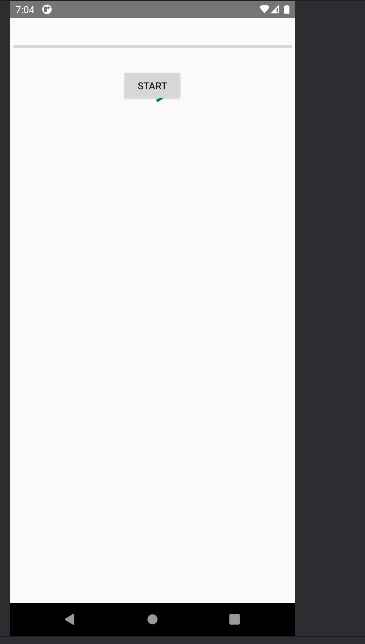
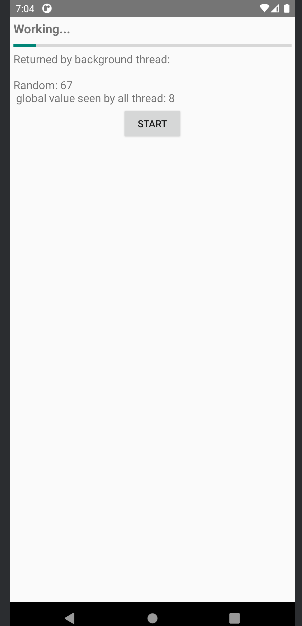
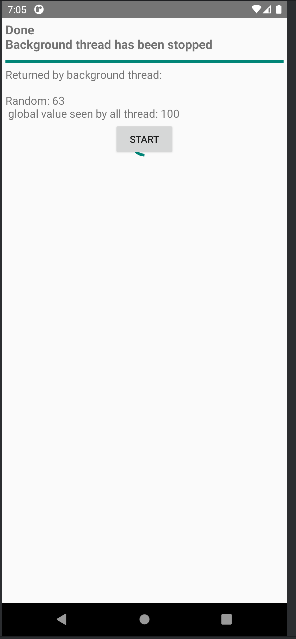
Bài 1:







Code MessageActivity.java

package com.example.lab6;

import android.os.Bundle;

import android.os.Handler;

import android.os.Looper;

import android.os.Message;

import android.widget.Button;

import android.widget.ProgressBar;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.util.Random;

public class MessageActivity extends AppCompatActivity {

private ProgressBar pbFirst, pbSecond;

private TextView tvMsgWorking, tvMsgReturned;

private Button btnStart;

private boolean isRunning;

private int intTest;

private Thread bgThread;

private Handler handler;

private static final int MSG\_UPDATE = 1;

private static final int MSG\_STOPPED = 2;

private static final int MAX\_PROGRESS = 100;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_message); // Gắn layout

// Tìm các View trong giao diện

pbFirst = findViewById(R.id.pb\_first);

pbSecond = findViewById(R.id.pb\_second);

tvMsgWorking = findViewById(R.id.tv\_working);

tvMsgReturned = findViewById(R.id.tv\_return);

btnStart = findViewById(R.id.btn\_start);

pbFirst.setMax(MAX\_PROGRESS);

pbSecond.setMax(MAX\_PROGRESS);

// Tạo Handler để nhận Message từ Thread nền

handler = new Handler(Looper.getMainLooper()) {

@Override

public void handleMessage(Message msg) {

if (msg.what == MSG\_UPDATE) {

int randomValue = (Integer) msg.obj;

intTest++;

pbFirst.setProgress(Math.min(intTest, MAX\_PROGRESS));

pbSecond.setProgress(Math.min(intTest, MAX\_PROGRESS));

tvMsgReturned.setText("Returned by background thread:\n\n" +

randomValue + "\nGlobal value seen by all thread: " + intTest);

} else if (msg.what == MSG\_STOPPED) {

tvMsgWorking.setText("Done\nBackground thread has been stopped");

isRunning = false;

}

}

};

// Khi nhấn nút Start

btnStart.setOnClickListener(v -> {

if (!isRunning) {

isRunning = true;

intTest = 0;

pbFirst.setProgress(0);

pbSecond.setProgress(0);

tvMsgWorking.setText("Working...");

startBackgroundThread();

}

});

}

// Hàm tạo Thread nền

private void startBackgroundThread() {

bgThread = new Thread(() -> {

Random random = new Random();

while (isRunning && intTest < MAX\_PROGRESS) {

int value = random.nextInt(101); // random 0–100

Message msg = handler.obtainMessage(MSG\_UPDATE, value);

handler.sendMessage(msg);

try {

Thread.sleep(500);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

handler.sendEmptyMessage(MSG\_STOPPED);

});

bgThread.start();

}

@Override

protected void onStop() {

super.onStop();

isRunning = false;

if (bgThread != null && bgThread.isAlive()) {

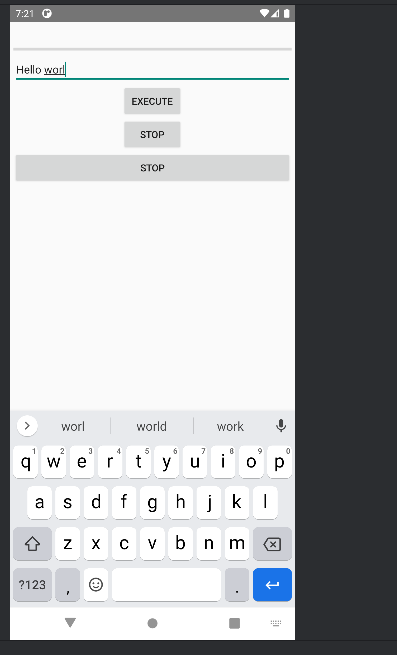
bgThread.interrupt();

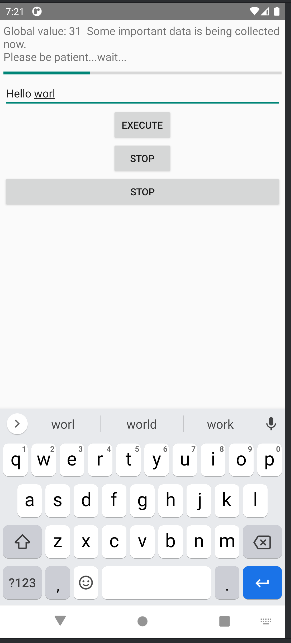
}

}

}

Bài 2:



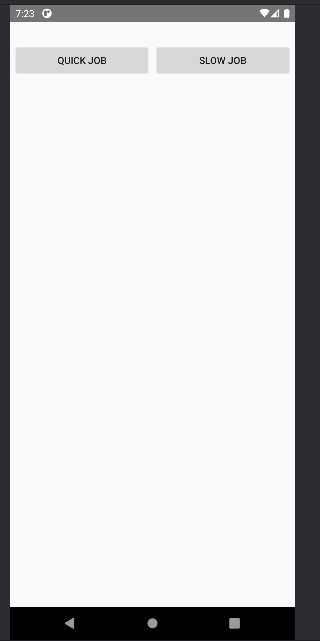


Code PostActivity.java :

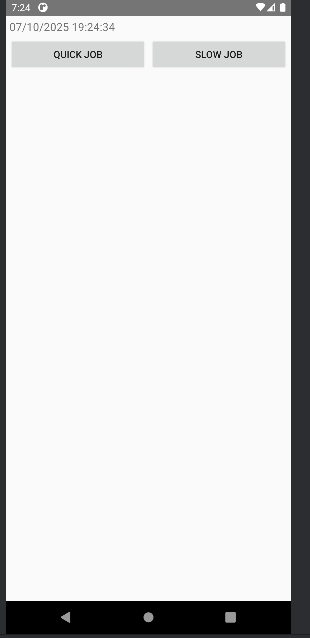
package com.example.lab6;  
  
import android.os.Bundle;  
import android.os.Handler;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class PostActivity extends AppCompatActivity {  
  
 private ProgressBar pbWaiting;  
 private TextView tvTopCaption;  
 private EditText etInput;  
 private Button btnExecute, btnStop;  
  
 private int globalValue, accum;  
 private long startTime;  
 private final String PATIENCE = "Some important data is being collected now.\nPlease be patient...wait...";  
 private Handler handler;  
 private Runnable fgRunnable, bgRunnable;  
 private Thread testThread;  
 private boolean isRunning = false; // Thêm biến kiểm soát luồng  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_post*);  
  
 findViewByIds();  
 initVariables();  
  
 // Khi nhấn Execute  
 btnExecute.setOnClickListener(v -> {  
 String text = etInput.getText().toString();  
 Toast.*makeText*(PostActivity.this, text, Toast.*LENGTH\_SHORT*).show();  
  
 if (!isRunning) {  
 isRunning = true;  
 testThread = new Thread(bgRunnable);  
 testThread.start();  
 }  
 });  
  
 // Khi nhấn Stop  
 btnStop.setOnClickListener(v -> {  
 if (isRunning) {  
 isRunning = false;  
 Toast.*makeText*(PostActivity.this, "Background work stopped!", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private void findViewByIds() {  
 tvTopCaption = findViewById(R.id.*tv\_top\_caption*);  
 pbWaiting = findViewById(R.id.*pb\_waiting*);  
 etInput = findViewById(R.id.*et\_input*);  
 btnExecute = findViewById(R.id.*btn\_execute*);  
  
 // 🔹 Thêm nút Stop mới  
 btnStop = new Button(this);  
 btnStop.setText("STOP");  
 ((android.widget.LinearLayout) pbWaiting.getParent()).addView(btnStop);  
 }  
  
 private void initVariables() {  
 globalValue = 0;  
 accum = 0;  
 startTime = System.*currentTimeMillis*();  
 handler = new Handler();  
  
 fgRunnable = () -> {  
 tvTopCaption.setText("Global value: " + globalValue + " " + PATIENCE);  
 pbWaiting.setProgress(globalValue % 100);  
 };  
  
 bgRunnable = () -> {  
 while (isRunning) {  
 globalValue++;  
 handler.post(fgRunnable);  
 try {  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 // Khi dừng thread  
 handler.post(() -> tvTopCaption.setText("Background work is over!"));  
 };  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 isRunning = false; // Dừng thread khi thoát activity  
 }  
}

Bài 3:

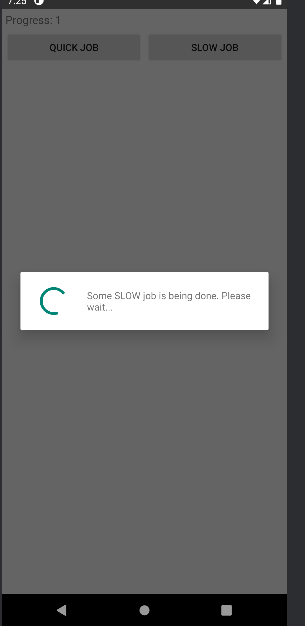
Menu

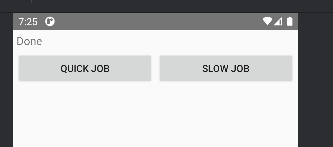


Quick job:



Slow Job





Code:

AsyncActivity.java

package com.example.lab6;  
  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public class AsyncActivity extends AppCompatActivity {  
  
 private Button btnQuickJob, btnSlowJob;  
 private TextView tvStatus;  
 private SlowTask slowTask;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_async*);  
 findViewByIds();  
  
 slowTask = new SlowTask(AsyncActivity.this, tvStatus);  
  
 btnQuickJob.setOnClickListener(v -> {  
 SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
 tvStatus.setText(sdf.format(new Date()));  
 });  
  
 btnSlowJob.setOnClickListener(v -> slowTask.execute());  
 }  
  
 private void findViewByIds() {  
 btnQuickJob = findViewById(R.id.*btn\_quick\_job*);  
 btnSlowJob = findViewById(R.id.*btn\_slow\_job*);  
 tvStatus = findViewById(R.id.*tv\_status*);  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 if (slowTask != null && !slowTask.isCancelled()) slowTask.cancel(true);  
 }  
}

SlowTask.java

package com.example.lab6;  
  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public class AsyncActivity extends AppCompatActivity {  
  
 private Button btnQuickJob, btnSlowJob;  
 private TextView tvStatus;  
 private SlowTask slowTask;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_async*);  
 findViewByIds();  
  
 slowTask = new SlowTask(AsyncActivity.this, tvStatus);  
  
 btnQuickJob.setOnClickListener(v -> {  
 SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
 tvStatus.setText(sdf.format(new Date()));  
 });  
  
 btnSlowJob.setOnClickListener(v -> slowTask.execute());  
 }  
  
 private void findViewByIds() {  
 btnQuickJob = findViewById(R.id.*btn\_quick\_job*);  
 btnSlowJob = findViewById(R.id.*btn\_slow\_job*);  
 tvStatus = findViewById(R.id.*tv\_status*);  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 if (slowTask != null && !slowTask.isCancelled()) slowTask.cancel(true);  
 }  
}