

## Friday (chat bot) App:

### Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/boot_back"
    tools:context=".MainActivity">
    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:id="@+id/recycler_view"
        android:layout_weight="1"
        android:layout_gravity="center"
        android:layout_above="@+id/bottom_layout"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="300sp"
        android:gravity="center"
        android:text="Welcome to Friday! use Trivia keyword to get your trivia Historical questions"
        android:textColor="#EDF0F3"
        android:id="@+id/welcome_text"
        android:layout_gravity="center"
        android:textSize="28dp"/>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    <EditText
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_marginLeft="15dp"
        android:layout_height="wrap_content"
        android:id="@+id/message_edit_text"
        android:layout_centerInParent="true"
        android:hint="How can i help you to day"
```

```

        android:textColor="#123"
        android:layout_toLeftOf="@+id/send_btn"
        android:padding="16dp"
        android:background="@drawable/rounded_corner"/>
<ImageButton
    android:id="@+id/send_btn"
    android:layout_width="49dp"
    android:layout_height="49dp"
    android:layout_alignParentEnd="true"
    android:layout_centerInParent="true"
    android:layout_marginStart="10dp"
    android:padding="8dp"
    android:src="@drawable/send"
    android:background="?attr/selectableItemBackgroundBorderless"
/>

</LinearLayout>
</LinearLayout>

```

### MainActivity.java

```

package com.example.friday;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.text.Html;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;

public class MainActivity extends AppCompatActivity {
    RecyclerView recyclerView;
    TextView welcomeTextView;
    EditText messageEditText;
    ImageButton sendButton;
    List<Message> messageList;
    MessageAdapter messageAdapter;

```

```

// Declare global variables for trivia
String triviaQuestion = null;
String correctAnswer = null;

public static final MediaType JSON = MediaType.get("application/json; charset=utf-8");
OkHttpClient client = new OkHttpClient();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    messageList = new ArrayList<>();
    recyclerView = findViewById(R.id.recycler_view);
    welcomeTextView = findViewById(R.id.welcome_text);
    messageEditText = findViewById(R.id.message_edit_text);
    sendButton = findViewById(R.id.send_btn);

    messageAdapter = new MessageAdapter(messageList);
    recyclerView.setAdapter(messageAdapter);
    LinearLayoutManager llm = new LinearLayoutManager(this);
    llm.setStackFromEnd(true);
    recyclerView.setLayoutManager(llm);

    sendButton.setOnClickListener(view -> {
        String message = messageEditText.getText().toString().trim();
        addToChat(message, Message.SENT_BY_ME);
        messageEditText.setText("");
        welcomeTextView.setVisibility(View.GONE);

        // Check if the user is answering a trivia
        if (correctAnswer != null && message.equalsIgnoreCase(correctAnswer)) {
            addToChat("Correct! 🎉 Well done!", Message.SENT_BY_BOT);
            correctAnswer = null; // Reset after answering
        }
        else if (message.toLowerCase().contains("i don't know")){
            addToChat("The Correct Answer is \""+correctAnswer+"\" 🤔 Let Me Know if you need another question ", Message.SENT_BY_BOT);
        }
        else if (correctAnswer != null && !message.equalsIgnoreCase(correctAnswer)) { //Handle incorrect answers
            addToChat("❌ Wrong answer! \""+correctAnswer+"\" the correct answer is \""+correctAnswer, Message.SENT_BY_BOT);
        }
        else if (message.toLowerCase().contains("trivia") || message.toLowerCase().contains("question")) {
            Log.d("BotDebug", "Calling Trivia API..."); // Debugging callAPI();
        }
        else if (message.toLowerCase().contains("hi") || message.toLowerCase().contains("hello") || correctAnswer != null){
            addToChat("Hi! Welcome to Friday. Please use \"trivia\" keyword to get your question?", Message.SENT_BY_BOT);
        }
        else {
            addToChat("You Are using ❌ Wrong! keywords. please use trivia keyword to get your question", Message.SENT_BY_BOT);
        }
    });
}

```

```

}

void addToChat(String message, String sentBy) {
    runOnUiThread(() -> {
        messageList.add(new Message(message, sentBy));
        messageAdapter.notifyDataSetChanged();
        recyclerView.smoothScrollToPosition(messageAdapter.getItemCount());
    });
}

void callAPI() {
    Request request = new Request.Builder()
        .url(" https://opentdb.com/api.php?amount=50&category=23&type=multiple")
        .build();

    client.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            addToChat("Failed to fetch trivia question.\nThe issue may be:\n" +
                "\n" +
                "No internet connection\n" +
                "\n" +
                "API server issues\n" +
                "\n" +
                "Timeout due to slow response\n", Message.SENT_BY_BOT);
        }
    });
    @Override
    public void onResponse(Call call, Response response) throws IOException {
        if (response.isSuccessful()) {
            String result = response.body().string();
            Log.d("BotDebug", "Raw API Response: " + result); // Debugging log
            try {
                JSONObject jsonObject = new JSONObject(result);
                JSONArray results = jsonObject.getJSONArray("results");
                if (results.length() > 0) {
                    JSONObject questionObj = results.getJSONObject(0);

                    // Store both question & correct answer
                    triviaQuestion = questionObj.getString("question");
                    correctAnswer = questionObj.getString("correct_answer");
                    // Ensure proper text formatting
                    triviaQuestion =
                        Html.fromHtml(triviaQuestion, Html.FROM_HTML_MODE_LEGACY).toString();
                    correctAnswer =
                        Html.fromHtml(correctAnswer, Html.FROM_HTML_MODE_LEGACY).toString();

                    addToChat("Alright! Here's a trivia challenge for you: \n" + triviaQuestion,
                        Message.SENT_BY_BOT);
                } else {
                    addToChat("No trivia question found.", Message.SENT_BY_BOT);
                }
            } catch (JSONException e) {
                addToChat("Error parsing trivia question.", Message.SENT_BY_BOT);
            }
        } else {
            addToChat("Error from server.", Message.SENT_BY_BOT);
        }
    }
}
}
}
}

```

### activity\_splash.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_height="match_parent"
    android:background="@drawable/splash_background"
    tools:context=".Splash">
    <ImageView
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:src="@drawable/splash2"
        android:layout_marginLeft="22dp"
        android:layout_gravity="center"/>

</LinearLayout>
```

### Chat\_iteam.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <LinearLayout
        android:id="@+id/left_chat_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="9dp"
        android:background="@drawable/rounded_corner"
        android:backgroundTint="#03A9F5"
        android:layout_marginEnd="80dp">

        <TextView
            android:layout_margin="3dp"
            android:id="@+id/left_chat_text_view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:backgroundTint="#03A9BE"
            android:textSize="18sp"
            android:textColor="@color/white" />
    </LinearLayout>
</RelativeLayout>
```

```

        android:id="@+id/right_chat_view"
        android:layout_alignParentEnd="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="9dp"
        android:background="@drawable/rounded_corner"
        android:backgroundTint="#4CAF50"
        android:layout_marginStart="80dp">

        <TextView
            android:layout_margin="3dp"
            android:id="@+id/right_chat_text_view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:backgroundTint="#03A9BE"
            android:textSize="18sp"
            android:textColor="@color/white" />
    </LinearLayout>

</RelativeLayout>

```

## Message.java

```

package com.example.friday;

public class Message {
    public static String SENT_BY_ME="me";
    public static String SENT_BY_BOT="bot";

    String message;
    String sendBy;

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public String getSendBy() {
        return sendBy;
    }

    public void setSendBy(String sendBy) {

```

```

        this.sendBy = sendBy;
    }

    public Message(String message, String sendBy) {
        this.message = message;
        this.sendBy = sendBy;
    }
}

```

### MessageAdapter.java

```

package com.example.friday;

import android.text.Layout;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.MyViewHolder> {

    List<Message> messageList;

    public MessageAdapter(List<Message> messageList) {
        this.messageList =messageList;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View chatView= LayoutInflater.from(parent.getContext()).inflate(R.layout.chat_iteam,null);
        MyViewHolder myViewHolder = new MyViewHolder(chatView);
        return myViewHolder;
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
        Message message=messageList.get(position);
        if (message.getSendBy().equals(Message.SENT_BY_ME)){
            holder.leftChatView.setVisibility(View.GONE);

```

```

holder.rightChatView.setVisibility(View.VISIBLE);
holder.rightTextView.setText(message.getMessage());
}else {
    holder.rightChatView.setVisibility(View.GONE);
    holder.leftChatView.setVisibility(View.VISIBLE);
    holder.leftTextView.setText(message.getMessage());
}
}

@Override
public int getItemCount() {
    return messageList.size();
}

public class MyViewHolder extends RecyclerView.ViewHolder {
    LinearLayout leftChatView,rightChatView;
    TextView leftTextView,rightTextView;
    public MyViewHolder(@NonNull View itemView) {
        super(itemView);
        leftChatView=itemView.findViewById(R.id.left_chat_view);
        rightChatView=itemView.findViewById(R.id.right_chat_view);
        leftTextView=itemView.findViewById(R.id.left_chat_text_view);
        rightTextView=itemView.findViewById(R.id.right_chat_text_view);
    }
}
}

```

## **Splash.java**

```

package com.example.friday;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class Splash extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        new Handler().postDelayed(new Runnable() {
            @Override

```



```

        public void run() {
            Intent i= new Intent(Splash.this, MainActivity.class);
            startActivity(i);
        }
    },3000);
}
}

```

## Drawable:

### send.xml

```

<vector android:autoMirrored="true" android:height="24dp"
    android:tint="#E2850B" android:viewportHeight="24"
    android:viewportWidth="24" android:width="24dp"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="@android:color/white" android:pathData="M2.01,21L23,12 2.01,3
2,10l15,2 -15,2z"/>
</vector>

```

### round\_corner.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle"
    >
    <corners android:radius="10dp"/>
    <stroke android:width="1dp"/>
    <solid android:color="@color/white"/>
</shape>

```

### Build.gradle.kts(Module:app)

```

plugins {
    id("com.android.application")
}

```

```

android {
    namespace = "com.example.friday"
    compileSdk = 34

    defaultConfig {
        applicationId = "com.example.friday"
        minSdk = 26
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"
    }
}

```

```

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation("androidx.appcompat:appcompat:1.7.1")
    implementation("com.google.android.material:material:1.12.0")
    implementation("androidx.constraintlayout:constraintlayout:2.2.1")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.2.1")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.6.1")
    implementation("com.squareup.okhttp3:okhttp:4.10.0")
}

```

### AndroidManifest.xml

```

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

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/bot_logo"
        android:label="@string/app_name"
        android:roundIcon="@drawable/send"
        android:supportRtl="true"

```

```
    android:theme="@style/Theme.Friday"
    tools:targetApi="31" >
    <activity
        android:name=".MainActivity"
        android:exported="false" />
    <activity
        android:name=".Splash"
        android:exported="true"
        android:theme="@style/Theme.Design.NoActionBar"
        tools:ignore="DuplicateActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```