activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/rootLayout"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="#F8F8F8"
  android:gravity="center"
  android:orientation="vertical"
  android:padding="20dp">

  <!-- Title -->
  <TextView
    android:id="@+id/titleText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingBottom="20dp"
    android:text="Simple Media Player"
    android:textColor="#6200EE"
    android:textSize="22sp"
    android:textStyle="bold" />

  <!-- SeekBar for tracking progress -->
  <SeekBar
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp" />

  <!-- Button Layout for Rewind, Play, and Forward -->
  <LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:orientation="horizontal">

    <!-- Rewind Button -->
    <Button
      android:id="@+id/btnRewind"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginEnd="20dp"
      android:backgroundTint="#6200EE"
      android:text="⏪"
      android:textColor="#FFFFFF" />

    <!-- Play Button (Centered) -->
    <Button
```

```
            android:id="@+id/btnPlay"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:backgroundTint="#4CAF50"
            android:text="Play"
            android:textColor="#FFFFFF" />

        <!-- Forward Button -->
        <Button
            android:id="@+id/btnForward"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="20dp"
            android:backgroundTint="#6200EE"
            android:text="⏩"
            android:textColor="#000000" />
    </LinearLayout>

    <!-- Pause Button -->
    <Button
        android:id="@+id/btnPause"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:backgroundTint="#FF9800"
        android:text="Pause"
        android:textColor="#FFFFFF" />

    <!-- Stop Button -->
    <Button
        android:id="@+id/btnStop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:backgroundTint="#F44336"
        android:text="Stop"
        android:textColor="#FFFFFF" />

</LinearLayout>
```

MainActivity.java

```
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TextView;
```

```java
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private MediaPlayer mediaPlayer;
    private SeekBar seekBar;
    private Handler handler = new Handler();

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

        // Initialize UI elements
        Button btnPlay = findViewById(R.id.btnPlay);
        Button btnPause = findViewById(R.id.btnPause);
        Button btnStop = findViewById(R.id.btnStop);
        Button btnRewind = findViewById(R.id.btnRewind);
        Button btnForward = findViewById(R.id.btnForward);
        seekBar = findViewById(R.id.seekBar);
        TextView titleText = findViewById(R.id.titleText);

        // Initialize MediaPlayer with a sample audio file from raw folder
        mediaPlayer = MediaPlayer.create(this, R.raw.sample_audio);
        seekBar.setMax(mediaPlayer.getDuration());

        // Play Button
        btnPlay.setOnClickListener(v -> {
            if (!mediaPlayer.isPlaying()) {
                mediaPlayer.start();
                updateSeekBar();
            }
        });

        // Pause Button
        btnPause.setOnClickListener(v -> {
            if (mediaPlayer.isPlaying()) {
                mediaPlayer.pause();
            }
        });

        // Stop Button (Now resets track to the start)
        btnStop.setOnClickListener(v -> {
            if (mediaPlayer.isPlaying()) {
                mediaPlayer.stop();
                mediaPlayer.prepareAsync();
                mediaPlayer.setOnPreparedListener(mp -> {
                    mp.seekTo(0); // Reset to start
                    seekBar.setProgress(0); // Reset SeekBar
                });
            }
        });

        // Rewind Button (Rewind 10 seconds)
        btnRewind.setOnClickListener(v -> {
            int currentPosition = mediaPlayer.getCurrentPosition();
            int rewindPosition = Math.max(currentPosition - 10000, 0); // Prevent negative time
            mediaPlayer.seekTo(rewindPosition);
            seekBar.setProgress(rewindPosition);
        });

        // Forward Button (Forward 10 seconds)
        btnForward.setOnClickListener(v -> {
            int currentPosition = mediaPlayer.getCurrentPosition();
            int forwardPosition = Math.min(currentPosition + 10000, mediaPlayer.getDuration()); // Prevent
overflow
            mediaPlayer.seekTo(forwardPosition);
            seekBar.setProgress(forwardPosition);
        });

        // SeekBar Change Listener
        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                if (fromUser) {
                    mediaPlayer.seekTo(progress);
                }
            }
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {}

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {}
        });
    }

    // Function to update SeekBar as audio plays
    private void updateSeekBar() {
        runOnUiThread(() -> {
            if (mediaPlayer.isPlaying()) {
                seekBar.setProgress(mediaPlayer.getCurrentPosition());
                handler.postDelayed(this::updateSeekBar, 500);
            }
        });
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (mediaPlayer != null) {
            mediaPlayer.release();
```

```
      mediaPlayer = null;
    }
  }
}
```