**Pimpri Chinchwad Education Trust's**
**Pimpri Chinchwad College of Engineering**

## Assignment-04

**Roll No: 123M1H044**
**Name of Student: Drimesh Pendam** **Submission Date:** 12/10/2024

1. Create an Android application that issues a simple notification when a button is clicked. The notification should display a title, message, and small icon. Ensure that the notification appears in the status bar and can be dismissed by the user. Use the NotificationCompat.Builder class to build and issue the notification.

**Solution:**

XML FILE:

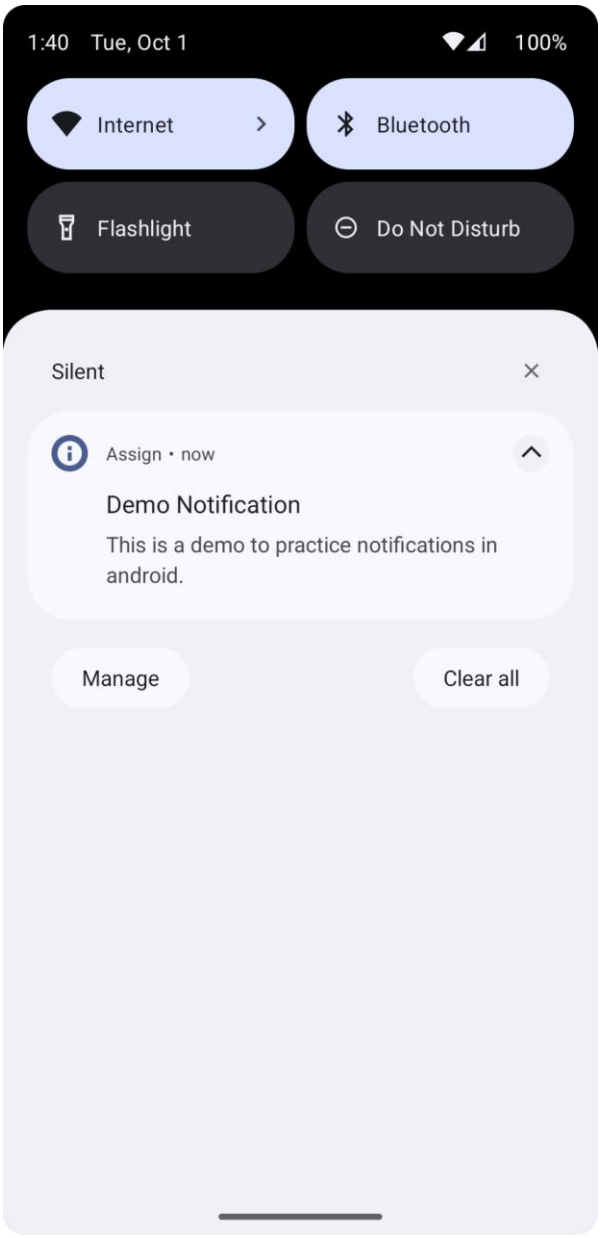
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="notify"
       android:id="@+id/b"/>

</LinearLayout>
```

JAVA FILE:

```
package com.example.assign;

import  android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
```

```
public class MainActivity extends AppCompatActivity {

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

       createchannel();
       b = findViewById(R.id.b);
       b.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               notifyuser();
           }
       });
   }

   protected void createchannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           NotificationChannel ch = new NotificationChannel("channel", "Channel
Name", NotificationManager.IMPORTANCE_DEFAULT);
           NotificationManager nm = getSystemService(NotificationManager.class);
           if (nm != null) {
               nm.createNotificationChannel(ch);
           }
       }
   }


   protected void notifyuser(){
       NotificationCompat.Builder b = new
NotificationCompat.Builder(MainActivity.this, "channel")
               .setSmallIcon(android.R.drawable.ic_dialog_info)
               .setContentTitle("Demo Notification")
               .setContentText("This is a demo to practice notifications in
android.")
               .setPriority(NotificationCompat.PRIORITY_HIGH)
               .setAutoCancel(true);

       NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
       notificationManager.notify(1, b.build());
   }
}
```

**Output:**


1:40 Tue, Oct 1
100%
Internet
Bluetooth
Flashlight
Do Not Disturb
Silent
Assign • now
Demo Notification
This is a demo to practice notifications in android.
Manage
Clear all

2. Design an app that triggers a basic notification with a clickable action. The notification should have a “View” button that, when clicked, opens a specific activity within the app. Use an Intent to handle the notification action, and display the action result within the new activity.

**Solution:**

**XML FILE 1:**

```
<?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">

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="notify"
       android:id="@+id/b"/>

</LinearLayout>
```

**XML FILE 2:**

```
<?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">

   <TextView
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="2nd activity. Navigated through notification"
       android:textSize="30sp"/>

</LinearLayout>
```

**JAVA FILE 1:**

```
package com.example.assign;


import  android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.*;


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

```
import androidx.core.app.NotificationCompat;



public class MainActivity extends AppCompatActivity {


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


       createchannel();
       b = findViewById(R.id.b);
       b.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               notifyuser();
           }
       });
   }


   protected void createchannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           NotificationChannel ch = new NotificationChannel("channel", "Channel
Name", NotificationManager.IMPORTANCE_DEFAULT);
           NotificationManager nm = getSystemService(NotificationManager.class);
           if (nm != null) {
               nm.createNotificationChannel(ch);
           }
       }
   }




   protected void notifyuser(){
       NotificationCompat.Builder b = new
NotificationCompat.Builder(MainActivity.this, "channel")
               .setSmallIcon(android.R.drawable.ic_dialog_info)
               .setContentTitle("Demo Notification")
               .setContentText("This is a demo to practice notifications in
android.")
               .setPriority(NotificationCompat.PRIORITY_HIGH)
               .setAutoCancel(true);


       NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
       notificationManager.notify(1, b.build());
   }
}
```

**JAVA FILE 2:**

```
package com.example.assign;
```

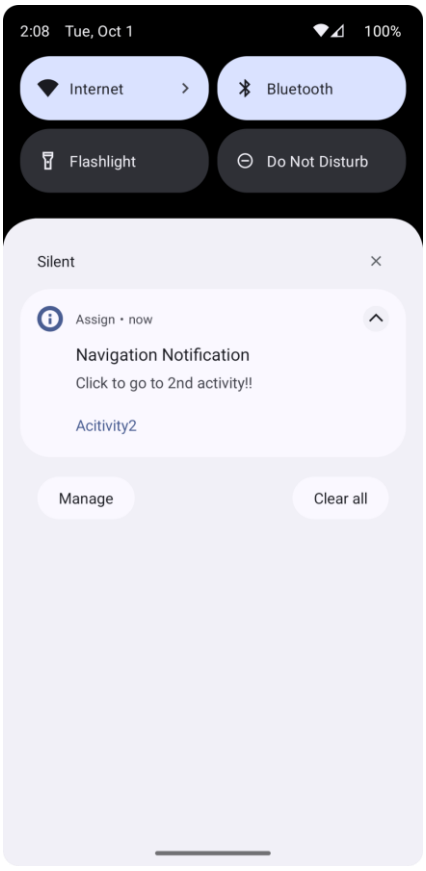
```
import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
    }
}
```

**Notification class:**

```
package com.example.assign;

import static androidx.core.content.ContextCompat.startActivity;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class NotificationActionReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if ("ACTION_BUTTON_CLICKED".equals(intent.getAction())) {
            Intent i = new Intent(context, MainActivity2.class);
            i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            context.startActivity(i);
        }
    }
}
```

**Output:**


2:07
notify


2:08 Tue, Oct 1
100%
Internet
Bluetooth
Flashlight
Do Not Disturb
Silent
Assign • now
Navigation Notification
Click to go to 2nd activity!!
Acitivity2
Manage
Clear all


3. Create an Android application that triggers a simple notification when a button is clicked. Use the NotificationCompat.Builder class to build the notification and set its properties, such as title, text, and icon. Ensure that the notification appears in the status bar and can be expanded to show additional content.

**Solution:**

XML FILE:

```
<?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">


   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="notify"
       android:id="@+id/b"/>
```

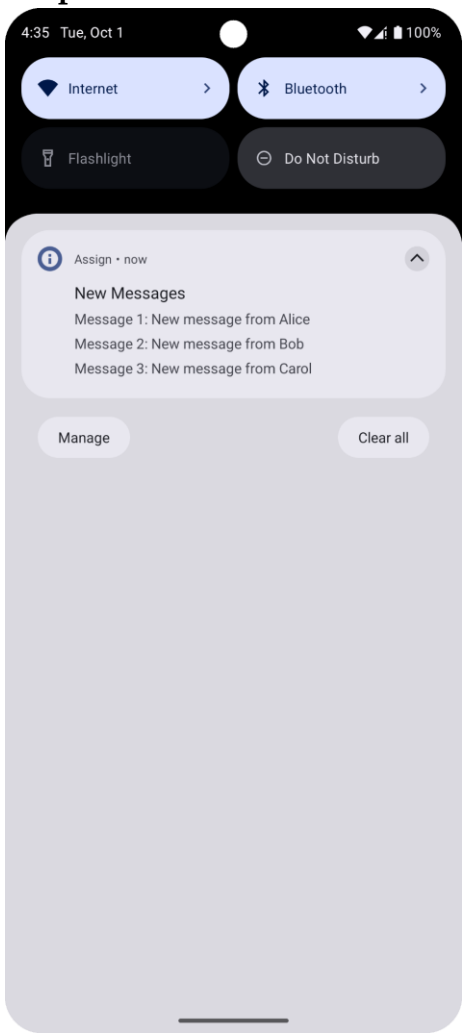
```
</LinearLayout>
```

JAVA FILE:

```
package com.example.assign;


import  android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.*;


import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;




public class MainActivity extends AppCompatActivity {


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


       createchannel();
       b = findViewById(R.id.b);
       b.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               notifyuser();
           }
       });
   }


   protected void createchannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           NotificationChannel ch = new NotificationChannel("channel", "Channel
Name", NotificationManager.IMPORTANCE_DEFAULT);
           NotificationManager nm = getSystemService(NotificationManager.class);
           if (nm != null) {
               nm.createNotificationChannel(ch);
           }
       }
   }




   protected void notifyuser(){
       NotificationCompat.Builder b = new
NotificationCompat.Builder(MainActivity.this, "channel")
```

```
                .setSmallIcon(android.R.drawable.ic_dialog_info)
                .setContentTitle("Demo Notification")
                .setContentText("Pull down to see extra content.")
                .setStyle(new NotificationCompat.InboxStyle()
                        .addLine("Message 1: New message from Alice")
                        .addLine("Message 2: New message from Bob")
                        .addLine("Message 3: New message from Carol")
                        .setBigContentTitle("New Messages"))
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setAutoCancel(true);


        NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        notificationManager.notify(1, b.build());
    }
}
```

**Output:**




4. Build an application that generates a notification with custom properties such as sound, vibration, and LED light color. Use the NotificationCompat.Builder class to set these

properties. The app should allow the user to configure these properties through a settings screen and preview the notification with the chosen settings.

**Solution:**

XML FILE:

```
<?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="vertical">


   <CheckBox
       android:layout_height="wrap_content"
       android:layout_width="wrap_content"
       android:id="@+id/cb1"
       android:text="Vibrations"/>

   <CheckBox
       android:layout_height="wrap_content"
       android:layout_width="wrap_content"
       android:id="@+id/cb2"
       android:text="Sound"/>

   <CheckBox
       android:layout_height="wrap_content"
       android:layout_width="wrap_content"
       android:id="@+id/cb3"
       android:text="LED Light"/>

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:id="@+id/button"
       android:text="apply"/>


</LinearLayout>
```

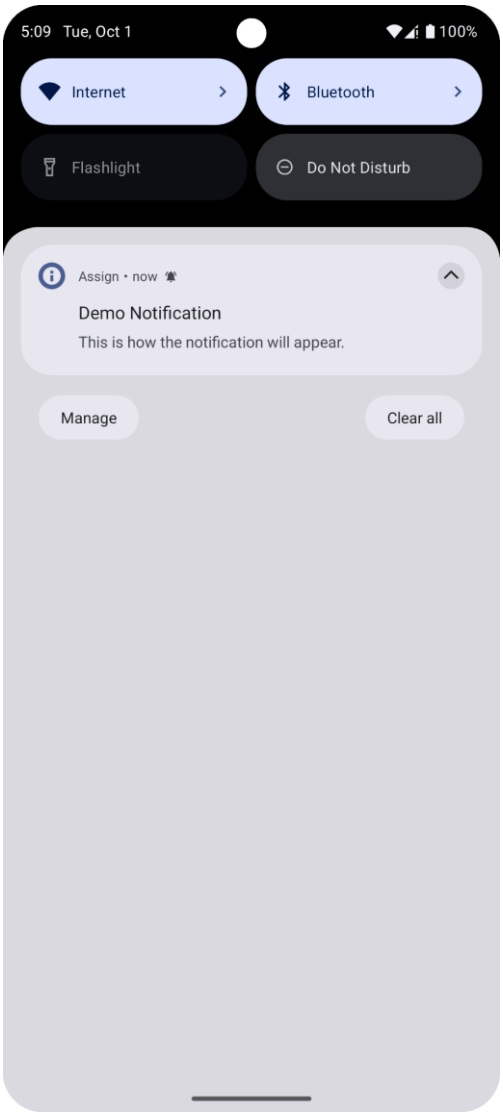
JAVA FILE:

```
package com.example.assign;


import  android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.graphics.Color;
import android.media.RingtoneManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.*;
```

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;




public class MainActivity extends AppCompatActivity {


   Button b;
   CheckBox cb1, cb2, cb3;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);


       createchannel();
       cb1 = findViewById(R.id.cb1);
       cb2 = findViewById(R.id.cb2);
       cb3 = findViewById(R.id.cb3);
       b = findViewById(R.id.button);

       b.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               NotificationCompat.Builder b = new
NotificationCompat.Builder(MainActivity.this, "channel")
                       .setSmallIcon(android.R.drawable.ic_dialog_info)
                       .setContentTitle("Demo Notification")
                       .setContentText("This is how the notification will
appear.")
                       .setPriority(NotificationCompat.PRIORITY_HIGH)
                       .setAutoCancel(true);



               if(cb1.isChecked()){
                   b.setVibrate(new long[]{0, 200, 100, 200});
               }
               if(cb2.isChecked()){

b.setSound(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION));
               }
               if(cb3.isChecked()){
                   b.setLights(Color.BLUE, 300, 1000);
               }
               NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
               notificationManager.notify(1, b.build());
           }
       });

   }


   protected void createchannel() {
       if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
           NotificationChannel ch = new NotificationChannel("channel", "Channel
Name", NotificationManager.IMPORTANCE_DEFAULT);
           NotificationManager nm = getSystemService(NotificationManager.class);
           if (nm != null) {
               nm.createNotificationChannel(ch);
```

```
                }
            }
        }
}
```

## Output:




5. Create a notification that includes action buttons. For example, build a media player notification with “Play”,”Pause”, and & ”Stop” buttons. Use the NotificationCompat.Builder class to attach these actions and handle the corresponding intents when the user interacts with the notification.

**Solution:**

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_height="match_parent"
   android:layout_width="match_parent">
```

```
</LinearLayout>
```

JAVA FILE:

```
package com.example.assign;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

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


        createchannel();
        notifyuser();
        mediaPlayer = MediaPlayer.create(this, R.raw.audio);

    }

    protected void createchannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel ch = new NotificationChannel("channel", "Channel
Name", NotificationManager.IMPORTANCE_DEFAULT);
            NotificationManager nm = getSystemService(NotificationManager.class);
            if (nm != null) {
                nm.createNotificationChannel(ch);
            }
        }
    }

    protected void notifyuser() {
        // Create PendingIntent for "Play" action
        Intent playIntent = new Intent(this, NotificationActionReceiver.class);
        playIntent.setAction("ACTION_PLAY");
        PendingIntent playPendingIntent = PendingIntent.getBroadcast(this, 0,
playIntent, PendingIntent.FLAG_IMMUTABLE);

        // Create PendingIntent for "Pause" action
        Intent pauseIntent = new Intent(this, NotificationActionReceiver.class);
        pauseIntent.setAction("ACTION_PAUSE");
        PendingIntent pausePendingIntent = PendingIntent.getBroadcast(this, 1,
pauseIntent, PendingIntent.FLAG_IMMUTABLE);

        // Create PendingIntent for "Stop" action
```

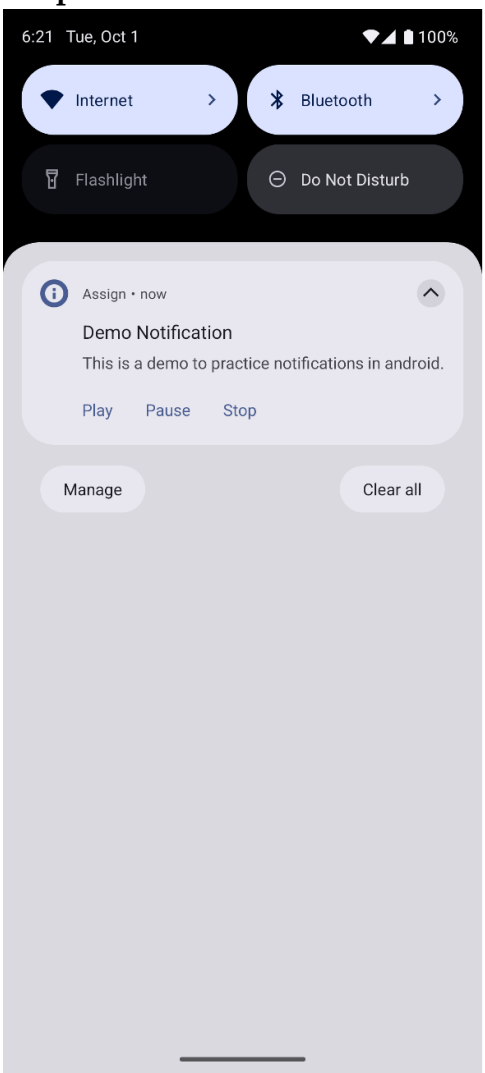
```
        Intent stopIntent = new Intent(this, NotificationActionReceiver.class);
        stopIntent.setAction("ACTION_STOP");
        PendingIntent stopPendingIntent = PendingIntent.getBroadcast(this, 2,
stopIntent, PendingIntent.FLAG_IMMUTABLE);

        // Build the notification with action buttons
        NotificationCompat.Builder b = new
NotificationCompat.Builder(MainActivity.this, "channel")
                .setSmallIcon(android.R.drawable.ic_dialog_info)
                .setContentTitle("Demo Notification")
                .setContentText("This is a demo to practice notifications in
android.")
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setAutoCancel(true)
                .addAction(android.R.drawable.ic_media_play, "Play",
playPendingIntent)   // Add Play button
                .addAction(android.R.drawable.ic_media_pause, "Pause",
pausePendingIntent) // Add Pause button
                .addAction(android.R.drawable.ic_menu_close_clear_cancel, "Stop",
stopPendingIntent);  // Add Stop button

        NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        notificationManager.notify(1, b.build());
    }
}
```

NOTIFICATION CLASS:

```
package com.example.assign;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.widget.Toast;

public class NotificationActionReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        MediaPlayer mediaPlayer = MainActivity.mediaPlayer;  // Access the media
player from MainActivity

        if (action != null && mediaPlayer != null) {
            switch (action) {
                case "ACTION_PLAY":
                    if (!mediaPlayer.isPlaying()) {
                        mediaPlayer.start();  // Start playing audio
                        Toast.makeText(context, "Audio Playing",
Toast.LENGTH_SHORT).show();
                    }
                    break;

                case "ACTION_PAUSE":
                    if (mediaPlayer.isPlaying()) {
                        mediaPlayer.pause();  // Pause the audio
```

```
                        Toast.makeText(context, "Audio Paused",
Toast.LENGTH_SHORT).show();
                    }
                    break;

                case "ACTION_STOP":
                    if (mediaPlayer.isPlaying()) {
                        mediaPlayer.stop(); // Stop the audio
                        Toast.makeText(context, "Audio Stopped",
Toast.LENGTH_SHORT).show();
                        try {
                            mediaPlayer.prepare();  // Prepare mediaPlayer to play
again
                        } catch (Exception e) {
                            e.printStackTrace();
                        }
                    }
                    break;
            }
        }
    }
}
```

**Output:**



6. **Develop an app that triggers a "Big Picture Style" notification. The notification should display a large image when expanded. UseNotificationCompat.BigPictureStyle to implement the expanded notification and ensure it includes both a title and a summary text when collapsed.**

**Solution:**

**MainActivity.java:**

```
package com.example.myapplication;


import android.app.NotificationChannel;

import android.app.NotificationManager;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.os.Build;

import android.os.Bundle;

import android.view.View;

import android.widget.Toast;


import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.app.NotificationCompat;

import androidx.core.app.NotificationManagerCompat;


public class MainActivity extends AppCompatActivity {
```

```
    private static final String CHANNEL_ID = "channel_id";
    private static final int NOTIFICATION_PERMISSION_CODE = 100;

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

        createNotificationChannel();

        findViewById(R.id.button_notify).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (ActivityCompat.checkSelfPermission(MainActivity.this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{android.Manifest.permission.POST_NOTIFICATIONS},
NOTIFICATION_PERMISSION_CODE);
                } else {
                    sendBigPictureNotification();
                }
            }
        });
    }
```

```
    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
                    "Channel Name",
                    NotificationManager.IMPORTANCE_DEFAULT);
            NotificationManager manager = getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel);
        }
    }

    private void sendBigPictureNotification() {
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }
        Bitmap bigPicture = BitmapFactory.decodeResource(getResources(),
R.drawable.flower);

        NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
CHANNEL_ID)
                .setSmallIcon(R.mipmap.notify)
                .setContentTitle("Notification Title")
                .setContentText("This is a summary text.")
                .setStyle(new NotificationCompat.BigPictureStyle()
                        .bigPicture(bigPicture)
                        .setBigContentTitle("Expanded Title")
```

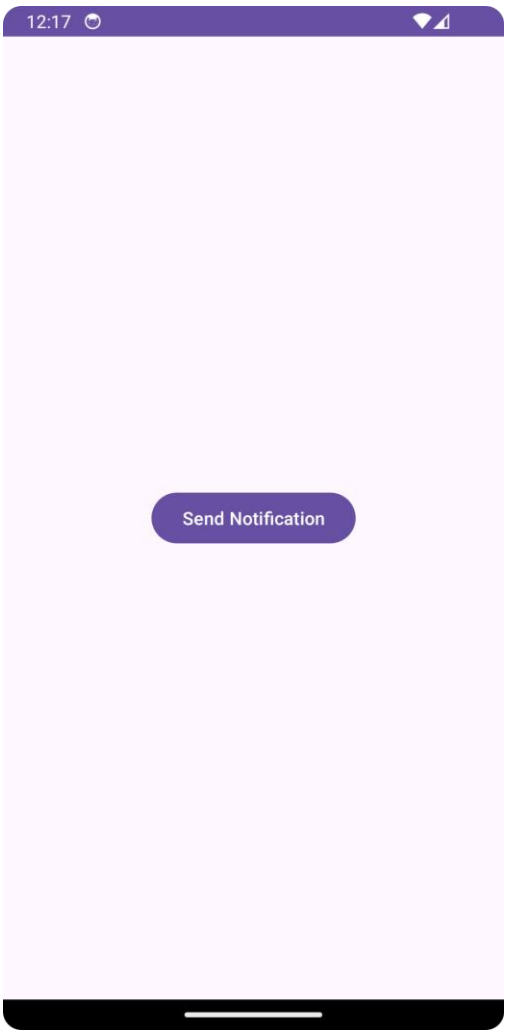
```
                .setSummaryText("This is expanded summary text."))
            .setPriority(NotificationCompat.PRIORITY_DEFAULT);

        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
        notificationManager.notify(1, builder.build());
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == NOTIFICATION_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                sendBigPictureNotification();
            } else {
                Toast.makeText(MainActivity.this, "Permission denied!",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

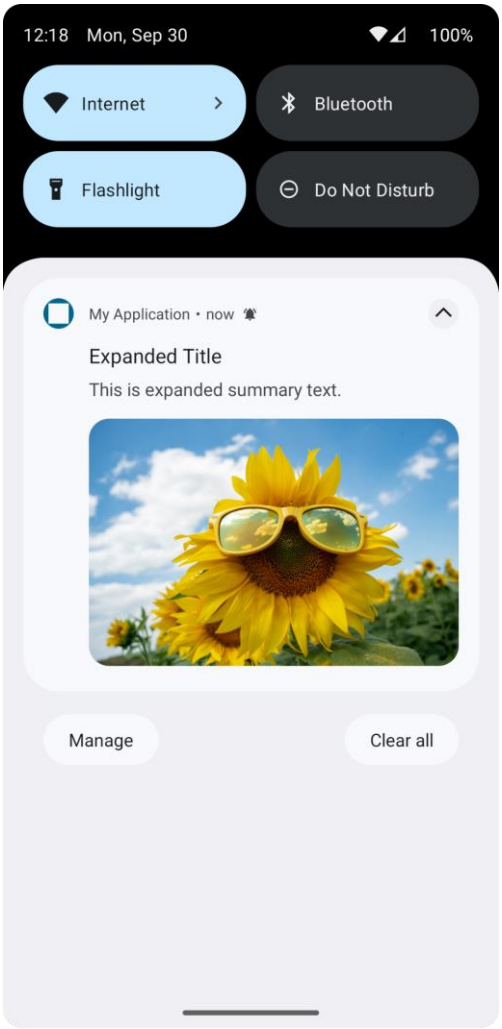
**activity_main.xml:**

```
<?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="match_parent">


    <Button
        android:id="@+id/button_notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Notification"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

**Output:**



**7. Build an app that generates a heads-up notification (high-priority notification that pops up as an overlay). Set up the notification to appear when an urgent event occurs, such as receiving an important message or a time-sensitive alert. Customize the notification to include an action, such as "Dismiss" or "Snooze".**

**Solution:**

**MainActivity.java:**

```
package com.example.myapplication;


import android.app.Notification;

import  android.app.NotificationChannel;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.os.Build;

import android.os.Bundle;

import android.view.View;


import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.app.NotificationCompat;
```

```
import androidx.core.app.NotificationManagerCompat;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "urgent_channel";
    private static final int NOTIFICATION_ID = 001;
    private static final int REQUEST_NOTIFICATION_PERMISSION = 123;

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

        createNotificationChannel();
        checkNotificationPermission();
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "Urgent Notifications";
            String description = "Channel for urgent notifications";
            int importance = NotificationManager.IMPORTANCE_HIGH;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name,
importance);
            channel.setDescription(description);
```

```
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);

            notificationManager.createNotificationChannel(channel);

        }

    }


    private void checkNotificationPermission() {

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {

            if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS)

                    != PackageManager.PERMISSION_GRANTED) {

                ActivityCompat.requestPermissions(this,

                        new String[]{android.Manifest.permission.POST_NOTIFICATIONS},
REQUEST_NOTIFICATION_PERMISSION);

            } else {

                setupNotificationButton();

            }

        } else {

            setupNotificationButton();

        }

    }


    @Override

    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {

        super.onRequestPermissionsResult(requestCode, permissions, grantResults);

        if (requestCode == REQUEST_NOTIFICATION_PERMISSION) {
```

```
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                setupNotificationButton();
            }
        }
    }

    private void setupNotificationButton() {
        findViewById(R.id.buttonNotify).setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                sendUrgentNotification();
            }
        });
    }

    private void sendUrgentNotification() {
        // Create the intent for "Dismiss" action
        Intent dismissIntent = new Intent(this, DismissNotificationReceiver.class);
        dismissIntent.putExtra("notification_id", NOTIFICATION_ID);  // Pass notification
ID to the receiver
        PendingIntent dismissPendingIntent = PendingIntent.getBroadcast(
                this, 1, dismissIntent, PendingIntent.FLAG_CANCEL_CURRENT |
PendingIntent.FLAG_IMMUTABLE);

        Intent snoozeIntent = new Intent(this, SnoozeNotificationReceiver.class);
```

```
        PendingIntent snoozePendingIntent = PendingIntent.getBroadcast(

                this, 2, snoozeIntent, PendingIntent.FLAG_CANCEL_CURRENT |
PendingIntent.FLAG_IMMUTABLE);



        NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
CHANNEL_ID)

                .setSmallIcon(android.R.drawable.ic_dialog_info)

                .setContentTitle("Urgent Alert")

                .setContentText("This is a time-sensitive alert!")

                .setPriority(NotificationCompat.PRIORITY_HIGH)

                .setDefaults(Notification.DEFAULT_ALL)

                .setCategory(NotificationCompat.CATEGORY_MESSAGE)

                .addAction(R.drawable.dismiss, "Dismiss", dismissPendingIntent)  // Add
Dismiss action

                .addAction(R.drawable.snooze, "Snooze", snoozePendingIntent)    // Add
Snooze action

                .setAutoCancel(true)

                .setFullScreenIntent(null, true); // Heads-up display



        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);

        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {

            return;

        }

        notificationManager.notify(NOTIFICATION_ID, builder.build());

    }

}
```

**DismissNotificationReceiver.java:**

```
package com.example.myapplication;

import android.app.NotificationManager;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.util.Log;


public class DismissNotificationReceiver extends BroadcastReceiver {


   private static final String TAG = "DismissReceiver";


   @Override

   public void onReceive(Context context, Intent intent) {

      // Log the dismiss action for debugging

      Log.d(TAG, "Notification dismissed");


      // Cancel the notification using NotificationManager

      NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

      if (notificationManager != null) {

         int notificationId = intent.getIntExtra("notification_id", -1);

         if (notificationId != -1) {

            notificationManager.cancel(notificationId); // Cancel the specific notification

            Log.d(TAG, "Notification with ID " + notificationId + " cancelled.");
```

```
            } else {
                Log.e(TAG, "No valid notification ID provided to cancel.");
            }
        } else {
            Log.e(TAG, "NotificationManager is null.");
        }
    }
}
```

**SnoozeNotificationReceiver.java:**

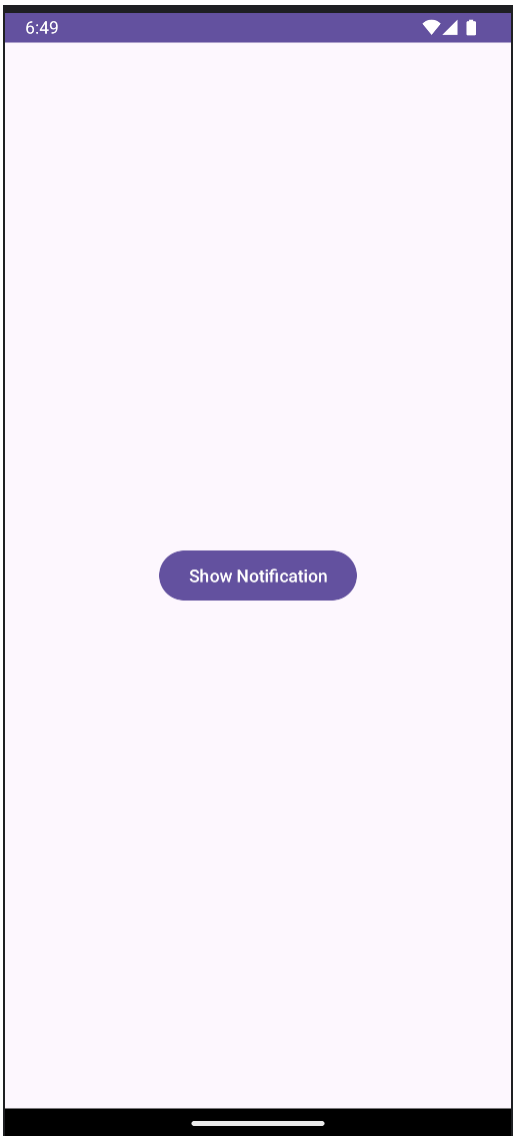
```
package com.example.myapplication;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class SnoozeNotificationReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
    }
}
```

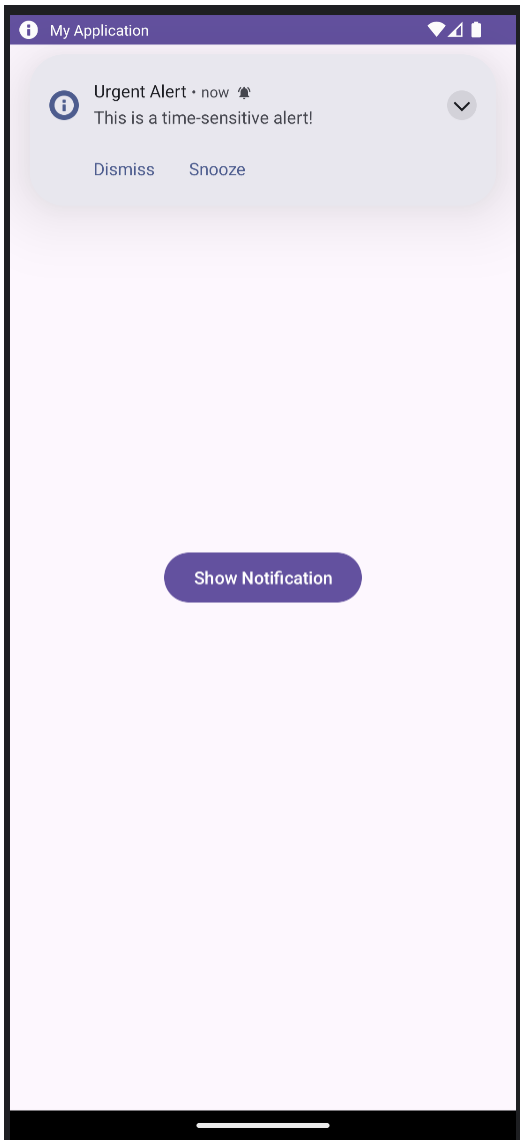
**activity_main.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<Button
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/buttonNotify"
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:text="Show Notification"
    android:layout_gravity="center"/>
```

**Output:**




**8. Develop an Android application that creates notification channels for different categories of notifications (e.g., “Messages”, “Alerts”, “Promotions”). Use the NotificationChannel class to define channel properties like importance, sound, and vibration. Ensure notifications are issued under the appropriate channel, and allow the user to customize channel seflings.**

**Solution:**

**Output:**

**MainActivity.java:**

```
package com.example.myapplication;

import  android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

public class MainActivity extends AppCompatActivity {

   private static final String CHANNEL_MESSAGES_ID = "messages_channel";
   private static final String CHANNEL_ALERTS_ID = "alerts_channel";
   private static final String CHANNEL_PROMOTIONS_ID = "promotions_channel";

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

      createNotificationChannels();

      findViewById(R.id.buttonSendMessagesNotification).setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View v) {
            sendNotification("Messages Channel", "You have a new message!",
CHANNEL_MESSAGES_ID);
         }
      });

      findViewById(R.id.buttonSendAlertsNotification).setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View v) {
            sendNotification("Alerts Channel", "This is an important alert!",
CHANNEL_ALERTS_ID);
         }
```

```
        });

        findViewById(R.id.buttonSendPromotionsNotification).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendNotification("Promotions Channel", "Check out our latest promotions!",
CHANNEL_PROMOTIONS_ID);
            }
        });

        findViewById(R.id.buttonOpenNotificationSettings).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openNotificationSettings();
            }
        });
    }

    private void createNotificationChannels() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);

            NotificationChannel messagesChannel = new NotificationChannel(
                    CHANNEL_MESSAGES_ID,
                    "Messages",
                    NotificationManager.IMPORTANCE_DEFAULT
            );
            messagesChannel.setDescription("Channel for message notifications");
            notificationManager.createNotificationChannel(messagesChannel);

            NotificationChannel alertsChannel = new NotificationChannel(
                    CHANNEL_ALERTS_ID,
                    "Alerts",
                    NotificationManager.IMPORTANCE_HIGH
            );
            alertsChannel.setDescription("Channel for important alerts");
            alertsChannel.enableVibration(true);
            notificationManager.createNotificationChannel(alertsChannel);

            NotificationChannel promotionsChannel = new NotificationChannel(
                    CHANNEL_PROMOTIONS_ID,
```

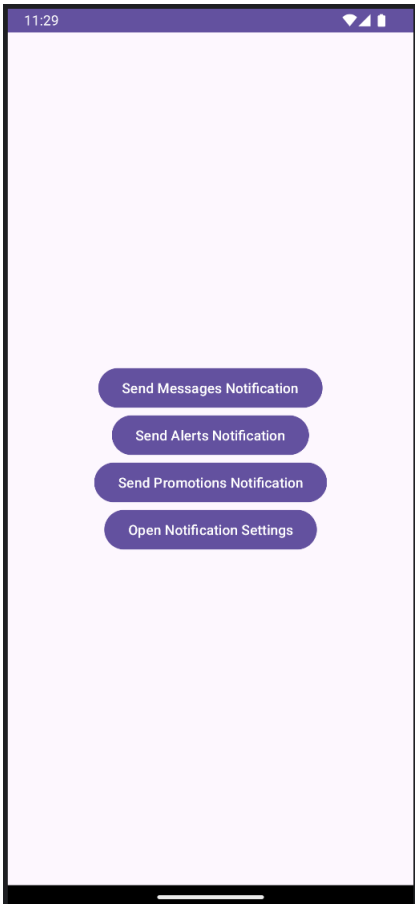
```
                "Promotions",
                NotificationManager.IMPORTANCE_LOW
            );
            promotionsChannel.setDescription("Channel for promotional notifications");
            notificationManager.createNotificationChannel(promotionsChannel);
        }
    }

    private void sendNotification(String title, String content, String channelId) {
        NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
channelId)
                .setSmallIcon(android.R.drawable.ic_dialog_info)
                .setContentTitle(title)
                .setContentText(content)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setAutoCancel(true);

        NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }
        notificationManager.notify((int) System.currentTimeMillis(), builder.build());
    }

    private void openNotificationSettings() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            Intent intent = new
Intent(android.provider.Settings.ACTION_APP_NOTIFICATION_SETTINGS);
            intent.putExtra(android.provider.Settings.EXTRA_APP_PACKAGE,
getPackageName());
            startActivity(intent);
        }
    }
}
```

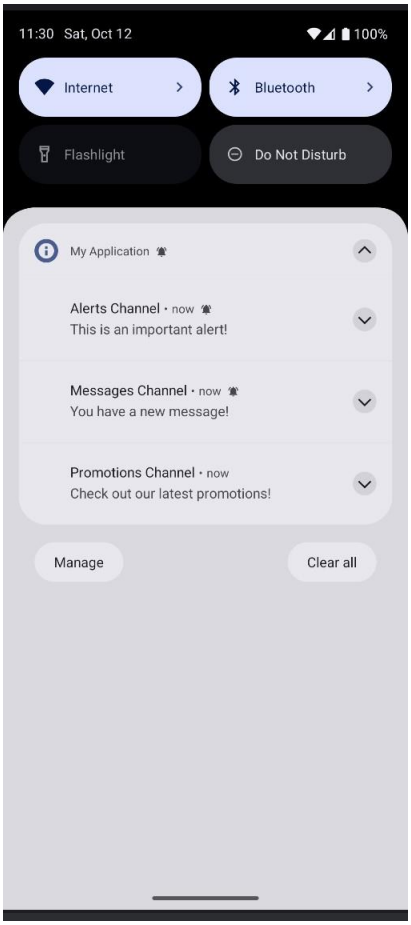
**activity_main.xml:**

```
<?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:gravity="center"
    android:padding="16dp">

    <Button
        android:id="@+id/buttonSendMessagesNotification"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Messages Notification" />

    <Button
        android:id="@+id/buttonSendAlertsNotification"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Alerts Notification" />

    <Button
        android:id="@+id/buttonSendPromotionsNotification"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Promotions Notification" />

    <Button
        android:id="@+id/buttonOpenNotificationSettings"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Open Notification Settings" />

</LinearLayout>
```

**Output:**




**9. Create an application that issues multiple notifications and groups them into a single expandable notification. Use NotificationCompat.Builder and NotificationCompat.InboxStyle to group notifications, such as showing a list of recent messages in a messaging app. Implement functionality to expand and collapse the group.**

Solution:

MainActivity.java:

```
package com.example.myapplication;
import  android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "message_channel";
```

```
    private static final String GROUP_KEY_MESSAGES = "group_key_messages";

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

        createNotificationChannel();
        issueNotifications();
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "Message Channel";
            String description = "Channel for message notifications";
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name,
importance);
            channel.setDescription(description);
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }

    private void issueNotifications() {
        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

        for (int i = 0; i < 5; i++) {
            String message = "Message " + (i + 1);
            NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
CHANNEL_ID)
                    .setSmallIcon(android.R.drawable.ic_dialog_info) // Replace with your icon
                    .setContentTitle("New Message")
                    .setContentText(message)
                    .setGroup(GROUP_KEY_MESSAGES)
                    .setAutoCancel(true);
            notificationManager.notify(i, builder.build());
        }

        // Create the summary notification
        NotificationCompat.Builder summaryBuilder = new
NotificationCompat.Builder(this, CHANNEL_ID)
```

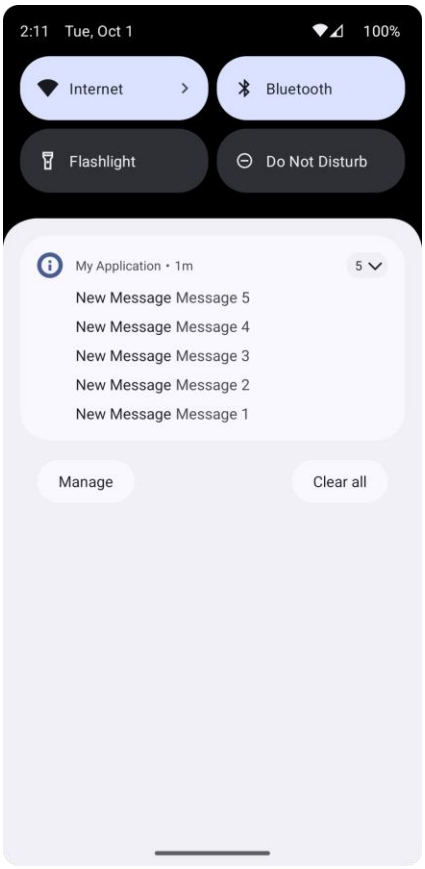
```
            .setContentTitle("You have new messages")
            .setContentText("You have " + 5 + " new messages.")
            .setSmallIcon(android.R.drawable.ic_dialog_info) // Replace with your icon
            .setStyle(new NotificationCompat.InboxStyle()
                .addLine("Message 1")
                .addLine("Message 2")
                .addLine("Message 3")
                .addLine("Message 4")
                .addLine("Message 5")
                .setBigContentTitle("New Messages"))
            .setGroup(GROUP_KEY_MESSAGES)
            .setGroupSummary(true);

        notificationManager.notify(100, summaryBuilder.build());
    }
}
```

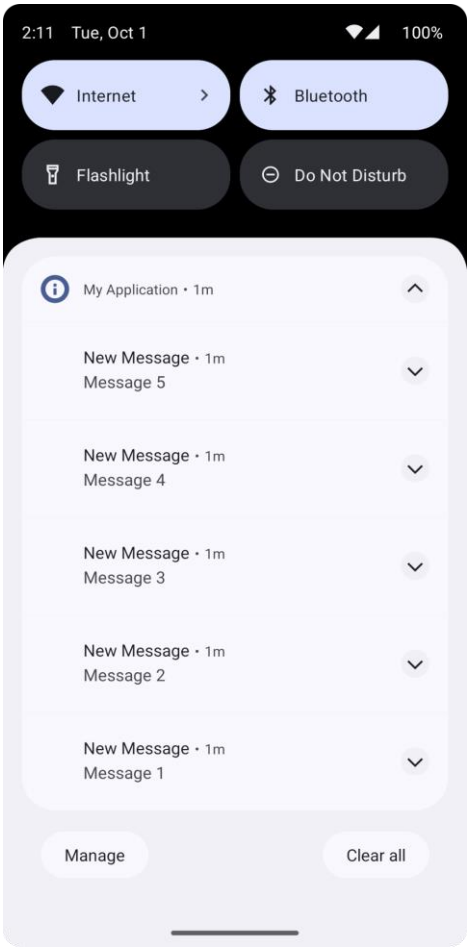
activity_main.xml:

```
<?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="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notifications Example"
        android:textSize="24sp"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

Output:


2:10
Notifications Example


2:11 Tue, Oct 1
100%
Internet
Bluetooth
Flashlight
Do Not Disturb
My Application • 1m
5
New Message Message 5
New Message Message 4
New Message Message 3
New Message Message 2
New Message Message 1
Manage
Clear all


2:11 Tue, Oct 1
100%
Internet
Bluetooth
Flashlight
Do Not Disturb
My Application • 1m
New Message • 1m
Message 5
New Message • 1m
Message 4
New Message • 1m
Message 3
New Message • 1m
Message 2
New Message • 1m
Message 1
Manage
Clear all

**10. Design an application that schedules and triggers notifications at a specific time or interval (e.g., daily reminders). Use AlarmManager or WorkManager to schedule the notifications, and issue them using NotificationCompat.Builder. Ensure that notifications are triggered even when the app is in the background or closed.**

**Solution:**
**MainActivity:**

```
package com.example.myapplication;

import android.Manifest;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

  private Button setAlarmButton;
  private static final int REQUEST_CODE_POST_NOTIFICATIONS = 1;

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

    checkNotificationPermission();

    setAlarmButton = findViewById(R.id.setAlarmButton);

    // Check and request POST_NOTIFICATIONS permission if necessary
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
      if (ContextCompat.checkSelfPermission(this,
Manifest.permission.POST_NOTIFICATIONS)
```

```
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.POST_NOTIFICATIONS},
                    REQUEST_CODE_POST_NOTIFICATIONS);
        }
    }

    setAlarmButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showTimePickerDialog();
        }
    });
  }

  @Override
  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_CODE_POST_NOTIFICATIONS) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            // Permission granted for POST_NOTIFICATIONS
            Toast.makeText(this, "Notification permission granted",
Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(this, "Notification permission denied",
Toast.LENGTH_SHORT).show();
        }
    }
  }

  private void checkNotificationPermission() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) { // API 33
        if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS)
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                    new String[]{android.Manifest.permission.POST_NOTIFICATIONS},
                    1);
        }
    }
  }
```

```
  private void showTimePickerDialog() {
     Calendar calendar = Calendar.getInstance();
     int hour = calendar.get(Calendar.HOUR_OF_DAY);
     int minute = calendar.get(Calendar.MINUTE);

     // Show TimePickerDialog to select the time
     new android.app.TimePickerDialog(this, (view, selectedHour, selectedMinute) -> {
        // Schedule alarm with the selected time
        scheduleDailyReminder(selectedHour, selectedMinute);
        Toast.makeText(this, "Reminder set for " + selectedHour + ":" + selectedMinute,
Toast.LENGTH_SHORT).show();
     }, hour, minute, true).show();
  }

  private void scheduleDailyReminder(int hour, int minute) {
     AlarmManager alarmManager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
     Calendar calendar = Calendar.getInstance();

     // Set the calendar to the selected time
     calendar.set(Calendar.HOUR_OF_DAY, hour);
     calendar.set(Calendar.MINUTE, minute);
     calendar.set(Calendar.SECOND, 0);
     calendar.set(Calendar.MILLISECOND, 0);

     Intent intent = new Intent(this, ReminderBroadcastReceiver.class);
     PendingIntent pendingIntent = PendingIntent.getBroadcast(
           this,
           0,
           intent,
           PendingIntent.FLAG_UPDATE_CURRENT |
PendingIntent.FLAG_IMMUTABLE // Updated line
     );

     if (alarmManager != null) {
        // Set alarm to trigger daily at the selected time
        alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,
calendar.getTimeInMillis(),
              AlarmManager.INTERVAL_DAY, pendingIntent);
     }
  }
```

```
}
```

**NotificationHelper.java:**

```
package com.example.myapplication;

import  android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.Context;
import androidx.core.app.NotificationCompat;

public class NotificationHelper {
  private static final String CHANNEL_ID = "reminder_channel";

  public static NotificationCompat.Builder createNotification(Context context, String title,
String message) {
    // Create a notification channel for Android O and above
    if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
      NotificationChannel channel = new NotificationChannel(
          CHANNEL_ID,
          "Reminder Notifications",
          NotificationManager.IMPORTANCE_HIGH
      );
      NotificationManager manager =
context.getSystemService(NotificationManager.class);
      if (manager != null) {
        manager.createNotificationChannel(channel);
      }
    }

    // Build the notification
    return new NotificationCompat.Builder(context, CHANNEL_ID)
        .setContentTitle(title)
        .setContentText(message)
        .setSmallIcon(R.drawable.ic_launcher_foreground) // Replace with your app's
notification icon
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setAutoCancel(true); // Automatically remove the notification when tapped
  }
}
```

ReminderBroadcastReceiver.java:

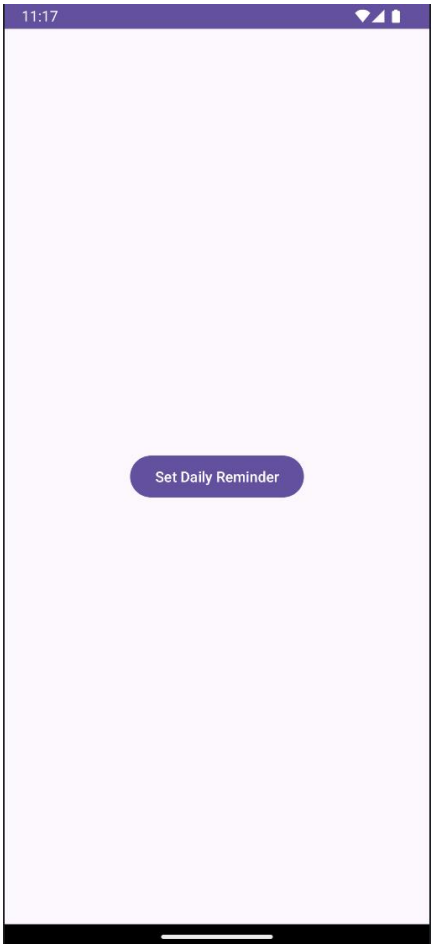
```
package com.example.myapplication;

import android.content.BroadcastReceiver;
```

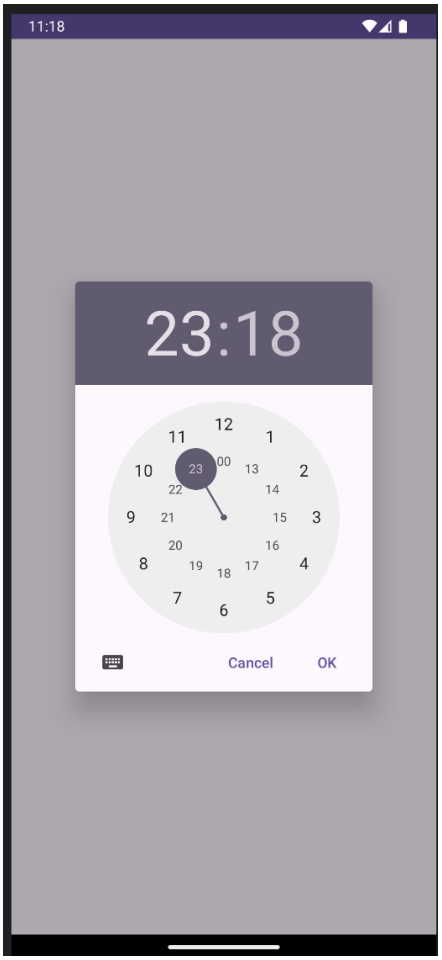
```
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;

import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

public class ReminderBroadcastReceiver extends BroadcastReceiver {
  @Override
  public void onReceive(Context context, Intent intent) {
     String title = "Daily Reminder";
     String message = "This is your scheduled reminder!";

     NotificationCompat.Builder notification =
NotificationHelper.createNotification(context, title, message);
     NotificationManagerCompat manager = NotificationManagerCompat.from(context);

     // Check if the POST_NOTIFICATIONS permission is granted
     if (ActivityCompat.checkSelfPermission(context,
android.Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
        return; // Permission not granted, return without showing notification
     }

     manager.notify(1001, notification.build()); // Unique notification ID
  }
}
```

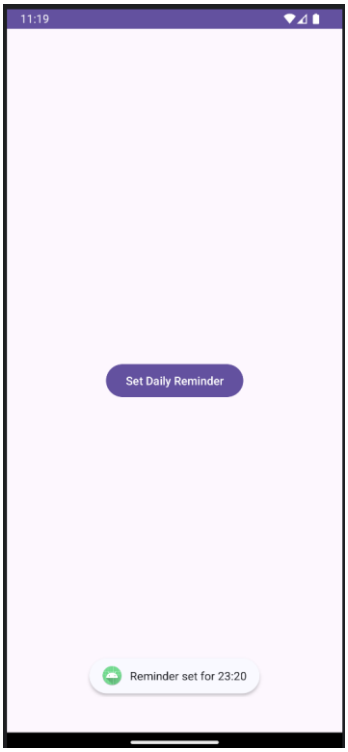
xml file:

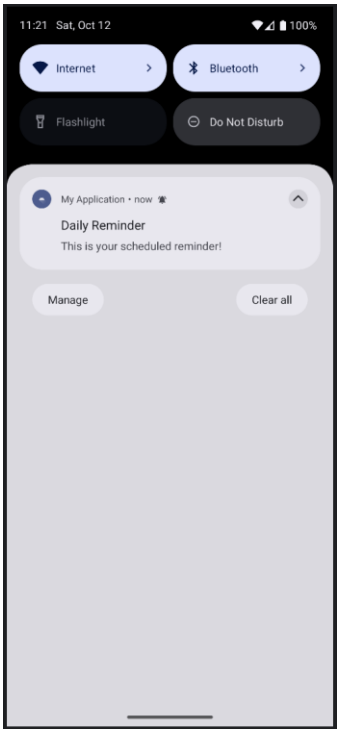
```
<?xml version="1.0" encoding="utf-8"?>
<!-- activity_main.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:gravity="center"
  android:orientation="vertical">

  <Button
     android:id="@+id/setAlarmButton"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:text="Set Daily Reminder" />
</LinearLayout>
```

**Output:**


11:17
Set Daily Reminder


11:18
23:18
Cancel
OK


11:19
Set Daily Reminder
Reminder set for 23:20


11:21 Sat, Oct 12
100%
Internet
Bluetooth
Flashlight
Do Not Disturb
My Application • now
Daily Reminder
This is your scheduled reminder!
Manage
Clear all