Experiment 08: Develop an application that uses GUI components, Fonts and colors

PART A

A.1 Aim: To develop an application that uses GUI components, Fonts and colors.

A.2 Objectives: **To introduce students with various tools like** Android Studio, NS2, Wire-shark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcome:

After successful completion of this experiment students will be able to Develop an application that uses GUI components, Fonts and colours **A.4 Theory:**

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

1 Open android studio and select new android project .

2 Give project name and select next 3

Choose the android version.

4 Enter the package name. package name must be two word separated by comma and click finish

5 Go to package explorer in the left hand side and select our project.

6 Go to res folder and select layout. Double click the main.xml file 7 Now you can see the Graphics layout window.

Source code:

CodeforActivity_main.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"android:orient
  ati on="vertical"
  android:layout_width="match_parent"android:layout_height="mat ch_parent">

  <TextViewandroid:id="@+id/textVi
    ew"android:layout_width="match_
    parent"
    android:layout_height="wrap_content"
    a
    ndroid:layout_margin="30dp"android:
    g ravity="center"android:text="Hello
    World!"android:textSize="25sp"androi
    d:textStyle="bold" />
  <Button
    android:id="@+id/button1"android:lay
    out_width="match_parent"android:layo
```

```
    ut_height="wrap_content"android:layou
    t_margin="20dp"android:gravity="cent
    er"android:text="Change font size"android:textSize="25sp"
    />

  <Button
    android:id="@+id/button2"android:lay
    out_width="match_parent"android:lay o

    ut_height="wrap_content"android:layo
    u

    t_margin="20dp"android:gravity="cent
    er"android:text="Change color"android:textSize="25sp"
    />

< /LinearLayout >
```

CodeforMainActivity.java:

```
ackagecom.example.exno1;


importandroid.graphics.Col
or; import

android.support.v7.app.AppCompatActivity
; im port android.os.Bundle; import
android.view.View;import
android.widget.Button;impo rta
ndroid.widget.TextView;


publicclassMainActivityextendsAppCompatActivity
```

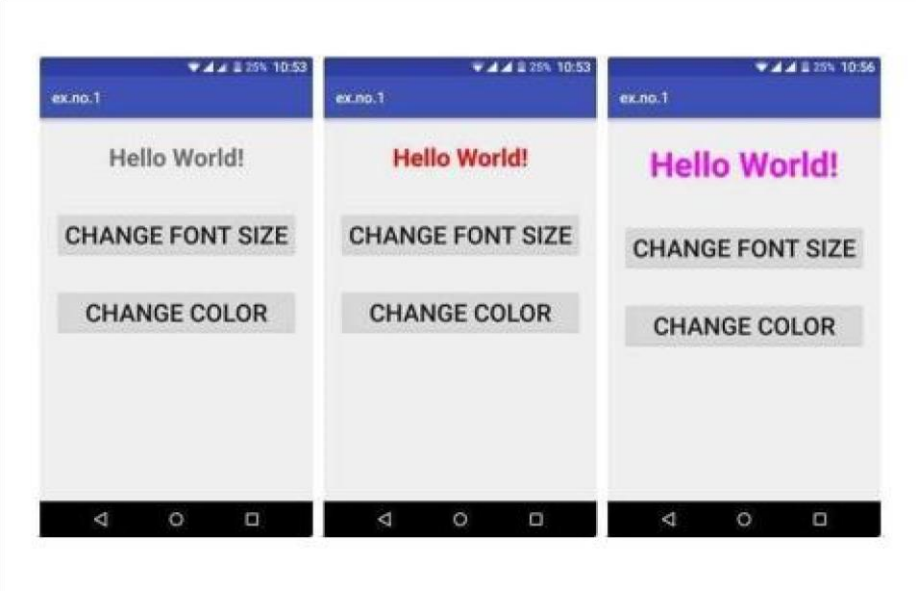
```
{
  intch=1;floa
  t
  font=30;@
  O verride
  protectedvoidonCreate(BundlesavedInstanceState)
  { super.onCreate(savedInstanceState);setContentView(R.layout.a ctivity_main); final
  TextView t= (TextView)
    findViewById(R.id.textView);Button  b1=  (Button)
    findViewById(R.id.button1);b1.setOnClickListener
    ( ne wView.OnClickListener(){
      @Override  public  void
      onClick(View
        v)
        {t.setTextSize(font); font
        = font +
        5;if (font
        == 50)font
        =30;
      }
    });
    Button b2= (Button)
    findViewById(R.id.button2);b2.setOnClickList
    en er(newView.OnClickListener(){
      @Override
```

```
public void onClick(View
  v)

    {switch(ch){ case1:


      t.setTextColor(Color.RED); break

    case2:
      t.setTextColor(Color.GREEN); break;

    case3:
      t.setTextColor(Color.BLUE)
      ; break;

    case4:
      t.setTextColor(Color.CYAN); break;

    case5:
      t.setTextColor(Color.YELLOW); break;
```

TEC[Computer Engg.] Page 123

```
case6: t.setTextColor(Color.MAGENTA);
  break;
}
}
});
}
}
ch++;
if ( ch == 7 ): ch
  == 1;
```

Output:



PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch:B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

```
package com.example.exno8;

import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;

public class MainActivity extends AppCompatActivity {
    private int colorIndex = 0;
    private float fontSize = 30f;
```

```
  private final int[] colors = {Color.RED, Color.GREEN, Color.BLUE, Color.CYAN, Color.YELLOW,
Color.BLACK};

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Load theme preference
    SharedPreferences preferences = getSharedPreferences("AppPrefs", MODE_PRIVATE);
    boolean isDarkMode = preferences.getBoolean("DarkMode", false);
    AppCompatDelegate.setDefaultNightMode(isDarkMode ?
        AppCompatDelegate.MODE_NIGHT_YES : AppCompatDelegate.MODE_NIGHT_NO);

    setContentView(R.layout.activity_main);

    // Initialize views
    TextView textView = findViewById(R.id.textView);
    Button buttonSize = findViewById(R.id.button1);
    Button buttonColor = findViewById(R.id.button2);
    Button buttonTheme = findViewById(R.id.button3);

    // Increase font size on button click
    buttonSize.setOnClickListener(v -> {
      fontSize += 5;
      if (fontSize > 50) {
        fontSize = 30;
      }
      textView.setTextSize(fontSize);
    });

    // Change text color on button click
    buttonColor.setOnClickListener(v -> {
      textView.setTextColor(colors[colorIndex]);
      colorIndex = (colorIndex + 1) % colors.length;
    });

    // Toggle Dark Mode
    buttonTheme.setOnClickListener(v -> {
```

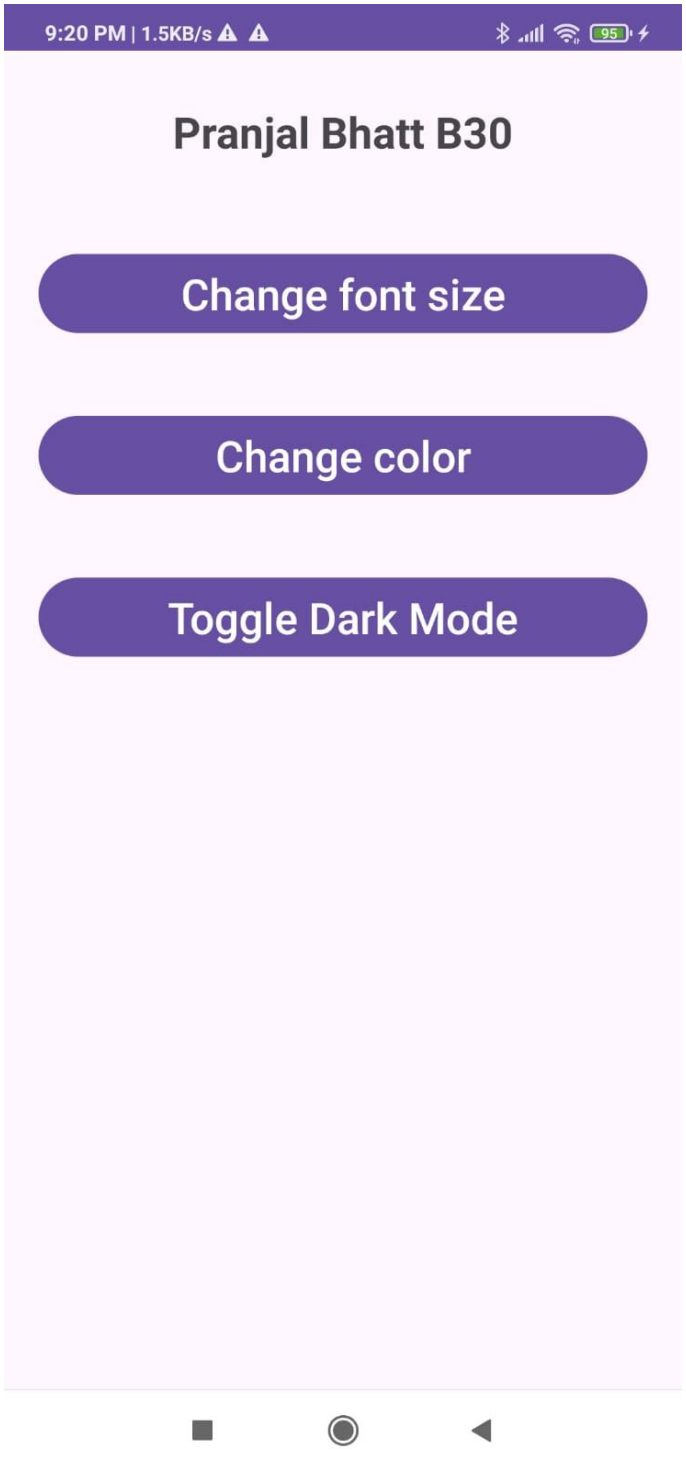
```
        boolean isDark = AppCompatDelegate.getDefaultNightMode() ==
AppCompatDelegate.MODE_NIGHT_YES;
        AppCompatDelegate.setDefaultNightMode(isDark ?
            AppCompatDelegate.MODE_NIGHT_NO : AppCompatDelegate.MODE_NIGHT_YES);

        // Save preference
        SharedPreferences.Editor editor = preferences.edit();
        editor.putBoolean("DarkMode", !isDark);
        editor.apply();

        // Restart activity to apply theme change
        recreate();
      });
   }
}
```

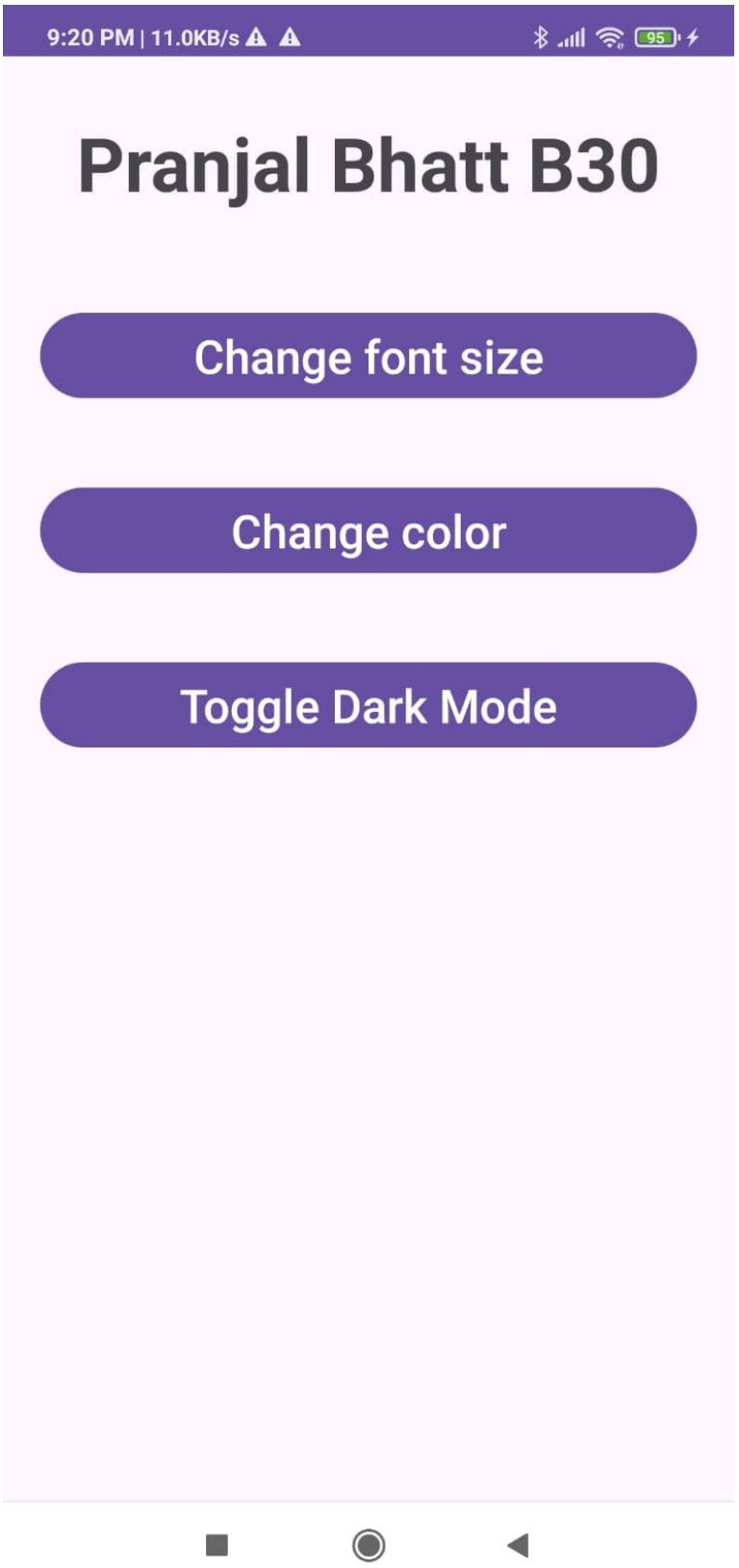
AndroidManifest.xml:

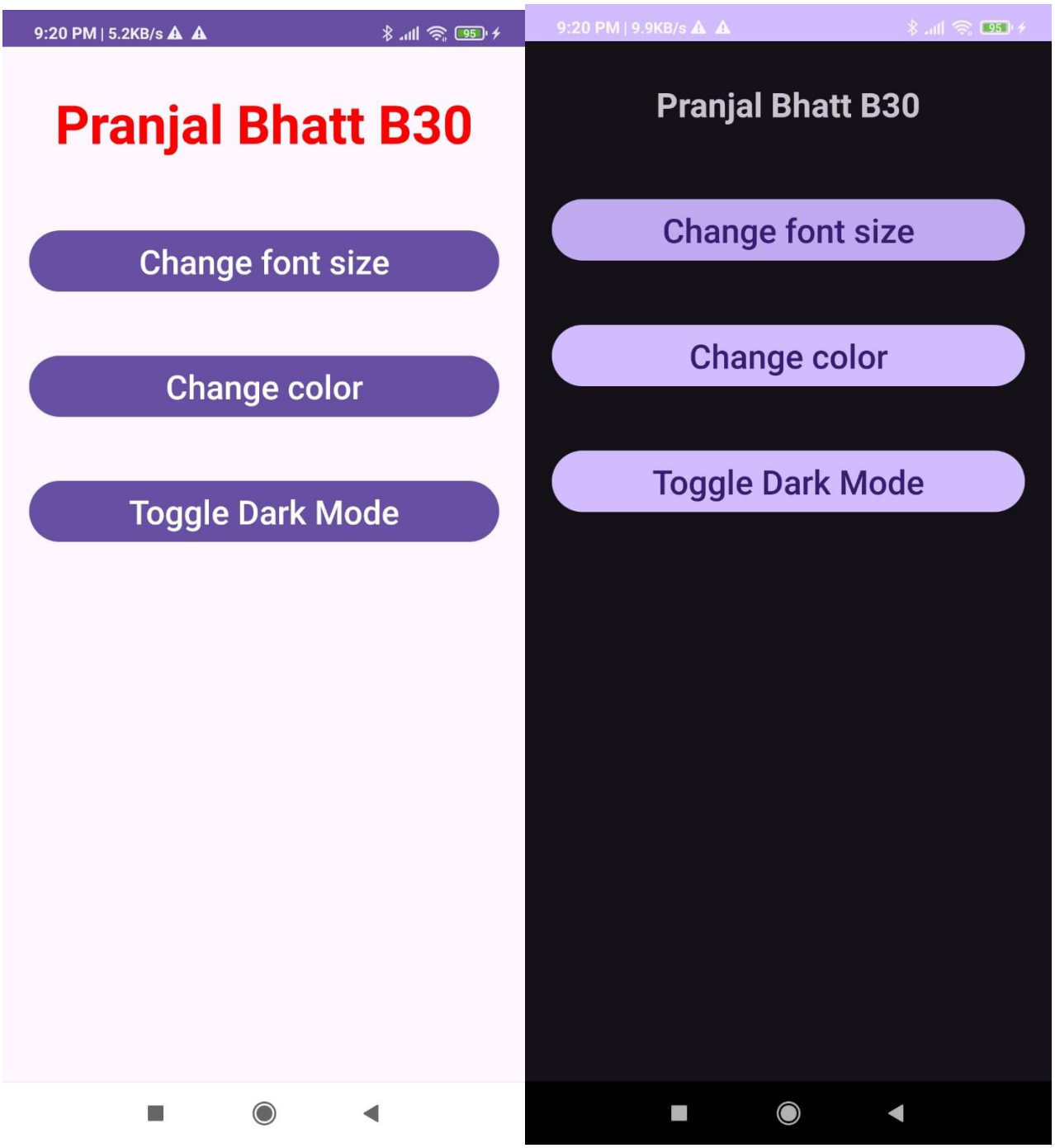
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools">

   <application
      android:allowBackup="true"
      android:dataExtractionRules="@xml/data_extraction_rules"
      android:fullBackupContent="@xml/backup_rules"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/Theme.Exno8"
      tools:targetApi="31">
      <activity
         android:name=".MainActivity"
         android:exported="true">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

```
</application>

</manifest>
```

B.2 Input and Output:




9:20 PM | 11.0KB/s
Pranjal Bhatt B30
Change font size
Change color
Toggle Dark Mode



B.3 Observations and learning:

During the experiment, an Android application was developed using Android Studio to implement GUI components, fonts, and colors. Various UI elements like TextView, Button, EditText, and ImageView were added and customized using different fonts and colors. XML and Java/Kotlin code were used to define the layout and functionality. The project successfully displayed a user interface with custom-styled

text, buttons, and background colors. Testing was conducted on an emulator and a physical device to verify proper rendering of the UI components.

B.4 Conclusion:

The experiment provided hands-on experience in designing Android user interfaces with GUI components, fonts, and colors. Students learned how to customize UI elements using XML properties and Java/Kotlin code. This experiment helped them understand the importance of UI design in application development and how to enhance the visual appeal of an Android application.

B.5 Question of Curiosity

1) Explain different steps required to build up this project?

Steps Required to Build the Project:

Set Up Development Environment:

1. Install Android Studio, Java Development Kit (JDK), and Android SDK.
2. Configure Gradle dependencies.

Create a New Android Project:

1. Open Android Studio and select "New Project."
2. Choose an appropriate project template (e.g., Empty Activity).
3. Enter the application name, package name, and select the minimum Android version.
4. Click “Finish” to create the project.

Modify `activity_main.xml` to Add GUI Components:

1. Navigate to `res/layout/activity_main.xml`.
2. Add UI components like `TextView`, `EditText`, `Button`, and `ImageView`:

Customize Fonts and Colors:

- Define custom fonts by placing `.ttf` files inside `res/font/`.
- Apply the custom font in `TextView`

Write Java/Kotlin Code to Handle Button Clicks:

- Open `MainActivity.java` and modify button functionality:

Run and Test the Application:

- Launch the emulator or use a physical device.
- Verify that UI components display correctly.
- Test button functionality and color changes.

Debug and Optimize:

- Check logs in Logcat for debugging.
- Adjust font sizes and colors for better readability.