

2. Create “Hello World” application. That will display “Hello World” in the middle of the screen using TextView Widget in the red color.

MainActivity.java

```
package com.practical.second;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"> <!-- White background -->

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="Hello World!"
        android:textColor="#FF0000" /> <!-- Red text color -->

</RelativeLayout>
```

OUTPUT



Hello World!

3. Create Registration page to demonstration of Basic widgets available in android.

MainActivity.java

```
package com.practical.third;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText etUsername, etEmail, etPassword;
    private CheckBox cbAgree;
    private Button btnRegister;

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

        // Initialize the views
        etUsername = findViewById(R.id.etUsername);
        etEmail = findViewById(R.id.etEmail);
        etPassword = findViewById(R.id.etPassword);
        cbAgree = findViewById(R.id.cbAgree);
        btnRegister = findViewById(R.id.btnRegister);

        // Set a click listener for the register button
        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Get the entered values
                String username = etUsername.getText().toString().trim();
                String email = etEmail.getText().toString().trim();
                String password = etPassword.getText().toString().trim();
                boolean isAgreeChecked = cbAgree.isChecked();

                // Validate the input
                if (username.isEmpty() || email.isEmpty() || password.isEmpty()) {
                    Toast.makeText(MainActivity.this, "All fields are required",
                        Toast.LENGTH_SHORT).show();
                } else if (!isAgreeChecked) {
                    Toast.makeText(MainActivity.this, "You must agree to the terms and conditions",
                        Toast.LENGTH_SHORT).show();
                } else if (!isValidEmail(email)) {
```

```

        Toast.makeText(MainActivity.this, "Invalid email address",
Toast.LENGTH_SHORT).show();
    } else if (password.length() < 8) {
        Toast.makeText(MainActivity.this, "Password must be at least 8 characters long",
Toast.LENGTH_SHORT).show();
    } else {
        // Proceed with registration (for now, just show a toast)
        Toast.makeText(MainActivity.this, "Registration successful",
Toast.LENGTH_SHORT).show();
    }
}
});
}

private boolean isValidEmail(String email) {
    String emailRegex = "^[A-Z0-9._%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,6}$";
    return email.matches(emailRegex);
}
}

```

activity_main.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/etUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username" />

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:hint="Email" />

    <EditText
        android:id="@+id/etPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:hint="Password" />

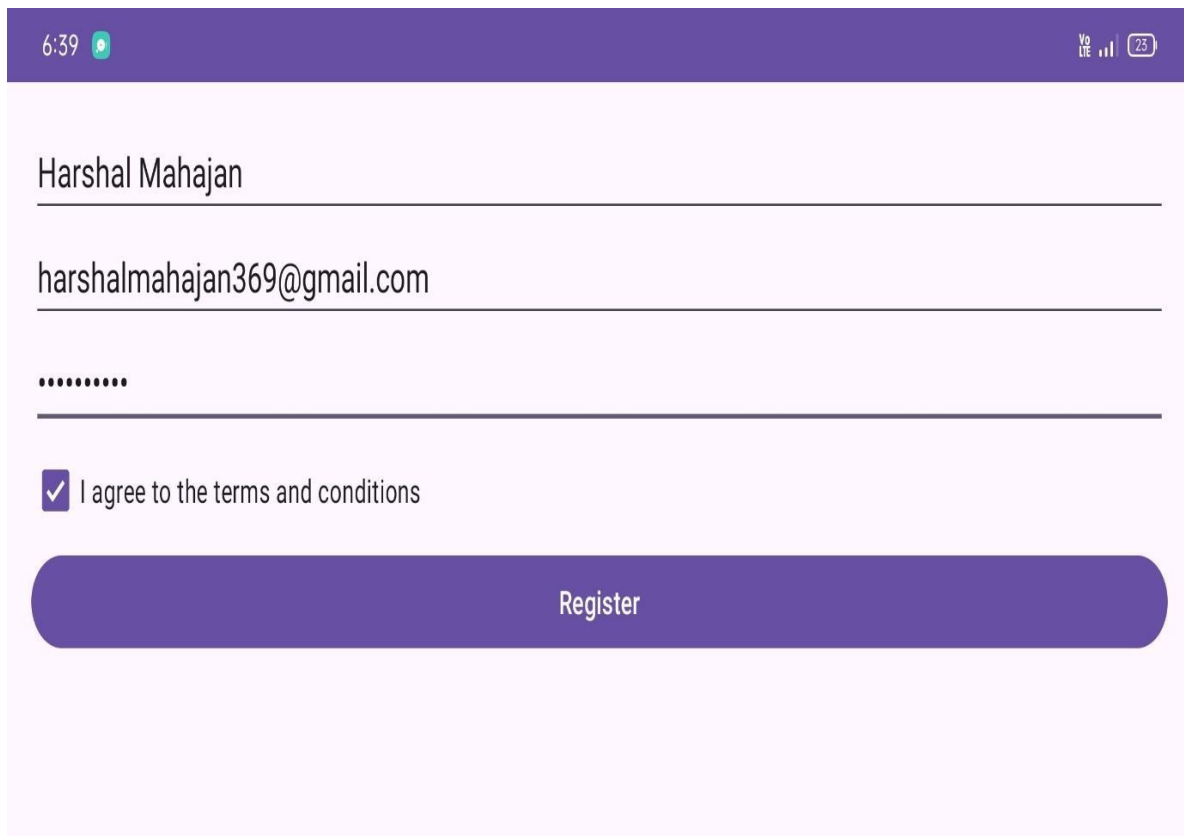
```

```
<CheckBox
    android:id="@+id/cbAgree"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="I agree to the terms and conditions" />
```

```
<Button
    android:id="@+id/btnRegister"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Register" />
```

```
</LinearLayout>
```

OUTPUT



6:39 VoLTE 23

Harshal Mahajan

harshalmahajan369@gmail.com

.....

☒ I agree to the terms and conditions

Register

4. Create sample application with login module.(Check username and password) On successful login, Change TextView "Login Successful". And on failing login, alert user using Toast "Login fail".

activity_main.xml

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

    <!-- Title -->
    <TextView
        android:id="@+id/tvLoginTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login Module"
        android:textSize="24sp"
        android:layout_gravity="center"
        android:paddingBottom="20dp"/>

    <!-- Username -->
    <TextView
        android:id="@+id/tvUsername"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Username:"
        android:textSize="16sp"/>

    <EditText
        android:id="@+id/etUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter username"
        android:inputType="text"/>

    <!-- Password -->
    <TextView
        android:id="@+id/tvPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password:"
        android:textSize="16sp"
        android:layout_marginTop="16dp"/>

    <EditText
        android:id="@+id/etPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter password">
```

```

        android:inputType="textPassword"/>

<!-- Login Button -->
<Button
    android:id="@+id/btnLogin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Login"
    android:layout_marginTop="20dp"/>

<!-- Login Result -->
<TextView
    android:id="@+id/tvLoginResult"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="18sp"
    android:textColor="#008000"
    android:layout_marginTop="20dp"
    android:layout_gravity="center"/>

</LinearLayout>

```

MainActivity.java

```

package com.practical.four;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText etUsername, etPassword;
    private Button btnLogin;
    private TextView tvLoginResult;

    // Dummy credentials
    private static final String VALID_USERNAME = "user123";
    private static final String VALID_PASSWORD = "password123";

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

        // Initialize the views

```

```

etUsername = findViewById(R.id.etUsername);
etPassword = findViewById(R.id.etPassword);
btnLogin = findViewById(R.id.btnLogin);
tvLoginResult = findViewById(R.id.tvLoginResult);

// Set a click listener for the login button
btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Get the entered values
        String username = etUsername.getText().toString();
        String password = etPassword.getText().toString();

        // Validate the username and password
        if (username.equals(VALID_USERNAME) && password.equals(VALID_PASSWORD)) {
            // If credentials are correct, show "Login Successful"
            tvLoginResult.setText("Login Successful");
            tvLoginResult.setTextColor(getResources().getColor(android.R.color.holo_green_dark));
        } else {
            // If credentials are incorrect, show a Toast message
            tvLoginResult.setText("");
            Toast.makeText(MainActivity.this, "Login failed", Toast.LENGTH_SHORT).show();
        }
    }
});
}
}

```

OUTPUT

6:40

6:40

Y9 4G LTE 25

Login Module

Username:

Tushar Sonar

Password:

.....

Login

5. Create an application for demonstration of Scroll view in android.

activity_main.xml

```
<?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">

    <!-- Vertical ScrollView -->
    <ScrollView
        android:id="@+id/verticalScrollView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

                <!-- Add multiple buttons to demonstrate vertical scrolling -->
                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Button 1" />
                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Button 2" />
                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Button 3" />
                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Button 4" />
                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Button 5" />
                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Button 6" />
                <Button
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Button 7" />
            </LinearLayout>
        </ScrollView>
    </LinearLayout>
```



```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 8" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 9" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 10" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 11" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 12" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 13" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 14" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 15" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 16" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 17" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 18" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Button 19" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
        android:text="Button 20" />
```

```
</LinearLayout>
```

```
</ScrollView>
```

```
<!-- Horizontal ScrollView -->
```

```
<HorizontalScrollView
```

```
    android:id="@+id/horizontalScrollView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">
```

```
    <LinearLayout
```

```
        android:layout_width="wrap_content"  
        android:layout_height="match_parent"  
        android:orientation="horizontal">
```

```
        <!-- Add multiple buttons to demonstrate horizontal scrolling -->
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H1" />
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H2" />
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H3" />
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H4" />
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H5" />
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H6" />
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H7" />
```

```
        <Button
```

```
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="Button H8" />
```

```
</LinearLayout>
```

```
</HorizontalScrollView>
```

```
</LinearLayout>
```

MainActivity.java

```
package com.practical.fifth;
```

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

```
import android.os.Bundle;
```

```
import android.widget.ScrollView;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        // Example: Accessing the ScrollView programmatically
```

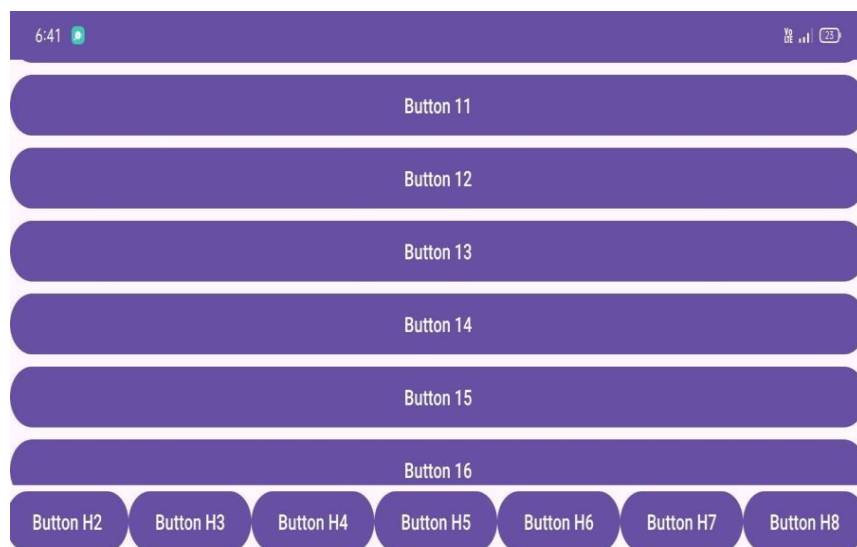
```
        ScrollView verticalScrollView = findViewById(R.id.verticalScrollView);
```

```
        // You can add listeners or modify properties here if needed
```

```
    }
```

```
}
```

OUTPUT



6. Create login application where you will have to validate username and passwords till the username and password is not validated, login button should remain disabled.

activity_main.xml

```
<?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"
    android:padding="20dp"
    android:gravity="center">

    <EditText
        android:id="@+id/editTextUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:inputType="text" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:layout_marginTop="20dp" />

    <Button
        android:id="@+id/buttonLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login"
        android:layout_marginTop="20dp"
        android:enabled="false" />

</LinearLayout>
```

MainActivity.java

```
package com.practical.sixth;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
```

```

private EditText editTextUsername, editTextPassword;
private Button buttonLogin;

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

    // Initialize the views
    editTextUsername = findViewById(R.id.editTextUsername);
    editTextPassword = findViewById(R.id.editTextPassword);
    buttonLogin = findViewById(R.id.buttonLogin);

    // Add TextChangedListeners to validate input
    editTextUsername.addTextChangedListener(new android.text.TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int start, int count, int after) {
        }

        @Override
        public void onTextChanged(CharSequence charSequence, int start, int before, int count) {
            validateInputs();
        }

        @Override
        public void afterTextChanged(android.text.Editable editable) {
        }
    });

    editTextPassword.addTextChangedListener(new android.text.TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int start, int count, int after) {
        }

        @Override
        public void onTextChanged(CharSequence charSequence, int start, int before, int count) {
            validateInputs();
        }

        @Override
        public void afterTextChanged(android.text.Editable editable) {
        }
    });

    // Set up click listener for login button (optional)
    buttonLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Handle login logic here (e.g., authenticate user)
            // This is just a placeholder for real authentication logic
            String username = editTextUsername.getText().toString();

```

```

String password = editTextPassword.getText().toString();

if (validateLogin(username, password)) {
    // Proceed with the login process
    // For now, just show a success message
    // Example: proceed to next activity
    // Intent intent = new Intent(MainActivity.this, HomeActivity.class);
    // startActivity(intent);
    // finish();

    // For now, show a simple message
    showMessage("Login Successful!");
} else {
    showMessage("Invalid credentials!");
}
}
});
}

private void validateInputs() {
    String username = editTextUsername.getText().toString();
    String password = editTextPassword.getText().toString();

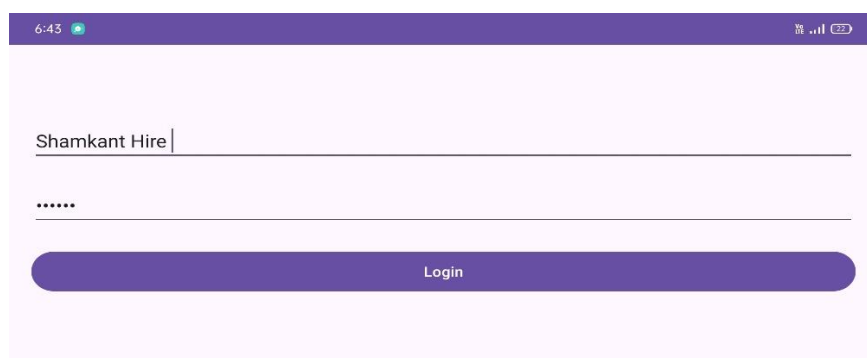
    // Enable login button only if both fields are non-empty
    buttonLogin.setEnabled(!TextUtils.isEmpty(username) && !TextUtils.isEmpty(password));
}

private boolean validateLogin(String username, String password) {
    // This is a placeholder logic for validation.
    // You can replace it with actual login logic.
    return username.equals("user") && password.equals("password");
}

private void showMessage(String message) {
    // Display a simple message (you can replace this with Toast or other UI elements)
    android.widget.Toast.makeText(MainActivity.this, message,
    android.widget.Toast.LENGTH_SHORT).show();
}
}

```

OUTPUT



7. Create an application for calculator.

MainActivity.java

```
package com.practical.seventh;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private EditText number1, number2;
    private Button add, subtract, multiply, divide, clear;
    private TextView result;

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

        // Initialize views
        number1 = findViewById(R.id.number1);
        number2 = findViewById(R.id.number2);
        add = findViewById(R.id.add);
        subtract = findViewById(R.id.subtract);
        multiply = findViewById(R.id.multiply);
        divide = findViewById(R.id.divide);
        clear = findViewById(R.id.clear);
        result = findViewById(R.id.result);

        // Set click listeners
        add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performOperation("+");
            }
        });

        subtract.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performOperation("-");
            }
        });

        multiply.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View v) {
            performOperation("*");
        }
    });

    divide.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            performOperation("/");
        }
    });

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

private void performOperation(String op) {
    try {
        double num1 = Double.parseDouble(number1.getText().toString());
        double num2 = Double.parseDouble(number2.getText().toString());

        double res = 0;
        switch (op) {
            case "+":
                res = num1 + num2;
                break;
            case "-":
                res = num1 - num2;
                break;
            case "*":
                res = num1 * num2;
                break;
            case "/":
                if (num2 != 0) {
                    res = num1 / num2;
                } else {
                    Toast.makeText(this, "Cannot divide by zero!", Toast.LENGTH_SHORT).show();
                    return;
                }
                break;
        }

        result.setText("Result: " + res);
    } catch (NumberFormatException e) {
        Toast.makeText(this, "Enter valid numbers!", Toast.LENGTH_SHORT).show();
    }
}

```



```

    }

    private void clearFields() {
        number1.setText("");
        number2.setText("");
        result.setText("Result: ");
    }
}

```

activity_main.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:id="@+id/number1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Number 1" />

    <EditText
        android:id="@+id/number2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Number 2" />

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

    <Button
        android:id="@+id/subtract"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-" />

    <Button
        android:id="@+id/multiply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="*" />

    <Button
        android:id="@+id/divide"
        android:layout_width="wrap_content"

```

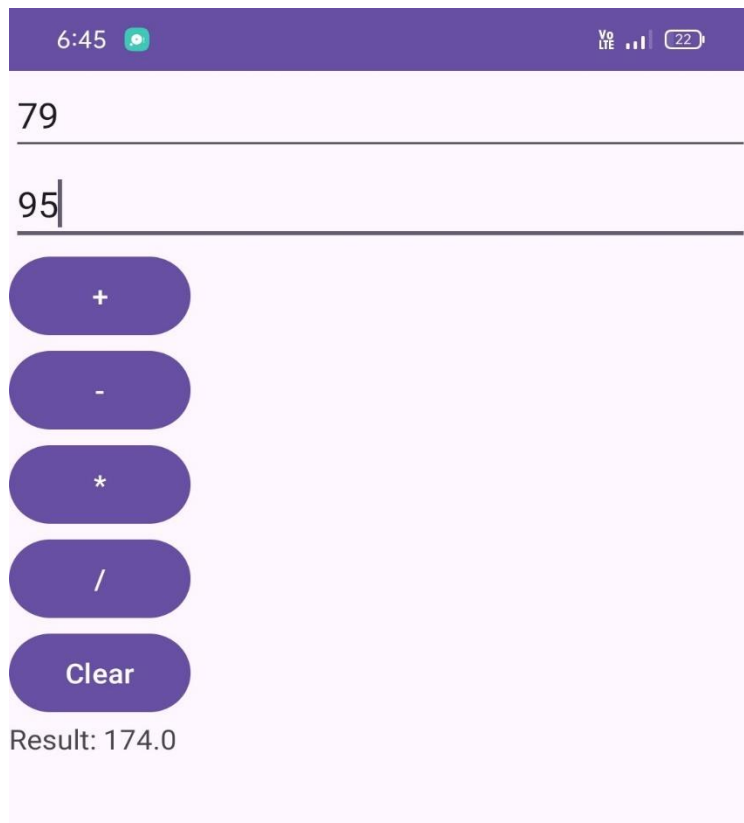
```
android:layout_height="wrap_content"  
android:text="/" />
```

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

```
<TextView  
    android:id="@+id/result"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Result: " />
```

```
</LinearLayout>
```

OUTPUT



8. Demonstrate use of scroll view.

activity_main.xml

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <!-- Add multiple views here to demonstrate scrolling -->
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 1" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 2" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 3" />

        <!-- Add more buttons to exceed the screen height -->
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 4" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 5" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 6" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 7" />

    </LinearLayout>

</ScrollView>
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button 8" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button 9" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button 10" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button 11" />
```

```
</LinearLayout>
```

```
</ScrollView>
```

MainActivity.java

```
package com.practical.eight;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
}
```

OUTPUT



9. Demonstrate use of intent in android

activity_second.xml

```
<?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:gravity="center"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Second Page!"
        android:textSize="20sp"/>
    <Button
        android:id="@+id/button_back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go Back"/>
</LinearLayout>
```

SecondActivity.java

```
package com.practical.ninth;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        Button backButton = findViewById(R.id.button_back);
        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish(); // This closes SecondActivity and returns to MainActivity
            }
        });
    }
}
```

activity_main.xml

```
<?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:gravity="center"
android:orientation="vertical">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
android:textSize="20sp"/>
<Button
android:id="@+id/button_next"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Go to Second Page"/>
</LinearLayout>

```

MainActivity.java

```
package com.practical.ninth;
```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button nextButton = findViewById(R.id.button_next);
        nextButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

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"

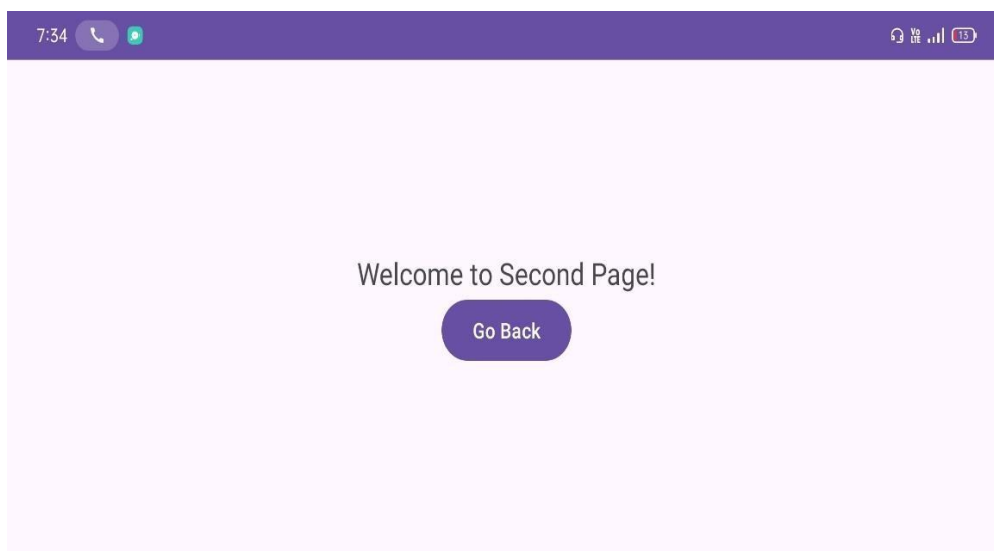
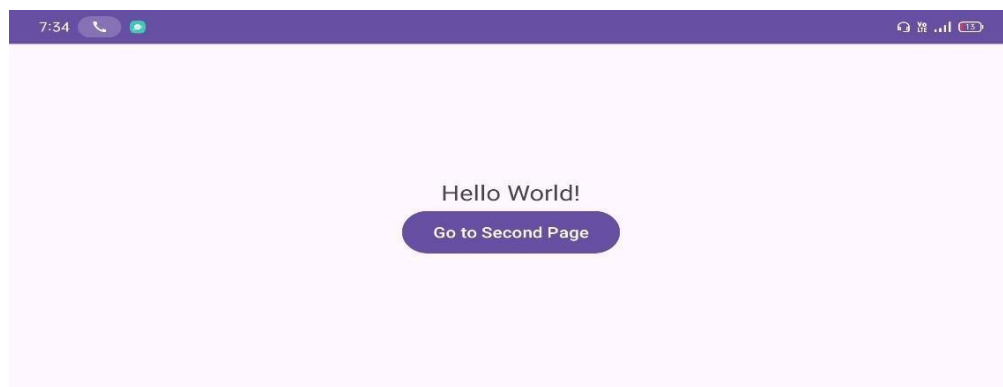
```

```
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/Theme.Ninth"
    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>
    <activity android:name=".SecondActivity" />
</application>

</manifest>
```

OUTPUT



10. Create application to demonstrate menu option.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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_height="match_parent"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/ThemeOverlay.MaterialComponents.Dark.ActionBar"
        app:layout_constraintTop_toTopOf="parent" >

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="#03A9F4"
            app:popupTheme="@style/ThemeOverlay.MaterialComponents.Light" />

    </com.google.android.material.appbar.AppBarLayout>

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Radhe Radhe"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

menu_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu_home"
        android:title="Home" />
    <item
        android:id="@+id/menu_help"
        android:title="Help" />
</menu>
```



```

<item
    android:id="@+id/menu_settings"
    android:title="Settings" />
<item
    android:id="@+id/menu_about"
    android:title="About Us" />
<item
    android:id="@+id/menu_exit"
    android:title="Exit" />
</menu>

```

MainActivity.java

```

package com.practical.tenth;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.menu_home) {
            Toast.makeText(this, "Home selected", Toast.LENGTH_SHORT).show();
        } else if (id == R.id.menu_help) {
            Toast.makeText(this, "Help selected", Toast.LENGTH_SHORT).show();
        } else if (id == R.id.menu_settings) {
            Toast.makeText(this, "Settings selected", Toast.LENGTH_SHORT).show();
        } else if (id == R.id.menu_about) {
            Toast.makeText(this, "About Us selected", Toast.LENGTH_SHORT).show();
        }
    }
}

```

```
    } else if (id == R.id.menu_exit) {  
        finish(); // Exit the app  
    }  
  
    return super.onOptionsItemSelected(item);  
}  
}
```

OUTPUT



11. Create application to demonstrate progress bar.

MainActivity.java

```
package com.practical.eleventh;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;

public class MainActivity extends AppCompatActivity {

    private ProgressBar progressBar;
    private Button startButton;
    private int progress = 0;

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

        progressBar = findViewById(R.id.progressBar);
        startButton = findViewById(R.id.startButton);

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

    private void startProgress() {
        // Reset progress if it's already completed
        if (progress >= progressBar.getMax()) {
            progress = 0;
            progressBar.setProgress(progress);
        }

        // Simulate progress update
        Thread thread = new Thread(new Runnable() {
            @Override
            public void run() {
                while (progress < progressBar.getMax()) {
                    try {
                        Thread.sleep(50); // Update every 50 milliseconds
                    } catch (InterruptedException e) {
                        Thread.currentThread().interrupt();
                    }
                }
            }
        });
    }
}
```

```

        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                progressBar.setProgress(progress);
                progress++;
            }
        });
    }
}
});
thread.start();
}
}

```

activity_main.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <ProgressBar
        android:id="@+id/progressBar"
        style="@style/Widget.AppCompat.ProgressBar.Horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:indeterminate="false"
        android:max="100"
        android:progress="0" />

    <Button
        android:id="@+id/startButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Progress" />

</LinearLayout>

```

OUTPUT

