Practical : 1

Create an Android application and understand the Project and file hierarchy.

- **Theory:**

1. Android Project Hierarchy:
2. app/src/main/java: This is where your Java (or Kotlin) source code resides.
3. app/src/main/res: Resources for your app, including XML layout files, images, strings, etc.
4. app/src/main/AndroidManifest.xml: The manifest file describes essential information about your app to the Android system, such as activities, permissions, and more.
5. app/build.gradle: The build file for your app, where you can configure various settings like dependencies, version codes, and more.
6. Project file hierarchy:

```
- app
  - src
    - main
      - java
        - com.example.simpleapp
          - MainActivity.java
      - res
        - layout
          - activity_main.xml
      - AndroidManifest.xml
      - build.gradle
```

- **Code:**

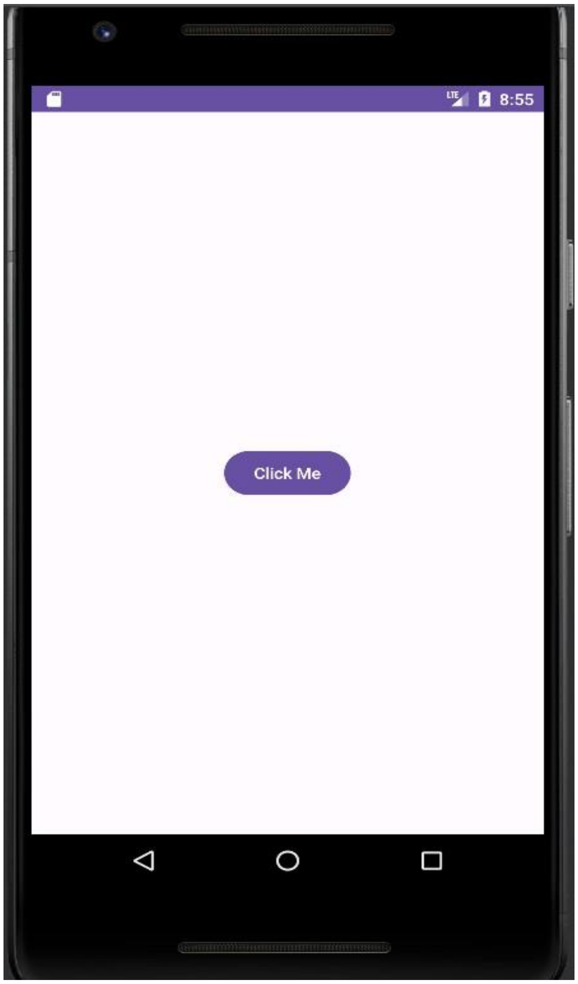
Activity_main.xml

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

    <Button
        android:id="@+id/clickButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Me"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

Main_Activity.java :

```
// com.example.simpleapp.MainActivity.java
package com.example.simpleapp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

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 clickButton = findViewById(R.id.clickButton);
        clickButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showToast();
            }
        });
    }

    private void showToast() {
        Toast.makeText(this, "Hello, Android!", Toast.LENGTH_SHORT).show();
    }
}
```

- **Output :**



Practical: 2
Develop an Android application that uses GUI components, Font, and Colors

- **Code:**

Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <color name="black">#FF000000</color>
   <color name="white">#FFFFFFFF</color>
   <color name="customTextColor">#FF5733</color>
   <color name="customButtonColor">#3377FF</color>
   <color name="defaultBackgroundColor">#FFFFFF</color>
   <color name="changedBackgroundColor">#FF5733</color>
</resources>
```

- activity_main.xml

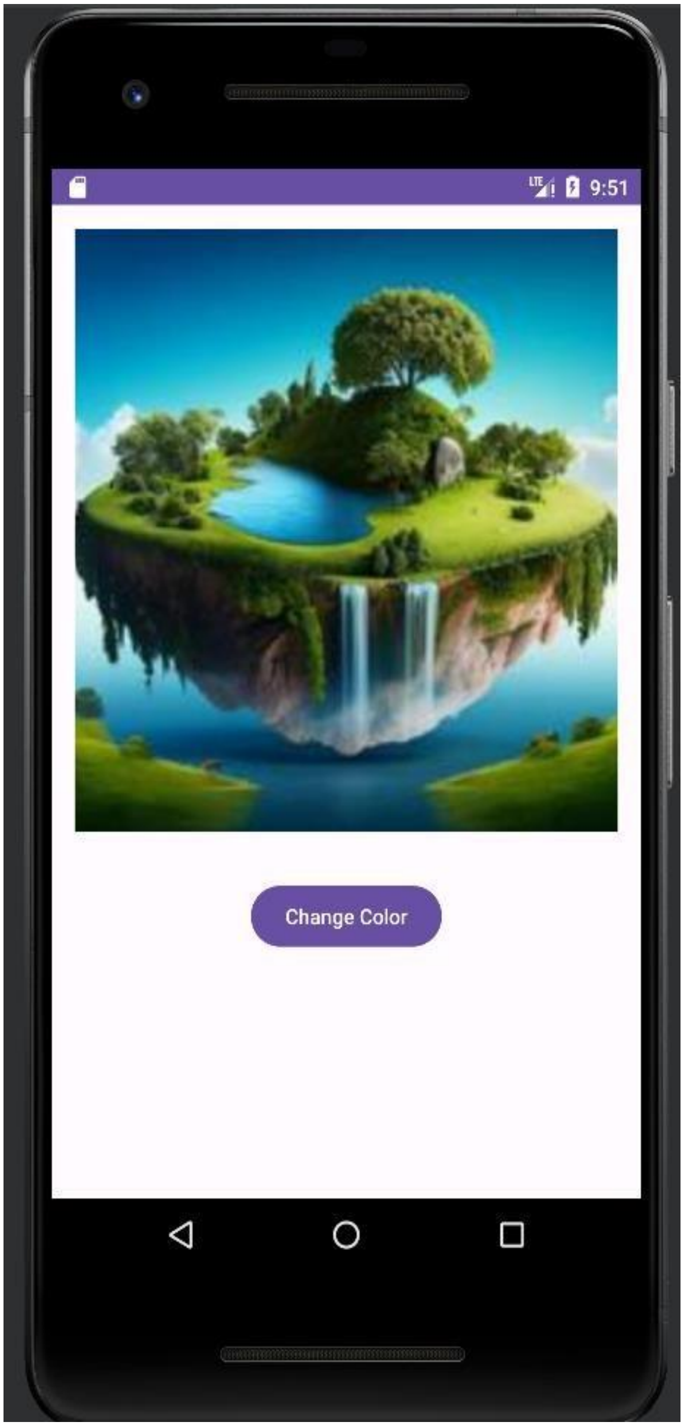
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:padding="16dp">

   <ImageView
      android:id="@+id/imageView"
      android:layout_width="match_parent"
      android:layout_height="400dp"
      android:src="@drawable/photo"
      android:scaleType="centerCrop"
      android:layout_centerHorizontal="true"
      android:layout_marginBottom="16dp"/>

   <Button
      android:id="@+id/colorChangeButton"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Change Color"
      android:layout_below="@id/imageView"
      android:layout_centerHorizontal="true"
```

```
        android:layout_marginTop="16dp"/>
</RelativeLayout>
```

- MainActivity.java

```
// com.example.colorchangeapp.MainActivity.java
package com.example.madp1;


import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;


import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {


    private ImageView imageView;
    private Button colorChangeButton;
    private boolean isColorChanged = false;


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


        imageView = findViewById(R.id.imageView);
        colorChangeButton = findViewById(R.id.colorChangeButton);
        colorChangeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                changeBackgroundColor();
            }
        });
    }

    private void changeBackgroundColor() {
        if (isColorChanged) {
            // Reset to the default color
            imageView.setBackgroundResource(R.color.defaultBackgroundColor);
        } else {
            // Change to a different color
            imageView.setBackgroundResource(R.color.changedBackgroundColor);
        }

        isColorChanged = !isColorChanged;
    }
}
```

- **Output:**



Practical: 2

Develop an Android application that uses Layout Managers and event listeners. Subscription form and Login form

- **Code:**

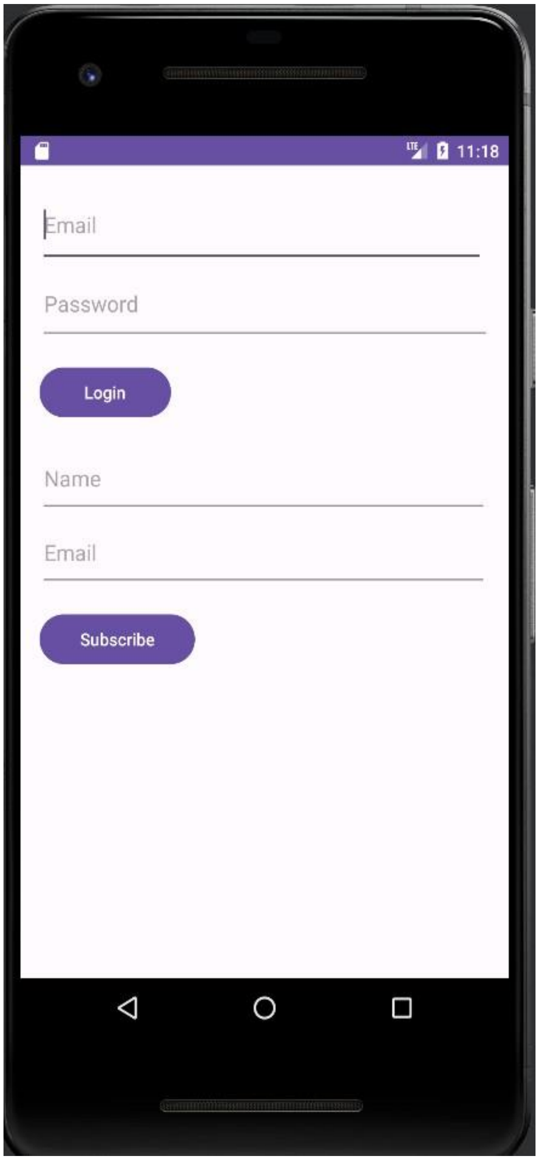
Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <!-- Login Form -->
    <EditText
        android:id="@+id/emailEditText"
        android:layout_width="375dp"
        android:layout_height="65dp"
        android:hint="Email" />

    <EditText
        android:id="@+id/passwordEditText"
        android:layout_width="380dp"
        android:layout_height="63dp"
        android:hint="Password"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/loginButton"
        android:layout_width="111dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Login" />
    <!-- Subscription Form -->
    <EditText
        android:id="@+id/nameEditText"
        android:layout_width="378dp"
        android:layout_height="60dp"
        android:layout_marginTop="16dp"
        android:hint="Name" />

    <EditText
        android:id="@+id/emailSubscribeEditText"
        android:layout_width="378dp"
        android:layout_height="60dp"
        android:hint="Email" />

    <Button
        android:id="@+id/subscribeButton"
        android:layout_width="131dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Subscribe" />

</LinearLayout>
```

MainActivity.java

```
// com.example.formsapp.MainActivity.java
package com.example.madp1;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText emailEditText, passwordEditText;
    private Button loginButton;
    private EditText nameEditText, emailSubscribeEditText;
    private Button subscribeButton;

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

        // Login Form
        emailEditText = findViewById(R.id.emailEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        loginButton = findViewById(R.id.loginButton);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                loginUser();
            }
        });

        // Subscription Form
        nameEditText = findViewById(R.id.nameEditText);
        emailSubscribeEditText = findViewById(R.id.emailSubscribeEditText);
        subscribeButton = findViewById(R.id.subscribeButton);

        subscribeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                subscribeUser();
            }
        });
    }

    private void loginUser() {
        String email = emailEditText.getText().toString();
        String password = passwordEditText.getText().toString();

        // Add your login logic here
        Toast.makeText(this, "Login: Email - " + email + ", Password - " + password,
Toast.LENGTH_SHORT).show();
    }
    private void subscribeUser() {
        String name = nameEditText.getText().toString();
        String email = emailSubscribeEditText.getText().toString();

        // Add your subscription logic here
        Toast.makeText(this, "Subscribe: Name - " + name + ", Email - " + email,
Toast.LENGTH_SHORT).show();
    }
}
```

Output:



Practical: 3

Create a fragment that has its own UI and enables your activities to communicate with fragments.

Code:

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">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/button"/>

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Replace Fragment"
        android:layout_alignParentBottom="true"/>
</RelativeLayout>
```

Fragment_example.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:gravity="center">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fragment Content"
```

```xml
        android:textSize="18sp"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click me in Fragment"/>
</LinearLayout>
```

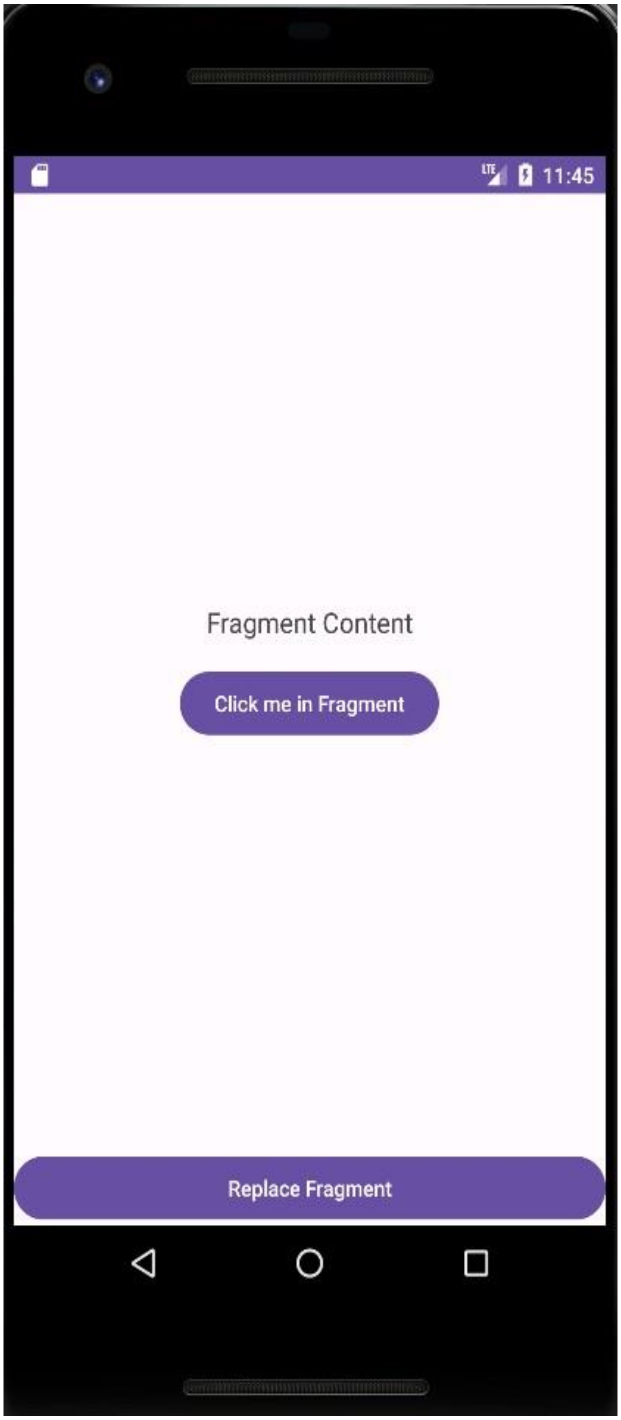
Fragment_example.java

```java
// com.example.fragmentcommunication.ExampleFragment.java
package com.example.madp1;

import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import androidx.fragment.app.Fragment;

public class fragment_example extends Fragment {

    private OnButtonClickListener mListener;

    public interface OnButtonClickListener {
        void onButtonClicked(String message);
    }

    public fragment_example() {
        // Required empty public constructor
    }

    public static fragment_example newInstance(String message) {
        fragment_example fragment_example = new fragment_example();
        Bundle args = new Bundle();
        args.putString("message", message);
        fragment_example.setArguments(args);
        return fragment_example;
    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnButtonClickListener) {
            mListener = (OnButtonClickListener) context;
        } else {
            throw new RuntimeException(context.toString()
                    + " must implement OnButtonClickListener");
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_example, container, false);

        Button button = view.findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mListener != null) {
                    mListener.onButtonClicked("Hello from Fragment!");
                }
            }
        });

        return view;
    }
}
```

MainActivity.java

```
// com.example.fragmentcommunication.MainActivity.java
package com.example.madp1;

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

public class MainActivity extends AppCompatActivity implements
fragment_example.OnButtonClickListener {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (savedInstanceState == null) {
            getSupportFragmentManager().beginTransaction()
                    .replace(R.id.fragment_container, new fragment_example())
                    .commit();
        }
    }

    @Override
    public void onButtonClicked(String message) {
        // This method is called when the button in the fragment is clicked
        // You can perform any action here based on the communication from the fragment
        getSupportFragmentManager().beginTransaction()
                .replace(R.id.fragment_container, fragment_example.newInstance(message))
                .addToBackStack(null)
                .commit();
    }
}
```

Output:


11:45
Fragment Content
Click me in Fragment
Replace Fragment

Practical: 4

Write an android code to make simple registration page using Intent

- **Code:**

Activity_registration.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">

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

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

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

    <Button
        android:id="@+id/registerButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register"
        android:layout_marginTop="16dp"/>
</LinearLayout>
```

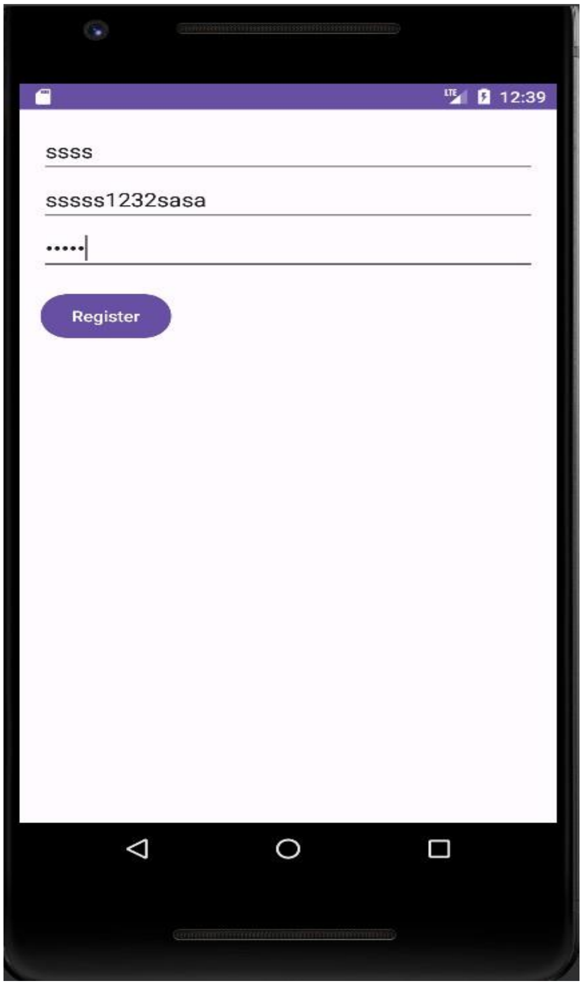
activity_registration.java

```
package com.example.mad5;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class activity_registration extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
    }
}
```

String.xml

```
<resources>
    <string name="welcome_message">Welcome, %1$s! Your email is %2$s.</string>
</resources>
```

Output:



ssss

sssss1232sasa

•••••

Practical: 5

Write an application to demonstrate Alert Dialog Box in android.

- **CODE:**

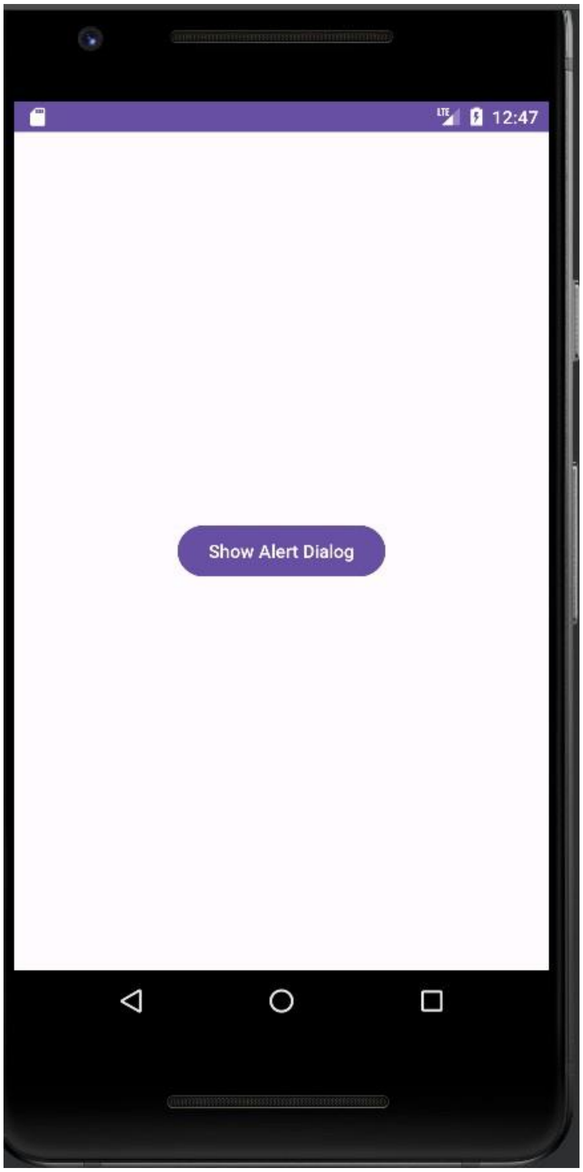
Activity_main.xml

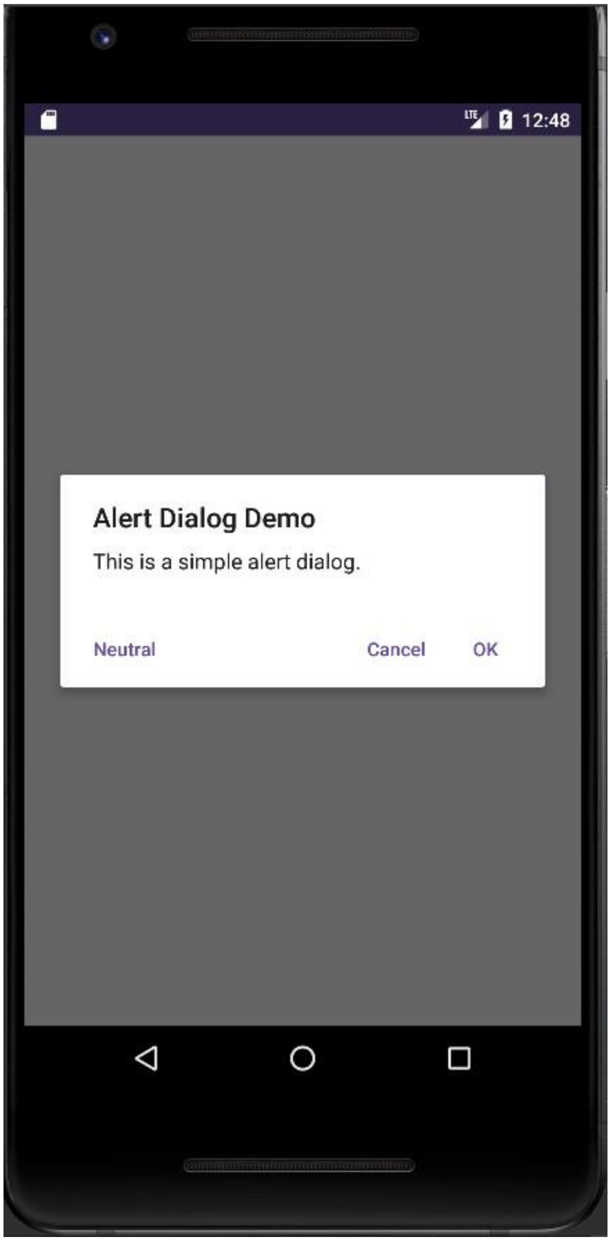
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <Button
        android:id="@+id/showDialogButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Alert Dialog"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

- MainActivity.java

```
// com.example.alertdialogdemo.MainActivity.java

package com.example.mad5;


import android.os.Bundle;

import android.view.View;

import android.widget.Button;


import androidx.appcompat.app.AlertDialog;

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 showDialogButton = findViewById(R.id.showDialogButton);

        showDialogButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showAlertDialog();
            }
        });
    }

    private void showAlertDialog() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Alert Dialog Demo")
                .setMessage("This is a simple alert dialog.")
                .setPositiveButton("OK", null)
                .setNegativeButton("Cancel", null)
                .setNeutralButton("Neutral", null);

        AlertDialog alertDialog = builder.create();
        alertDialog.show();
    }
}
```

Output:





Alert Dialog Demo

This is a simple alert dialog.

Practical: 6

Demonstrate Array Adapter using List View to display list of fruits.

- Code

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"
  android:padding="16dp">
  <ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
</RelativeLayout>
```

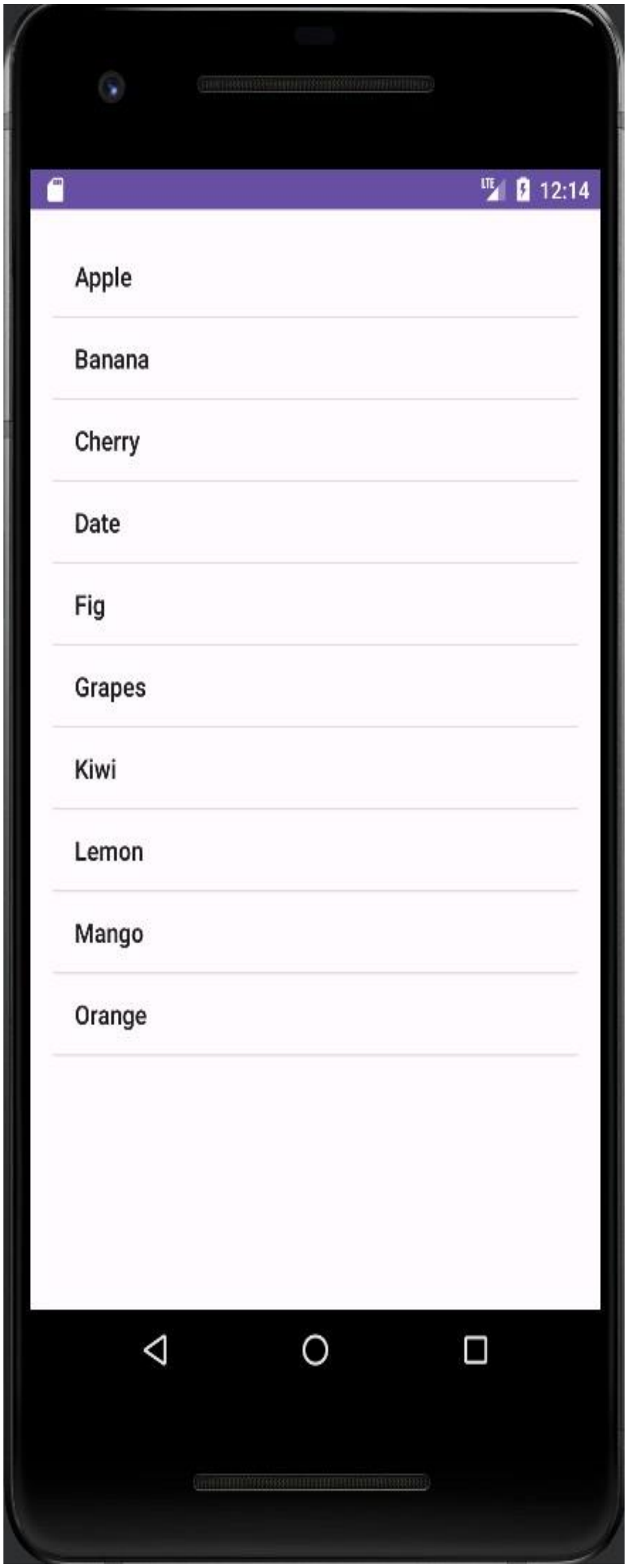
MainActity.java

```
package com.example.madp1;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Array of fruits
        String[] fruits = {"Apple", "Banana", "Cherry", "Date", "Fig", "Grapes", "Kiwi", "Lemon", "Mango", "Orange"};

        // Creating an ArrayAdapter
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, fruits);

        // Get a reference to the ListView
        ListView listView = findViewById(R.id.listView);

        // Set the ArrayAdapter on the ListView
        listView.setAdapter(adapter);
    }
}
```

strings.xml

```
<resources>
    <string name="app_name">ArrayAdapterDemo</string>
</resources>
```

Output :



Apple

Banana

Cherry

Date

Fig

Grapes

Kiwi

Lemon

Mango

Orange

Practical: 7

Write an application to produce Notification.

- **Code:**

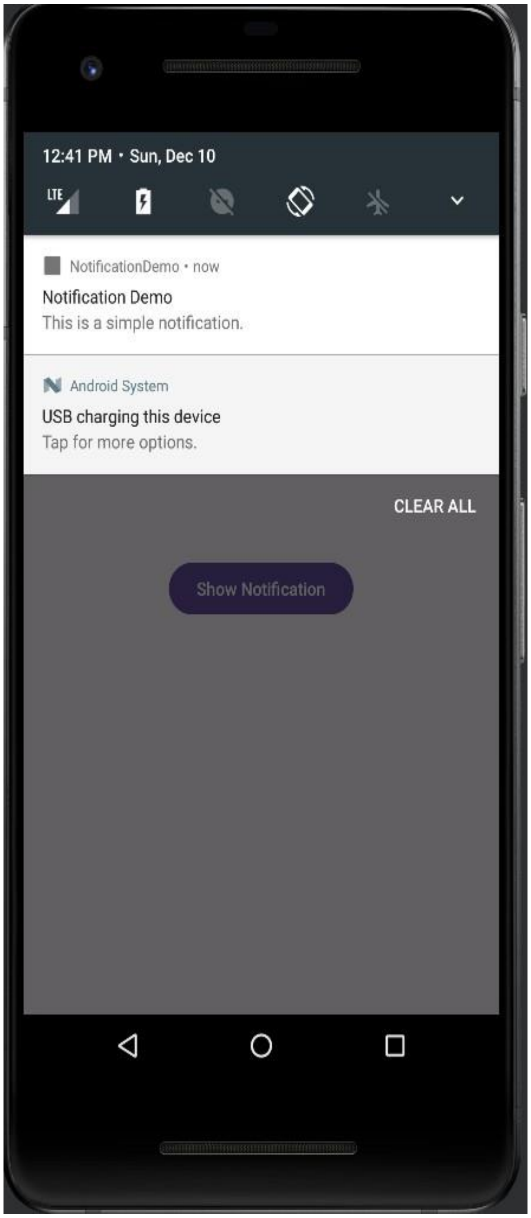
Activity_main.xml

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

    <Button
        android:id="@+id/showNotificationButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Notification"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

MainActivity.java

```
package com.example.madp1;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

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

public class MainActivity extends AppCompatActivity {

    private static final int NOTIFICATION_ID = 1;

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

        Button showNotificationButton = findViewById(R.id.showNotificationButton);

        showNotificationButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                showNotification();
            }
        });
    }

    private void showNotification() {
        // Create an Intent for the notification to open an activity when clicked
        Intent intent = new Intent(this, MainActivity.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE);

        // Create a notification
        Notification notification = new NotificationCompat.Builder(this, "default")
                .setSmallIcon(R.mipmap.ic_launcher)
                .setContentTitle("Notification Demo")
                .setContentText("This is a simple notification.")
                .setContentIntent(pendingIntent)
                .setAutoCancel(true)
                .build();

        // Get the notification manager
        NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

        // Notify with the ID and the notification
        notificationManager.notify(NOTIFICATION_ID, notification);
    }
}
```

- **Output:**



12:41 PM • Sun, Dec 10

NotificationDemo • now

Notification Demo

This is a simple notification.

Android System

USB charging this device

Tap for more options.

Practical: 8

Create an application which demonstrates radio button, check button, and Spinner.

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"
    android:padding="16dp">
    <!-- RadioGroup with RadioButtons -->
    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
    </RadioGroup>
    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Check me" />
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="393dp"
        android:layout_height="51dp"
        android:layout_below="@id/checkBox"
        android:layout_marginTop="50dp" />
    <Button
        android:id="@+id/submitButton"
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/spinner"
        android:layout_marginTop="173dp"
        android:text="Submit" />
    <RadioButton
        android:id="@+id/radioButtonOption2"
        android:layout_width="377dp"
        android:layout_height="97dp"
        android:text="Option 2" />
</RelativeLayout>
```

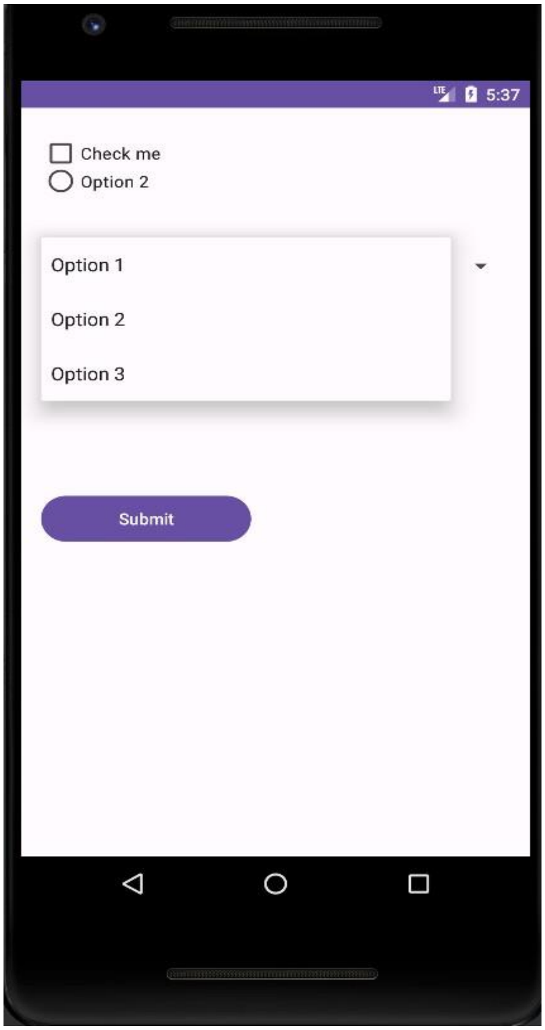
MainActivity.java

```
// com.example.formelementsdemo.MainActivity.java
package com.example.madp1;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private RadioGroup radioGroup;
    private CheckBox checkBox;
    private Spinner spinner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        radioGroup = findViewById(R.id.radioGroup);
        checkBox = findViewById(R.id.checkBox);
        spinner = findViewById(R.id.spinner);

        // Set up Spinner
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
                this,
                R.array.spinner_options,
                android.R.layout.simple_spinner_item
        );
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(adapter);

        Button submitButton = findViewById(R.id.submitButton);
        submitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                handleSubmit();
            }
        });
    }

    private void handleSubmit() {
        // Get selected RadioButton
        int selectedRadioButtonId = radioGroup.getCheckedRadioButtonId();
        RadioButton selectedRadioButton = findViewById(selectedRadioButtonId);
        // Get CheckBox state
        boolean checkBoxState = checkBox.isChecked();

        // Get selected Spinner item
        String selectedSpinnerItem = spinner.getSelectedItem().toString();

        // Display a toast with the selected values
        String toastMessage = "Radio Button: " + selectedRadioButton.getText() +
                "\nCheck Box: " + checkBoxState +
                "\nSpinner: " + selectedSpinnerItem;
        Toast.makeText(this, toastMessage, Toast.LENGTH_SHORT).show();
    }
}
```

strings.xml

```
<!-- res/values/strings.xml -->
<resources>
    <string name="app_name">FormElementsDemo</string>
    <string-array name="spinner_options">
        <item>Option 1</item>
        <item>Option 2</item>
        <item>Option 3</item>
    </string-array>
</resources>
```

- Output



Practical: 9

Create an application which demonstrates radio button, check button, and Spinner.

- Code

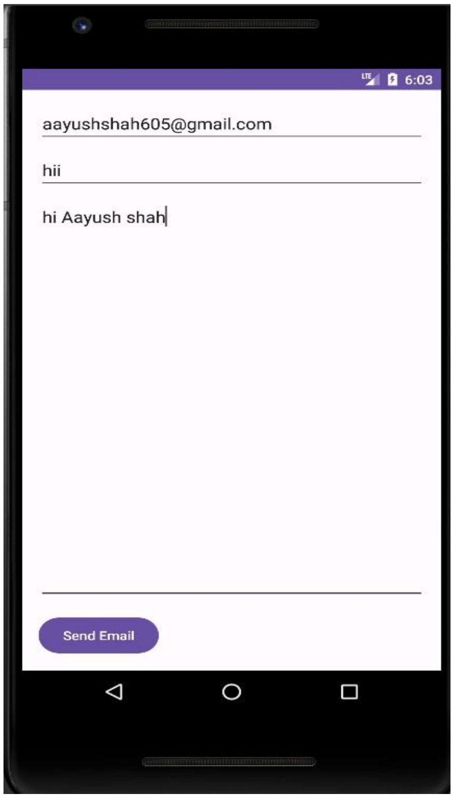
MainActivity.java

```
// com.example.emailcomposer.ComposeEmailActivity.java
package com.example.madp1;


import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;


import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {


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


        final EditText recipientEditText = findViewById(R.id.recipientEditText);
        final EditText subjectEditText = findViewById(R.id.subjectEditText);
    final EditText messageEditText = findViewById(R.id.messageEditText);

    Button sendButton = findViewById(R.id.sendButton);
    sendButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String recipient = recipientEditText.getText().toString();
            String subject = subjectEditText.getText().toString();
            String message = messageEditText.getText().toString();

            if (!recipient.isEmpty() && !subject.isEmpty() && !message.isEmpty()) {
                sendEmail(recipient, subject, message);
            } else {
                Toast.makeText(MainActivity.this, "Fill in all fields", Toast.LENGTH_SHORT).show();
            }
        }
    });
}

private void sendEmail(String recipient, String subject, String message) {
    Intent emailIntent = new Intent(Intent.ACTION_SENDTO);
    emailIntent.setData(Uri.parse("mailto:" + recipient));
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
    emailIntent.putExtra(Intent.EXTRA_TEXT, message);

    if (emailIntent.resolveActivity(getPackageManager()) != null) {
        startActivity(emailIntent);
        } else {
            Toast.makeText(this, "No email app found", Toast.LENGTH_SHORT).show();
        }
    }
}
```

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">
    <EditText
        android:id="@+id/recipientEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Recipient"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="8dp"/>
    <EditText
        android:id="@+id/subjectEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Subject"
        android:layout_marginBottom="8dp"/>
    <EditText
        android:id="@+id/messageEditText"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:hint="Message"
        android:gravity="top"
        android:layout_marginBottom="16dp"/>
    <Button
        android:id="@+id/sendButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Email"/>
</LinearLayout>
```

- Output:



aayushshah605@gmail.com

hii

hi Aayush shah

Practical: 10

Create application to demonstrate date and time picker.

- Code:

MainActivity.java

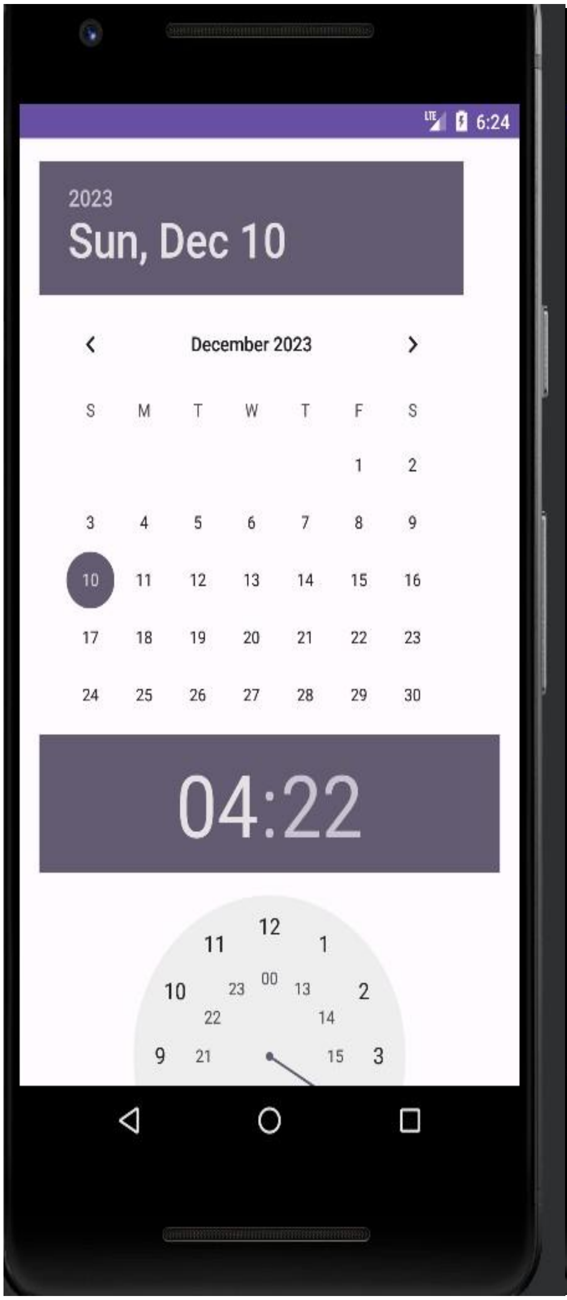
```
// com.example.datetimepickerdemo.DateTimePickerActivity.java

package com.example.madp1;


import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.DatePicker;

import android.widget.TimePicker;

import android.widget.Toast;


import androidx.appcompat.app.AppCompatActivity;


import java.util.Calendar;


public class MainActivity extends AppCompatActivity {


  @Override

  protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);


    final DatePicker datePicker = findViewById(R.id.datePicker);
```

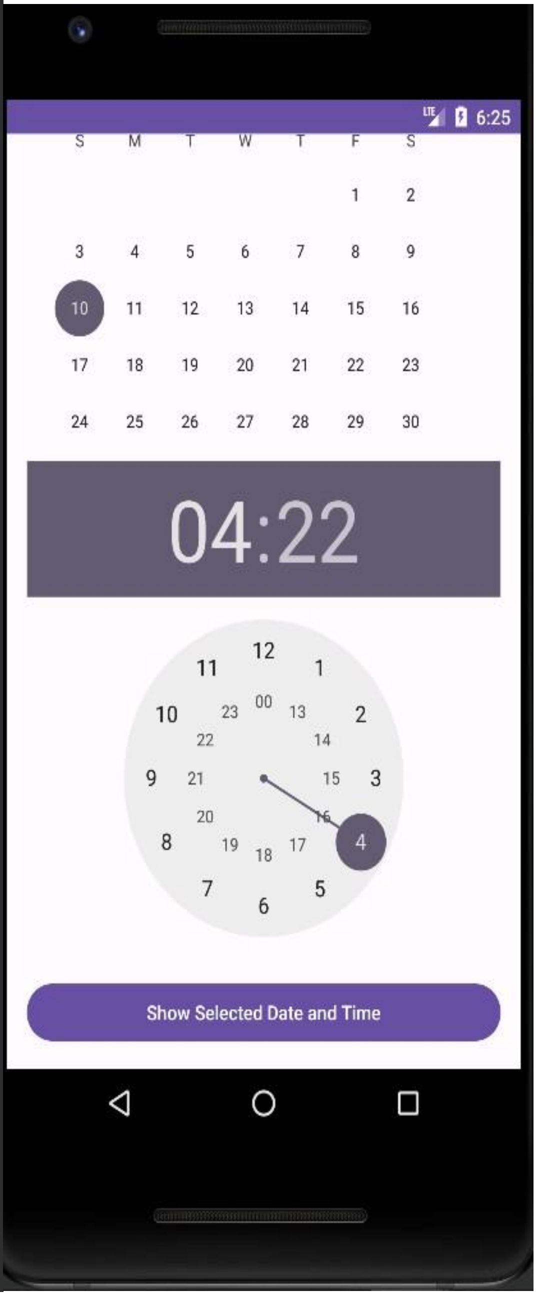
```
        final TimePicker timePicker = findViewById(R.id.timePicker);
        timePicker.setIs24HourView(true);

        Button showDateTimeButton = findViewById(R.id.showDateTimeButton);
        showDateTimeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                int year = datePicker.getYear();
                int month = datePicker.getMonth() + 1; // Month is zero-based
                int day = datePicker.getDayOfMonth();

                int hour = timePicker.getCurrentHour();
                int minute = timePicker.getCurrentMinute();

                // Display the selected date and time
                String dateTime = "Selected Date and Time: " + year + "-" + month + "-" + day + " " +
hour + ":" + minute;
                Toast.makeText(MainActivity.this, dateTime, Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

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"
        android:padding="16dp">

        <DatePicker
            android:id="@+id/datePicker"
            android:layout_width="385dp"
            android:layout_height="383dp"
            android:calendarViewShown="false" />

        <TimePicker
            android:id="@+id/timePicker"
            android:layout_width="wrap_content"
            android:layout_height="349dp"
            android:layout_marginTop="16dp" />
        <Button
            android:id="@+id/showDateTimeButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="Show Selected Date and Time" />
    </LinearLayout>
</ScrollView>
```

- Output


6:24
2023
Sun, Dec 10
December 2023
S M T W T F S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
04:22


6:25
S M T W T F S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
04:22
Show Selected Date and Time

Practical: 11

Design screen for payment board and display message on button click.

- Code:

Activity_main.xml

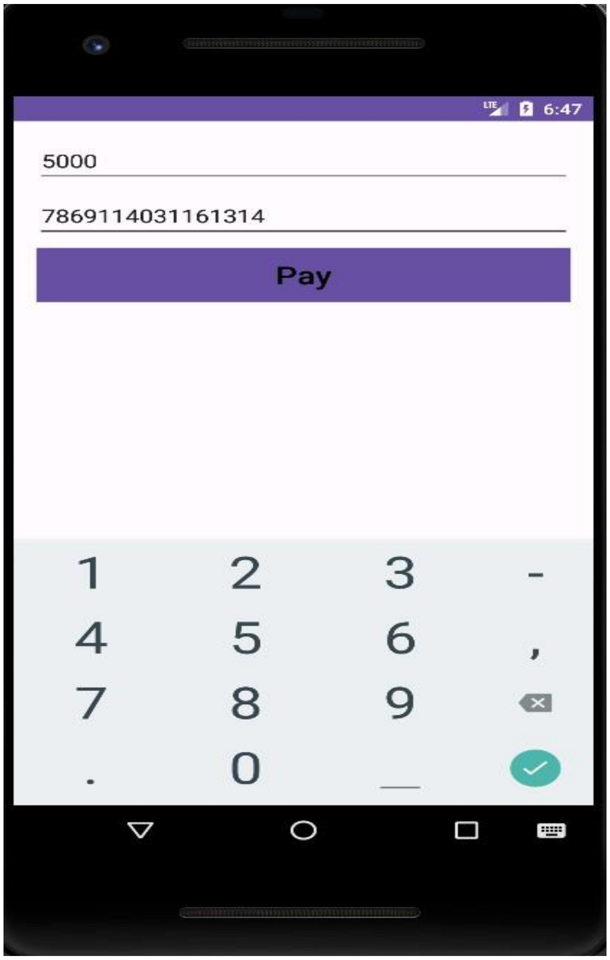
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <!-- Payment Amount -->
  <EditText
    android:id="@+id/amountEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Amount"
    android:inputType="numberDecimal"
    android:layout_marginBottom="8dp"/>

  <!-- Card Number -->
  <EditText
    android:id="@+id/cardNumberEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
        android:hint="Card Number"
        android:inputType="number"
        android:layout_marginBottom="8dp"/>

    <!-- Pay Button -->
    <Button
        android:id="@+id/paymentButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pay"
        android:textSize="24sp"
        android:background="#DDDDDD"
        android:textColor="@android:color/black"
        android:onClick="displayPaymentConfirmation" />

</LinearLayout>
```

MainActivity.xml

```
// com.example.paymentboarddemo.PaymentBoardActivity.java
package com.example.madp1;

import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void displayPaymentConfirmation(View view) {
        Toast.makeText(this, "Payment Successful!", Toast.LENGTH_SHORT).show();
    }
}
```

- Output:



5000

7869114031161314

Practical: 12

Write a program to demonstrate Options Menu in android for a restaurant and display appropriate message related to the dish user has opted from the menu on text field.

- Code:

Options_menu.xml

```
<!-- res/menu/options_menu.xml -->

<menu xmlns:android="http://schemas.android.com/apk/res/android">

  <item

    android:id="@+id/menu_pizza"

    android:title="Pizza" />

  <item

    android:id="@+id/menu_burger"

    android:title="Burger" />

</menu>
```

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:padding="16dp"

  tools:context=".MainActivity">


  <TextView

    android:id="@+id/messageTextView"

    android:layout_width="wrap_content"
```

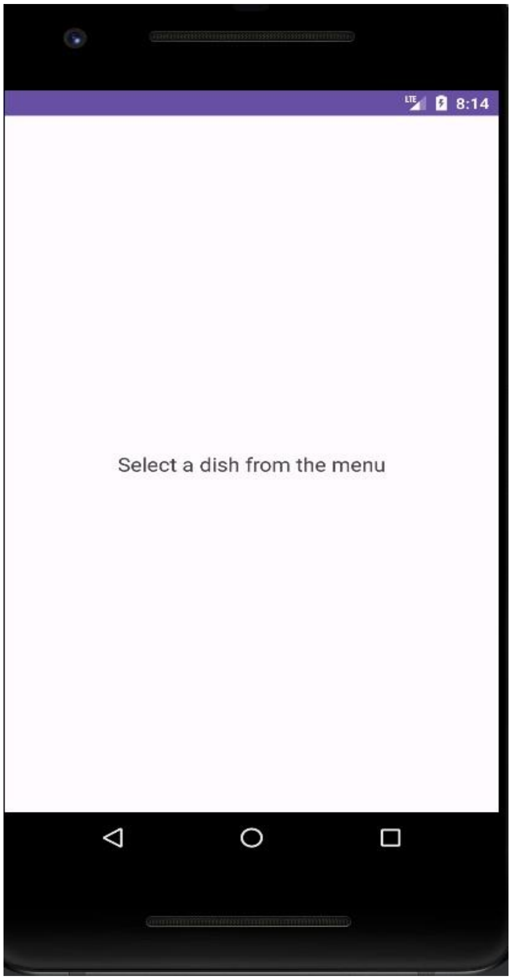
```
        android:layout_height="wrap_content"
        android:text="Select a dish from the menu"
        android:textSize="18sp"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

Main_activity.java

```
package com.example.madp1;

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

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private TextView messageTextView;

    private static final int MENU_PIZZA = R.id.menu_pizza;
    private static final int MENU_BURGER = R.id.menu_burger;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        messageTextView = findViewById(R.id.messageTextView);
    }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == MENU_PIZZA) {
            displayMessage("You selected Pizza");
            return true;
        } else if (item.getItemId() == MENU_BURGER) {
            displayMessage("You selected Burger");
            return true;
        } else {
            return super.onOptionsItemSelected(item);
        }
    }

    private void displayMessage(String message) {
        messageTextView.setText(message);
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }
```

```
}
```

- Output:



Select a dish from the menu

Practical: 13

Write a program to demonstrate Context Menu in android.

Code:

In manifest file check

```
 <activity
          android:name=".context_menu"
          android:exported="false" />
```

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:padding="16dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Long press me!"
        android:textSize="18sp"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

MainActivity.java

```
package com.example.madp1;
```

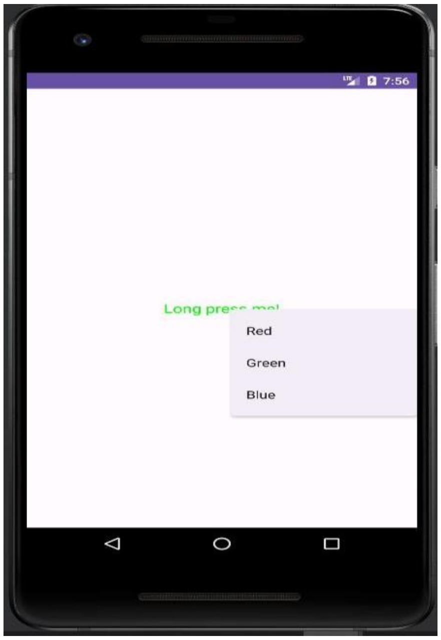
```
import android.graphics.Color;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private TextView textView;

    private static final int MENU_RED = R.id.menu_red;
    private static final int MENU_GREEN = R.id.menu_green;
    private static final int MENU_BLUE = R.id.menu_blue;

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

        textView = findViewById(R.id.textView);

        // Register the TextView for the context menu
        registerForContextMenu(textView);
    }
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        getMenuInflater().inflate(R.menu.context_menu, menu);
    }

    @Override
    public boolean onContextItemSelected(MenuItem item) {
        if (item.getItemId() == MENU_RED) {
            changeTextColor(Color.RED);
            return true;
        } else if (item.getItemId() == MENU_GREEN) {
            changeTextColor(Color.GREEN);
            return true;
        } else if (item.getItemId() == MENU_BLUE) {
            changeTextColor(Color.BLUE);
            return true;
        } else {
            return super.onContextItemSelected(item);
        } }
    private void changeTextColor(int color) {
        textView.setTextColor(color);
    }}
```

Res/menu/context_menu.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
```

```
    <item
        android:id="@+id/menu_red"
        android:title="Red" />
    <item
        android:id="@+id/menu_green"
        android:title="Green" />
    <item
        android:id="@+id/menu_blue"
        android:title="Blue" />
</menu>
```

- Output:



Practical: 14

Design android application for login activity. Write android code to check login credentials with username = "mca" and password = "android". Display appropriate toast message to the user.

- **Code:**

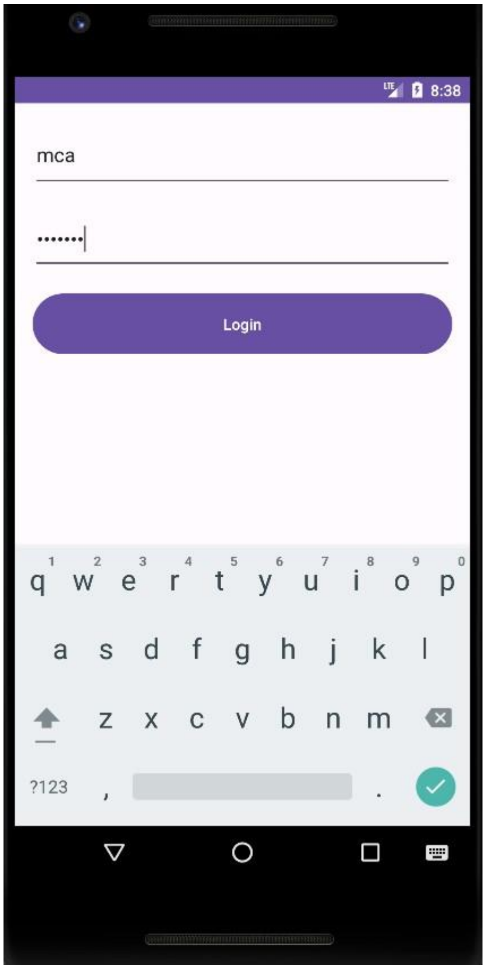
Activity_main.xml

```
<?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:padding="16dp"
  tools:context=".LoginActivity">

  <EditText
    android:id="@+id/usernameEditText"
    android:layout_width="393dp"
    android:layout_height="63dp"
    android:layout_marginBottom="16dp"
    android:hint="Username" />

  <EditText
    android:id="@+id/passwordEditText"
    android:layout_width="393dp"
    android:layout_height="59dp"
    android:layout_below="@id/usernameEditText"
    android:layout_marginBottom="16dp"
    android:hint="Password"
    android:inputType="textPassword" />
  <Button
    android:id="@+id/loginButton"
    android:layout_width="388dp"
    android:layout_height="63dp"
    android:layout_below="@id/passwordEditText"
    android:text="Login" />
</RelativeLayout>
```

MainActivity.java

```
package com.example.madp1;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private static final String CORRECT_USERNAME = "mca";
    private static final String CORRECT_PASSWORD = "android";

    private EditText usernameEditText;
    private EditText passwordEditText;
    private Button loginButton;

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

        usernameEditText = findViewById(R.id.usernameEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        loginButton = findViewById(R.id.loginButton);

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

    private void checkLoginCredentials() {
        String enteredUsername = usernameEditText.getText().toString();
        String enteredPassword = passwordEditText.getText().toString();
    if (enteredUsername.equals(CORRECT_USERNAME) &&
enteredPassword.equals(CORRECT_PASSWORD)) {
      showToast("Login successful!");
    } else {
      showToast("Invalid username or password. Please try again.");
    }
  }

  private void showToast(String message) {
    Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
  }
}
```

- **Output:**



mca

•••••••

Practical: 15

Write a program to demonstrate Popup Menu in android for cut, copy and paste options in it and display appropriate information related to operation.

- **Code**

Popup_menu.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

  <item

    android:id="@+id/menu_cut"

    android:title="Cut" />

  <item

    android:id="@+id/menu_copy"

    android:title="Copy" />

  <item

    android:id="@+id/menu_paste"

    android:title="Paste" />

</menu>
```

Mainacitivity.java

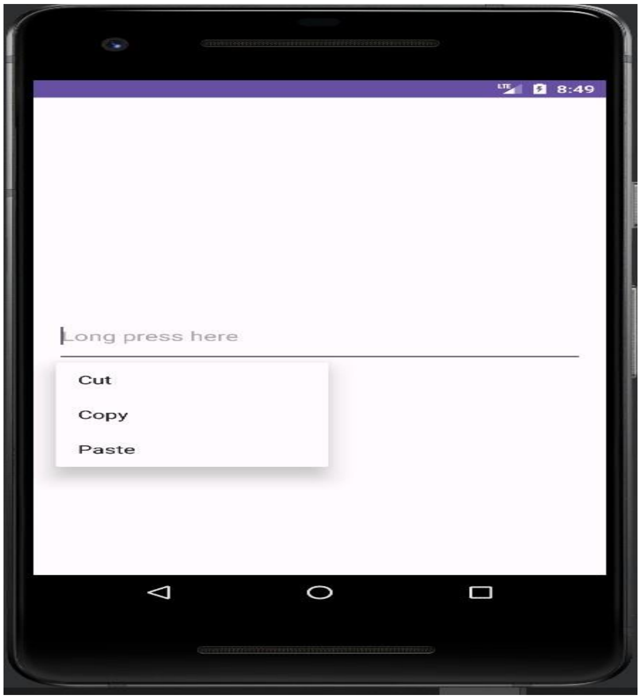
```
// MainActivity.java

package com.example.madp1;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.PopupMenu;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
  private EditText editText;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    editText = findViewById(R.id.editText);

    editText.setOnLongClickListener(new View.OnLongClickListener() {
      @Override
      public boolean onLongClick(View v) {
        showPopupMenu(v);
        return true;
      }
    });
  }

  private void showPopupMenu(View view) {
    PopupMenu popupMenu = new PopupMenu(this, view);
    popupMenu.inflate(R.menu.popup_menu);

    popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
      @Override
      public boolean onMenuItemClick(MenuItem item) {
        if (item.getItemId() == R.id.menu_cut) {
          showToast("Text cut");
          editText.setText("");  // Simulating cut by clearing the EditText
          return true;
        } else if (item.getItemId() == R.id.menu_copy) {
          showToast("Text copied");
          return true;
        } else if (item.getItemId() == R.id.menu_paste) {
          showToast("Text pasted");
          // In a real app, you would retrieve the text from the clipboard and set it in the
EditText
          return true;
        } else {
          return false;
        }
      }
    });
```

```
        popupMenu.show();
    }

    private void showToast(String message) {
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }
}
```

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:padding="16dp"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/editText"
        android:layout_width="399dp"
        android:layout_height="72dp"
        android:layout_centerInParent="true"
        android:hint="Long press here" />
</RelativeLayout>
```

- ➢ **Output:**



Practical: 16

Write an application to display image button.

- **Code:**

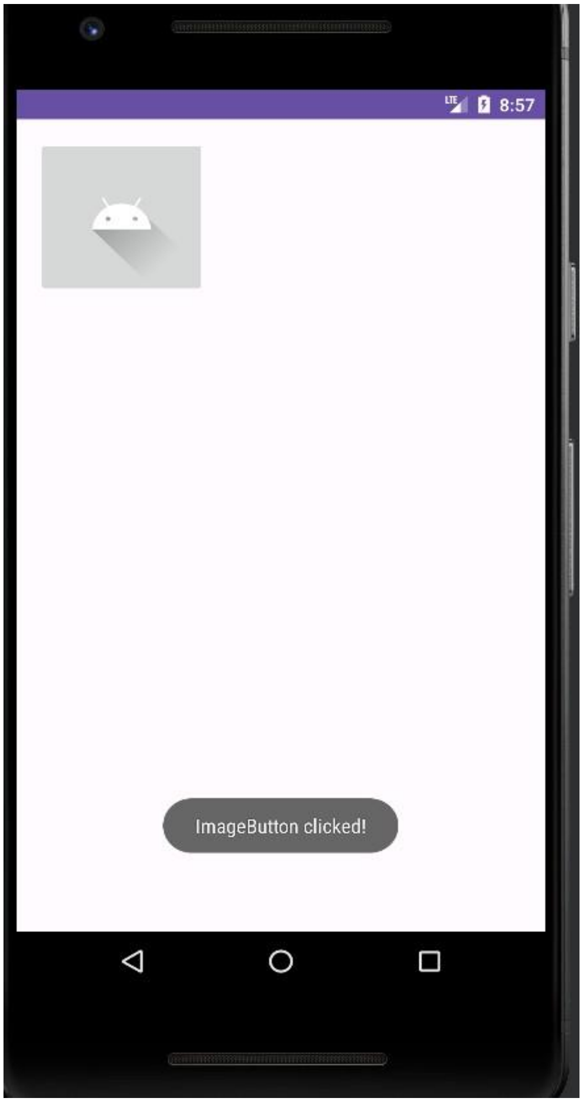
MainActivity.java

```
// MainActivity.java
package com.example.madp1;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private ImageButton imageButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageButton = findViewById(R.id.imageButton);
        imageButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showToast("ImageButton clicked!");
            }
        });
    }

    private void showToast(String message) {
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }
}
```

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:padding="16dp"
    tools:context=".MainActivity">

    <ImageButton
        android:id="@+id/imageButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher_foreground"
        android:contentDescription="ImageButton" />
</RelativeLayout>
```

- **Output**



ImageButton clicked!

Practical: 17

Demonstrate an application to implement web view in android.

<!-- AndroidManifest.xml -->

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

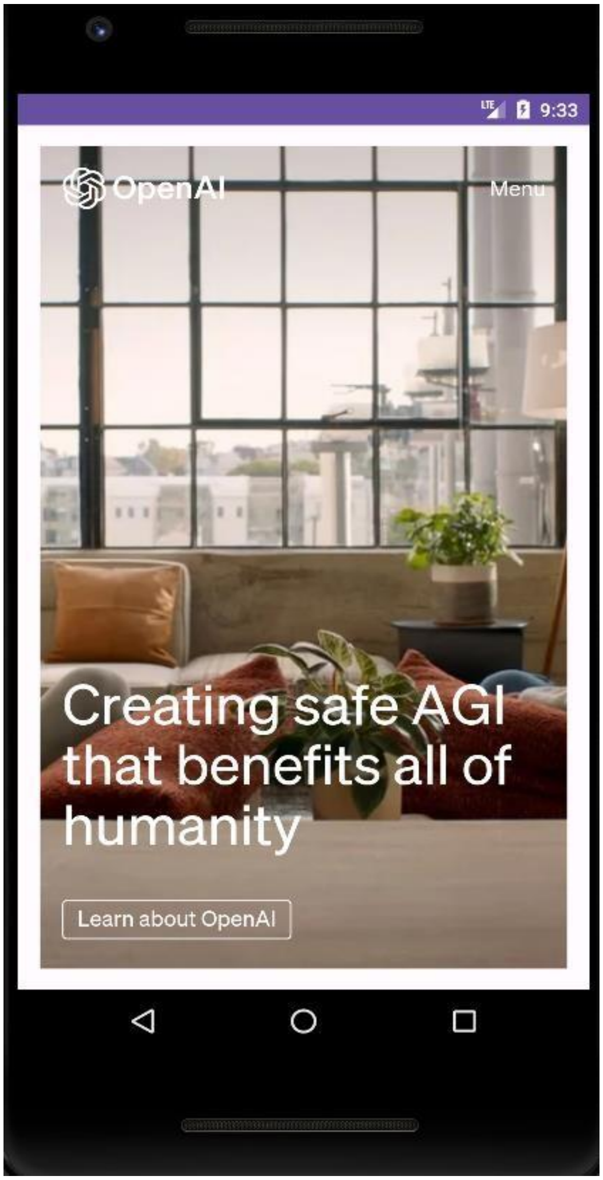
Activity_main.xml

```
<?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:padding="16dp"
    tools:context=".MainActivity">

    <WebView
        android:id="@+id/webView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</RelativeLayout>
```

ManinActivity.java

```
// MainActivity.java
package com.example.madp1;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        webView = findViewById(R.id.webView);
        // Enable JavaScript in the WebView
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        // Set a WebViewClient to handle links within the WebView
        webView.setWebViewClient(new WebViewClient());
        // Load a website (e.g., OpenAI's website)
        webView.loadUrl("https://www.openai.com");
    }
}
```

- **Output:**



Creating safe AGI that benefits all of humanity

Practical: 18

Write an android code to turn ON/OFF Bluetooth

- **Code:**

<!-- AndroidManifest.xml -->

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

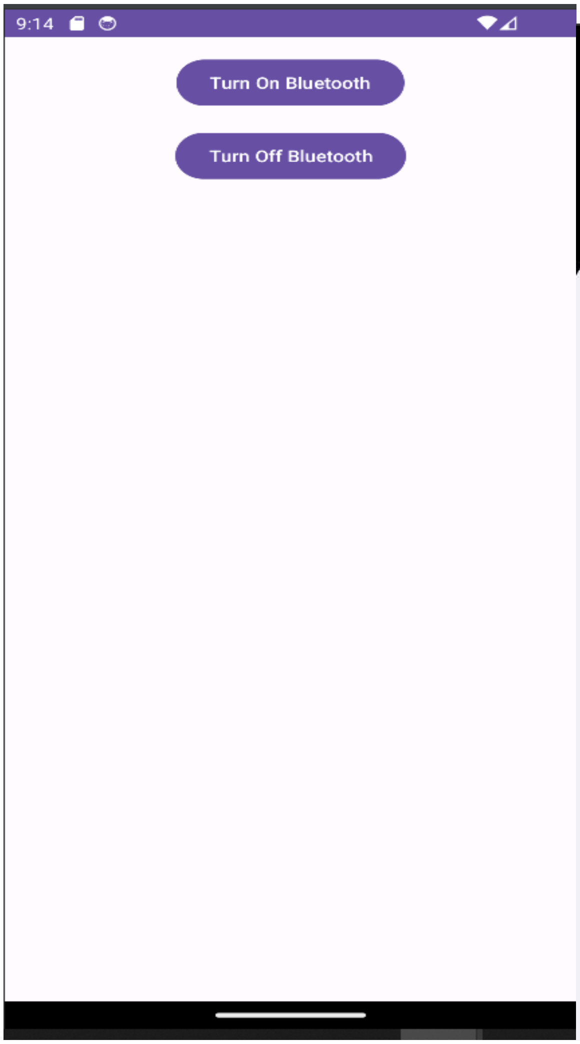
Activity_main.xml

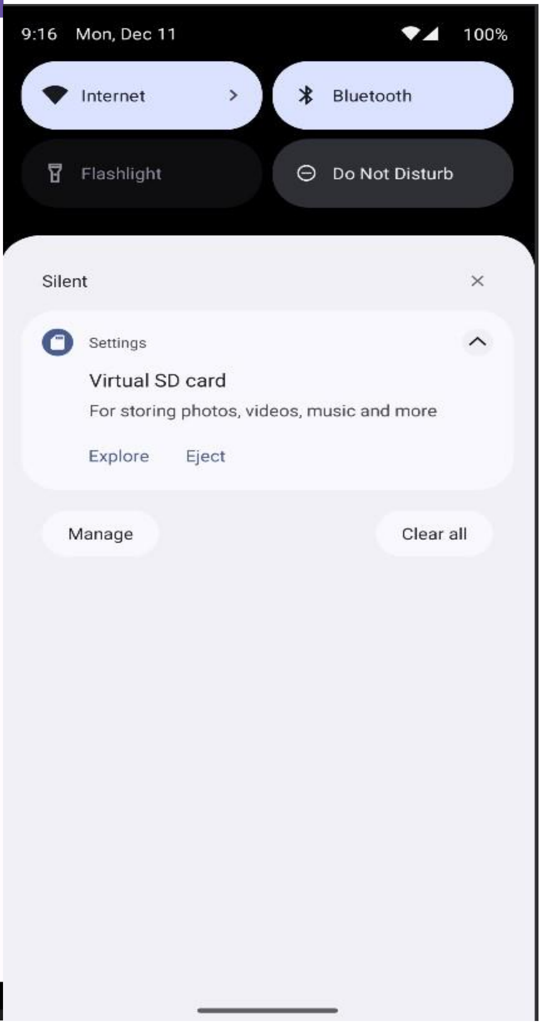
```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

   <ToggleButton
      android:id="@+id/toggleButtonBluetooth"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Bluetooth"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.bluetoothonoff;

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

import android.content.pm.PackageManager;
import android.os.Bundle;
import android.bluetooth.BluetoothAdapter;
import android.content.Intent;
import android.widget.CompoundButton;
import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {
  private ToggleButton toggleButtonBluetooth;
  private BluetoothAdapter bluetoothAdapter;

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

    toggleButtonBluetooth = findViewById(R.id.toggleButtonBluetooth);
    bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    toggleButtonBluetooth.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
      @Override
      public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
          enableBluetooth();
        } else {
          disableBluetooth();
        }
      }
    });

    // Set the initial state based on Bluetooth status
    toggleButtonBluetooth.setChecked(bluetoothAdapter.isEnabled());
  }

  private void enableBluetooth() {
    if (!bluetoothAdapter.isEnabled()) {
      Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
      if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.BLUETOOTH_CONNECT) !=
PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //  public void onRequestPermissionsResult(int requestCode, String[] permissions,
        //                                          int[] grantResults)
        // to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
      }
      startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
    }
  }

  private void disableBluetooth() {
    if (bluetoothAdapter.isEnabled()) {
      if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.BLUETOOTH_CONNECT) !=
PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //    ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //  public void onRequestPermissionsResult(int requestCode, String[] permissions,
        //                                          int[] grantResults)
        // to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
      }
      bluetoothAdapter.disable();
    }
  }

  private static final int REQUEST_ENABLE_BT = 1;
}
```

- **Output:**





Virtual SD card

For storing photos, videos, music and more

Practical: 19

Write an android application using SQLite to create table and perform CRUD operations (Example. COURSE table (ID, Name, Duration, Description), perform ADD, UPDATE, DELETE and READ operations)

- **Code:**

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:padding="16dp"
  tools:context=".MainActivity">

  <EditText
    android:id="@+id/nameEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Course Name"
    android:layout_marginBottom="8dp"/>

  <Button
    android:id="@+id/addButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add Course"
    android:layout_below="@id/nameEditText"
    android:layout_marginTop="16dp"/>

  <Button
    android:id="@+id/readButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Read Courses"
    android:layout_below="@id/addButton"
    android:layout_marginTop="16dp"/>

  <Button
```

```
        android:id="@+id/updateButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Update Course"
        android:layout_below="@id/readButton"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/deleteButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete Course"
        android:layout_below="@id/updateButton"
        android:layout_marginTop="16dp"/>

    <ListView
        android:id="@+id/courseListView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/deleteButton"
        android:layout_marginTop="16dp"/>
</RelativeLayout>
```

MainActivity.java

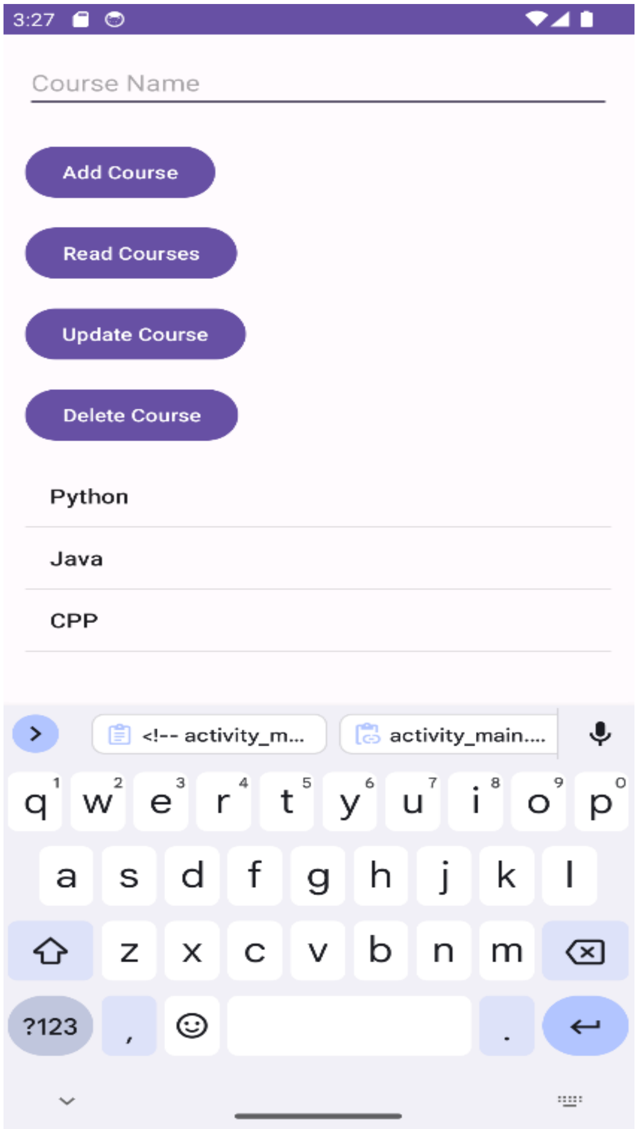
```
package com.example.sqldemmo;// MainActivity.java
import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import  android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import  android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

import  androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.List;
```

```
public class MainActivity extends AppCompatActivity {

    private SQLiteDatabase database;
    private List<String> courseList;
    private ArrayAdapter<String> adapter;
    private String selectedCourse = null;

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

        // Initialize database and UI elements
        initDatabase();
        initUI();

        // Load existing courses from the database
        loadCourses();
    }

    private void initDatabase() {
        try {
            // Open or create the database
            database = openOrCreateDatabase("courses.db", MODE_PRIVATE, null);

            // Create the 'courses' table if it doesn't exist
            database.execSQL("CREATE TABLE IF NOT EXISTS courses (name TEXT);");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void initUI() {
        final EditText nameEditText = findViewById(R.id.nameEditText);
        Button addButton = findViewById(R.id.addButton);
        Button readButton = findViewById(R.id.readButton);
        Button updateButton = findViewById(R.id.updateButton);
        Button deleteButton = findViewById(R.id.deleteButton);
        ListView courseListView = findViewById(R.id.courseListView);

        courseList = new ArrayList<>();
        adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, courseList);
        courseListView.setAdapter(adapter);
```

```
// Set click listener for list items to select the course
courseListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        selectedCourse = courseList.get(position);
        showToast("Selected Course: " + selectedCourse);
    }
});

addButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Add a new course to the database
        addCourse(nameEditText.getText().toString());
    }
});

readButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Read and display the list of courses
        loadCourses();
    }
});

updateButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Update the selected course
        if (selectedCourse != null) {
            updateCourse(selectedCourse);
        } else {
            showToast("Select a course to update");
        }
    }
});

deleteButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Delete the selected course
        if (selectedCourse != null) {
            deleteCourse(selectedCourse);
```

```
                } else {
                    showToast("Select a course to delete");
                }
            }
        });
    }

    private void addCourse(String name) {
        ContentValues values = new ContentValues();
        values.put("name", name);

        // Insert the new course into the database
        database.insert("courses", null, values);

        // Load and display the updated list of courses
        loadCourses();

        // Clear input fields
        clearInputFields();

        // Display a success message
        showToast("Course added successfully");
    }

    private void loadCourses() {
        courseList.clear();

        // Query all courses from the database
        Cursor cursor = database.query("courses", null, null, null, null, null, null);

        // Check if the cursor is not null
        if (cursor != null) {
            // Iterate through the cursor and add course names to the list
            if (cursor.moveToFirst()) {
                do {
                    @SuppressLint("Range") String name =
cursor.getString(cursor.getColumnIndex("name"));
                    courseList.add(name);
                } while (cursor.moveToNext());

                cursor.close();
            }
        }
        // Notify the adapter that the data has changed
        adapter.notifyDataSetChanged();
    }

    private void updateCourse(String courseName) {
        // Update the selected course in the database
        ContentValues values = new ContentValues();
        values.put("name", "Updated " + courseName);

        database.update("courses", values, "name = ?", new String[]{courseName});

        // Load and display the updated list of courses
        loadCourses();

        showToast("Course updated successfully");
    }

    private void deleteCourse(String courseName) {
        // Delete the selected course from the database
        database.delete("courses", "name = ?", new String[]{courseName});

        // Load and display the updated list of courses
        loadCourses();

        showToast("Course deleted successfully");

        // Clear the selected course
        selectedCourse = null;
    }

    private void clearInputFields() {
        EditText nameEditText = findViewById(R.id.nameEditText);
        nameEditText.getText().clear();
    }

    private void showToast(String message) {
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onDestroy() {
        // Close the database when the activity is destroyed
        if (database != null) {
            database.close();
        }
        super.onDestroy();
    }
}
```

- ➢ Output



Python

Java

CPP

Practical: 20

Create an Android app, powered by Firebase Realtime database that supports: Adding Data to Firebase Realtime database, Retrieving Data from Firebase and Deleting data from firebase data.

Step1: Create a new project on the Firebase Console.
Add an Android app to your Firebase project and follow the setup instructions to download the google-services.json file.
Add the google-services.json file to your Android Studio project.
Step 2: Add Firebase Dependencies
Add the necessary dependencies in your build.gradle files:
Sync your project with the updated Gradle files.
Step 3 : Design mainxml

- **Code:**

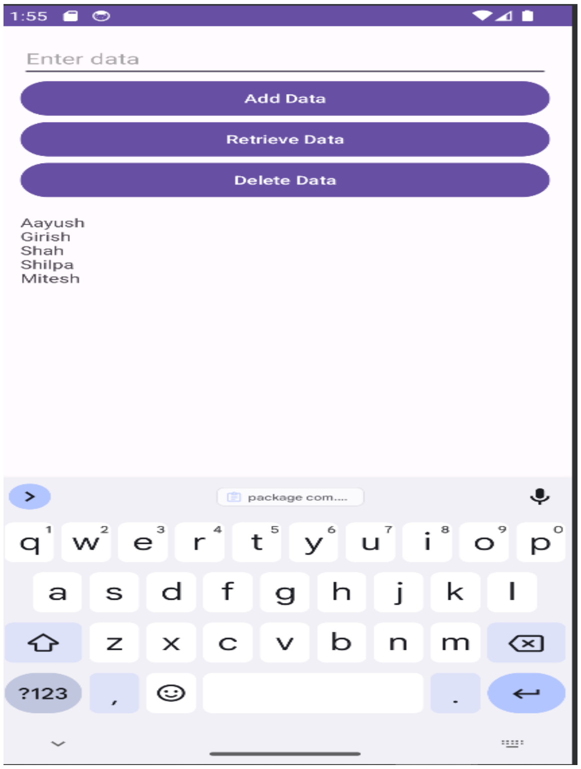
Activity_main.xml

```
<?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">
  <EditText
    android:id="@+id/dataEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter data" />
  <Button
    android:id="@+id/addButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Add Data" />
  <Button
    android:id="@+id/retrieveButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Retrieve Data" />
  <Button
    android:id="@+id/deleteButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Delete Data" />
  <!-- Display retrieved data -->
  <TextView
    android:id="@+id/displayTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Retrieved Data will be displayed here" />
</LinearLayout>
```

MainActivity.java

```
package com.example.practicaldemo;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class MainActivity extends AppCompatActivity {
    private EditText dataEditText;
    private Button addButton, retrieveButton, deleteButton;
    private TextView displayTextView;

    private DatabaseReference databaseReference;
    private FirebaseUser currentUser;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dataEditText = findViewById(R.id.dataEditText);
        addButton = findViewById(R.id.addButton);
        retrieveButton = findViewById(R.id.retrieveButton);
        deleteButton = findViewById(R.id.deleteButton);
        displayTextView = findViewById(R.id.displayTextView);

        // Get current user
        FirebaseAuth auth = FirebaseAuth.getInstance();
        currentUser = auth.getCurrentUser();

        // Get reference to Firebase Realtime Database
        databaseReference = FirebaseDatabase.getInstance().getReference().child("data");

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

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

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

    private void addData() {
        String newData = dataEditText.getText().toString().trim();
        if (!newData.isEmpty()) {
            // Generate a unique key for the new data
            String dataId = databaseReference.push().getKey();
            // Save the data to the database
            databaseReference.child(dataId).setValue(newData);
            dataEditText.setText("");
        } }
    private void retrieveData() {
        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                StringBuilder data = new StringBuilder();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                    String value = dataSnapshot.getValue(String.class);
                    data.append(value).append("\n");
                }
                displayTextView.setText(data.toString());
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                // Handle error
            }
        });
    }
    private void deleteData() {
        // Delete all data under the "data" node
        databaseReference.removeValue();
    }
}
```

- Output:



```
Aayush
Girish
Shah
Shilpa
Mitesh
```

Practical: 21

Write an android app to write JSON data into a file and read JSON data from created file.

- **Code:**

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:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context=".MainActivity">

   <ListView
      android:layout_width="wrap_content"
      android:id="@+id/list_view"
      android:layout_height="match_parent"/>

</RelativeLayout>
```

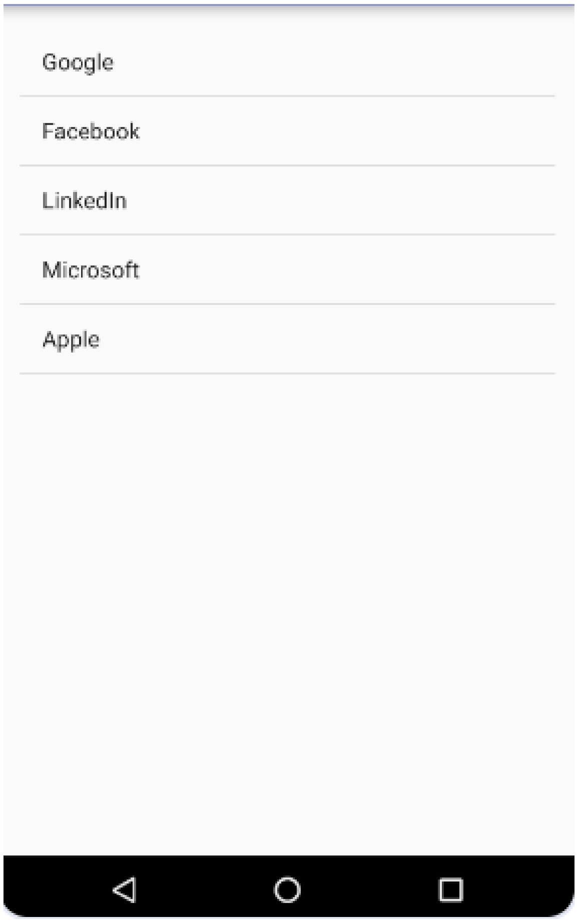
MainActivity.java

```
package com.example.practicaldemo;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
```

```
String json_string = "{\n" +
        " \"title\":\"JSONParserTutorial\",\n" +
        " \"array\":[\n" +
        "    {\n" +
        "    \"company\":\"Google\"\n" +
        "    },\n" +
        "    {\n" +
        "      \"company\":\"Facebook\"\n" +
        "    },\n" +
        "    {\n" +
        "    \"company\":\"LinkedIn\"\n" +
        "    },\n" +
        "    {\n" +
        "      \"company\" : \"Microsoft\"\n" +
        "    },\n" +
        "    {\n" +
        "      \"company\": \"Apple\"\n" +
        "    }\n" +
        "    ],\n" +
        "    \"nested\":{\n" +
        "    \"flag\": true,\n" +
        "    \"random_number\":1\n" +
        "    }\n" +
        "}";
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    try {
        ListView listView = (ListView) findViewById(R.id.list_view);
        List<String> items = new ArrayList<>();
        JSONObject root = new JSONObject(json_string);
        JSONArray array= root.getJSONArray("array");
        this.setTitle(root.getString("title"));
        for(int i=0;i<array.length();i++)
        {
            JSONObject object= array.getJSONObject(i);
            items.add(object.getString("company"));
        }
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
                android.R.layout.simple_list_item_1, items);
        if (listView != null) {
            listView.setAdapter(adapter);
        }
```

```
            JSONObject nested= root.getJSONObject("nested");
            Log.d("TAG","flag value "+nested.getBoolean("flag"));

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

- **Output:**



Google

Facebook

LinkedIn

Microsoft

Apple

Practical: 22

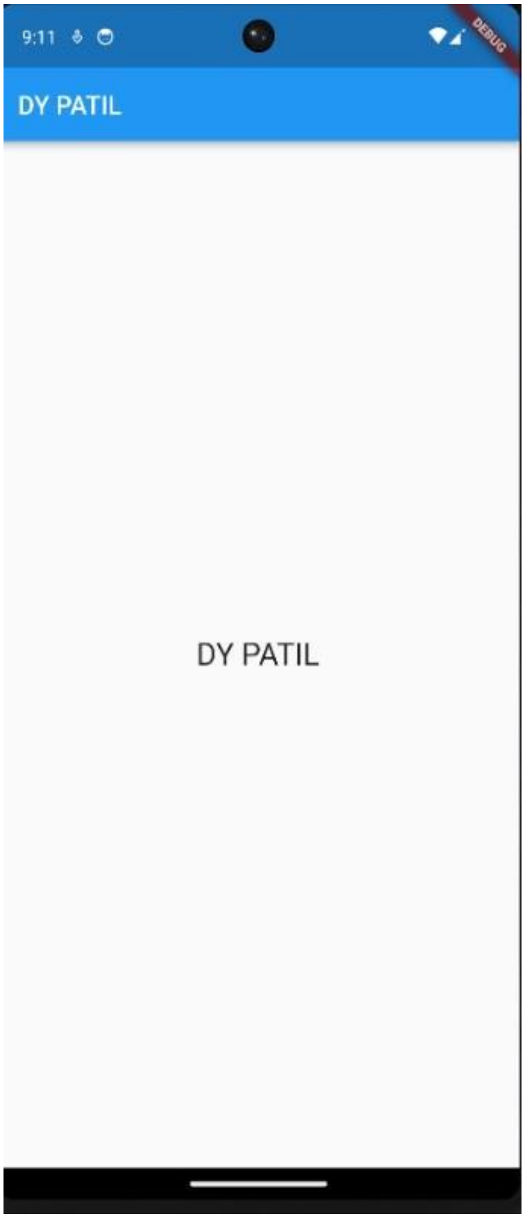
Demonstrate flutter application using android.

- **Code:**

Main.dart

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'DY PATIL',
      home: Scaffold(
      appBar: AppBar(
          title: const Text('DY PATIL'),
        ),
        body: const Center(
          child: Text(
            'DY PATIL',
            style: TextStyle(fontSize: 24),
          ),
        ),
      ),
    );
  }}
```

- **Output:**



DY PATIL