**Q1] Write a program to implement following functions in numpy library Array, arange, len, ndim, slicing, copy, view, reshape, concatenate, split**

import numpy as np

# Array function

def my\_array(arr\_list):

return np.array(arr\_list)

# arange function

def my\_arange(start, stop, step=1):

return np.arange(start, stop, step)

# len function

def my\_len(arr):

return len(arr)

# ndim function

def my\_ndim(arr):

return arr.ndim

# slicing function

def my\_slicing(arr, start=None, stop=None, step=None):

return arr[start:stop:step]

# copy function

def my\_copy(arr):

return arr.copy()

# view function

def my\_view(arr):

return arr.view()

# reshape function

def my\_reshape(arr, shape):

if np.prod(shape) != np.prod(arr.shape):

raise ValueError("Cannot reshape array into the specified shape")

return arr.reshape(shape)

# concatenate function

def my\_concatenate(arr1, arr2, axis=0):

return np.concatenate((arr1, arr2), axis=axis)

# split function

def my\_split(arr, indices, axis=0):

return np.split(arr, indices, axis=axis)

# Testing the functions

arr1 = my\_array([1, 2, 3, 4, 5])

arr2 = my\_array([6, 7, 8, 9, 10])

print("Array:")

print(arr1)

print("\narange:")

print(my\_arange(0, 10, 2))

print("\nlen:")

print(my\_len(arr1))

print("\nndim:")

print(my\_ndim(arr1))

print("\nslicing:")

print(my\_slicing(arr1, 1, 4, 1))

print("\ncopy:")

arr1\_copy = my\_copy(arr1)

print(arr1\_copy)

print("\nview:")

arr1\_view = my\_view(arr1)

print(arr1\_view)

print("\nreshape:")

arr1\_reshaped = my\_reshape(arr1, (1, 5)) # Corrected reshape to (1, 5)

print(arr1\_reshaped)

print("\nconcatenate:")

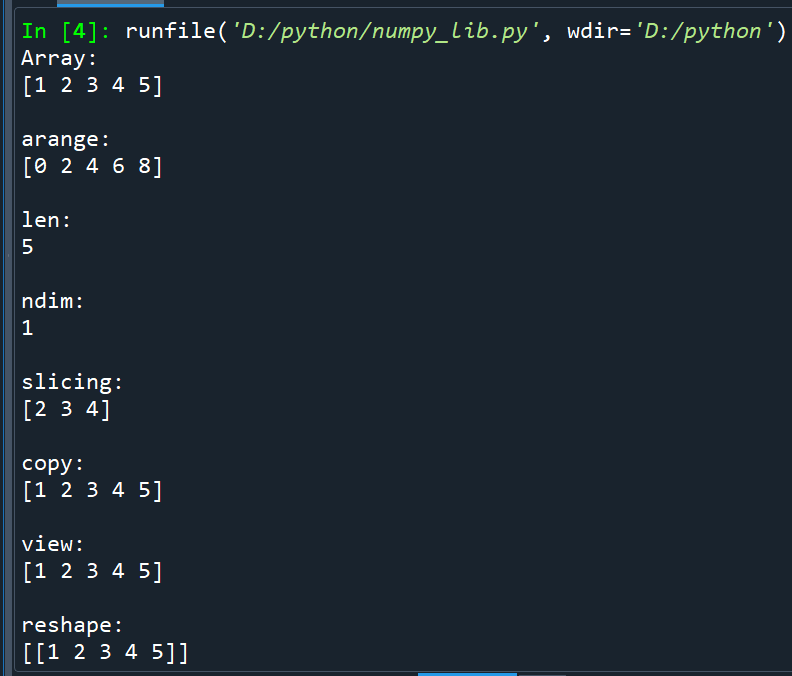
print(my\_concatenate(arr1, arr2))

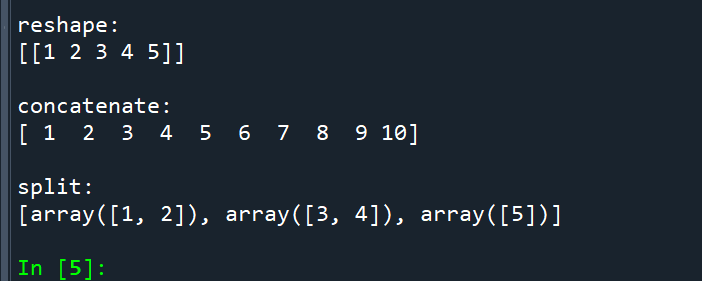
print("\nsplit:")

arr\_split = my\_split(arr1, [2, 4]) # Split into parts of sizes [2, 3]

print(arr\_split)

Output:





**Q2] Write a program to implement following functions in pandas library Dataframe, dropna, fillna, replace specific values**

import pandas as pd

import numpy as np

# DataFrame function

def my\_dataframe(data\_dict):

return pd.DataFrame(data\_dict)

# dropna function

def my\_dropna(df):

return df.dropna()

# fillna function

def my\_fillna(df, value):

return df.fillna(value)

# replace function

def my\_replace(df, old\_value, new\_value):

return df.replace(old\_value, new\_value)

# Testing the functions

data = {'A': [1, 2, np.nan, 4],

'B': [5, np.nan, 7, 8],

'C': [9, 10, 11, 12]}

df = my\_dataframe(data)

print("Original DataFrame:")

print(df)

print("\ndropna:")

print(my\_dropna(df))

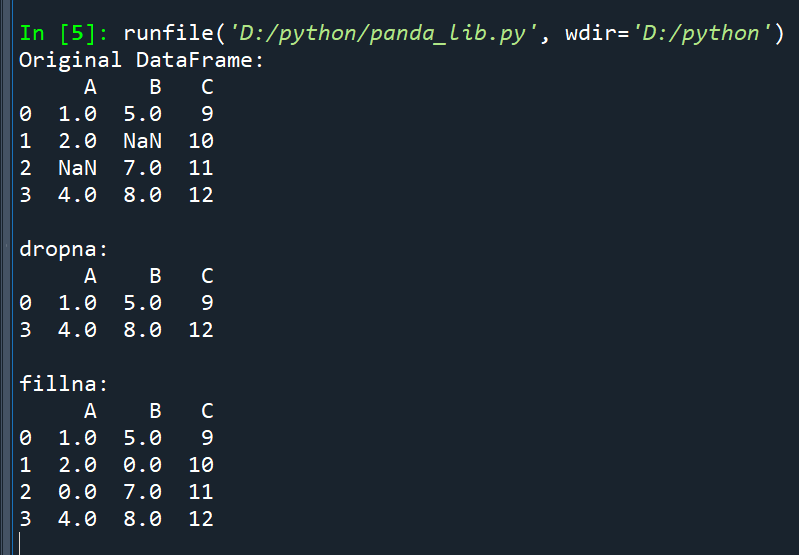
print("\nfillna:")

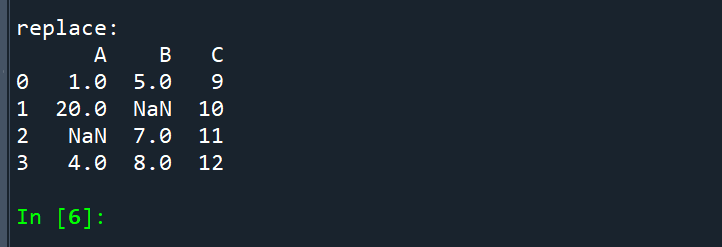
print(my\_fillna(df, 0))

print("\nreplace:")

print(my\_replace(df, 2, 20))

Output:





**Q3] Write a program to implement following graphs in matplotlib library Line chart, bar chart, scatterplot, pie chart, histogram**

import matplotlib.pyplot as plt

import numpy as np

# Create sample data

x = np.arange(1, 11)

y = np.array([2, 4, 6, 8, 10, 12, 14, 16, 18, 20])

# Line Chart

plt.figure(figsize=(8, 4))

plt.plot(x, y, marker='o', linestyle='-', color='b', label='Line Chart')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Line Chart Example')

plt.legend()

plt.grid(True)

plt.show()

# Bar Chart

categories = ['Category A', 'Category B', 'Category C', 'Category D']

values = [15, 24, 10, 30]

plt.figure(figsize=(8, 4))

plt.bar(categories, values, color='g', alpha=0.7)

plt.xlabel('Categories')

plt.ylabel('Values')

plt.title('Bar Chart Example')

plt.show()

# Scatterplot

x = np.random.rand(50)

y = np.random.rand(50)

plt.figure(figsize=(8, 4))

plt.scatter(x, y, c='r', marker='o', label='Scatterplot')

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Scatterplot Example')

plt.legend()

plt.grid(True)

plt.show()

# Pie Chart

labels = ['Category A', 'Category B', 'Category C', 'Category D']

sizes = [30, 15, 25, 20]

colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']

plt.figure(figsize=(6, 6))

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)

plt.axis('equal')

plt.title('Pie Chart Example')

plt.show()

# Histogram

data = np.random.randn(1000)

plt.figure(figsize=(8, 4))

plt.hist(data, bins=20, color='purple', alpha=0.7)

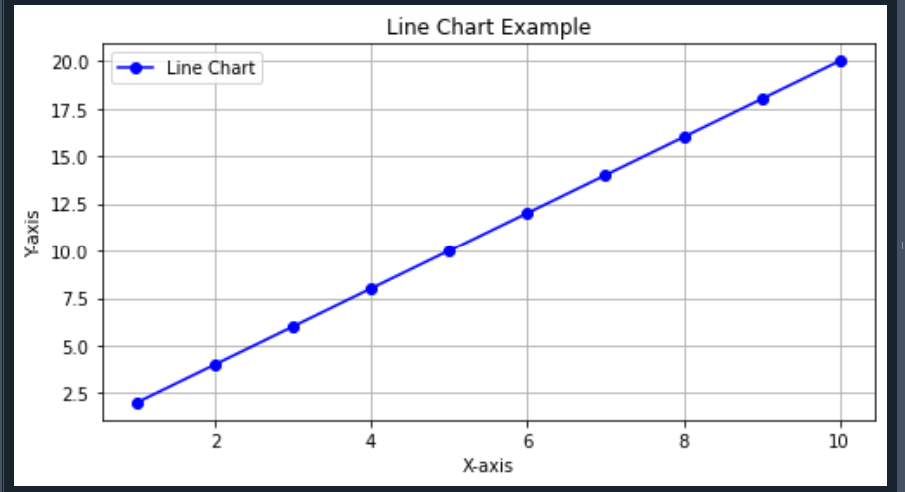
plt.xlabel('Value')

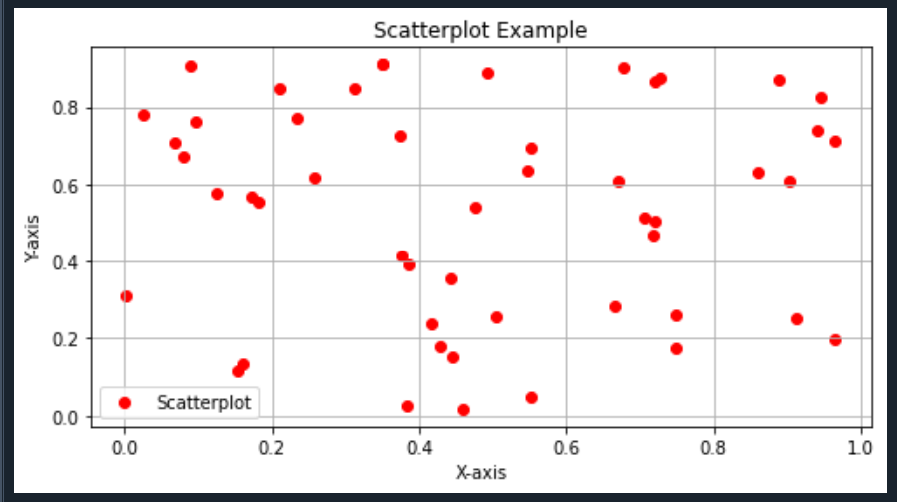
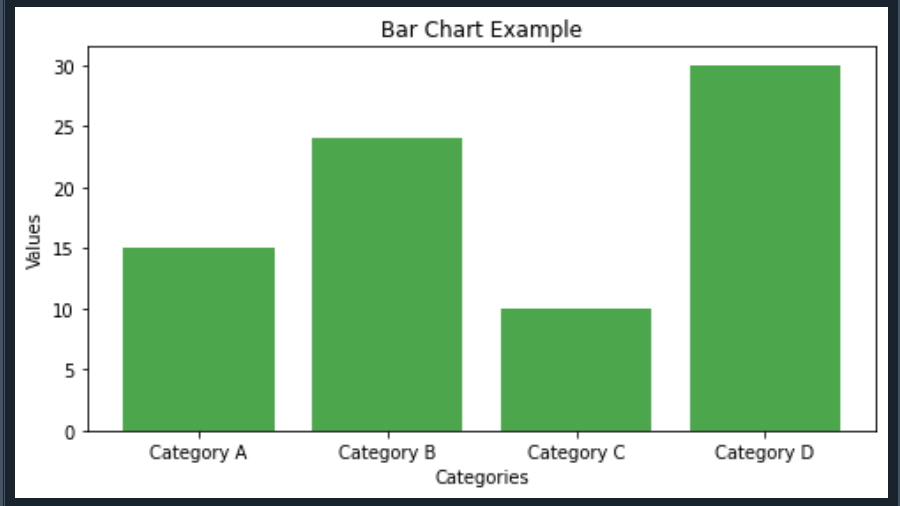
plt.ylabel('Frequency')

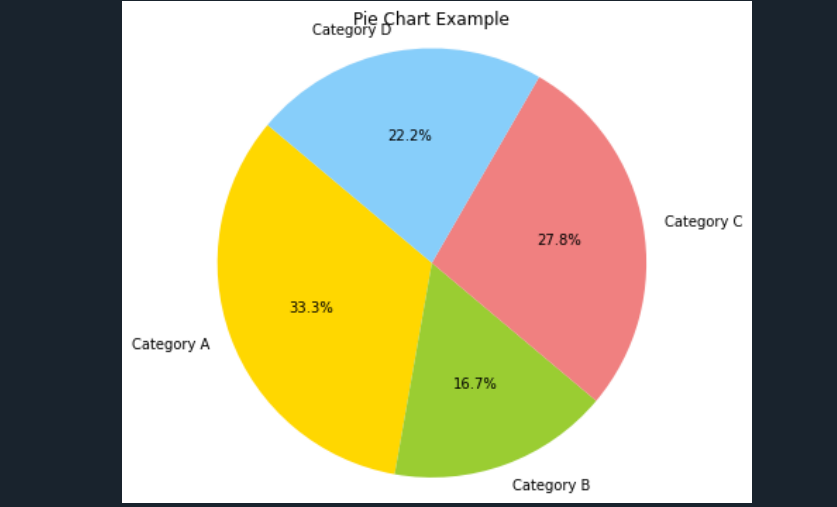
plt.title('Histogram Example')

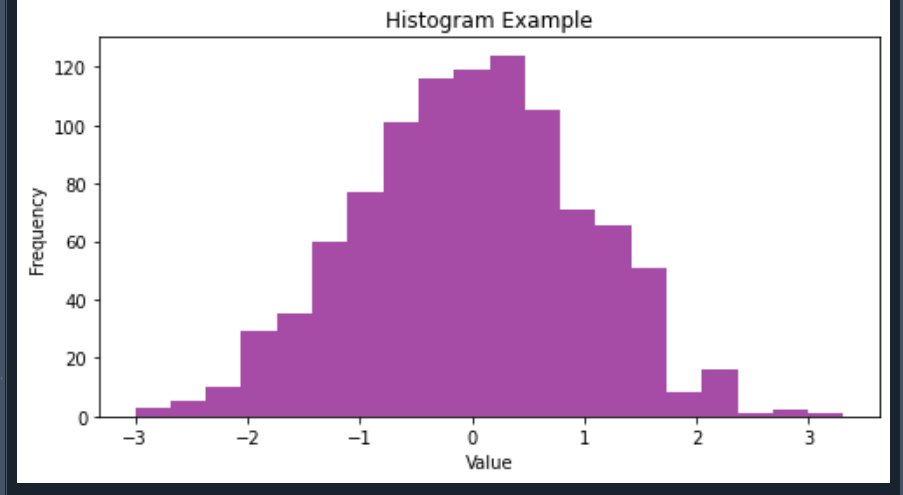
plt.show()

Output:









**Q4] Write a program to find the correlation matrix**

import pandas as pd

# Sample dataset

data = {

'A': [1, 2, 3, 4, 5],

'B': [2, 4, 1, 6, 8],

'C': [5, 7, 3, 8, 9],

'D': [1, 2, 3, 2, 1]

}

# Create a DataFrame

df = pd.DataFrame(data)

# Calculate the correlation matrix

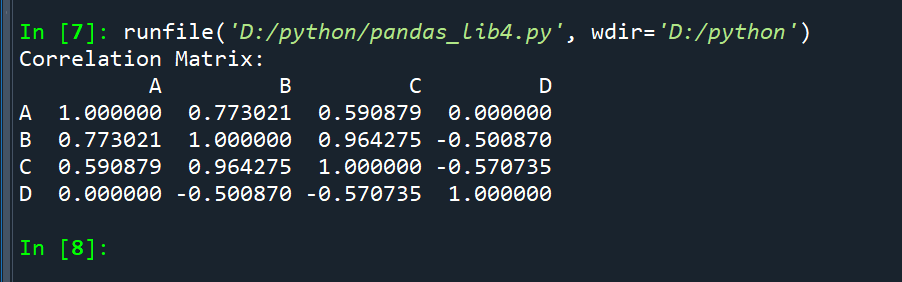
correlation\_matrix = df.corr()

# Display the correlation matrix

print("Correlation Matrix:")

print(correlation\_matrix)

Output:



**Q5] Create an Android application and understand the Project and file hierarchy.**

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">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.36" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="This is First program!"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.483" />

</androidx.constraintlayout.widget.ConstraintLayout>

Mainactivity.java:

package com.example.helloworld;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

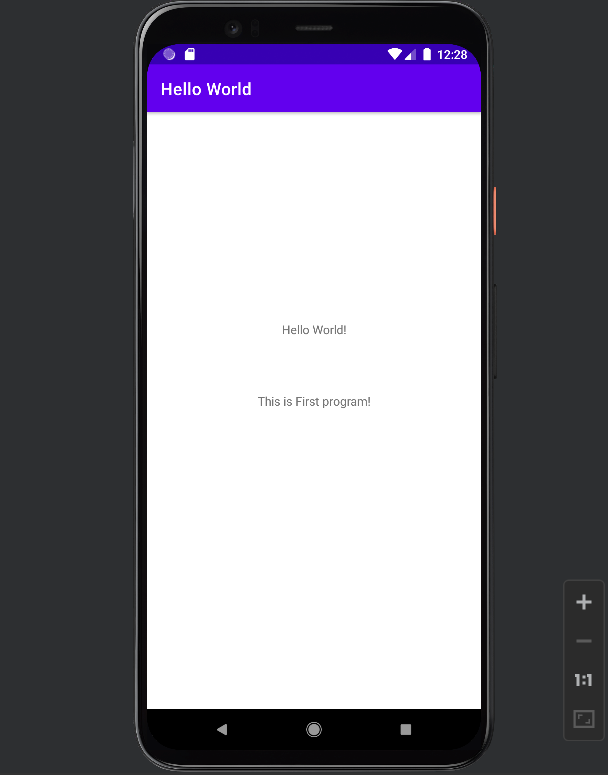
super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

Output:



**Q6] Develop an Android application that uses GUI components, Font and Colors**

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">

<CheckBox

android:id="@+id/cb1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="36dp"

android:text="Java"

android:textColor="#A6680B"

android:fontFamily="sans-serif-light"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.09" />

<CheckBox

android:id="@+id/cb2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="36dp"

android:text="DSA"

android:textColor="#A6680B"

android:fontFamily="sans-serif-light"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.175" />

<CheckBox

android:id="@+id/cb3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="36dp"

android:layout\_marginTop="4dp"

android:text="JavaScript"

android:textColor="#A6680B"

android:fontFamily="sans-serif-light"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/btn"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="50sp"

android:background="@color/black"

android:text="Selected Languages"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.245"

tools:layout\_editor\_absoluteX="16dp" />

<Button

android:id="@+id/clr"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="50sp"

android:background="@color/black"

android:text="Clear"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.448"

tools:layout\_editor\_absoluteX="16dp" />

<TextView

android:id="@+id/txtv1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=""

android:textColor="#FF3700B3"

android:textSize="30dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.691"

tools:layout\_editor\_absoluteX="0dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

Mainactivity.java:

@Override

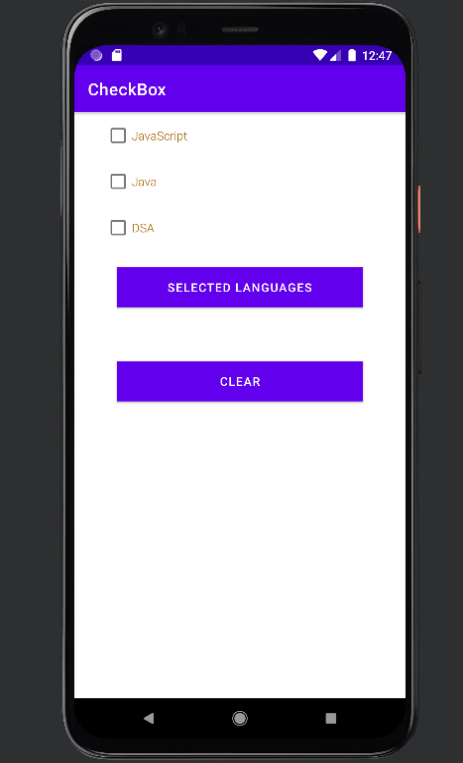
protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

Output:



**Q7] Develop an Android application that uses Layout Managers**

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\_height="match\_parent"

android:layout\_width="match\_parent"

tools:context=".MainActivity">

<TextView

android:id="@+id/firstNumberLabel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Enter First Number:"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:layout\_marginEnd="8dp"/>

<EditText

android:id="@+id/firstNumberEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_toEndOf="@+id/firstNumberLabel"

android:layout\_alignBaseline="@+id/firstNumberLabel"

android:hint="Enter the first number"/>

<TextView

android:id="@+id/secondNumberLabel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Enter Second Number:"

android:layout\_below="@+id/firstNumberLabel"

android:layout\_alignStart="@+id/firstNumberLabel"

android:layout\_marginTop="16dp"

android:layout\_marginEnd="8dp"/>

<EditText

android:id="@+id/secondNumberEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_toEndOf="@+id/secondNumberLabel"

android:layout\_alignBaseline="@+id/secondNumberLabel"

android:hint="Enter the second number"/>

<Button

android:id="@+id/multiplyButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Multiply"

android:layout\_centerHorizontal="true"

android:layout\_below="@+id/secondNumberLabel"

android:layout\_marginTop="16dp"/>

<TextView

android:id="@+id/resultTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text=""

android:layout\_centerHorizontal="true"

android:layout\_below="@+id/multiplyButton"

android:layout\_marginTop="16dp"/>

</RelativeLayout>

Mainactivity.java:

package com.example.layouts1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

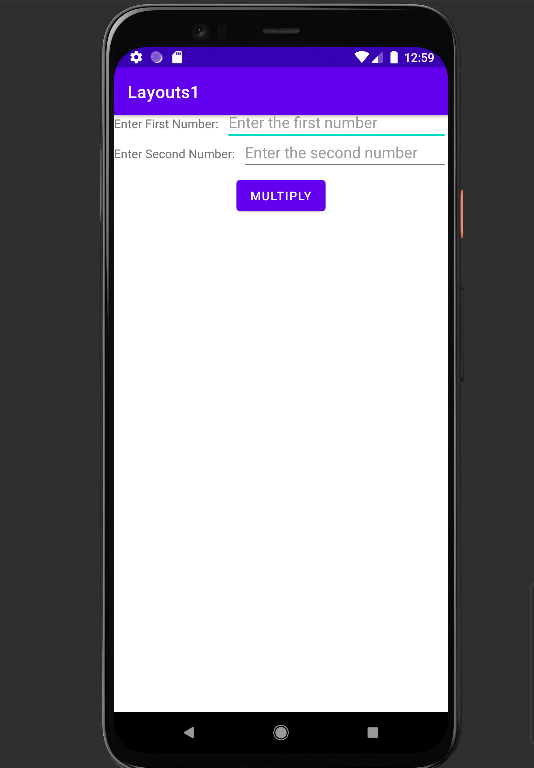
super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

Output:



**Q8] Develop an Android application that uses Layout Managers Login Form**

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/et\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Username"/>

<EditText

android:id="@+id/et\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Password"

android:inputType="textPassword"/>

<Button

android:id="@+id/btn\_show\_toast"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Login"

android:layout\_gravity="center"/>

</LinearLayout>

Mainactivity.java:

package com.example.layoutloginform;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

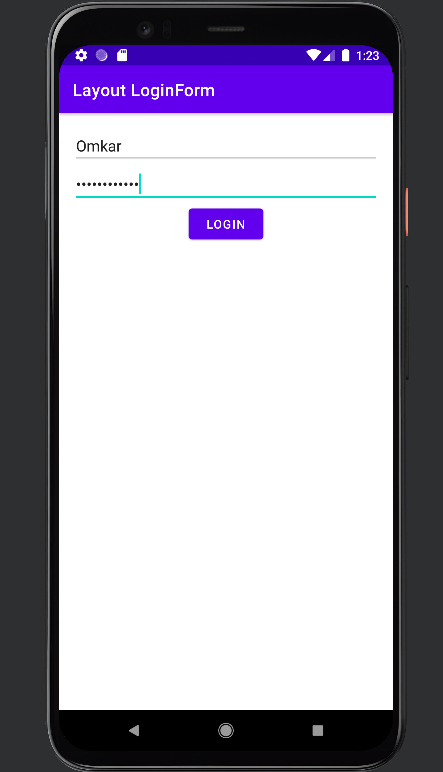
super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

Output:



**Q9] Develop an Android application that uses Layout Managers and event listeners. Registration form**

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/etName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Full Name" />

<EditText

android:id="@+id/etEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Email Address" />

<EditText

android:id="@+id/etPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Password"

android:inputType="textPassword" />

<Button

android:id="@+id/btnRegister"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Register" />

</LinearLayout>

Mainactivity.java:

package com.example.registrationform;

import androidx.appcompat.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

private EditText etName, etEmail, etPassword;

private Button btnRegister;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

etName = findViewById(R.id.etName);

etEmail = findViewById(R.id.etEmail);

etPassword = findViewById(R.id.etPassword);

btnRegister = findViewById(R.id.btnRegister);

btnRegister.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String name = etName.getText().toString();

String email = etEmail.getText().toString();

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

String message = "Name: " + name + "\nEmail: " + email + "\nPassword: " + password;

Toast.makeText(MainActivity.this, message, Toast.LENGTH\_LONG).show();

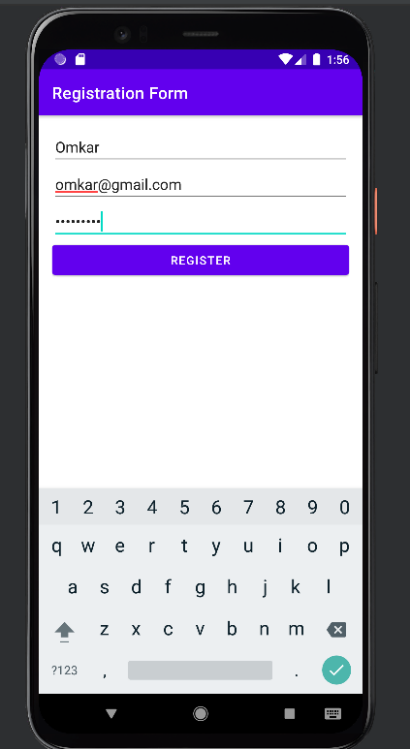
}

});

}

}

Output:



**Q10] Develop an Android application that uses Layout Managers and event listeners. Subscription form and Login form.**

Activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

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

android:id="@+id/subscriptionLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<!-- Subscription Form -->

<EditText

android:id="@+id/etSubscriptionEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Email Address" />

<Button

android:id="@+id/btnSubscribe"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Subscribe" />

<!-- Login Form -->

<LinearLayout

android:id="@+id/loginLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:orientation="vertical">

<EditText

android:id="@+id/etLoginUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Username" />

<EditText

android:id="@+id/etLoginPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Password"

android:inputType="textPassword" />

<Button

android:id="@+id/btnLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Login" />

</LinearLayout>

</LinearLayout>

Mainactivity.java:

package com.example.suscriptionform;

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 EditText etSubscriptionEmail, etLoginUsername, etLoginPassword;

private Button btnSubscribe, btnLogin;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

etSubscriptionEmail = findViewById(R.id.etSubscriptionEmail);

btnSubscribe = findViewById(R.id.btnSubscribe);

etLoginUsername = findViewById(R.id.etLoginUsername);

etLoginPassword = findViewById(R.id.etLoginPassword);

btnLogin = findViewById(R.id.btnLogin);

btnSubscribe.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// Handle subscription logic here

String email = etSubscriptionEmail.getText().toString();

String message = "Subscribed with email: " + email;

Toast.makeText(MainActivity.this, message, Toast.LENGTH\_LONG).show();

}

});

btnLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// Handle login logic here

String username = etLoginUsername.getText().toString();

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

// Display a toast message for login

String message = "Logged in as: " + username;

Toast.makeText(MainActivity.this, message, Toast.LENGTH\_LONG).show();

}

});

}

}

Output:

