**Name**

**:**

**Reg.No : Subject :\_**

**Semester:**

VSB ENGINEERING COLLEGE

**(An Autonomous Institution)**

### KARUDAYAMPALAYAM,KARUR-639 111



**RECORD NOTE BOOK**

VSB ENGINEERING COLLEGE

**(An Autonomous Institution)**

### VSB College of Engineering Technical Campus, CoimbatoreKARUDAYAMPALAYAM,KARUR-639 111

**RECORD**

**NAME :** \_ **CLASS** :

**REGISTER NO: BRANCH:**\_

**ROLL NO :**

***Certified that this is a bonafied record of work done by the above student during the year 20-----20-----***

**Date: Lab in-charge Head of the Department**

Submitted for the **“UNIVERSITY PRACTICAL EXAMINATION”**

held on**\_**

**Internal Examiner External Examiner**

**CONTENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Date** | **Name of the Experiment** | **Page. No** | **Marks Awarded** | **Sign** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**CONTENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Date** | **Name of the**  **Experiment** | **Page.**  **No** | **Marks**  **Awarded** | **Sign** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**V.S.B. ENGINEERING COLLEGE, KARUR**



**(An Autonomous Institution) DEPARTMENT OF INFORMATION TECHNOLOGY ACADEMIC YEAR: 2023-2024 (EVEN Semester)**

**VISION, MISSION, PEOs, POs and PSOs**

**Vision of the Institution:**

We endeavor to impart futuristic technical education of the highest quality to the student community and to inculcate discipline in them to face the world with self-confidence and thus we prepare them for life as responsible citizens to uphold human values and to be of service at large. We strive to bring of the Institution as an Institution of academic excellence of International standard.

**Mission of the Institution:**

We transform persons into personalities by the state-of the art infrastructure, time consciousness, quick response and the best academic practices through assessment and advice.

**Vision of the Department:**

To provide professional computing training to make competent IT engineers and thus prepare them to work in the emerging trends of information technology field.

**Mission of the Department:**

1. Producing quality IT engineers with good technical knowledge by effective teaching.
2. Making competent engineers to adapt to the dynamic needs of industries.
3. Developing extensive learning skills in studies.
4. Inculcating strong ethical values and professionalism to serve society with responsibility.

**Program Educational Objectives (PEOs)**

PEO 1: To equip graduates with proficiency in utilizing fundamental knowledge across various streams in engineering and technology, specifically focusing on mobile application development.

PEO 2: To enrich graduates with core competencies necessary for applying knowledge of mobile computing and technologies in the context of business enterprise.

PEO 3: To enable graduates to think logically and pursue lifelong learning to understand technical issues related to mobile computing systems and provide optimal solutions in the rapidly evolving field of mobile application development.

PEO 4: To enable graduates to develop hardware and software systems for mobile platforms, with a deep understanding of the importance of social, business, and environmental needs in the mobile computing.

**Program Outcomes (POs)**

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems & design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write

effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**Program Specific Outcome (PSOs)**

PSO1: Apply the mathematical and computing knowledge to identify and provide solutions for computing problems.

PSO2: Design and develop computer programs in the areas related to algorithms, networking, web design, and data analytics of varying complexity.

PSO3: Apply standard Engineering practices and strategies in software project using open source environment to deliver a quality product for business success.

**COURSE OBJECTIVES**

1. To learn the basic programming constructs in Flutter/Kotlin.

2. To practice various computing strategies for Flutter/Kotlin-based solutions

to real-world problems.

3. To use Flutter/Kotlin data structures - lists, tuples, dictionaries.

4. To perform input/output operations in Flutter/Kotlin applications.

**COURSE OUTCOMES**

1. Develop algorithmic solutions to simple computational problems.

2. Develop and execute simple Flutter/Kotlin programs.

3. Implement programs in Flutter/Kotlin using conditionals and loops for solving problems.

4. Deploy functions to decompose a Flutter/Kotlin program. Process compound data using Flutter/Kotlin data structures.

5. Utilize Flutter/Kotlin packages in developing mobile applications.

**CO-PO-PSO MATRIX**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Outcomes** | **Program Outcomes** | | | | | | | | | | | | **PSOs** | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **1** | **2** | **3** |
| **CO1** | 2 | 2 | 2 | - | 3 | - | - | - | - | - | - | 2 | 2 | 3 | 1 |
| **CO2** | 2 | 2 | 2 | - | 3 | - | - | - | - | - | - | 2 | 2 | 3 | 1 |
| **CO3** | 2 | 2 | 2 | - | 3 | - | - | - | - | - | - | 2 | 2 | 3 | 1 |
| **CO4** | 2 | 2 | 2 | - | 3 | - | - | - | - | - | - | 2 | 2 | 3 | 1 |
| **CO5** | 2 | 2 | 2 | - | 3 | - | - | - | - | - | - | 2 | 2 | 3 | 1 |
| **C110 (Average)** | **2** | **2** | **2** | **-** | **3** | **-** | **-** | **-** | **-** | **-** | **-** | **2** | **2** | **3** | **1** |

**EX. NO: 1**

**DATE:**

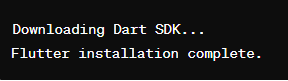
**STUDY AND INSTALLATION OF FLUTTER/KOTLIN MULTI-PLATFORM ENVIRONMENT**

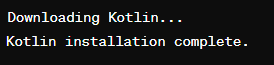
**Aim:**

To study and installation of Flutter/ Kotlin multi-platform environment.

**Procedure:**

Step 1: Install Flutter, Kotlin





Step 2: Create a Flutter Project:

Use the following command to create a new Flutter project:

flutter create my\_flutter\_project

cd my\_flutter\_project



Replace "my\_flutter\_project" with your desired project name.

Step 3: Configure Flutter for Kotlin:

Open the android/app/build.gradle file in your Flutter project and add the following lines to configure Kotlin:

def localProperties = new Properties()

def localPropertiesFile = rootProject.file('local.properties')

if (localPropertiesFile.exists()) {

localPropertiesFile.withReader('UTF-8') { reader ->

localProperties.load(reader)

}

}

android {

// ...

kotlin {

sourceSets {

commonMain {

dependencies {

implementation 'org.jetbrains.kotlin:kotlin-stdlib-common'

}

}

androidMain {

dependencies {

implementation 'org.jetbrains.kotlin:kotlin-stdlib'

}

}

}

}

}

This configures Kotlin for the common and Android source sets.



Step 4: Create a Kotlin Multiplatform Module:

Create a new directory for your Kotlin Multiplatform module:

mkdir my\_flutter\_project/kotlin\_module

cd my\_flutter\_project/kotlin\_module

Inside the kotlin\_module directory, create a build.gradle.kts file with your Kotlin Multiplatform configuration.

Step 5: Integrate Flutter and Kotlin

In your Flutter project, you can use the MethodChannel class to establish communication between Flutter and Kotlin.

Example Flutter code (Dart):

import 'package:flutter/services.dart';

const platform = const MethodChannel('your\_channel\_name');

Future<void> invokeKotlinFunction() async {

try {

final int result = await platform.invokeMethod('your\_kotlin\_function');

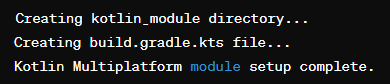
print(result);

} on PlatformException catch (e) {

print("Failed to invoke Kotlin function: ${e.message}");

}

}



Implement the Kotlin function that corresponds to the method channel in your Kotlin module.

Example Kotlin code:

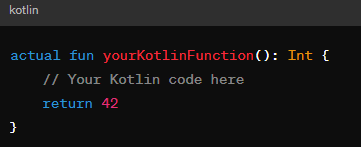
// Inside your Kotlin module

actual fun yourKotlinFunction(): Int {

// Your Kotlin code here

return 42

}

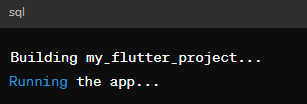


Step 6: Build and Run

Build and run your Flutter project:

cd my\_flutter\_project

flutter run



This is the set up of a Flutter/Kotlin multi-platform environment.

**EX. NO: 2 A**

**DATE:**

**DEVELOP AN APPLICATION THAT USES GUI COMPONENTS FONT AND COLORS**

**Aim:**

To develop a Simple Android Application that uses GUI components, Font and Colors.

**Procedure:**

**Flutter Project Setup:**

* Install Flutter SDK and set it up by following the instructions provided in the official Flutter documentation: Flutter Installation Guide
* Open Android Studio and install the Flutter plugin if you haven't already. You can install it via Settings/Preferences > Plugins > Search for Flutter > Install.
* Create a new Flutter project in Android Studio by selecting File > New > New Flutter Project.
* Choose a project name, Flutter SDK path, and other configurations as per your preference.
* Replace the lib/main.dart file with the provided Flutter UI code.
* Run the Flutter project by clicking the Run button in Android Studio or using the flutter run command in the terminal.

**Kotlin Project Setup:**

* Install Kotlin SDK by following the instructions provided in the official Kotlin documentation: Kotlin Installation Guide
* Create a new Kotlin project in Android Studio by selecting File > New > New Project.
* Choose Kotlin as the language and configure other settings as per your preference.
* Create a new Kotlin file for the backend code and paste the provided Kotlin backend code.
* Ensure that the Kotlin file is included in the project structure.
* Run the Kotlin project by clicking the Run button in Android Studio or using the terminal.
* Integration:
* Ensure that both the Flutter and Kotlin projects are open in Android Studio.
* Ensure that the Flutter project depends on the Kotlin project. You can do this by adding the Kotlin project as a module dependency in the Flutter project settings.
* Communicate between Flutter UI and Kotlin backend by using platform channels or other suitable methods as per your project requirements.

Test the integrated project thoroughly to ensure proper communication and functionality.

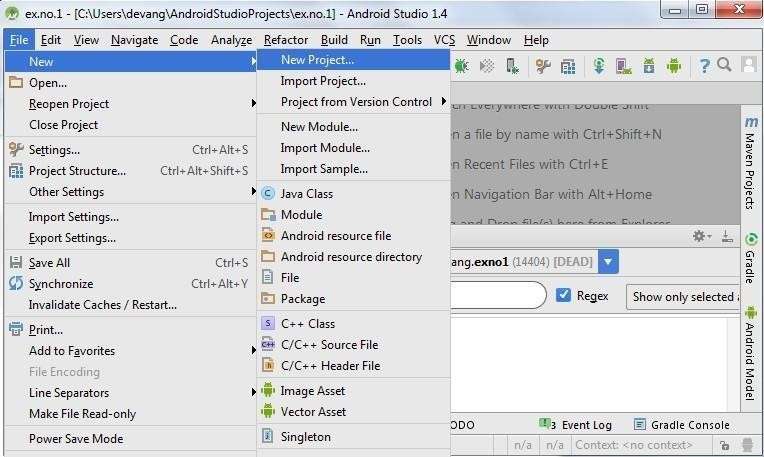
**Creating a New project:**

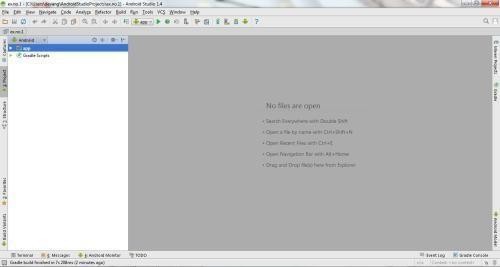
Open Android Studio and then click on **File -> New -> New project**.

Then type the Application name as **“exno1″**and click Next. Then **select the Minimum SDK** as shown below and click Next. Then select the **Empty Activity** and click Next.

Finally click **Finish**.

It will take some time to build and load the project. After completion it will look as given below.





**Designing layout for the Android Application:**

Click on **app -> res -> layout -> activity\_main.xml**. Now click on Text as shown below.

Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

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

<LinearLayout

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical"

android:background="#F09011">

<TextView

android:id="@+id/textView1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20sp"

android:gravity="center"

android:text="HELO!!!!! MOUNT ZION"

android:textSize="20sp"

android:fontFamily="sans-serif-medium"

android:textStyle="bold" />

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Change font size"

android:textSize="20sp" />

<Button

android:id="@+id/button2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Change color"

android:textSize="20sp" />

</LinearLayout>  
  
  
  
So now the designing part is completed.

**Java Coding for the Android Application:**

Click on **app -> java -> com.example.exno1 -> MainActivity**.

Then delete the code which is there and type the code as given below.

**Code for MainActivity.kt:**

package com.example.guiapp

import android.graphics.Color

import android.os.Bundle

import android.view.View

import android.widget.Button

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

var font = 24f

var al=26f

var i = 1

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val t1 = findViewById<TextView>(R.id.textView1)

val b1 = findViewById<Button>(R.id.button1)

b1.setOnClickListener {

t1.setTextSize(font)

font += 4

if (font == 40f) font = 20f

}

val b2 = findViewById<Button>(R.id.button2)

b2.setOnClickListener {

when (i) {

1 -> t1.setTextColor(Color.parseColor("#0000FF"))

2 -> t1.setTextColor(Color.parseColor("#00FF00"))

3 -> t1.setTextColor(Color.parseColor("#FF0000"))

4 -> t1.setTextColor(Color.parseColor("#800000"))

}

i++

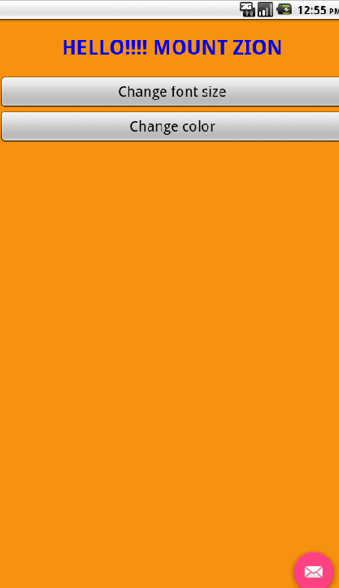
if (i == 5) i = 1

}

}

}

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result:**

**EX. NO: 2 B**

**DATE:**

**DEVELOP AN APPLICATION THAT USES GUI COMPONENTS FONT AND COLOURS**

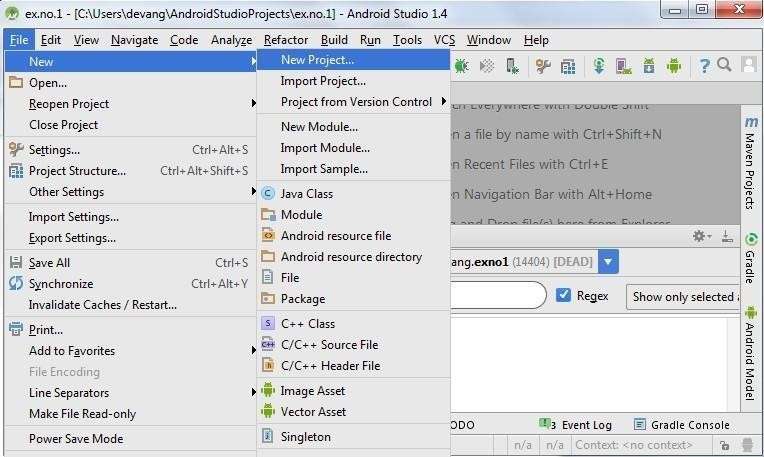
**Aim:**

To develop a Simple Android Application that uses GUI components

**Procedure:**

**Creating a New project:**

Open Android Studio and then click on **File -> New -> New project**.



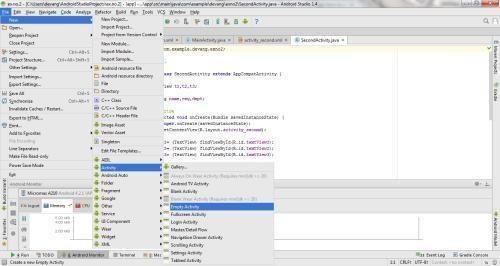
Then type the Application name as **“exno2″** and click Next. Then **select the Minimum SDK** as shown below and click Next. Then **select the Empty Activity** and click Next.

Finally click **Finish**.

It will take some time to build and load the project. After completion it will look as given below.

**Creating Second Activity for the Android Application:**

Click on **File -> New -> Activity -> Empty Activity**.



Type the Activity Name as **SecondActivity** and click Finish button. Thus Second Activity For the application is created.

**Designing Layout for Main Activity:**

Click on **app -> res -> layout -> activity\_main.xml**. Now click on Text as shown below.

Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

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

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="126dp"

android:gravity="center"

tools:context=".MainActivity">

<TextView

android:id="@+id/textView3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Widget"

android:textSize="40dp"

android:layout\_marginBottom="60dp"

android:layout\_gravity="top"/>

<EditText

android:id="@+id/editText"

android:layout\_marginBottom="30dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter text" />

<Button

android:id="@+id/changeButton"

android:layout\_marginBottom="30dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Change Text" />

<CheckBox

android:id="@+id/checkBox"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Bold" />

<CheckBox

android:id="@+id/textcolor"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="color-red" />

<CheckBox

android:id="@+id/checkBoxItalic"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Italic" />

<Spinner

android:id="@+id/fontSizeSpinner"

android:layout\_marginBottom="30dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:text="Hello World!"

android:textSize="24sp" />

</LinearLayout>

**Create: res/values/font\_sizes.xml**

<resources>

<string-array name="font\_sizes">

<item>12</item>

<item>16</item>

<item>20</item>

<item>24</item>

<item>28</item>

<item>32</item>

</string-array>

</resources>

**MainActivity.kt**

package com.example.widgetapp

import android.graphics.Color

import android.graphics.Typeface

import android.os.Bundle

import android.widget.\*

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

lateinit var textView: TextView

lateinit var editText: EditText

lateinit var changeButton: Button

lateinit var checkBox: CheckBox

lateinit var textcolor: CheckBox

lateinit var checkBoxItalic: CheckBox

lateinit var fontSizeSpinner: Spinner

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

textView = findViewById<TextView>(R.id.textView)

editText = findViewById<EditText>(R.id.editText)

changeButton = findViewById<Button>(R.id.changeButton)

checkBox = findViewById<CheckBox>(R.id.checkBox)

textcolor = findViewById<CheckBox>(R.id.textcolor)

checkBoxItalic = findViewById<CheckBox>(R.id.checkBoxItalic)

fontSizeSpinner = findViewById<Spinner>(R.id.fontSizeSpinner)

// Populate font size spinner

val fontSizeAdapter = ArrayAdapter.createFromResource(this, R.array.font\_sizes, android.R.layout.simple\_spinner\_item)

fontSizeAdapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item)

fontSizeSpinner.adapter = fontSizeAdapter

changeButton.setOnClickListener {

val newText = editText.text.toString()

if (newText.isNotEmpty()) {

textView.text = newText

}

// Set text size

val selectedSize = fontSizeSpinner.selectedItem.toString().toFloat()

textView.textSize = selectedSize

// Set text color based on checkbox state

if (textcolor.isChecked) {

textView.setTextColor(Color.RED)

} else {

textView.setTextColor(Color.BLACK)

}

// Set text style

val style = when {

checkBox.isChecked && checkBoxItalic.isChecked -> Typeface.BOLD\_ITALIC

checkBox.isChecked -> Typeface.BOLD

checkBoxItalic.isChecked -> Typeface.ITALIC

else -> Typeface.NORMAL

}

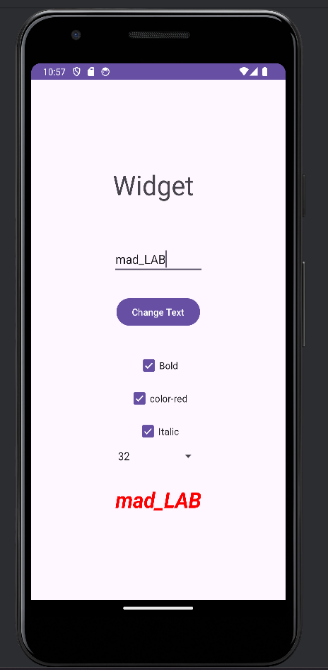
textView.setTypeface(null, style)

}

}

}

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result:**

**EX.NO.3**

**DATE:**

**DEVELOP A NATIVE CALCULATOR APPLICATION**

**Aim:**

To develop a Simple Android Application for the native calculator with its functions

**Procedure:**

**Creating a New project:**

Open Android Studio and then click on **File -> New -> New project**. Then type the Application name as **“exno3″** and click Next.

Then **select the Minimum SDK** as shown below and click Next. Then **select the Empty Act**ivity and click Next.

Finally click **Finish**.

It will take some time to build and load the project. After completion it will look as given below.

**Designing layout for the Android Application:**

Click on **app -> res -> layout -> activity\_main.xml**. Now click on Text as shown below.

Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

<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:gravity="center"

android:background="#FFFFDD">

<EditText

android:id="@+id/result\_id"

android:layout\_width="370dp"

android:textSize="30sp"

android:textStyle="bold"

android:gravity="right"

android:hint="0"

android:layout\_height="70dp"/>

<Button

android:id="@+id/Btn7\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/result\_id"

android:text="7"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn7Clicked"/>

<Button

android:id="@+id/Btn8\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/result\_id"

android:layout\_toRightOf="@id/Btn7\_id"

android:text="8"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn8Clicked"/>

<Button

android:id="@+id/Btn9\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/result\_id"

android:layout\_toRightOf="@id/Btn8\_id"

android:text="9"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn9Clicked"/>

<Button

android:id="@+id/Btnclear\_id"

android:layout\_width="100dp"

android:layout\_height="60dp"

android:layout\_below="@id/result\_id"

android:layout\_toRightOf="@id/Btn9\_id"

android:text="CLEAR"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btnclearClicked"/>

<Button

android:id="@+id/Btn4\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn7\_id"

android:text="4"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn4Clicked"/>

<Button

android:id="@+id/Btn5\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn8\_id"

android:layout\_toRightOf="@id/Btn4\_id"

android:text="5"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn5Clicked"/>

<Button

android:id="@+id/Btn6\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn9\_id"

android:layout\_toRightOf="@id/Btn5\_id"

android:text="6"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn6Clicked"/>

<Button

android:id="@+id/Btnplus\_id"

android:layout\_width="100dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btnclear\_id"

android:layout\_toRightOf="@id/Btn6\_id"

android:text="+"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btnplusClicked"/>

<Button

android:id="@+id/Btn1\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn4\_id"

android:text="1"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn1Clicked"/>

<Button

android:id="@+id/Btn2\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn5\_id"

android:layout\_toRightOf="@id/Btn1\_id"

android:text="2"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn2Clicked"/>

<Button

android:id="@+id/Btn3\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn6\_id"

android:layout\_toRightOf="@id/Btn2\_id"

android:text="3"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn3Clicked"/>

<Button

android:id="@+id/Btnminus\_id"

android:layout\_width="100dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btnplus\_id"

android:layout\_toRightOf="@id/Btn3\_id"

android:text="-"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btnminusClicked"/>

<Button

android:id="@+id/Btn0\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn1\_id"

android:text="0"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btn0Clicked"/>

<Button

android:id="@+id/Btnequal\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn1\_id"

android:layout\_toRightOf="@id/Btn0\_id"

android:text="="

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btnequalClicked"/>

<Button

android:id="@+id/Btndivide\_id"

android:layout\_width="90dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btn1\_id"

android:layout\_toRightOf="@id/Btnequal\_id"

android:text="/"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btndivideClicked"/>

<Button

android:id="@+id/Btnmulti\_id"

android:layout\_width="100dp"

android:layout\_height="60dp"

android:layout\_below="@id/Btnminus\_id"

android:layout\_toRightOf="@id/Btndivide\_id"

android:text="\*"

android:textSize="20sp"

android:textStyle="bold"

android:onClick="btnmultiClicked"/>

</RelativeLayout>

**Java Coding for the Android Application:**

Click on **app -> java -> com.example.exno3 -> MainActivity**.

Then delete the code which is there and type the code as given below.

Mainactivity.kt

package com.example.calculator

import android.app.Activity

import android.os.Bundle

import android.view.View

import android.widget.EditText

class MainActivity : Activity() {

private var str = ""

private var op: Char = 'q'

private var num = 0

private var numtemp = 0

private lateinit var showResult: EditText

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

showResult = findViewById(R.id.result\_id)

}

fun btn0Clicked(v: View) {

insert(0)

}

fun btn1Clicked(v: View) {

insert(1)

}

fun btn2Clicked(v: View) {

insert(2)

}

fun btn3Clicked(v: View) {

insert(3)

}

fun btn4Clicked(v: View) {

insert(4)

}

fun btn5Clicked(v: View) {

insert(5)

}

fun btn6Clicked(v: View) {

insert(6)

}

fun btn7Clicked(v: View) {

insert(7)

}

fun btn8Clicked(v: View) {

insert(8)

}

fun btn9Clicked(v: View) {

insert(9)

}

fun btnplusClicked(v: View) {

perform('+')

}

fun btnminusClicked(v: View) {

perform('-')

}

fun btndivideClicked(v: View) {

perform('/')

}

fun btnmultiClicked(v: View) {

perform('\*')

}

fun btnequalClicked(v: View) {

calculate()

}

fun btnclearClicked(v: View) {

reset()

}

private fun reset() {

str = ""

op = 'q'

num = 0

numtemp = 0

showResult.setText("")

}

private fun insert(j: Int) {

str += j.toString()

num = str.toInt()

showResult.setText(str)

}

private fun perform(operation: Char) {

str = ""

op = operation

numtemp = num

}

private fun calculate() {

num = when (op) {

'+' -> numtemp + num

'-' -> numtemp - num

'/' -> numtemp / num

'\*' -> numtemp \* num

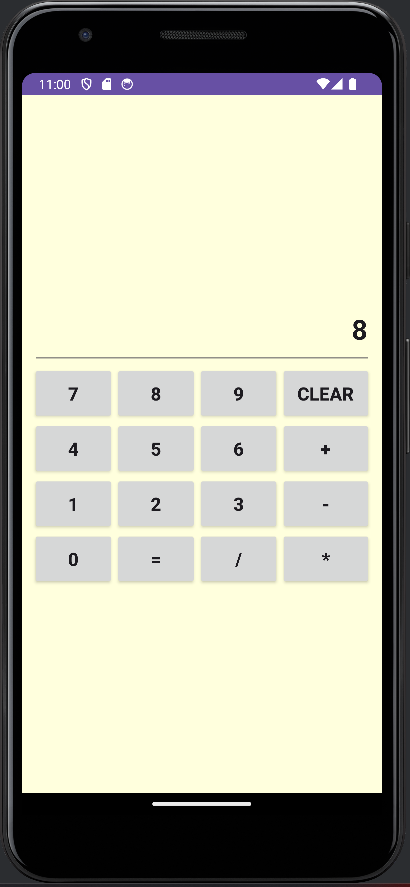
else -> 0

}

showResult.setText(num.toString())

}

}

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result:**

**EX.NO:4**

**DATE:**

**CREATE A GAMING APPLICATION THAT USES 2D GESTURES**

**Aim:**

To develop a Simple Android Application that makes use of gestures.

**Procedure:**

**Creating a New project:**

Open Android Studio and then click on **File -> New -> New project**. Then type the Application name as **“exno4″** and click Next.

Then **select the Minimum S**DK as shown below and click Next. Then **select the Empty Activity** and click Next.

Finally click **Finish**.

It will take some time to build and load the project. After completion it will look as given below.

**Designing layout for the Android Application:**

Click on **app -> res -> layout -> activity\_main.xml**. Now click on Text as shown below.

Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

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

<AbsoluteLayout xmlns:android="<http://schemas.android.com/apk/res/android>" android:layout\_width="match\_parent" android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_x="50dp" android:layout\_y="20dp" android:text="Student Details" android:textSize="30sp" />

<TextView

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_x="20dp" android:layout\_y="110dp" android:text="Enter Rollno:" android:textSize="20sp" />

<EditText

android:id="@+id/Rollno" android:layout\_width="150dp" android:layout\_height="wrap\_content" android:layout\_x="175dp" android:layout\_y="100dp" android:inputType="number" android:textSize="20sp" />

<TextView

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_x="20dp" android:layout\_y="160dp" android:text="Enter Name:" android:textSize="20sp" />

<EditText

android:id="@+id/Name" android:layout\_width="150dp" android:layout\_height="wrap\_content" android:layout\_x="175dp" android:layout\_y="150dp" android:inputType="text" android:textSize="20sp" />

<TextView

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_x="20dp" android:layout\_y="210dp" android:text="Enter Marks:" android:textSize="20sp" />

<EditText

android:id="@+id/Marks" android:layout\_width="150dp" android:layout\_height="wrap\_content"

android:layout\_x="175dp" android:layout\_y="200dp" android:inputType="number" android:textSize="20sp" />

<Button

android:id="@+id/Insert" android:layout\_width="150dp" android:layout\_height="wrap\_content" android:layout\_x="25dp" android:layout\_y="300dp" android:text="Insert" android:textSize="30dp" />

<Button

android:id="@+id/Delete" android:layout\_width="150dp" android:layout\_height="wrap\_content" android:layout\_x="200dp" android:layout\_y="300dp" android:text="Delete" android:textSize="30dp" />

<Button

android:id="@+id/Update" android:layout\_width="150dp" android:layout\_height="wrap\_content" android:layout\_x="25dp" android:layout\_y="400dp" android:text="Update" android:textSize="30dp" />

<Button

android:id="@+id/View" android:layout\_width="150dp" android:layout\_height="wrap\_content" android:layout\_x="200dp" android:layout\_y="400dp" android:text="View"

android:textSize="30dp" />

<Button

android:id="@+id/ViewAll" android:layout\_width="200dp" android:layout\_height="wrap\_content" android:layout\_x="100dp" android:layout\_y="500dp" android:text="View All" android:textSize="30dp" />

</AbsoluteLayout>

Now click on Design and your application will look as given below. So now the designing part is completed.

**Java Coding for the Android Application:**

Click on **app -> java -> com.example.exno4 -> MainActivity**.

Then delete the code which is there and type the code as given below.

**Code for MainActivity.kt :** packagecom.example.exno4; import android.app.Activity;

import android.app.AlertDialog.Builder; import android.content.Context; import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase; import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener

{

EditText Rollno,Name,Marks;

Button Insert,Delete,Update,View,ViewAll; SQLiteDatabase db;

/\*\* Called when the activity is first created. \*/ @Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Rollno=(EditText)findViewById(R.id.Rollno); Name=(EditText)findViewById(R.id.Name); Marks=(EditText)findViewById(R.id.Marks); Insert=(Button)findViewById(R.id.Insert); Delete=(Button)findViewById(R.id.Delete); Update=(Button)findViewById(R.id.Update); View=(Button)findViewById(R.id.View); ViewAll=(Button)findViewById(R.id.ViewAll);

Insert.setOnClickListener(this); Delete.setOnClickListener(this); Update.setOnClickListener(this); View.setOnClickListener(this); ViewAll.setOnClickListener(this)

;

// Creating database and table

db=openOrCreateDatabase("StudentDB", Context.MODE\_PRIVATE, null); db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name VARCHAR,marks

VARCHAR);");

}

public void onClick(View view)

{

// Inserting a record to the Student table if(view==Insert)

{

// Checking for empty fields if(Rollno.getText().toString().trim().length()==0||

Name.getText().toString().trim().length()==0|| Marks.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter all values"); return;

}

db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+"','"+Marks.getText()+"');");

showMessage("Success", "Record added"); clearText();

}

// Deleting a record from the Student table if(view==Delete)

{

// Checking for empty roll number if(Rollno.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE rollno='"+Rollno.getText()+"'", null); if(c.moveToFirst())

{

db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'"); showMessage("Success", "Record Deleted");

}

else

{

showMessage("Error", "Invalid Rollno");

}

clearText();

}

// Updating a record in the Student table if(view==Update)

{

// Checking for empty roll number if(Rollno.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE rollno='"+Rollno.getText()+"'", null); if(c.moveToFirst()) {

db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" + Marks.getText() + "' WHERE rollno='"+Rollno.getText()+"'");

showMessage("Success", "Record Modified");

}

else {

showMessage("Error", "Invalid Rollno");

}

clearText();

}

// Display a record from the Student table if(view==View)

{

// Checking for empty roll number if(Rollno.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno"); return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE rollno='"+Rollno.getText()+"'", null); if(c.moveToFirst())

{

Name.setText(c.getString(1)); Marks.setText(c.getString(2));

}

else

{

showMessage("Error", "Invalid Rollno");clearText();

}

}

// Displaying all the records if(view==ViewAll)

{

Cursor c=db.rawQuery("SELECT \* FROM student", null); if(c.getCount()==0)

{

showMessage("Error", "No records found"); return;

}

StringBuffer buffer=new StringBuffer(); while(c.moveToNext())

{

buffer.append("Rollno: "+c.getString(0)+"\n"); buffer.append("Name: "+c.getString(1)+"\n"); buffer.append("Marks: "+c.getString(2)+"\n\n");

}

showMessage("Student Details", buffer.toString());

}

}

public void showMessage(String title,String message)

{

Builder builder=new Builder(this); builder.setCancelable(true); builder.setTitle(title);

builder.setMessage(message); builder.show();

}

public void clearText()

{

Rollno.setText("");

Name.setText("");

Marks.setText(""); Rollno.requestFocus();

}

}

So now the Coding part is also completed. Now run the application to see the output.



**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result****:**

**EX.NO.5**

**DATE:**

**IMPLEMENT A WEB SERVER SUPPORTING PUSH NOTIFICATIONS**

Aim:

To develop an Android Application that makes use of Notification Manager.

**Procedure:**

**Creating a New project:**

Open Android Studio and then click on **File -> New -> New project**. Then type the Application name as **“exno5″**and click Next.

Then **select the Minimum SDK** as shown below and click Next. Then **select the Empty Activity** and click Next.

Finally click **Finish**.

It will take some time to build and load the project. After completion it will look as given below.

**Designing layout for the Android Application:**

Click on **app -> res -> layout -> activity\_main.xml.**

Now click on Text as shown below.

Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

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

<LinearLayout xmlns:android=["h](http://schemas.android.com/apk/res/android)t[tp://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" xmlns:app=["h](http://schemas.android.com/apk/res-auto)t[tp://schemas.android.com/apk/res-auto](http://schemas.android.com/apk/res-auto)" xmlns:tools=["h](http://schemas.android.com/tools)t[tp://schemas.android.com/tools](http://schemas.android.com/tools)" android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" android:gravity="center" tools:context=".MainActivity">

<Button android:id="@+id/btnSimpleNotification" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Simple Notification" />

<Button android:id="@+id/btnNotificationIcon" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:text="Notification With Icon" />

<Button android:id="@+id/btnNotificationImage" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Notification With Image" />

<Button android:id="@+id/btnNotificationWithGroupConvo" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Notification With Group Conversation" />

<Button android:id="@+id/btnNotificationSemantic" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Notification Semantic Action" />

</LinearLayout>

**Java Coding for the Android Application:**

Click on **app -> java -> com.example.exno5 -> MainActivity**.

Then delete the code which is there and type the code as given below.

**Code for MainActivity.Kt:**

packagecom.example.exno5;

import android.app.NotificationChannel; import android.app.NotificationManager; import android.app.PendingIntent; import android.content.Context;

import android.content.Intent; import android.net.Uri;

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

import androidx.core.graphics.drawable.IconCompat; import android.os.Bundle;

import android.view.View; import android.widget.Button;

import android.widget.Toast; import java.util.Date;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

NotificationManager notificationManager; NotificationCompat.Builder builder; NotificationChannel channel;

CharSequence charSequence = ""; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

Button btnSimpleNotification = findViewById(R.id.btnSimpleNotification); Button btnNotificationIcon = findViewById(R.id.btnNotificationIcon); Button btnNotificationImage = findViewById(R.id.btnNotificationImage);

Button btnNotificationWithGroupConvo = findViewById(R.id.btnNotificationWithGroupConvo); Button btnNotificationSemantic = findViewById(R.id.btnNotificationSemantic);

charSequence = btnNotificationIcon.getText();

btnSimpleNotification.setOnClickListener(this); btnNotificationIcon.setOnClickListener(this); btnNotificationImage.setOnClickListener(this); btnNotificationWithGroupConvo.setOnClickListener(this); btnNotificationSemantic.setOnClickListener(this);

notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE); CharSequence name = "My Notification";

String description = "yadda yadda";

int importance = NotificationManager.IMPORTANCE\_DEFAULT;

channel = new NotificationChannel("1", name, importance); channel.setDescription(description);

builder = new NotificationCompat.Builder(MainActivity.this, channel.getId())

.setSmallIcon(R.mipmap.ic\_launcher); notificationManager.createNotificationChannel(channel);

}

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.btnSimpleNotification: simpleNotification();

break;

case R.id.btnNotificationIcon: notificationWithIcon(); break;

case R.id.btnNotificationImage: notificationWithImage(); break;

case R.id.btnNotificationWithGroupConvo: notificationWithGroupConvo();

break;

case R.id.btnNotificationSemantic: notificationSemantic();

break;

}

}

private void simpleNotification() {

Person jd = new Person.Builder().setName("JournalDev ") .setImportant(true) .build();

new NotificationCompat.MessagingStyle(jd)

.addMessage("Check me out", new Date().getTime(), jd) .setBuilder(builder);

notificationManager.notify(1, builder.build());

}

private void notificationWithIcon() { Person anupam = new Person.Builder()

.setName("Anupam")

.setIcon(IconCompat.createWithResource(this, R.drawable.index))

.setImportant(true) .build();

new NotificationCompat.MessagingStyle(anupam)

.addMessage("Check out my latest article!", new Date().getTime(), anupam)

.setBuilder(builder);

notificationManager.notify(2, builder.build());

}

private void notificationWithImage() { Person bot = new Person.Builder()

.setName("Bot") .setImportant(true)

.setBot(true) .build();

Uri uri = Uri.parse("android.resource://com.journaldev.androidpnotifications/drawable/"+R.drawable.bg);

NotificationCompat.MessagingStyle.Message message =new NotificationCompat.MessagingStyle.Message("Check out my latest article!", new Date().getTime(), bot); message.setData("image/\*",uri);

new NotificationCompat.MessagingStyle(bot)

.addMessage(message) .setGroupConversation(true).setBuilder(builder);

notificationManager.notify(3, builder.build());

}

private void notificationWithGroupConvo()

{

Person jd = new Person.Builder()

.setName("JournalDev") .build();

Person anupam = new Person.Builder()

.setName("Anupam")

.setIcon(IconCompat.createWithResource(this, R.drawable.samindexple\_photo))

.setImportant(true).build();

Person bot = new Person.Builder()

.setName("Bot").setBot(true) .build();

Uri uri = Uri.parse("android.resource://com.journaldev.androidpnotifications/drawable/"+R.drawable.bg);

NotificationCompat.MessagingStyle.Message message=new NotificationCompat.MessagingStyle.Message("", new Date().getTime(), bot); message.setData("image/\*",uri);

new NotificationCompat.MessagingStyle(bot)

.addMessage("Hi. How are you?", new Date().getTime(), anupam)

.addMessage(message)

.addMessage("Does this image look good?", new Date().getTime(), bot)

.addMessage("Looks good!", new Date().getTime(), jd)

.setGroupConversation(true)

.setConversationTitle("Sample Conversation")

.setBuilder(builder); notificationManager.notify(4, builder.build());

}

private void notificationSemantic()

{

Person jd = new Person.Builder()

.setName("JournalDev")

.build();

Person anupam = new Person.Builder()

.setName("Anupam")

.setIcon(IconCompat.createWithResource(this, R.drawable.index))

.setImportant(true)

.build();

Person bot = new Person.Builder()

.setName("Bot")

.setBot(true)

.build();

Uri uri = Uri.parse("android.resource://com.journaldev.androidpnotifications/drawable/"+R.drawable.bg); Intent intent = new Intent(this, MainActivity.class);

intent.putExtra("hi","Notifications were read");

PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, 0);

NotificationCompat.MessagingStyle.Message message=new NotificationCompat.MessagingStyle.Message("", new Date().getTime(), bot); message.setData("image/\*",uri);

NotificationCompat.Action replyAction = new NotificationCompat.Action.Builder(

R.drawable.bg, "MARK READ", pendingIntent)

.setSemanticAction(NotificationCompat.Action.SEMANTIC\_ACTION\_MARK\_AS\_READ)

.build();

NotificationCompat.Builder separateBuilder = builder; separateBuilder.addAction(replyAction);

new NotificationCompat.MessagingStyle(bot)

.addMessage("Hi. How are you?", new Date().getTime(), anupam)

.addMessage(message)

.addMessage("Does this image look good?", new Date().getTime(), bot)

.addMessage("Looks good!", new Date().getTime(), jd)

.setGroupConversation(true)

.setConversationTitle("Sample Conversation")

.setBuilder(separateBuilder);

notificationManager.notify(5, separateBuilder.build());

}

@Override

protected void onResume() { super.onResume();

if(getIntent()!=null && getIntent().getExtras()!=null)

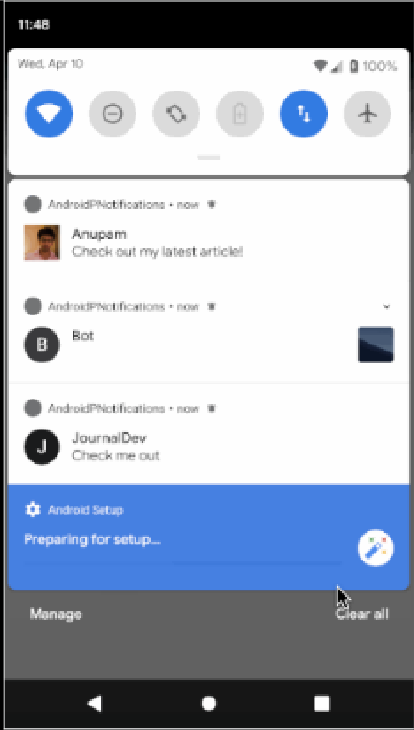
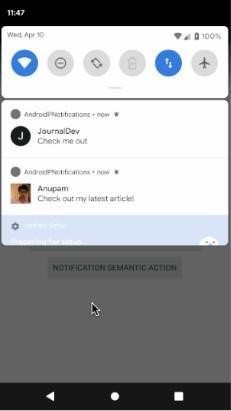
{

String value = getIntent().getStringExtra("hi"); Toast.makeText(getApplicationContext(),value,Toast.LENGTH\_LONG).show();

}

}

}

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result:**

**EX.NO.6**

**DATE:**

**DEVELOP A MOVIE RATING APPLICATION (SIMILAR TO IMDB)**

**Aim:**

To develop an Android Application that implements Movie rating .

**Procedure:**

**Creating a New project:**

Open Android Studio and then click on **File -> New -> New project**. Then type the Application name as **“exno6″**and click Next.

Then **select the Minimum SDK** as shown below and click Next. Then **select the Empty Activity** and click Next.

Finally click **Finish**.

It will take some time to build and load the project. After completion it will look as given below.

**Designing layout for the Android Application:**

Click on **app -> res -> layout -> activity\_main.xml.**

Now click on Text as shown below.

Then delete the code which is there and type the code as given below.

**Code for 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">

<EditText

android:id="@+id/editTextMovieName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Movie Name" />

<RatingBar

android:id="@+id/ratingBar"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:numStars="5"

android:rating="0" />

<Button

android:id="@+id/buttonRate"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Rate Movie" />

<LinearLayout

android:id="@+id/linearLayoutMovies"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

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

</LinearLayout>

Now click on Design and your application will look as given below. So now the designing part is completed.



**Java Coding for the Android Application:**

Click on **app -> java -> com.example.exno6 -> MainActivity**.

Then delete the code which is there and type the code as given below.

**Code for MainActivity.java:**

**package com.example.movie**

import android.os.Bundle

import android.view.View

import android.widget.Button

import android.widget.EditText

import android.widget.LinearLayout

import android.widget.RatingBar

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

data class Movie(val name: String, val ratings: MutableList<Float> = mutableListOf()) {

fun averageRating(): Float {

return if (ratings.isNotEmpty()) {

ratings.average().toFloat()

} else {

0f

}

}

}

class MainActivity : AppCompatActivity() {

private lateinit var editTextMovieName: EditText

private lateinit var ratingBar: RatingBar

private lateinit var buttonRate: Button

private lateinit var linearLayoutMovies: LinearLayout

private val movies = mutableListOf<Movie>()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

editTextMovieName = findViewById(R.id.editTextMovieName)

ratingBar = findViewById(R.id.ratingBar)

buttonRate = findViewById(R.id.buttonRate)

linearLayoutMovies = findViewById(R.id.linearLayoutMovies)

buttonRate.setOnClickListener {

rateMovie()

}

}

private fun rateMovie() {

val movieName = editTextMovieName.text.toString()

val rating = ratingBar.rating

if (movieName.isNotEmpty()) {

val existingMovie = movies.find { it.name == movieName }

if (existingMovie != null) {

// Add rating to existing movie

existingMovie.ratings.add(rating)

} else {

// Add new movie with rating

val movie = Movie(movieName)

movie.ratings.add(rating)

movies.add(movie)

}

displayMovies()

editTextMovieName.text.clear()

ratingBar.rating = 0f

}

}

private fun displayMovies() {

linearLayoutMovies.removeAllViews()

movies.forEach { movie ->

val textViewMovie = TextView(this)

textViewMovie.text = "Movie: ${movie.name}, Average Rating: ${movie.averageRating()}"

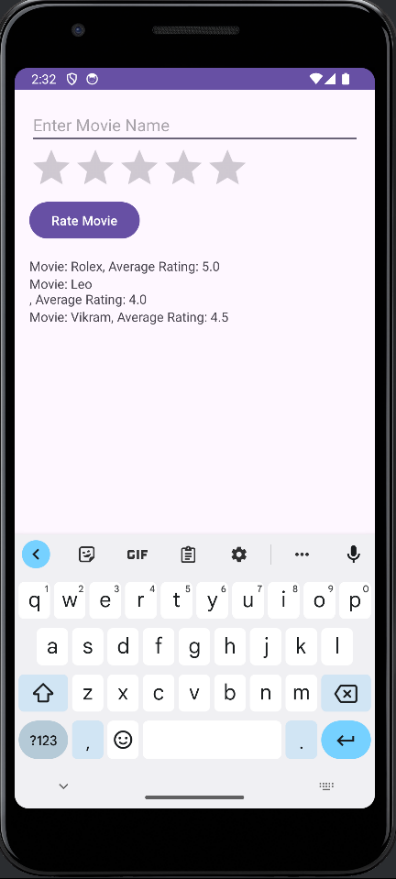
linearLayoutMovies.addView(textViewMovie)

}

}

}

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

\

**Result:**

**EX.NO:7**

**DATE:**

**DEVELOP A NATIVE APPLICATION THAT USES GPS LOCATION INFORMATION**

**Aim:**

To develop an Android Application that uses GPS location information.

**Procedure:**

**Creating a New project:**

Open Android Studio and then click on **File -> New -> New project**. Then type the Application name as **“exno7″**and click Next.

Then **select the Minimum SDK** as shown below and click Next. Then **select the Empty Activity** and click Next.

Finally click **Finish**.

It will take some time to build and load the project. After completion it will look as given below.

**Designing layout for the Android Application:**

Click on **app -> res -> layout -> activity\_main.xml**. Now click on Text as shown below.

Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

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

<LinearLayout xmlns:android = ["h](http://schemas.android.com/apk/res/android)t[tp://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:layout\_width = "fill\_parent"

android:layout\_height = "fill\_parent" android:orientation = "vertical" >

<Button

android:id = "@+id/button" android:layout\_width = "fill\_parent" android:layout\_height = "wrap\_content" android:text = "getlocation"/>

</LinearLayout>

Now click on Design and your application will look as given below. So now the designing part is completed.

**Following will be the content of res/values/strings.xml to define two new constants −**

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

<resources>

<string name = "app\_name">Tutorialspoint</string>

</resources>

**Adding permissions in Manifest for the Android Application:**

Click on **app -> manifests -> AndroidManifest.xml.**

**Code for AndroidManifest.xml:**

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

<manifest xmlns:android = ["h](http://schemas.android.com/apk/res/android)t[tp://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"package = "com.example.tutorialspoint7.myapplication">

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

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

<application android:allowBackup = "true"

android:icon = "@mipmap/ic\_launcher" android:label = "@string/app\_name" android:supportsRtl = "true" android:theme = "@style/AppTheme">

<activity android:name = ".MainActivity">

<intent-filter>

<action android:name = "android.intent.action.MAIN" />

<category android:name = "android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**Java Coding for the Android Application:**

Click on **app -> java -> com.example.exno7 -> MainActivity**.

Then delete the code which is there and type the code as given below.

**Code for MainActivity.kt:**

packagecom.example.exno7;

import android.Manifest; import android.app.Activity; import android.os.Bundle;

import android.support.v4.app.ActivityCompat; import android.test.mock.MockPackageManager; import android.view.View;

import android.widget.Button; import android.widget.Toast;

public class MainActivity extends Activity { Button btnShowLocation;

private static final int REQUEST\_CODE\_PERMISSION = 2;

String mPermission = Manifest.permission.ACCESS\_FINE\_LOCATION;

// GPSTracker classGPSTracker gps;

@Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

try {

if (ActivityCompat.checkSelfPermission(this, mPermission)

!= MockPackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(this, new String[]{mPermission}, REQUEST\_CODE\_PERMISSION);

// If any permission above not allowed by user, this condition will execute every time, else your else part will work

}

} catch (Exception e)

{

e.printStackTrace();

}

btnShowLocation = (Button) findViewById(R.id.button);

// show location button click event btnShowLocation.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View arg0) {

// create class object

gps = new GPSTracker(MainActivity.this);

// check if GPS enabled if(gps.canGetLocation()){

double latitude = gps.getLatitude(); double longitude = gps.getLongitude();

// \n is for new line

Toast.makeText(getApplicationContext(), "Your Location is - \nLat: "

+ latitude + "\nLong: " + longitude, Toast.LENGTH\_LONG).show();

}else{

// can't get location

// GPS or Network is not enabled

// Ask user to enable GPS/network in settings gps.showSettingsAlert();

}

}

});

}

}

Following is the content of the modified main activity file **GPSTracker.java**.

**Code for GPDTracker.kt**

package com.example.maps

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import com.google.android.gms.maps.CameraUpdateFactory

import com.google.android.gms.maps.GoogleMap

import com.google.android.gms.maps.MapView

import com.google.android.gms.maps.OnMapReadyCallback

import com.google.android.gms.maps.model.LatLng

import com.google.android.gms.maps.model.MarkerOptions

class MainActivity : AppCompatActivity(), OnMapReadyCallback {

private lateinit var mapView: MapView

private lateinit var map: GoogleMap

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

mapView = findViewById(R.id.mapView)

mapView.onCreate(savedInstanceState)

mapView.getMapAsync(this)

}

override fun onResume() {

super.onResume()

mapView.onResume()

}

override fun onPause() {

super.onPause()

mapView.onPause()

}

override fun onDestroy() {

super.onDestroy()

mapView.onDestroy()

}

override fun onLowMemory() {

super.onLowMemory()

mapView.onLowMemory()

}

override fun onMapReady(googleMap: GoogleMap) {

map = googleMap

val location = LatLng(10.956889, 77.952307) // San Francisco

map.addMarker(MarkerOptions().position(location).title("Marker in San Francisco"))

map.moveCamera(CameraUpdateFactory.newLatLngZoom(location, 12f))

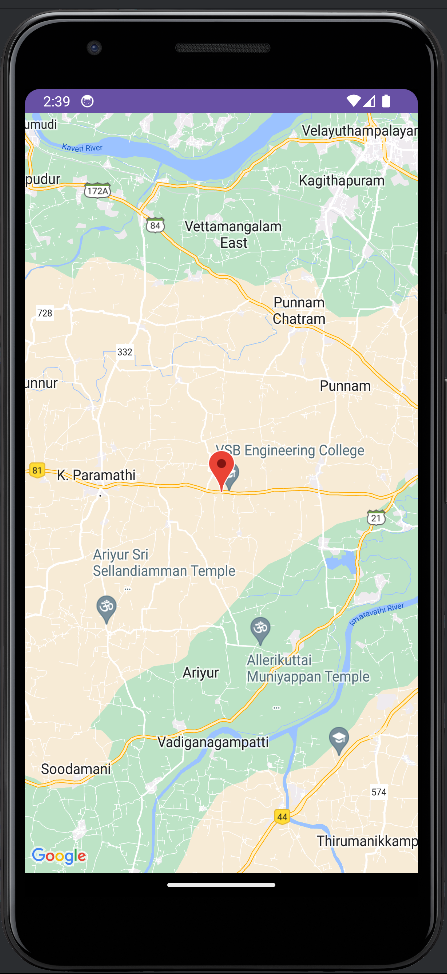
}

}

So now the Coding part is also completed.

* Now run the application to see the output.

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result:**

**EX.NO:8**

**DATE:**

**DEVELOP AN ANDROID APPLICATION THAT READS THE WEB SERVER MESSAGE**

**Aim:**

To develop an Android Application that writes the sever notification.

**Procedure:**

**Creating a New project:**

* Open Android Studio and then click on **File -> New -> New project**.
* Then type the Application name as **“exno8**″ and click Next.
* Then **select the Minimum SDK** as shown below and click Next.
* Then **select the Empty Activity** and click Next.
* Finally click **Finish**.
* It will take some time to build and load the project.
* After completion it will look as given below.

**Designing layout for the Android Application:**

* Click on **app -> res -> layout -> activity\_main.xml**.
* Now click on Text as shown below.
* Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

* Now click on Design and your application will look as given below.
* So now the designing part is completed.

**Adding permissions in Manifest for the Android Application:**

* Click on **app -> manifests -> AndroidManifest.xml.**
* Now include the WRITE\_EXTERNAL\_STORAGE permissions in the AndroidManifest.xml file as shownbelow.

**Code for AndroidManifest.xml:**

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

<manifest xmlns:android=["h](http://schemas.android.com/apk/res/android)t[tp://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" package="com.example.exno8" >

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<application

android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name"

android:supportsRtl="true" android:theme="@style/AppTheme" >

<activity android:name=".MainActivity" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

* So now the Permissions are added in the Manifest.

**Java Coding for the Android Application:**

* Click on **app -> java -> com.example.exno8 -> MainActivity**.
* Then delete the code which is there and type the code as given below.

**Code for MainActivity.kt:**

package com.example.myapplication

import android.os.Bundle

import android.util.Log

import android.widget.Button

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

import retrofit2.Call

import retrofit2.Callback

import retrofit2.Response

class MainActivity : AppCompatActivity() {

private lateinit var textView: TextView

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

// Find the TextView using findViewById()

textView = findViewById(R.id.textView)

// Find the Refresh Button using findViewById()

val refreshButton: Button = findViewById(R.id.refreshButton)

// Set OnClickListener to the Refresh Button

refreshButton.setOnClickListener {

fetchDataFromServer()

}

// Fetch data from the server initially

fetchDataFromServer()

}

private fun fetchDataFromServer() {

// Making a network request using Retrofit

val call: Call<DataResponse> = ApiClient.apiService.getData()

call.enqueue(object : Callback<DataResponse> {

override fun onResponse(call: Call<DataResponse>, response: Response<DataResponse>) {

if (response.isSuccessful) {

// Handle successful response

val dataResponse: DataResponse? = response.body()

// Update UI with the dataResponse

dataResponse?.let {

textView.text = "Response: ${it.toString()}"

}

} else {

// Log error message

val errorBody = response.errorBody()?.string()

Log.e("API\_ERROR", "Failed to get data: $errorBody")

// Handle unsuccessful response

textView.text = "Failed to get data"

}

}

override fun onFailure(call: Call<DataResponse>, t: Throwable) {

// Log failure message

Log.e("API\_ERROR", "Error: ${t.message}", t)

// Handle failure

textView.text = "Error: ${t.message}"

}

})

}

}

Server.js :  
const express = require('express');

const app = express();

const port = 3000;

app.get('/', (req, res) => {

// Define your sample data

const data = {

message: "Hello,sankar ala!!!"

};

// Set the response header to indicate JSON content

res.setHeader('Content-Type', 'application/json');

// Send the JSON response

res.send(JSON.stringify(data));

});

// Start the server

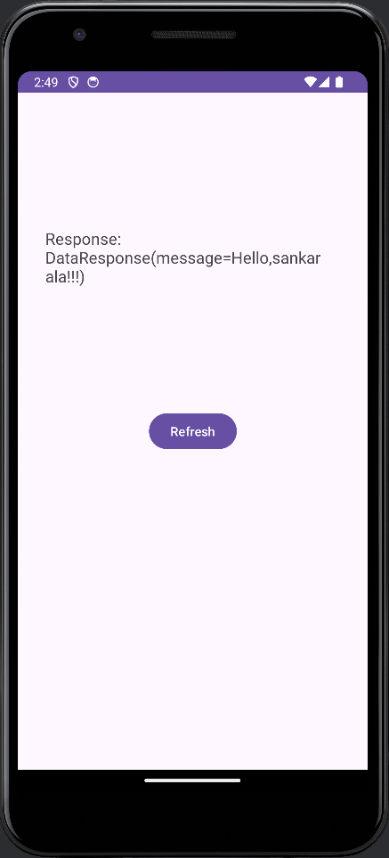
app.listen(port, () => {

console.log(`Server running at http://localhost:${port}`);

});

* So now the Coding part is also completed.
* Now run the application to see the output.

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result :**

**EX.NO.9**

**DATE:**

**IMPLEMENT AN APPLICATION FOR A SIMPLE ECOMMERCE SITE**

**Aim:**

To develop an Android Application that creates a simple shopping app

**Procedure:**

**Creating a New project:**

* Open Android Studio and then click on **File -> New -> New project**.
* Then type the Application name as **“ex.no9″**and click Next.
* Then **select the Minimum SDK** as shown below and click Next.
* Then **select the Empty Activity** and click Next.
* Finally click **Finish.**
* It will take some time to build and load the project.
* After completion it will look as given below.

**Creating Second Activity for the Android Application:**

* Click on **File -> New -> Activity -> Empty Activity**.
* Type the Activity Name as SecondActivity and click Finish button.
* Thus Second Activity For the application is created.

**Designing layout for the Android Application:**

* Click on **app -> res -> layout -> activity\_main.xml**.
* Now click on Text as shown below.
* Then delete the code which is there and type the code as given below.

**Code for 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">**

**<!-- ListView for displaying products -->**

**<ListView**

**android:id="@+id/productsListView"**

**android:layout\_width="match\_parent"**

**android:layout\_height="0dp"**

**android:layout\_weight="1"**

**android:choiceMode="singleChoice" />**

**<!-- Button to add selected item to the cart -->**

**<Button**

**android:id="@+id/addToCartButton"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="Add to Cart" />**

**<!-- ListView for displaying cart items -->**

**<ListView**

**android:id="@+id/cartListView"**

**android:layout\_width="match\_parent"**

**android:layout\_height="0dp"**

**android:layout\_weight="1"**

**android:choiceMode="singleChoice" />**

**<!-- Button to delete selected item from the cart -->**

**<Button**

**android:id="@+id/deleteFromCartButton"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="Delete from Cart" />**

**<!-- Button to purchase items in the cart -->**

**<Button**

**android:id="@+id/purchaseButton"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="Purchase" />**

**</LinearLayout>**

* Now click on Design and your application will look as given below.
* So now the designing part is completed.

**Java Coding for the Android Application:**

* Click on **app -> java -> com.example.exno9 -> MainActivity**.
* Then delete the code which is there and type the code as given below.

**Code for MainActivity.kt:**

import android.os.Bundle

import android.widget.ArrayAdapter

import android.widget.Button

import android.widget.ListView

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

data class Product(val name: String, val price: Double)

class Cart {

val items = mutableListOf<Product>()

fun add(product: Product) {

items.add(product)

}

fun total(): Double {

return items.sumByDouble { it.price }

}

fun remove(product: Product) {

items.remove(product)

}

}

class MainActivity : AppCompatActivity() {

private lateinit var productsListView: ListView

private lateinit var addToCartButton: Button

private lateinit var purchaseButton: Button

private lateinit var deleteFromCartButton: Button

private lateinit var cartListView: ListView

private val productsList = listOf(

Product("Product 1", 10.0),

Product("Product 2", 20.0),

Product("Product 3", 30.0)

)

private lateinit var productsAdapter: ArrayAdapter<Product>

private lateinit var cartAdapter: ArrayAdapter<Product>

private val cart = Cart()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

productsListView = findViewById(R.id.productsListView)

addToCartButton = findViewById(R.id.addToCartButton)

purchaseButton = findViewById(R.id.purchaseButton)

deleteFromCartButton = findViewById(R.id.deleteFromCartButton)

cartListView = findViewById(R.id.cartListView)

productsAdapter = ArrayAdapter(this, android.R.layout.simple\_list\_item\_single\_choice, productsList)

productsListView.adapter = productsAdapter

addToCartButton.setOnClickListener {

val selectedItemPosition = productsListView.checkedItemPosition

if (selectedItemPosition != ListView.INVALID\_POSITION) {

val selectedProduct = productsList[selectedItemPosition]

cart.add(selectedProduct)

updateCartList()

Toast.makeText(this, "Added ${selectedProduct.name} to cart", Toast.LENGTH\_SHORT).show()

} else {

Toast.makeText(this, "Please select a product to add to cart", Toast.LENGTH\_SHORT).show()

}

}

deleteFromCartButton.setOnClickListener {

val selectedItemPosition = cartListView.checkedItemPosition

if (selectedItemPosition != ListView.INVALID\_POSITION) {

val selectedProduct = cartAdapter.getItem(selectedItemPosition)

if (selectedProduct != null) {

cart.remove(selectedProduct)

updateCartList()

Toast.makeText(this, "Removed ${selectedProduct.name} from cart", Toast.LENGTH\_SHORT).show()

}

} else {

Toast.makeText(this, "Please select a product to remove from cart", Toast.LENGTH\_SHORT).show()

}

}

purchaseButton.setOnClickListener {

val total = cart.total()

Toast.makeText(this, "Total Purchase Amount: $total", Toast.LENGTH\_SHORT).show()

}

}

private fun updateCartList() {

cartAdapter = ArrayAdapter(this, android.R.layout.simple\_list\_item\_single\_choice, cart.items)

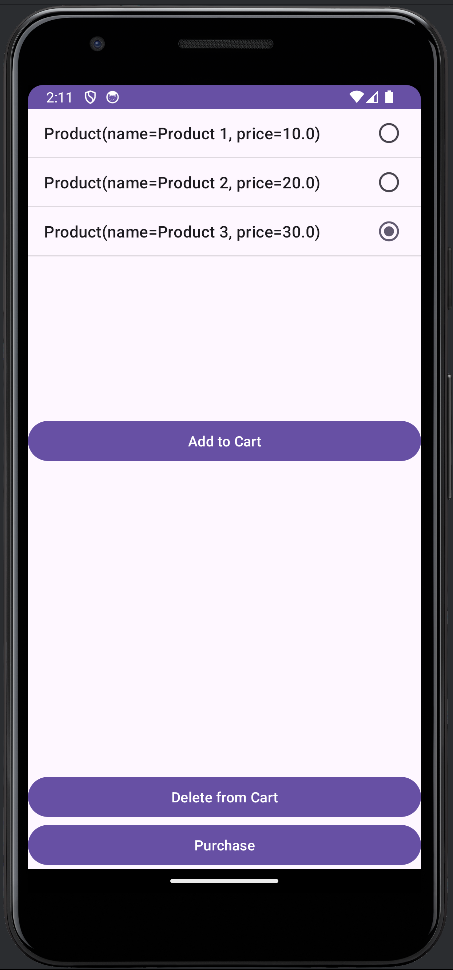
cartListView.adapter = cartAdapter

}

}

* So now the coding part is also completed.
* Now run the application to see the output.

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result:**

**EX.NO.10**

**DATE:**

**MINI PROJECTS INVOLVING FLUTTER/KOTLIN APPLICATIONS (TIC-TAC-TOE)**

**Aim:**

To develop an Android Application by using Kotlin a mini-project\_ Tic-tac-toe

**Procedure:**

**Creating a New project:**

* Open Android Studio and then click on **File -> New -> New project**.
* Then type the Application name as **“exno10″**and click Next.
* Then **select the Minimum SDK** as shown below and click Next.
* Then select the **Empty Activity** and click Next.
* Finally click **Finish**.
* It will take some time to build and load the project.
* After completion it will look as given below.

**Designing layout for the Android Application:**

* Click on **app -> res -> layout -> activity\_main.xml.**
* Now click on Text as shown below.
* Then delete the code which is there and type the code as given below.

**Code for Activity\_main.xml:**

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

<LinearLayout xmlns:android=["h](http://schemas.android.com/apk/res/android)t[tp://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"android:layout\_width="fill\_parent" android:layout\_height="fill\_parent"

android:orientation="vertical" >

<ListView

android:id="@+id/listView" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" />

</LinearLayout>

* Now click on Design and your application will look as given below.
* So now the designing part is completed.

**Adding permissions in Manifest for the Android Application:**

* Click on **app -> manifests -> AndroidManifest.xml.**
* Now include the INTERNET permissions in the AndroidManifest.xml file as shown below.

**Code for AndroidManifest.xml:**

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

<manifest xmlns:android=["h](http://schemas.android.com/apk/res/android)t[tp://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" package="com.example.exno10" >

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

<application

android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name" android:supportsRtl="true" android:theme="@style/AppTheme" >

<activity android:name=".MainActivity" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

* So now the Permissions are added in the Manifest.

**Java Coding for the Android Application:**

* Click on app -> java -> com.example.exno10 -> MainActivity.
* Then delete the code which is there and type the code as given below.

**Code for MainActivity.kt :**package com.example.tic\_tac

import android.graphics.Color

import android.os.Bundle

import android.view.View

import android.widget.Button

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity(), View.OnClickListener {

private var playerOneActive = true

private lateinit var playerOneScore: TextView

private lateinit var playerTwoScore: TextView

private lateinit var playerStatus: TextView

private lateinit var buttons: Array<Button>

private lateinit var reset: Button

private lateinit var playAgain: Button

private var gameState = intArrayOf(2, 2, 2, 2, 2, 2, 2, 2, 2)

private val winningPositions = arrayOf(

intArrayOf(0, 1, 2), intArrayOf(3, 4, 5), intArrayOf(6, 7, 8), intArrayOf(0, 3, 6),

intArrayOf(1, 4, 7), intArrayOf(2, 5, 8), intArrayOf(0, 4, 8), intArrayOf(2, 4, 6)

)

private var rounds = 0

private var playerOneScoreCount = 0

private var playerTwoScoreCount = 0

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

playerOneScore = findViewById(R.id.score\_Player1)

playerTwoScore = findViewById(R.id.score\_Player2)

playerStatus = findViewById(R.id.textStatus)

reset = findViewById(R.id.btn\_reset)

playAgain = findViewById(R.id.btn\_play\_again)

buttons = Array(9) { index ->

findViewById<Button>(resources.getIdentifier("btn$index", "id", packageName))

}

buttons.forEach { it.setOnClickListener(this) }

playerOneScoreCount = 0

playerTwoScoreCount = 0

playerOneActive = true

rounds = 0

reset.setOnClickListener {

playAgain()

playerOneScoreCount = 0

playerTwoScoreCount = 0

updatePlayerScore()

}

playAgain.setOnClickListener { playAgain() }

}

override fun onClick(view: View) {

val button = view as Button

if (!button.text.toString().isEmpty() || checkWinner()) {

return

}

val buttonID = resources.getResourceEntryName(button.id)

val gameStatePointer = buttonID.substring(buttonID.length - 1).toInt()

if (playerOneActive) {

button.text = "X"

button.setTextColor(Color.parseColor("#ffc34a"))

gameState[gameStatePointer] = 0

} else {

button.text = "O"

button.setTextColor(Color.parseColor("#70fc3a"))

gameState[gameStatePointer] = 1

}

rounds++

if (checkWinner()) {

if (playerOneActive) {

playerOneScoreCount++

updatePlayerScore()

playerStatus.text = "Player-1 has won"

} else {

playerTwoScoreCount++

updatePlayerScore()

playerStatus.text = "Player-2 has won"

}

} else if (rounds == 9) {

playerStatus.text = "No Winner"

} else {

playerOneActive = !playerOneActive

}

}

private fun checkWinner(): Boolean {

for (winningPosition in winningPositions) {

if (gameState[winningPosition[0]] == gameState[winningPosition[1]] &&

gameState[winningPosition[1]] == gameState[winningPosition[2]] &&

gameState[winningPosition[0]] != 2

) {

return true

}

}

return false

}

private fun playAgain() {

rounds = 0

playerOneActive = true

for (i in buttons.indices) {

gameState[i] = 2

buttons[i].text = ""

}

playerStatus.text = "Status"

}

private fun updatePlayerScore() {

playerOneScore.text = playerOneScoreCount.toString()

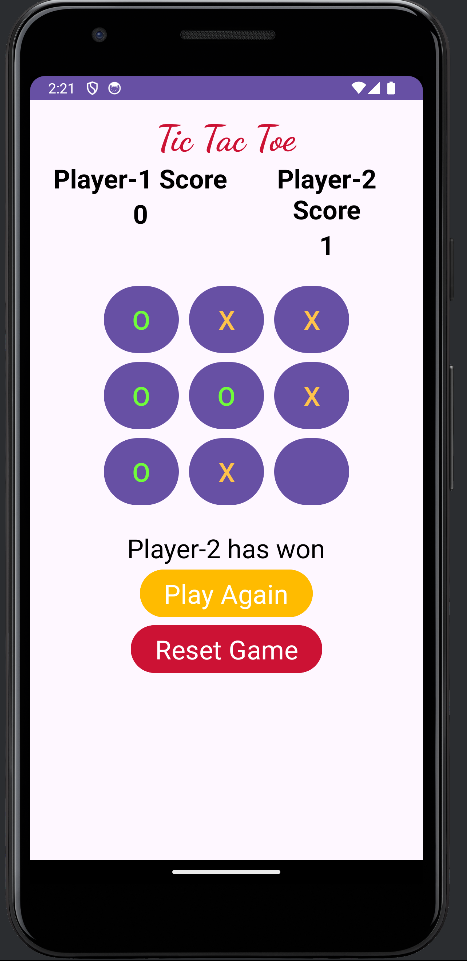
playerTwoScore.text = playerTwoScoreCount.toString()

}

}

* So now the Coding part is also completed.
* Now run the application to see the output.

**Output:**



|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Particulars** | **Marks Allotted** | **Marks Obtained** |
| **1** | **Performance** | **50** |  |
| **2** | **Viva** | **10** |  |
| **3** | **Record** | **15** |  |
| **4** | **Total** | **75** |  |

**Result:**

**Ex. No. 12**

**Date:**

**Develop a Mobile application for simple needs (Mini Project)**