

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 014



A Mobile Application Development Mini Project Report On

“MULTIPURPOSE CALCULATOR”

Submitted in Partial fulfillment of the Requirements for the VI Semester of the Degree of

**Bachelor of Engineering
In
Computer Science & Engineering**

By

**SHRUTHA V BHAT (4MW19CS094)
VIKRAM GIRISH THUNGA (4MW19CS116)**

Under the Guidance of

**Ms. SAVITHA A SHENOY
Asst. Prof, Dept. of CSE**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**Shri Madhwa Vadiraja Institute of Technology & Management
Vishwothama Nagar, Bantakal-574115**

July, 2022

SHRI MADHWA VADIRAJA
INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(A Unit of Shri Sode Vadiraja Mutt Education Trust ®, Udupi)
Vishwothama Nagar, BANTAKAL – 574 115, Udupi District, Karnataka, INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

Certified that the Mobile Application Development Mini Project work entitled “**MULTIPURPOSE CALCULATOR**” has been carried out by **SHRUTHA V BHAT (4MW19CS094)** and **VIKRAM GIRISH THUNGA (4MW19CS094)** who are the bonafide students of Shri Madhwa Vadiraja Institute of Technology and Management, in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2021-2022**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Graphics Project Report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

MS. SAVITHA A SHENOY

Project Guide

Dept. of CSE

DR. NAGARAJ BHAT

Professor and Head

Dept. of CSE

External Viva

Name of the examiners

Signature with date

1

2.

ACKNOWLEDGEMENT

It is our duty to acknowledge the help rendered to us by various persons in completing this Mobile Application Development Mini Project titled “**MULTIPURPOSE CALCULATOR**”

We would like to express our heartfelt thanks to **Dr. Thirumaleshwara Bhat, Principal, SMVITM, Bantakal**, for extending his support.

We would also like to express our deepest gratitude to **Dr. Nagaraj Bhat, HOD, Computer Science and Engineering** whose guidance and support was truly invaluable in completing this project on time.

It is our privilege to express our sincerest regards to our project coordinator, **Ms. Savitha A Shenoy**, for her valuable inputs, guidance, encouragement, and whole-hearted cooperation throughout the duration of our project.

We take this opportunity to thank all our teaching and non-teaching staff of Department of CSE who has directly or indirectly helped the completion of our project.

We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout our student life.

Thanking you all,

SHRUTHA V BHAT

VIKRAM GIRISH THUNGA

ABSTRACT

The Android mobile platform has developed from its first phone in October 2008 to being the most popular smart phone operating system in the world by 2012. The explosive growth of the platform has been a significant win for consumers with respect to competition and features.

The market has been booming in the past few years that, there are now over 1,195,932 applications on the Android market. Due to the wide usage, it is necessary to provide users with security applications to manage the data in their personal smart phones.

Computers have become a powerful medium for the rapid and economical production of pictures. Mobile app development is a relatively new phenomenon that is increasing rapidly due to the ubiquity and popularity of smartphones among end-users. The goal of our study is to gain an understanding of the main challenges developers face in practice when they build apps for different mobile devices.

Our project Multipurpose Calculator provides a working, functional app which contains calculator to many basic things we need in our day-to-day life, from a Standard Math Calculator to Financial Calculators -which will include EMI calculator, Simple and Compound Interest calculator, Currency convertor, To Health and Fitness Calculators and many other. The sole purpose of this application is to provide fast, comprehensive, convenient, free calculators in a plethora of areas, such that it can be used by people in their daily life in order to ease the work of students, shopkeepers and others who need a single application for all their computational needs.

TABLE OF CONTENTS

Contents

ACKNOWLEDGEMENT	III
ABSTRACT	IV
TABLE OF CONTENTS	V
LIST OF FIGURES	VI
CHAPTER 1	1
INTRODUCTION	1
1.1 ABOUT ANDROID STUDIO	1
1.2 ABOUT MULTIPURPOSE CALCULATOR APPLICATION	1
1.3 OBJECTIVES	2
CHAPTER 2	3
REQUIREMENT ANALYSIS	3
2.1 SOFTWARE REQUIREMENTS	3
2.2 HARDWARE REQUIREMENTS	3
CHAPTER 3	4
DESIGN	4
3.1 FLOW CHART	4
3.2 ANDROID UI CONTROLS:	8
3.3 XML-BASED LAYOUTS IN ANDROID:	8
<i>Relative Layout:</i>	9
<i>Absolute Layout:</i>	9
<i>Frame Layout:</i>	10
<i>Linear Layout:</i>	10
<i>Grid Layout:</i>	10
<i>Constraint layout:</i>	10
CHAPTER 4	11
IMPLEMENTATION	11
4.1 APP COMPONENTS	11
<i>Activity:</i>	11
<i>Services:</i>	11
<i>Broadcast Receivers:</i>	12
<i>Content Providers:</i>	12
4.2 FUNCTIONS	12
<i>protected void onCreate(Bundle savedInstanceState):</i>	12
<i>super.onCreate(savedInstanceState):</i>	12
<i>setContentView setContentView(R.layout.main):</i>	13
<i>protected void onCreate(Bundle savedInstanceState):</i>	13
<i>public void movepageh(View v):</i>	13

Multipurpose Calculator

<i>public void movepagef(View v):</i>	13
<i>public void movepagem(View v):</i>	13
<i>intent(context, class):</i>	13
<i>startActivity(intentObject):</i>	13
<i>findViewById:</i>	14
<i>public void bmicalc (View v):</i>	14
<i>public void bfatcalc (View v):</i>	14
<i>public void emcalc (View v):</i>	14
<i>public void sintcalc (View v):</i>	14
<i>public void cintcalc (View v):</i>	14
<i>public void simpcalc (View v):</i>	14
<i>public void pcalc (View v):</i>	14
<i>public void qcalc (View v):</i>	15
<i>Float.parseFloat(String):</i>	15
<i>id.setText(String):</i>	15
CHAPTER 5	16
SNAPSHOTS	16
CHAPTER 6	22
CONCLUSION AND FUTURE SCOPE	22
FUTURE SCOPE:	22
BIBLIOGRAPHY/REFERENCE	23
USER GUIDE	24
<i>Run on a real device:</i>	24
<i>Run the app on the Emulator as follows:</i>	24

LIST OF FIGURES

Figure No.	Label
3.1.0	Flowchart representing the process of selecting categories.
3.1.1	Flowchart representing the process of selecting type of Financial Calculator.
3.1.2	Flowchart representing the process of selecting type of Health Calculator.
3.1.3	Flowchart representing the process of selecting type of Math Calculator.
5.0	Dashboard
5.1	Activity to select the type of Health Calculator
5.2	BMI Calculator Activity
5.3	BMI Calculator Activity with result for a given input.
5.4	Body Fat Percentage Calculator Activity
5.5	Body Fat Percentage Calculator Activity with result for a given input.
5.6	Activity to select the type of Financial Calculator
5.7	EMI Calculator Activity
5.8	EMI Calculator Activity with result for a given input.
5.9	Simple Interest Calculator Activity
5.10	Simple Interest Activity with result for a given input.
5.11	Compound Interest Calculator Activity
5.12	Compound Interest Activity with result for a given input.
5.13	Activity to select the type of Math Calculator
5.14	Simple Calculator Activity
5.15	Simple Activity with result for a given input.
5.16	Root and Power Calculator Activity
5.17	Root and Power Calculator Activity with result for a given input (Root Calculation).
5.18	Root and Power Calculator Activity with result for a given input (Power Calculation).

Multipurpose Calculator

5.19	Roots of a Quadratic Equation Calculator Activity
5.20	Roots of a Quadratic Equation Calculator Activity with result for a given input.

INTRODUCTION

1.1 About Android Studio

The Android operating system is the largest installed base among various mobile platforms across the globe. Hundreds of millions of mobile devices are powered by Android in more than 190 countries of the world. It conquered around 75% of the global market share by the end of 2020, and this trend is growing bigger every other day. The company named Open Handset Alliance developed Android for the first time that is based on the modified version of the Linux kernel and other open-source software. Google sponsored the project at initial stages and in the year 2005, it acquired the whole company. In September 2008, the first Android-powered device launched in the market.

Android dominates the mobile OS industry because of the long list of features it provides. It's user-friendly, has huge community support, provides a greater extent of customization, and a large number of companies build Android-compatible smartphones. As a result, the market observes a sharp increase in the demand for developing Android mobile applications, and with that companies need smart developers with the right skill set.

At first, the purpose of Android was thought of as a mobile operating system. However, with the advancement of code libraries and its popularity among developers of the divergent domain, Android is an absolute set of software for all devices like tablets, wearables, set-top boxes, smart TVs, notebooks, etc.

.

1.2 About Multipurpose Calculator Application

Creation of software intended to run on mobile devices and optimized to take advantage of those products' unique features and hardware. While the in-built calculator in most Android smartphones only have a simple calculator with an exception of few smartphones which also has scientific calculator, we decided why not provide a variety of calculators in a single place, easily accessed by anyone and everyone.

Multipurpose Calculator provides a good GUI for logging purposes. After the successful logging the list of calculators will be displayed. The user can select any calculator which he wishes to use. Once the user selects a calculator, the app takes the user to a different page, which is done using intent mechanism. Every calculator has its own XML and each and every operation has their own Java code. If the user wishes to come back to main page, he can either select back button from the system or use the 'home' symbol on the left top corner.

In this the XML helps to support the structure of the App and Java acts as a backend tool. Using Java, we can perform control actions like taking inputs, giving the required output with respect to the user request, and maintaining the overall process.

1.3 Objectives

Our Multipurpose Calculator provides a working, functional app which contains calculator to many basic things we need in our day-to-day life, from a Standard Math Calculator to Financial Calculators -which will include EMI calculator, Simple and Compound Interest calculator, Currency convertor, To Health and Fitness Calculators and many other. The sole purpose of this application is to provide fast, comprehensive, convenient, free calculators in a plethora of areas, such that it can be used by people in their daily life in order to ease the work of students, shopkeepers and others who need a single application for all their computational needs.

REQUIREMENT ANALYSIS

2.1 Software Requirements

- Programming Platform: OpenJDK 1.8.0_332, Android Studio SDK
- Android Studio: Latest Version
- Emulator (Version Oreo x26, Pixel 3a XL (Edited)_2 API 26)
- Programming Languages: Java and XML
- Operating System (OS): Windows-10

2.2 Hardware Requirements

- Microsoft Windows 7/8/10 (64 bit).
- Memory: Minimum 4GB RAM. 8GB RAM recommended.
- 8 GB disk space.
- Processor: Intel® Core™ i5-8250U @1.60GHz 1.80GHz
- Android Phone
- 1 GB for Android SDK.

DESIGN

The basic requirements include the hardware and software requirements, keyboard keys are suitably implemented to provide an interactive interface to the user by which the user can give values for some calculation purpose, which are then executed on the standard output device, the screen. User interaction is provided through keyboard. The basic concepts of project development such as menus, display, and much more functions, functions are implemented to design the source code. The project is very presentable and thus, easy to understand.

3.1 Flow Chart

Data flow design is as shown below- covering the flow of the data in the system. It describes the relation between user input and the system behavior.

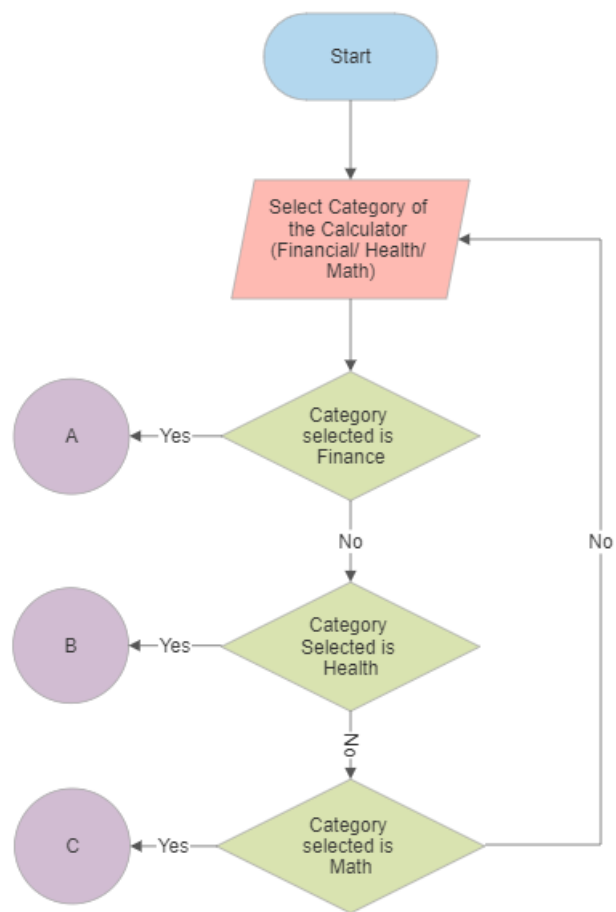


Figure 3.1.0

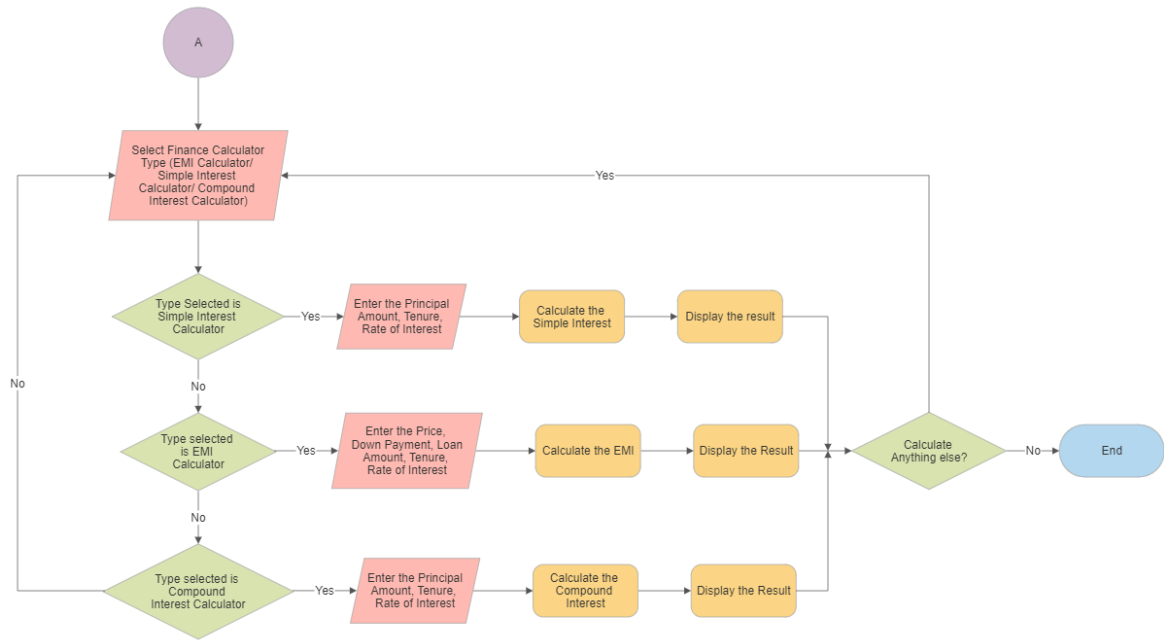


Figure 3.1.1

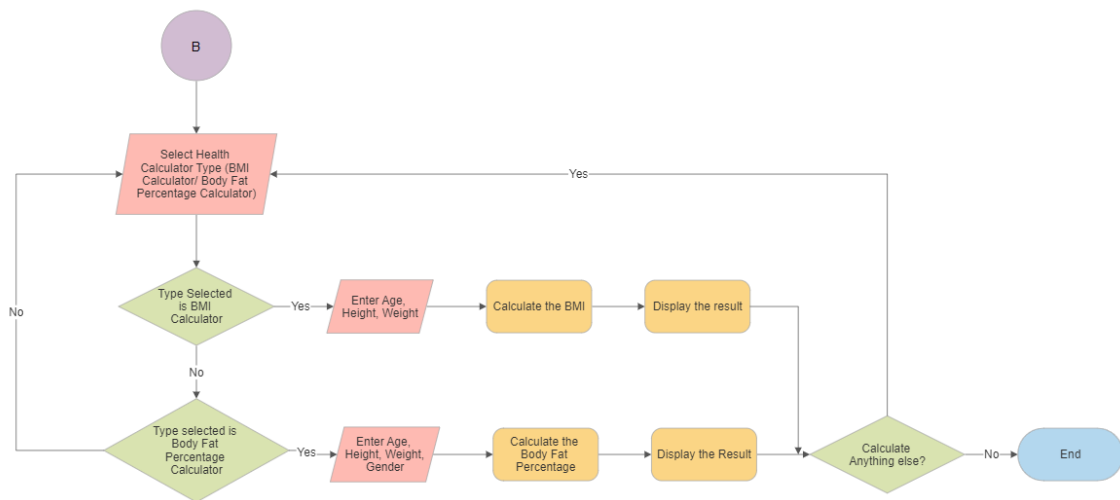


Figure 3.1.2

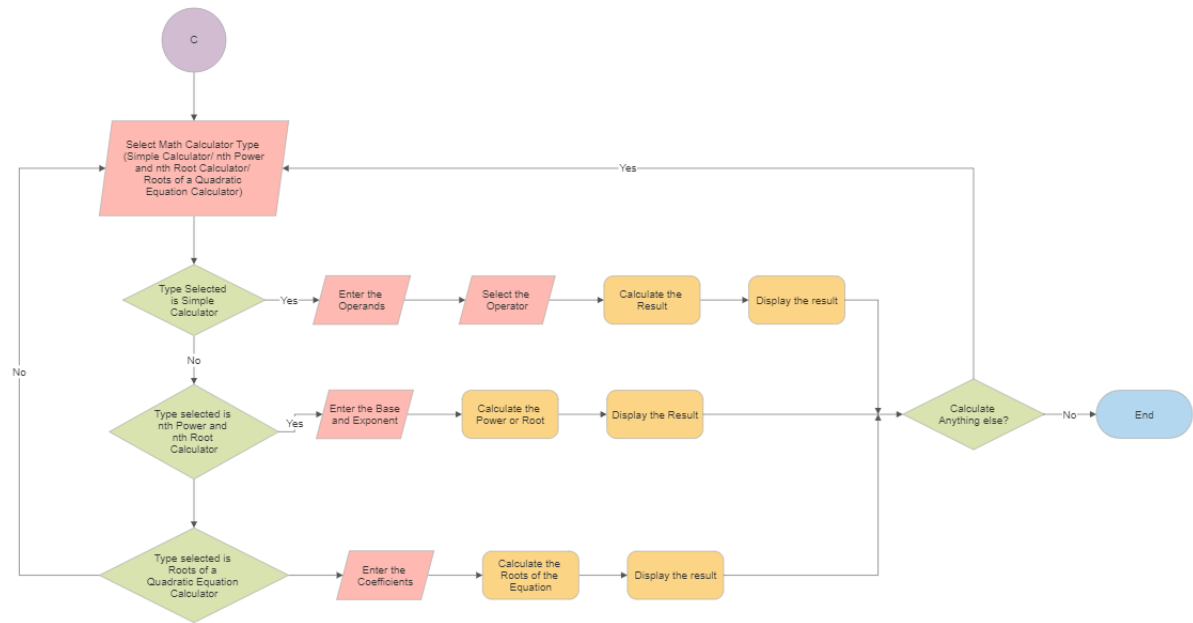


Figure 3.1.3

3.2 Android UI Controls:

Android provides solid support for the development of UI-based applications. Android provides a variety of widgets that the application programmer can use to create a desired layout and interface. These layout elements can be created via the programming language directly, or through XML layout files.

Sr. No.	UI Control & Description
1	<u>Text View</u> This control is used to display text to the user.
2	<u>Edit Text</u> Edit Text is a predefined subclass of Text View that includes rich editing capabilities.
3	<u>Auto Complete Text View</u> The Auto Complete Text View is a view that is similar to Edit Text, except that it shows a list of completion suggestions automatically while the user is typing.
4	<u>Button</u> A push-button that can be pressed, or clicked, by the user to perform an action.
5	<u>ImageButton</u> An ImageButton is an Absolute Layout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user.

3.3 XML-Based Layouts in Android:

❖ XML is a very popular and widely-used format. Hence, a lot of developers are quite comfortable with it.

- ❖ It helps to provide separation of the UI from the code logic. This provided flexibility to change one without affecting much the other.
- ❖ Generating XML output is easier than writing direct code, making it easier to have drag-and-drop

UI tools to generate interfaces for android apps, i.e.:

- Absolute Layout
- Frame Layout
- Linear Layout
- Relative Layout
- Table Layout
- Percent Relative Layout
- Grid Layout
- Coordinator Layout
- Constraint Layout
- Toolbar Layout

In our project, we make use of only one XML Layout, i.e, Relative Layout.

Relative Layout:

Using relative layout, we can specify the position of the elements in relation to other elements, or in relation to the parent container.

Other Layouts:

Absolute Layout:

In absolute layout, we can specify the exact coordinates of each control that we want to place. In absolute layout, we will give the exact X and Y coordinates of each control. The following is an example of an absolute layout.

Frame Layout:

Frame layout is used when you want to show one item on each screen. Using frame layout, we can have multiple items, but they will be overlapping and only, displaying themselves one at a time. Frame Layout is particularly useful when you want to create animation or movement on screen.

Linear Layout:

Linear layout is used to place one element on each line. So, all the elements will be place in an orderly top-to-bottom fashion. This is a very widely-used layout for creating forms on Android. We are now going to create a small app to display a basic form using the linear layout. The layout.xml file is as follows.

Grid Layout:

Android Grid View shows items in two-dimensional scrolling grid (rows & columns) and the grid items are not necessarily predetermined but they automatically inserted to the layout using a List Adapter.

Constraint layout:

A Constraint Layout is a View Group which allows you to position and size widgets in a flexible way.

IMPLEMENTATION

4.1 App Components

App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others. There are four different types of app components:

- Activities
- Services
- Broadcast receivers
- Content providers

Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed. The following sections describe the four types of app components.

Activity:

An activity is the entry point for interacting with the user. It represents a single screen with a user interface. An activity facilitates the following key interactions between system and app:

- Keeping track of what the user currently cares about (what is on screen) to ensure that the system keeps running the process that is hosting the activity.
- Knowing that previously used processes contain things the user may return to (stopped activities), and thus more highly prioritize keeping those processes around.
- Helping the app handle having its process killed so the user can return to activities with their previous state restored.
- Providing a way for apps to implement user flows between each other, and for the system to coordinate these flows.

Services:

A service is a general-purpose entry point for keeping an app running in the background for all kinds of reasons. It is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface. For example, a service might play music in the background while the user is in a

different app, or it might fetch data over the network without blocking user interaction with an activity. Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it.

Broadcast Receivers:

A broadcast receiver is a component that enables the system to deliver events to the app outside of a regular user flow, allowing the app to respond to system-wide broadcast announcements. Because broadcast receivers are another well-defined entry into the app, the system can deliver broadcasts even to apps that aren't currently running. So, for example, an app can schedule an alarm to post a notification to tell the user about an upcoming event... and by delivering that alarm to a `BroadcastReceiver` of the app, there is no need for the app to remain running until the alarm goes off.

Content Providers:

A content provider manages a shared set of app data that you can store in the file system, in an SQLite database, on the web, or on any other persistent storage location that your app can access. Through the content provider, other apps can query or modify the data if the content provider allows it. For example, the Android system provides a content provider that manages the user's contact information.

4.2 Functions

protected void onCreate(Bundle savedInstanceState):

The `savedInstanceState` is a reference to a `Bundle` object that is passed into the `onCreate` method of every Android Activity. Activities have the ability, under special circumstances, to restore themselves to a previous state using the data stored in this bundle.

super.onCreate(savedInstanceState):

By calling `super.onCreate(savedInstanceState);` you tell the Dalvik VM to run your code in addition to the existing code in the `onCreate()` of the parent class. The code in the framework classes handle stuff like UI drawing, house cleaning and maintaining the Activity and application lifecycles.

setContentView setContentView(R.layout.main):

R means Resource. layout means design. main is the xml you have created under res->layout->main.xml. Whenever you want to change the current look of an Activity or when you move from one Activity to another, the new Activity must have a design to show.

protected void onCreate(Bundle savedInstanceState):

The savedInstanceState is a reference to a Bundle object that is passed into the onCreate method of every Android Activity. Activities have the ability, under special circumstances, to restore themselves to a previous state using the data stored in this bundle.

public void movepageh(View v):

This function is present in the Dashboard of our application. This is called to set a new Intent and move into activity_select_use_health Activity.

public void movepagef(View v):

This function is present in the Dashboard of our application. This is called to set a new Intent and move into activity_select_use_finance Activity.

public void movepagem(View v):

This function is present in the Dashboard of our application. This is called to set a new Intent and move into activity_select_use_finance Activity.

intent(context, class):

An Intent object carries information that the Android system uses to determine which component to start (such as the exact component name or component category that should receive the intent), plus information that the recipient component uses in order to properly perform the action (such as the action to take

startActivity(intentObject):

An Activity represents a single screen in an app. You can start a new instance of an Activity by passing an Intent to startActivity() .

findViewById:

Finds a view that was identified by the id attribute from the XML layout resource.

public void bmicalc (View v):

This function is present in the activity_select_use_health Activity of our application. This is called to set a new Intent and move into BMI calculator.

public void bfatcalc (View v):

This function is present in the activity_select_use_health Activity of our application. This is called to set a new Intent and move into Body Fat calculator.

public void emcalc (View v):

This function is present in the activity_select_use_finance Activity of our application. This is called to set a new Intent and move into EMI calculator.

public void sintcalc (View v):

This function is present in the activity_select_use_finance Activity of our application. This is called to set a new Intent and move into Simple Interest calculator.

public void cintcalc (View v):

This function is present in the activity_select_use_finance Activity of our application. This is called to set a new Intent and move into Compound Interest calculator.

public void simpcalc (View v):

This function is present in the activity_select_use_math Activity of our application. This is called to set a new Intent and move into Simple calculator.

public void pcalc (View v):

This function is present in the activity_select_use_math Activity of our application. This is called to set a new Intent and move into nth Power and Root calculator.

public void qcalc (View v):

This function is present in the activity_select_use_math Activity of our application. This is called to set a new Intent and move into nth Quadratic Equation calculator.

Float.parseFloat(String):

This function converts given String Type input from the EditText into Float type.

id.setText(String):

This function sets a String to any given ID of TextView or EditText.

SNAPSHOTS

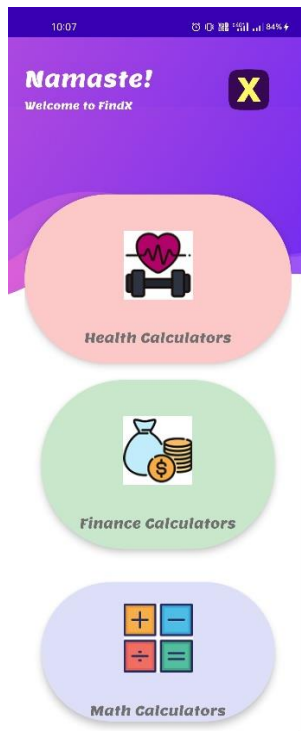


Figure 5.0

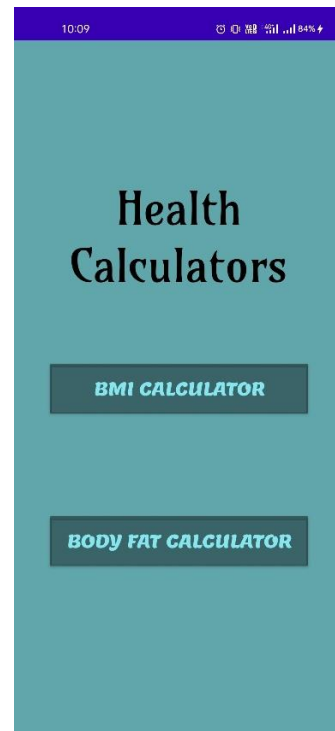


Figure 5.1



Figure 5.2



Figure 5.3

10:15

Body Fat Calculator

AGE Age (2-120)

GENDER F or M

HEIGHT Height (cm)

WEIGHT Weight (kg)

CALCULATE

BODY FAT Result (%)

Figure 5.4

10:15

Body Fat Calculator

AGE 21

GENDER F

HEIGHT 154

WEIGHT 60

CALCULATE

BODY FAT 20.13

Figure 5.5

10:16

Finance Calculators

EMI CALCULATOR

SIMPLE INTEREST CALCULATOR

COMPOUND INTEREST CALCULATOR

Figure 5.6

10:16

EMI Calculator

PRICE Price

DOWN PAYMENT Down Payment

LOAN AMOUNT Calculate/Enter Amount

CALCULATE LOAN AMT/DOWN PAYMENT

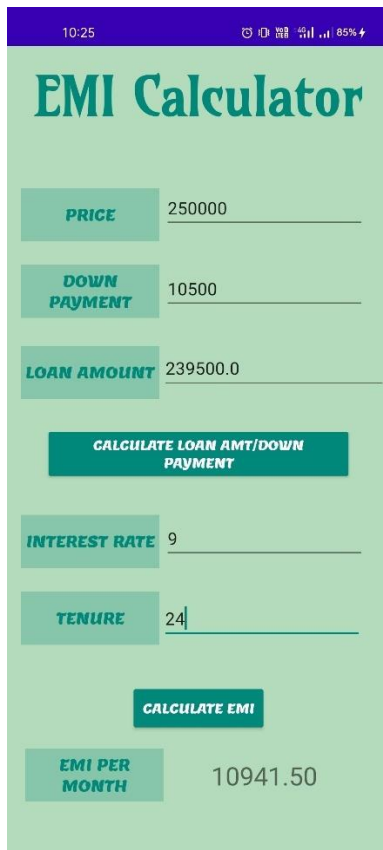
INTEREST RATE Interest Rate PA

TENURE Tenure in Months

CALCULATE EMI

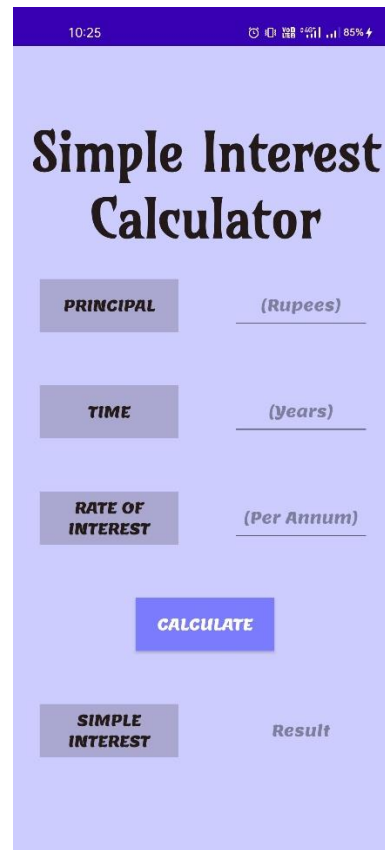
EMI PER MONTH

Figure 5.7



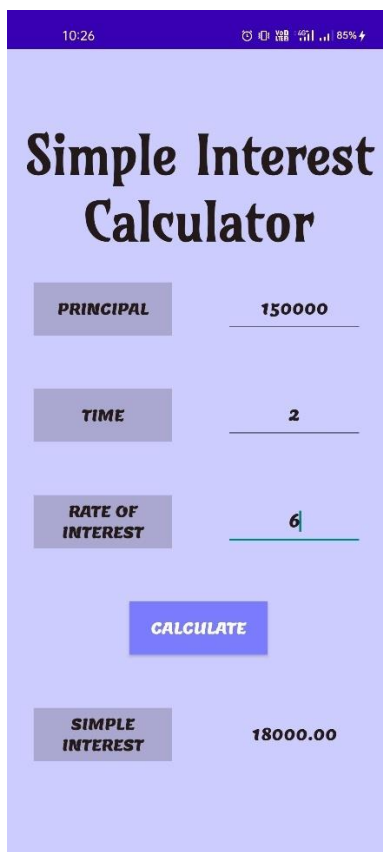
The EMI Calculator app interface features a green background. At the top, the title "EMI Calculator" is displayed in a large, bold, teal font. Below the title, there are three input fields: "PRICE" with the value 250000, "DOWN PAYMENT" with the value 10500, and "LOAN AMOUNT" with the value 239500.0. A teal button labeled "CALCULATE LOAN AMT/DOWN PAYMENT" is positioned below these fields. Further down, there are two more input fields: "INTEREST RATE" with the value 9 and "TENURE" with the value 24. A teal button labeled "CALCULATE EMI" is located below these fields. At the bottom, the result "EMI PER MONTH" is shown as 10941.50.

Figure 5.8



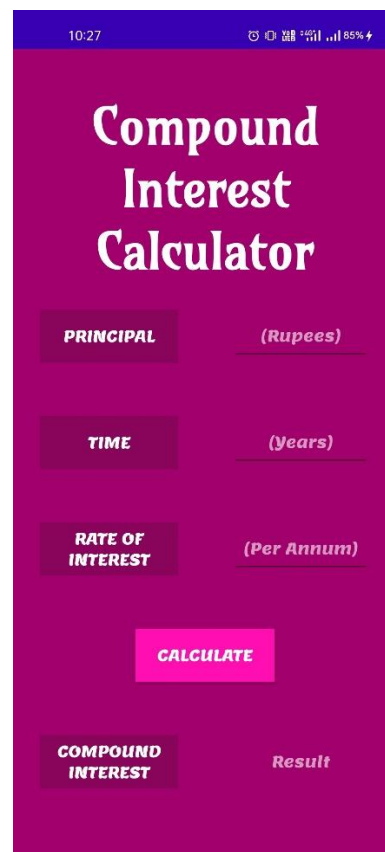
The Simple Interest Calculator app interface has a light purple background. The title "Simple Interest Calculator" is at the top in a large, bold, black font. Below the title, there are three input fields: "PRINCIPAL" with the placeholder "(Rupees)", "TIME" with the placeholder "(Years)", and "RATE OF INTEREST" with the placeholder "(Per Annum)". A blue button labeled "CALCULATE" is positioned below these fields. At the bottom, the result "SIMPLE INTEREST" is shown as "Result".

Figure 5.9



The Simple Interest Calculator app interface has a light purple background. The title "Simple Interest Calculator" is at the top in a large, bold, black font. Below the title, there are three input fields: "PRINCIPAL" with the value 150000, "TIME" with the value 2, and "RATE OF INTEREST" with the value 6. A blue button labeled "CALCULATE" is positioned below these fields. At the bottom, the result "SIMPLE INTEREST" is shown as 18000.00.

Figure 5.10



The Compound Interest Calculator app interface has a magenta background. The title "Compound Interest Calculator" is at the top in a large, bold, white font. Below the title, there are three input fields: "PRINCIPAL" with the placeholder "(Rupees)", "TIME" with the placeholder "(Years)", and "RATE OF INTEREST" with the placeholder "(Per Annum)". A pink button labeled "CALCULATE" is positioned below these fields. At the bottom, the result "COMPOUND INTEREST" is shown as "Result".

Figure 5.11

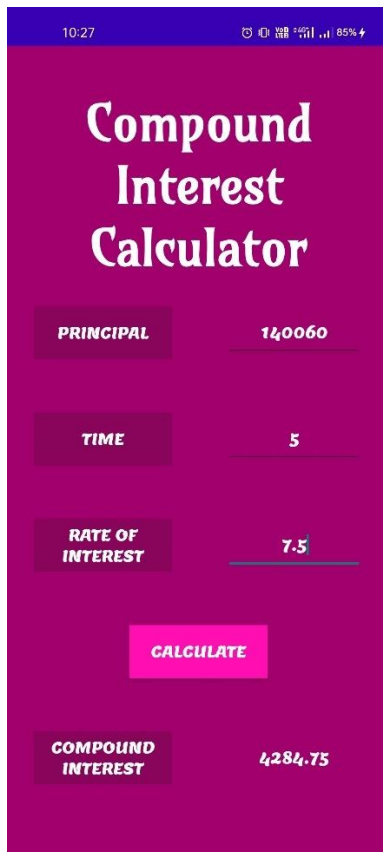


Figure 5.12

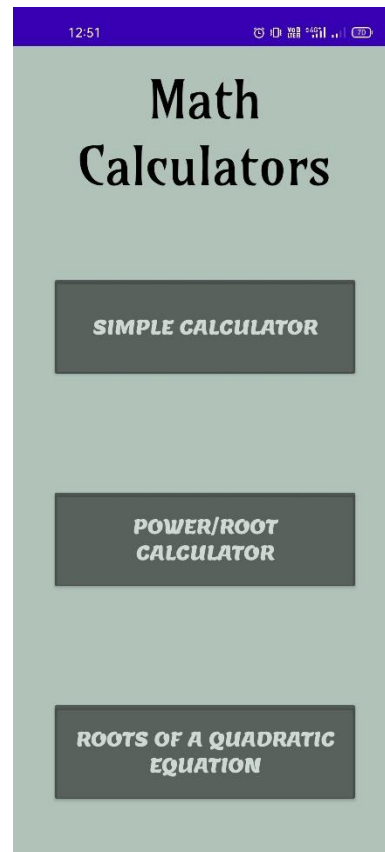


Figure 5.13

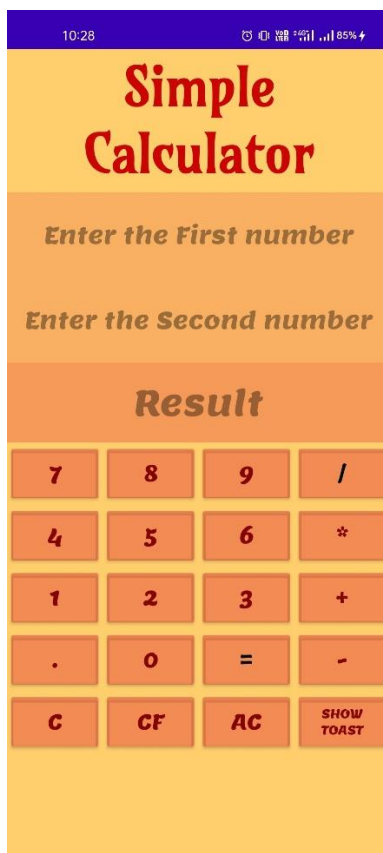


Figure 5.14

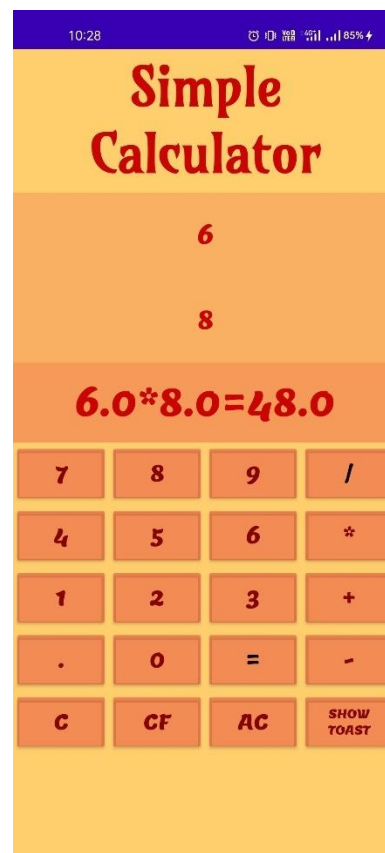


Figure 5.15

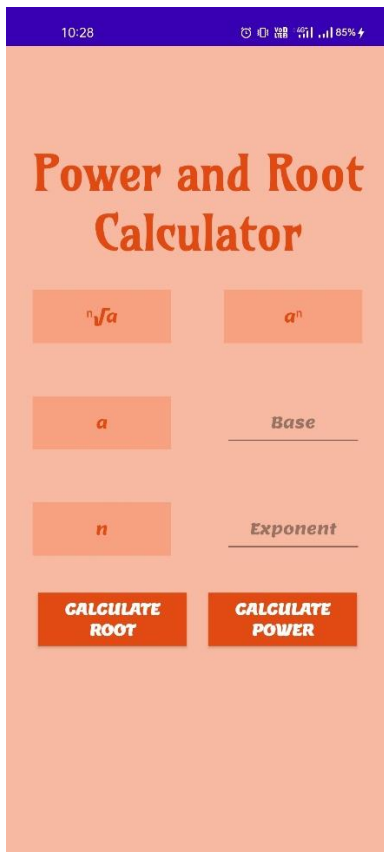


Figure 5.16

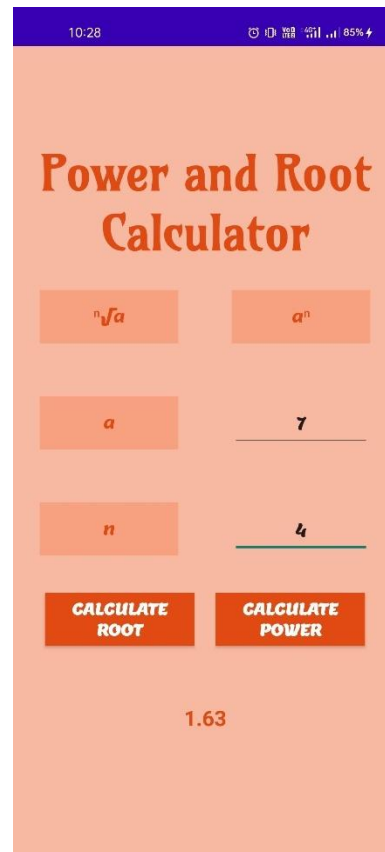


Figure 5.17

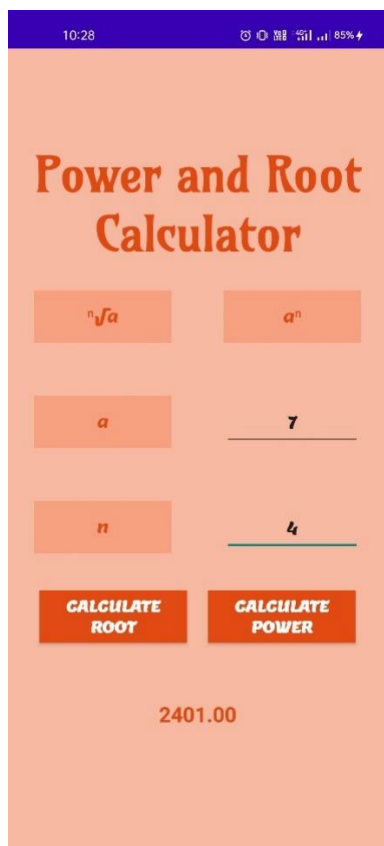


Figure 5.18

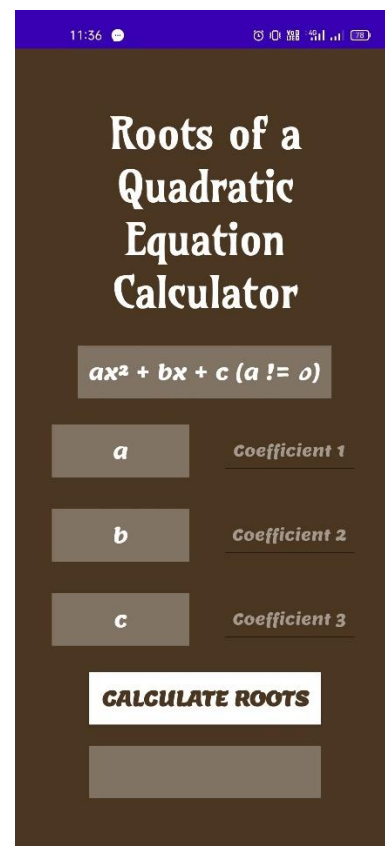


Figure 5.19

11:36

Roots of a Quadratic Equation Calculator

$ax^2 + bx + c (a \neq 0)$

a	2
b	5
c	3

CALCULATE ROOTS

-1.00, -1.50

Figure 5.20

CONCLUSION AND FUTURE SCOPE

This project allows the user to have multiple calculators inside a single application at his disposal. List of calculators:

- Simple Calculator
- Root and Power Calculator
- Roots of a Quadratic Equation Calculator
- EMI Calculator
- Simple Interest Calculator
- Compound Interest Calculator
- BMI Calculator
- Body Fat Calculator

FUTURE SCOPE:

- **CROSS PLATFORM COMPATIBILITY:**
We can develop the application using flutter to have cross platform compatibility. The app then can run on both IOS and Android seamlessly.
- **ADDITIONAL CALCULATORS:**
We can develop and implement many other types of calculators in our application.

BIBLIOGRAPHY/REFERENCE

1. H.M.Deitel and P.J.Deital, Java How to program: SixthEdition
2. Herbert Schildt, The Complete Reference: Fifth edition

USER GUIDE

Run on a real device:

1. Set up your device as follows:
2. Connect your device to your development machine with a USB cable. If you developed on
3. Windows, you might need to install the appropriate USB driver for your device.
4. Perform the following steps to enable USB debugging in the Developer options window:
5. Open the Settings app.
6. If your device uses Android v8.0 or higher, select System. Otherwise, proceed to the next step.
7. Scroll to the bottom and select About phone.
8. Scroll to the bottom and tap Build number seven times.
9. Return to the previous screen, scroll to the bottom, and tap Developer options.
10. In the Developer options window, scroll down to find and enable USB debugging.

Run the app on the Emulator as follows:

1. In Android Studio, select your app from the run/debug configurations drop-down menu in the toolbar.
2. In the toolbar, select the device that you want to run your app on from the target device drop-down menu.