

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**“Jnana Sangama”, Belagavi – 590 014**



**MOBILE APPLICATION DEVELOPMENT (18CSMP68)**  
**REPORT ON**

**“RETURN ON INVESTMENT CALCULATOR”**

Submitted in partial fulfilment of the requirements for Mobile Application development  
(18CSMP68) the award of the degree of

**Bachelor of Engineering**

In

Computer Science and Engineering

By

Mushfiq K  
Shaheer

4BP19CS046  
4BP19CS056

Under the guidance of:

Prof. Afsar Baig M  
Head of Department of CSE, BIT

**Academic Year: 2021 -22**



Bearys  
Institute  
of Technology  
MANGALORE

**Department of Computer Science and Engineering,**  
**Bearys Institute of Technology,**  
**MANGALURU - 574 153**

# BEARYS INSTITUTE OF TECHNOLOGY

*(Affiliated to Visvesvaraya Technological University & Recognized by AICTE)*

INNOLI, MANGALURU – 574153, KARNATAKA

DEPARTMENT OF

**COMPUTER SCIENCE AND ENGINEERING**



Bearys  
Institute  
of Technology  
MANGALORE

## CERTIFICATE

This is to certify that the mini project work entitled “**Return On Investment Calculator**” carried out by **Mushfiq K (4BP19CS046)** and **Shaheer (4BP19CS056)** in partial fulfilments of the requirements of Mobile Application Development Laboratory (18CSMP68) prescribed by the Visvesvaraya Technological University, Belagavi for the VI Semester B.E (Computer Science and Engineering) Degree course during the academic year of 2021-2022.

---

**Signature of the H.O.D**  
**Prof. Afsar Baig M,**  
Head of Department,  
Department of CSE, BIT

## **DECLARATION**

We, the students of the sixth semester of Computer Science and Engineering, Bearys Institute of Technology, Mangalore, declare that the work entitled “**Return On Investment Calculator**” has been successfully completed under the guidance of Mr. Afsar Baig M, Computer Science and Engineering Department, Bearys Institute of Technology. This dissertation work is submitted to Visvesvaraya Technological University in partial fulfilment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science during the academic year 2021 - 2022. Further, the matter embodied in the project report has not been submitted previously by anyone for the award of any degree or diploma to any university.

### **Team members:**

- 1. Mushfiq K (4BP19CS046)**
- 2. Shaheer (4BP19CS056)**

**Place:**

**Date:**

# ACKNOWLEDGEMENT

The knowledge and satisfaction that accompanies a successful completion of a project is hard to describe. Behind any successful project there are wise people guiding throughout. We thank them for guiding us, correcting our mistakes, and providing valuable feedback. We would consider it as our privilege to express our gratitude and respect to all those who guided and encouragement.

We extend our heartfelt gratitude to our chairman, **MR. SAYYED MUHAMMAD BEARY** and also to our beloved principal, **DR. S.I MANJUR BASHA** for the success of this project.

We are grateful to **MR. AFSAR BAIG M**, Professor & Head of CSE Department, Bearys Institute of Technology, for guidance and encouragement at all times needed.

**MUSHFIQ K (4BP19CS046)**  
**SHAHEER (4BP19CS056)**  
**6<sup>th</sup> SEM CSE**

## **ABSTRACT**

This project deals with the demonstration of the 'Return On Investment Calculator. ROI Calculator is the easiest and excellent app for calculating the profit percentage from an investment. With an Annualized ROI. In finance, Return on Investment, known as ROI, is a common, metric used to evaluate the forecasted profitability on an investment. It is simply a calculator-like app developed using Java programming language. Such application can be used by any user, but it is mainly useful for business, shares, and finance related areas where money transfer and currency exchange takes place on a daily basis. ROI calculator is a kind of investment calculator that enables you to estimate the profit or loss on your investment. If the calculation process of ROI is known, then it's easier to foresee the results of an investment.

# TABLE OF CONTENTS

<b>Titles</b>	<b>Page No.</b>
<b>List of Figures</b>	<b>I</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>01</b>
1.1 Mobile Application Development	01
1.2 Java and XML in Mobile Application	02
1.3 Introduction to the Project	03
<b>CHAPTER 2 SYSTEM REQUIREMENTS</b>	<b>04</b>
2.1 Functional Requirements	04
2.2 Non-Functional Requirements	04
2.3 Hardware Requirements	05
2.4 Software Requirements	05
<b>CHAPTER 3 DESIGN</b>	<b>06</b>
<b>CHAPTER 4 IMPLEMENTATION</b>	<b>08</b>
4.1 Implementation Requirements	08
4.2 Code Implementation	08
<b>CHAPTER 5 RESULTS</b>	<b>15</b>
<b>CONCLUSION</b>	<b>21</b>
<b>REFERENCES</b>	<b>22</b>

## LIST OF FIGURES

<b>Fig. No.</b>	<b>Description</b>	<b>Page No.</b>
5.1.1	Simple ROI Calculator	15
5.1.2	Investment Calculated page	16
5.1.3	Investment Calculator on Stock	17
5.1.4	Stock Investment calculated page	18
5.1.5	Investment calculator on Real Estate	19
5.1.6	Real Estate Investment Calculated page	20

## **CHAPTER 1**

# **INTRODUCTION**

## **1.1 Mobile Application Development**

Mobile application development is the process to making software for smartphones and digital assistants, most commonly for Android and iOS. The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used for this kind of software development include Java, Swift, C# and HTML5. Mobile app development is rapidly growing. From retail, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transactions and access information [1]. Today, mobile devices—and the mobile applications that unlock their value—are the most popular way for people and businesses to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications that their customers, partners and employee's demand.



## 1.2 Java and XML in Mobile Application Development

Android is an open-source software platform and Linux-based operating system for mobile devices. The Android platform allows developers to write managed code using Java to manage and control the Android device. Android applications can be developed by using the Java programming language and the Android SDK. So, familiarity with the basics of the Java programming language is a prerequisite for programming on the Android platform.

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. XML tags are not predefined in XML. We must define our own Tags. XML as itself is well readable both by human and machine. Also, it is scalable and simple to develop. In Android we use xml for designing our layouts because xml is lightweight language so it doesn't make our layout heavy.

### **1.3 Introduction to the Project**

An application is to be developed to represent dynamic functionalities like to calculate return on investment. ROI Calculator is the most simple and excellent app for calculating. In finance, Return on Investment, usually abbreviated as ROI, is a common, widespread metric used to evaluate the forecasted profitability on different investments. ROI calculator is a kind of investment calculator that enables you to estimate the profit or loss on your investment. This is just an app development like a calculator using Java programming language. This application can be used by any user, but it is mainly useful for business, shares and finance-related areas where money transfer and currency exchange takes place daily. ROI calculator is a kind of investment calculator that enables you to estimate the profit or loss on your investment. If the calculation process of ROI is known, then it's easier to foresee the results of an investment.

## CHAPTER 2

# SYSTEM REQUIREMENTS

The package is designed such that users with a computer having minimum configuration can also use it, which does not require complex android studio packages. The package requires simple in-built functions found in the import statements along with a few user defined functions.

### 2.1 Functional Requirements

Functional Requirements describe a product must behave and what its features and function. Generally, functional requirement describes system behavior under specific condition. The proposed project has module login module, on-going module, newsfeed module, and the stories module.

### 2.2 Non-Functional Requirements

Nonfunctional requirements describe the general characteristics of a system. They are also known as quality attributes. Nonfunctional requirements describe how a system must behave and establish constraints of its functionality. Some typical nonfunctional requirements are:

- **Scalability:** Application supports for large number of users.
- **Safety:** The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

- **Security:** System will use secured database. Normal users can just read the information but they cannot edit or modify anything except their personal and some other information and every user has access constraints.
- **User friendly:** System is very interactive.
- **Availability:** The application is made available all time.

## 2.3 Hardware Requirements

- Operating System - Ubuntu 14.04 or Windows
- Language Tool – XML, Java (jdk 14.0 or newer)
- Documentation Tool - MS Word
- IDE – Android Studio

## 2.4 Software Requirements

- Processor - Compatible with and after i3
  - RAM – 4 GB RAM (minimum)
  - Hard Disk – 4 GB (minimum)
- Monitor – VGA Compatible

## CHAPTER 3

# DESIGN

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. In this process requirements are translated into a representation of software. Initially, the representation depicts a holistic view of software. Subsequent refinement leads to a design representation that is very close to source code. One could see it as the application of system theory to product development. There is some overlap with the disciplines of system analysis, system architecture and system engineering.

The design of Return on Investment Calculator app is similar to a simple calculator app. The designed app is presented through an emulator.

It consists of 6 pages:

1. Simple Return On Investment page.

In this page user enters Amount invested, Amount Returned and Investment Period then clicks on Calculate.

2. Investment Calculated Page.

In this page user gets the display of Amount Invested, Amount Returned, Period Of Investment, Net-Profit/Loss, Profit/Loss percentage, Annual Growth Rate, Profit in next one year and Total Amount after one Year.

3. Stock Investment Calculator page.

In this page user enters Buying Price Per Stock, Current Price Per Stock, Quantity, Period of investment. And there will be a button as Calculate

#### 4. Stock Investment Calculated page.

In this page user gets the display of Buying Price Per Stock, Current Price Per Stock, Quantity, Period of investment, Total buying price, Current buying price, Net-Profit/Loss and Compound monthly growth rate.

#### 5. Real Estate Investment calculator page.

In this page user enters Number of Square yards, Buying price, Current price, Time Period And there will be button as Calculate.

#### 6. Real Estate Investment calculated page

In this page user gets the display of Number of Square yards, Buying price, Current price, Time Period, total buying value, total current value, Net-Profit/loss, Profit/loss percentage And compound annual growth rate.

## CHAPTER 4

# IMPLEMENTATION

### 4.1 Implementation Requirements

For Simple Return on Investment Calculator, The application should contain a space to enter the amount to be Invested, Amount Returned and Investment period. As for Stocks, It should contain enters Buying Price Per Stock, Current Price Per Stock, Quantity, Period of investment, As for Real Estate It should Contain Number of Square yards, Buying price, Current price, Time Period. Each page Contains a 'Calculate' Button, When user clicks on it, it provides Net-Profit/loss, Percentage of Profit and Loss on Investment . For the implementation of this application we used Android Studio. The test here was to make the application user-friendly by making interface basic, easy to use and helpful for its fundamental reason.

### 4.1 Code Implementation

#### 4.1.1 MainActivity.java

```
package com.example.roicalc;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import java.nio.channels.SeekableByteChannel;

public class MainActivity extends AppCompatActivity {

    public static final String EXTRA_NUMBER =
        "com.example.application.roicalc.EXTRA_NUMBER";
    public static final String EXTRA_NUMBER1 =
        "com.example.application.roicalc.EXTRA_NUMBER1";
    public static final String EXTRA_NUMBER2 =
        "com.example.application.roicalc.EXTRA_NUMBER2";
    SeekBar sb;
    TextView tv1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sb = (SeekBar) findViewById(R.id.seekBar);
        tv1 = (TextView) findViewById(R.id.years);
        sb.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                tv1.setText(progress+"");
            }
        });

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
```



```
    }  
    });
```

```
Button button = (Button) findViewById(R.id.btn_calc);  
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
        openactivity2();  
  
    }  
});  
Button button1 = (Button) findViewById(R.id.btn_stock);  
button1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        openstock();  
    }  
});  
Button button2 = (Button) findViewById(R.id.btn_re);  
button2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        openreal();  
    }  
});  
  
}  
public void openactivity2(){  
    EditText editText1 = (EditText) findViewById(R.id.amt_in);  
    int number = Integer.parseInt(editText1.getText().toString());  
  
    EditText editText2 = (EditText) findViewById(R.id.amt_ret);  
    int number1 = Integer.parseInt(editText2.getText().toString());  
  
    int number2 = Integer.parseInt(tv1.getText().toString());  
  
    Intent intent = new Intent(this,calculate.class);
```

```
intent.putExtra(EXTRA_NUMBER,number);
intent.putExtra(EXTRA_NUMBER1,number1);
intent.putExtra(EXTRA_NUMBER2,number2);

startActivity(intent);

}
public void openstock(){
Intent intent = new Intent(this,stock.class);
startActivity(intent);
}
public void openreal(){
Intent intent = new Intent(this,real.class);
startActivity(intent);
}
}
```

#### **4.1.2 stock.java**

```
package com.example.roicalc;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class stock extends AppCompatActivity {
    public static final String EXTRA_NUMBER =
    "com.example.application.roicalc.EXTRA_NUMBER";
    public static final String EXTRA_NUMBER1 =
    "com.example.application.roicalc.EXTRA_NUMBER1";
    public static final String EXTRA_NUMBER2 =
    "com.example.application.roicalc.EXTRA_NUMBER2";
    public static final String EXTRA_NUMBER3 =
    "com.example.application.roicalc.EXTRA_NUMBER3";

    @Override
```

---

```
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_stock);

Button button = (Button) findViewById(R.id.sbtn);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        openstockcalc();
    }
});
}

public void openstockcalc(){
EditText sbuy = (EditText) findViewById(R.id.sbuy);
Double number = Double.parseDouble(sbuy.getText().toString());

EditText ssell = (EditText) findViewById(R.id.ssell);
Double number1 = Double.parseDouble(ssell.getText().toString());

EditText sqnt = (EditText) findViewById(R.id.sqnt);
int number2 = Integer.parseInt(sqnt.getText().toString());

EditText smon = (EditText) findViewById(R.id.smon);
int number3 = Integer.parseInt(smon.getText().toString());
Intent intent = new Intent(this,stockcalc.class);

intent.putExtra(EXTRA_NUMBER,number);
intent.putExtra(EXTRA_NUMBER1,number1);
intent.putExtra(EXTRA_NUMBER2,number2);
intent.putExtra(EXTRA_NUMBER3,number3);

startActivity(intent);
}
```

### 4.2.3 real.java

```
package com.example.roicalc;

import androidx.appcompat.app.AppCompatActivity;

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

public class real extends AppCompatActivity {
    public static final String EXTRA_NUMBER =
"com.example.application.roicalc.EXTRA_NUMBER";
    public static final String EXTRA_NUMBER1 =
"com.example.application.roicalc.EXTRA_NUMBER1";
    public static final String EXTRA_NUMBER2 =
"com.example.application.roicalc.EXTRA_NUMBER2";
    public static final String EXTRA_NUMBER3 =
"com.example.application.roicalc.EXTRA_NUMBER3";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_real);
        Button button = (Button) findViewById(R.id.rebtn);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                openrealcalc();

            }
        });
    }

    public void openrealcalc(){

        EditText reyar = (EditText) findViewById(R.id.reyar);
```

---

```
int number =Integer.parseInt(reyar.getText().toString());

EditText rebuy = (EditText) findViewById(R.id.rebuy);

Double number1 = Double.parseDouble(rebuy.getText().toString());

EditText resell = (EditText) findViewById(R.id.resell);
Double number2 = Double.parseDouble(resell.getText().toString());

EditText reyears = (EditText) findViewById(R.id.reyears);
int number3 = Integer.parseInt(reyears.getText().toString());

Intent intent = new Intent(this,realcalc.class);

intent.putExtra(EXTRA_NUMBER,number);
intent.putExtra(EXTRA_NUMBER1,number1);
intent.putExtra(EXTRA_NUMBER2,number2);
intent.putExtra(EXTRA_NUMBER3,number3);

startActivity(intent);
}
```

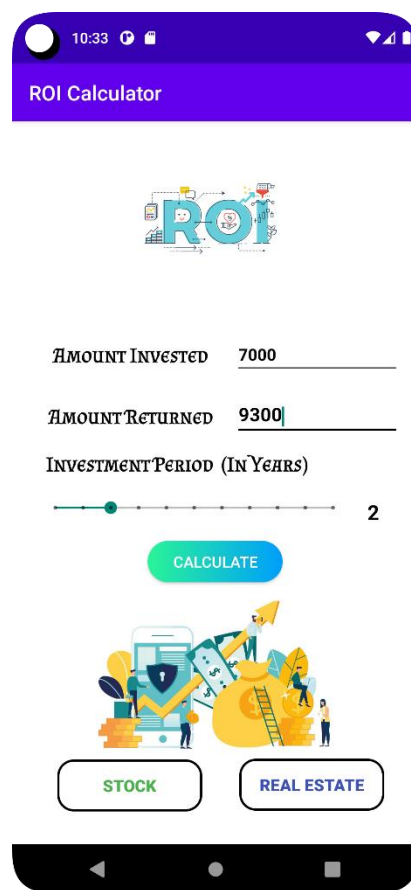
## CHAPTER 5

# RESULTS

### 5.1 Screenshots

#### 5.1.1 Simple ROI Calculator

In this page user enters Amount invested, Amount Returned and Investment Period then clicks on Calculate..

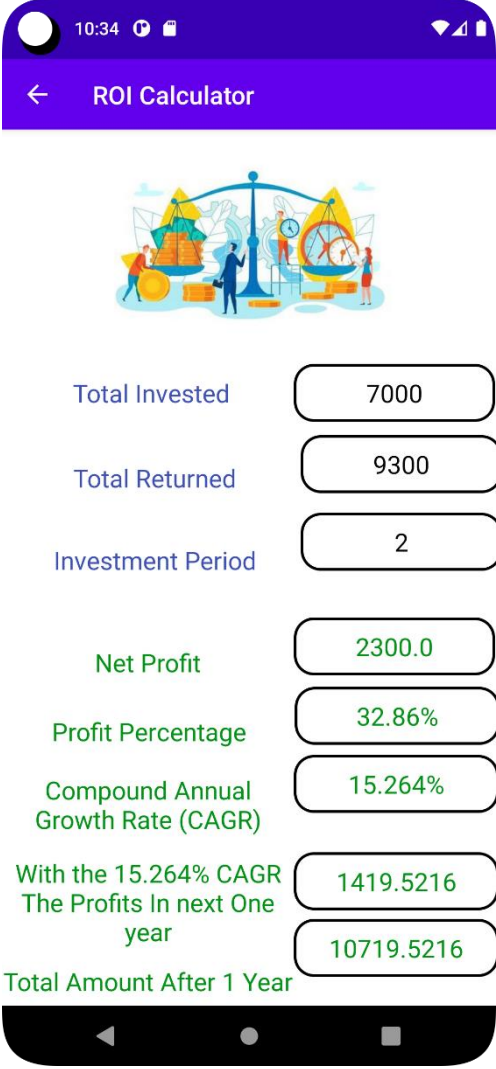


The screenshot shows a mobile application interface for an ROI Calculator. At the top, the status bar displays the time 10:33 and various icons. Below the status bar, the app title "ROI Calculator" is visible. The main interface features a large, stylized "ROI" graphic. Below this, there are three input fields: "AMOUNT INVESTED" with the value 7000, "AMOUNT RETURNED" with the value 9300, and "INVESTMENT PERIOD (IN YEARS)" with a slider set to 2. A green "CALCULATE" button is positioned below the input fields. At the bottom, there are two buttons labeled "STOCK" and "REAL ESTATE". The interface is decorated with financial-themed illustrations, including a laptop, a bar chart, and a large gold coin.

**Fig 5.1.1: Simple ROI Calculator**

### 5.1.2 Investment Calculated Page

Here it will show the Calculated result of Investment.



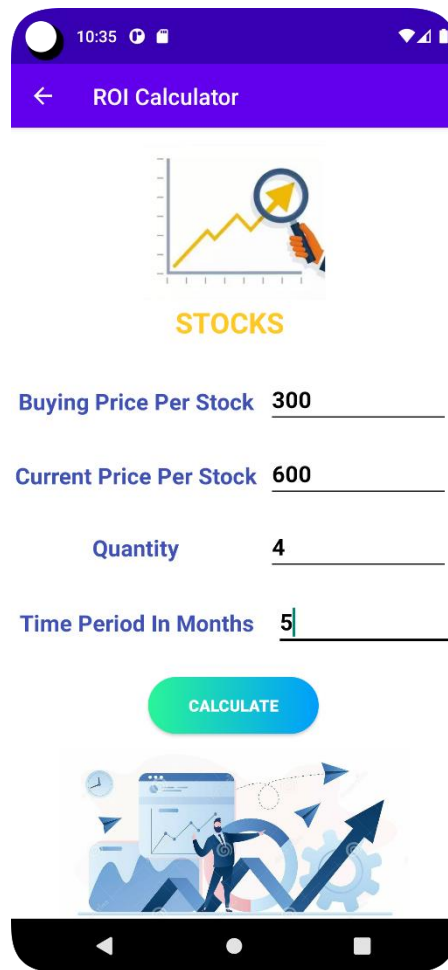
The screenshot displays the 'ROI Calculator' app interface. At the top, there is a purple header with a back arrow and the title 'ROI Calculator'. Below the header is an illustration of a balance scale with people and money. The main area contains input fields for 'Total Invested' (7000), 'Total Returned' (9300), and 'Investment Period' (2). Below these are calculated results: 'Net Profit' (2300.0), 'Profit Percentage' (32.86%), 'Compound Annual Growth Rate (CAGR)' (15.264%), and 'With the 15.264% CAGR The Profits In next One year' (1419.5216). The final result, 'Total Amount After 1 Year', is 10719.5216. The bottom of the screen shows a black navigation bar with three icons.

Input	Value
Total Invested	7000
Total Returned	9300
Investment Period	2
Net Profit	2300.0
Profit Percentage	32.86%
Compound Annual Growth Rate (CAGR)	15.264%
With the 15.264% CAGR The Profits In next One year	1419.5216
Total Amount After 1 Year	10719.5216

**Figure 5.1.2: Investment calculated Page**

### 5.1.3 Investment Calculator On Stocks

Here user enters Buying Price Per Stock, Current Price Per Stock, Quantity, Period of investment.



The screenshot shows a mobile application interface for an ROI Calculator. At the top, there is a purple header bar with a back arrow and the text "ROI Calculator". Below the header is a yellow line graph with a magnifying glass icon and the word "STOCKS" in yellow. The main form contains four input fields: "Buying Price Per Stock" with the value 300, "Current Price Per Stock" with the value 600, "Quantity" with the value 4, and "Time Period In Months" with the value 5. A green "CALCULATE" button is positioned below the input fields. At the bottom, there is a blue illustration of a person standing next to a large upward-pointing arrow, with a clock and a gear icon nearby. The entire interface is displayed on a black background representing a smartphone screen.

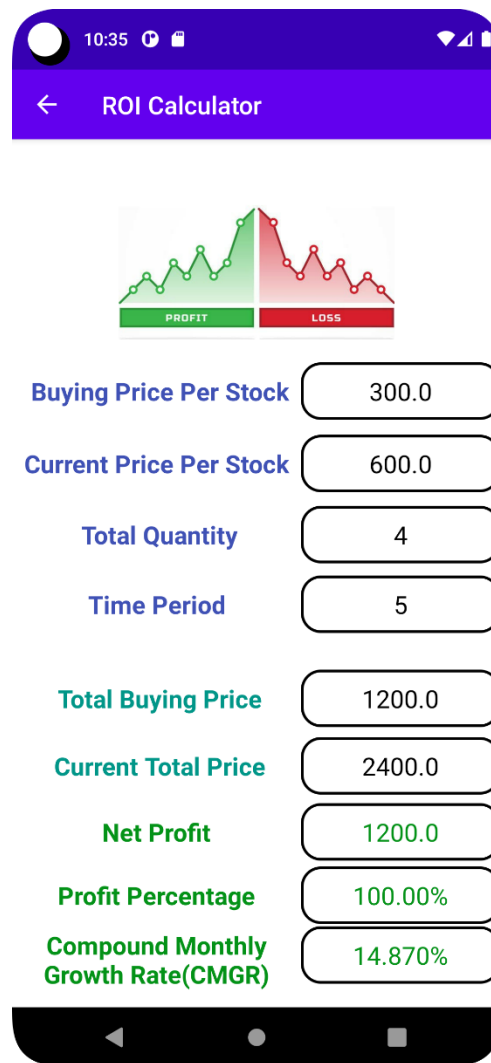
Field	Value
Buying Price Per Stock	300
Current Price Per Stock	600
Quantity	4
Time Period In Months	5

**Figure 5.1.3 Investment calculator on stocks**



### 5.1.4 Stock Investment Calculated Page

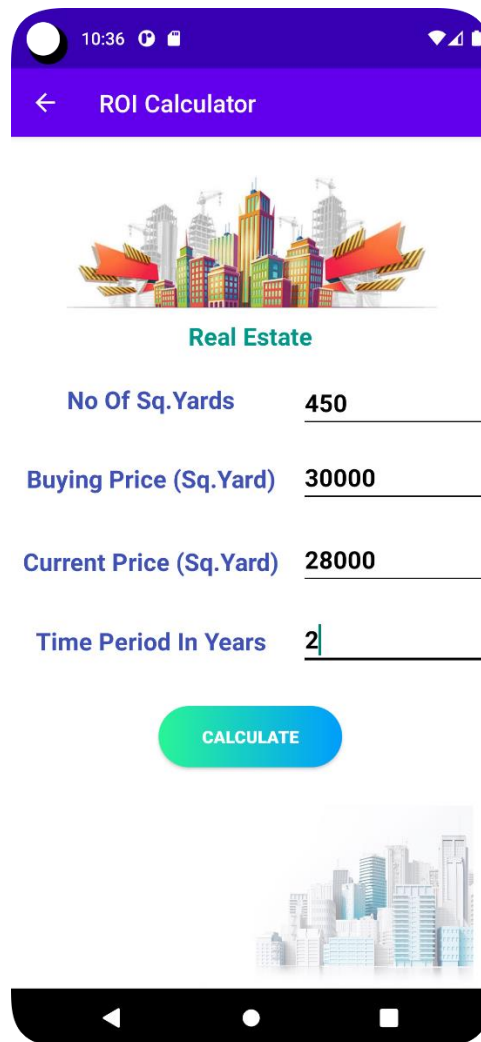
Here it will show the Calculated result of Investment on Stocks



### 5.1.4 Stock Investment Calculated Page

### 5.1.5 Investment Calculator On Real Estate

Here user enters Number of Square yards, Buying price, Current price, Time Period



10:36

← ROI Calculator

Real Estate

No Of Sq.Yards 450

Buying Price (Sq.Yard) 30000

Current Price (Sq.Yard) 28000

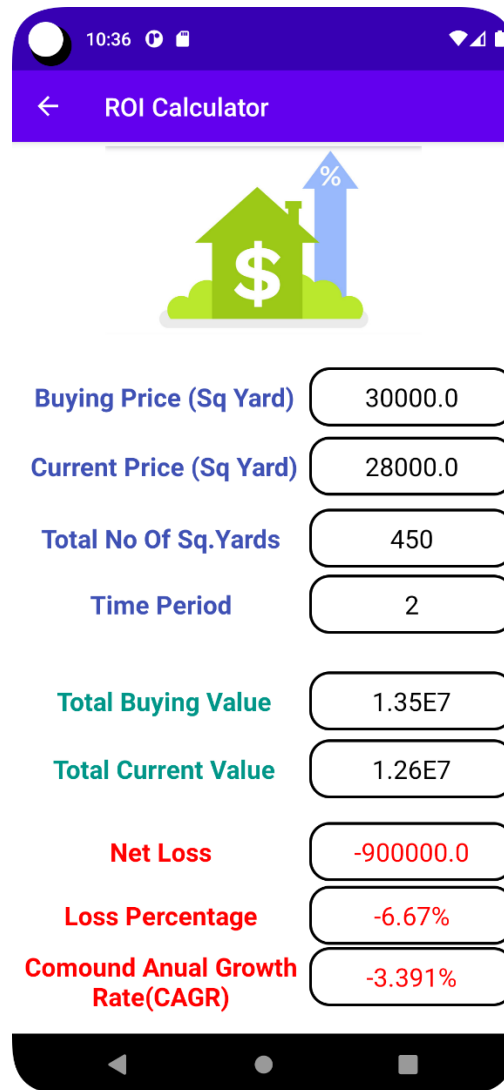
Time Period In Years 2

CALCULATE

### 5.1.5 Investment Calculator On Real Estate

### 5.1.6 Real Estate Investment Calculated Page

Here it will show the Calculated result of Investment on Real Estate



The screenshot displays a mobile application interface for an ROI Calculator. The app has a purple header with a back arrow and the title "ROI Calculator". Below the header is a green house icon with a white dollar sign and a blue upward arrow with a percentage sign. The main area contains a list of input and output fields. The input fields are: "Buying Price (Sq Yard)" with value 30000.0, "Current Price (Sq Yard)" with value 28000.0, "Total No Of Sq.Yards" with value 450, and "Time Period" with value 2. The output fields are: "Total Buying Value" with value 1.35E7, "Total Current Value" with value 1.26E7, "Net Loss" with value -900000.0, "Loss Percentage" with value -6.67%, and "Comound Annual Growth Rate(CAGR)" with value -3.391%. The bottom of the screen shows a black navigation bar with three icons: a back arrow, a circle, and a square.

Buying Price (Sq Yard)	30000.0
Current Price (Sq Yard)	28000.0
Total No Of Sq.Yards	450
Time Period	2
Total Buying Value	1.35E7
Total Current Value	1.26E7
Net Loss	-900000.0
Loss Percentage	-6.67%
Comound Annual Growth Rate(CAGR)	-3.391%

### 5.1.6 Real Estate Investment Calculated Page

## CONCLUSION

A wonderful experience developing this project. By working on the project, we gained hands-on experience in using the Android Studio software using which we experimented with, in this project. We have also learned about the crucial things or tools needed to develop an application such as the Emulator, XML, and Java.

The project helped in understanding the working of a mobile application using the Android Studio and various concepts, functions, and methodologies for the development of application. The proposed system will serve its purpose without any hassles.

---

## REFERENCES

- [1]. Google Developer Training, “Android Developer Fundamentals Course – Concept Reference”, Google Developer Training Team, 2017.
- [2]. Erik Hellman, “Android Programming – Pushing the Limits”, 1<sup>st</sup> Edition, Wiley India Pvt Ltd, 2014. ISBN: 978-8126547197.
- [3]. [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
- [4]. [www.stackoverflow.com](http://www.stackoverflow.com)
- [5]. [www.github.com](http://www.github.com)

