CHAPTER 1

INTRODUCTION

1.1 Scope and Application of Mini Project

With colourful charts and instant results, our EMI calculator is easy to use, intuitive to understand and is quick perform. You can calculate home loan or bike loan or personal loan EMI with this calculator. If you are planning to avail any kind of loan, use our EMI calculator to determine. EMI payments against the principal loan amount and the interest on the loan and judge how affordable a loan can be for you. Always use the calculator to get a quick quote on your EMIs.
Enter the following information in the EMI calculator:

- Principal loan amount you wish to avail (rupees).
- Loans per month.

Rate of interest (percentage) EMI Calculator is a simple loan calculation tool that helps the user to quickly calculate EMI and view payment schedule. Use this Loan EMI Calculator to calculate your EMI Equated Monthly Instalment, plan your loan repayment in an effective way. You can also easily compare two loans by using Loan EMI Calculator. Emi Calculator : EMI ,TDR , STDR, RD, All Calculator in one App Easily Get Maturity Amount & instalments.

* Helps you take informed decisions about your finance

* No Internet connection is required

* Works on both Smart Phones and Tablets

Displays the Maturity amount.

Break-Up of the Due Amount is Provided: The bike loan EMI calculator helps you calculate the processing fees, interest that is paid, the total amount that must be paid, and the principal amount.
Your Budget can be Planned: Once you know the EMI that must be paid, you can plan your Budget accordingly. In case the loan amount that is being availed is large, you may think of opting for a longer tenure. These details can be determined by using the bike loan EMI calculator.
Accuracy: For the details that are being provided on the calculator, the results that are displayed are accurate. Manual calculations may not provide accurate results.

1.2About the Mini Project

EMI calculator is an simple and effective app built using android studio application and gives a result in no time with the perfect calculation.

The formula to calculate EMI is E = P * (r(1+r)n)/((1+r)n -1)

where ,

E = The EMI payable on the bike loan amount

P = The Bike loan Principal Amount

r = The interest rate value computed on a monthly basis

n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying the Bike. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four Edit Text to read the Principal Amount, Down Payment, Interest Rate, Loan Term (in months) and a button named as “Calculate Monthly EMI”. On click of this button, the result would be shown in a Text View. Also, calculate the EMI by varying the Loan Term and Interest Rate values

CHAPTER 2

LITERATURE SURVEY

Most of the research nowadays is focused towards problems that deal with complexity or are influenced by some kind of random events.

Interesting about these problems is that if they are deterministic, then a solution is expected to exist, at least a theoretical one. On the other hand the point of randomness involved in these problems increases the difficulty of prediction on the possible solution, or in some situations outcome.

This is why, there are certain methods of operations devised, that give some supplementary information to a decision maker. In most of the cases, the probability distribution of an even which took place randomly, it is possible to be affected by prior events.

1) This app is built on Android studio
2) Easy to reskin, change colours
3) Admob (App ID+ Interstitial + Native) only replace Admob App ID, Native ads id and Interstitial id's.
4) Replace your Icon, change the title, change package name, generate signed APK.

In such a situation, the Bike loan EMI calculator helps you in deciding the suitable bike loan amount and for managing your monthly finances accordingly. It is a hassle-free tool which assists you in planning your EMIs in advance.

Mobile Operating System is an operating system designed for smart phones, tablets and other mobile devices. It is just like a regular operating system for a computer but it combines other features along with camera, audio, video, radio, Bluetooth, speech recognition and many others.

2.1 Survey About the Project

Smart phones are extremely versatile gadgets, offering unlimited chances to utilize our time efficiently. On average, a smart phone has 41 applications, and every day many new phones are being activated. Perhaps the most startling statistic is that currently, 1.3 million Android devices are activated every day Which means that every 24 hours, more than four times as many new smart phones and tablets are set up. And once the devices are activated, they are checked on average 150 times a day once every 6.5 minutes. The apps in both Apple's and Google's app store combined is over 2 million. Admittedly, the apps are the purpose behind the smart phone. Furthermore, on average every smart phone contains 41

apps which are used for regular needs. Traditional way of calculating EMI is hardly to be used and moreover lack of knowledge with some people, so the Android App is the solution for this issue. A nice graphical user interface is responsible for displaying the EMI information to the user either on demand or proactively using sound or text alerting mechanisms. More intelligent applications have advanced features such as peer coordination for the scheduling of a meeting but they have not found wide acceptance in the market due to their complexity and computationally intensive operations. .However, there is still no application taking advantage of the ever increasing digital and networking convergence happening in our days.

2.2 RELATED WORK

Most of the research nowadays is focused towards problems that deal with complexity or are influenced by some kind of random events. Interesting about these problems is that if they are deterministic, then a solution is expected to exist, at least a theoretical one. On the other hand the point of randomness involved in these problems increases the difficulty of prediction on the possible solution, or in some situations outcome. This is why, there are certain methods of operations devised, that give some supplementary information to a decision maker.

This app is built on Android studio

1.Easy to reskin, change colours.
2.Admob (App ID+ Interstitial + Native) only replace Admob App ID, Native ads id and Interstitial.
3.Replace your Icon, change the title, change package name, generate signed APK.

In such a situation, the bike loan EMI calculator helps you in deciding the suitable bike loan amount and for managing your monthly finances accordingly. It is a hassle-free tool which assists you in planning your EMIs in advance.

2.3 Problem Statement:

Bike Loan Calculator. What is the calculator about? The bike loan calculator helps you calculate the equated monthly instalments' (EMIs) you have to pay the lender each month till the loan is fully paid .The EMI is based on the loan amount, the tenure, and interest rate. With each successive payment, you'll pay more towards the principal and less in interest. Here's the formula to calculate EMI: r is rate of interest calculated on monthly basis. (i.e., r = Rate of Annual interest/12/100. If rate of interest is 10.5% per annum, then r = 10.5/12/100=0.00875).

The interest component of the EMI would be larger during the initial months and gradually reduce with each payment. The exact percentage allocated towards payment of the principal depends on the interest rate. Even though your monthly EMI payment won't change, the proportion of principal and interest components will change with time.

2.4 Proposed System:

The proposed android app for the calculation of EMI is very effective and efficient. Without being into an issue , the android app provides the solution effortlessly. The EMI of any subject includes the Principal amount, Down Payment , Interest rate and the Loan in terms of month or years.

Here the Bike EMI Calculator app also considers all the four factors. Initially The principal amount is entered followed by Down Payment, Interest Rate and Loan in terms of month, and hence the EMI is calculated.

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATIONS.

1.System Requirements:

An android application can ideally be developed on any of the following platforms:

- ➢ Windows
- ➢ OS X
- ➢ Linux

2.Software requirements

- ➢ Java development kit(JDK)7
- ➢ Android SDK
- ➢ Android Studio

3.Hardware requirements:

- ➢ 4GB RAM minimum, 8 GB RAM recommended
- ➢ 400 MB hard disk space
- ➢ At least 1GB Android SDK, emulator system images, and caches
- ➢ 1280 x 800 minimum screen resolution.

CHAPTER 4

SYSTEM DESIGN AND ANAYLISIS

Before getting to implement the calculator directly we'll understand the files that we have created for this application

1. The first file that we have created is **activity_main.xml,** this file has the layout of the EMI Calculator we manage the interface of the Android Application.
2. The next important file is the **MainActivity.java** file. This file makes the application actually work.

It has all the methods that make functioning in the app possible.

Then we have developed some other files that were required for the interface of application that are listed below:

style.xml: This file has the description of cells for calculator board. It has height and width with the text size and colours described.

colors.xml: This file has the colours described so it would be easy to mention the colour name.

strings.xml: This describes the list of strings that are used in the application.

dimens.xml: This has the dimension of grid of our EMI android calculator.

System Design is the most creative and challenging phase in the system life cycle. Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. System design is a solution how to approach the creation of a new system. System design transforms a logic representation of what is required to do into the physical specification.

4.2 Logical design

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volumes etc.
- Prepares output specifications – that is, determines the format of report.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.

Specifies the implementation plan. Prepares a logical design walk through of the information flow, output, input, controls implementation. Reviews benefits, costs, target dates and system constraints.

4.3 Physical design

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

1. Design the physical system.
2. .Specify input and output media.
3. .Design the database and specify backup procedures.
4. Design physical information flow through the system and a physical designWalk through.
5. Plan system implementation.
6. .Prepare a conversion schedule and target date.
7. Determine training procedures, courses and timetable.
8. .Devise a test and implementation plan and specify any new hardware/software.

System Design MVC Model MVC Model

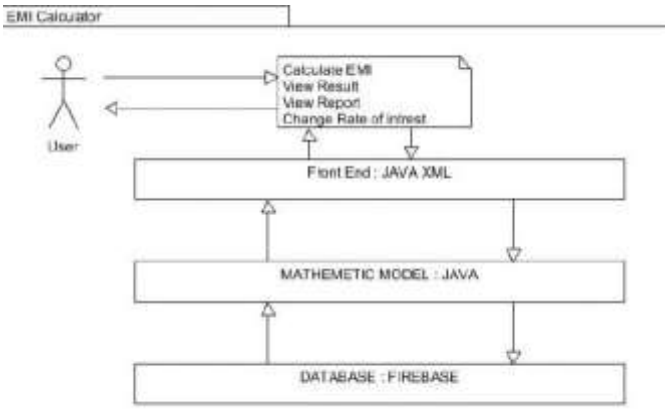
MVC model is one of the software architecture in the software engineering. Basically it is consisted of 2 part.

Model: package with the application's logic related to data as well as data processing method, which Directly manipulate data as well as record the action to be performed with the method of implementation.

View: the designed display. In general, it does not have any logic programming. But it will refresh the display constantly to show any message or annotation to the user. It also displays the result given by the model through Controller.

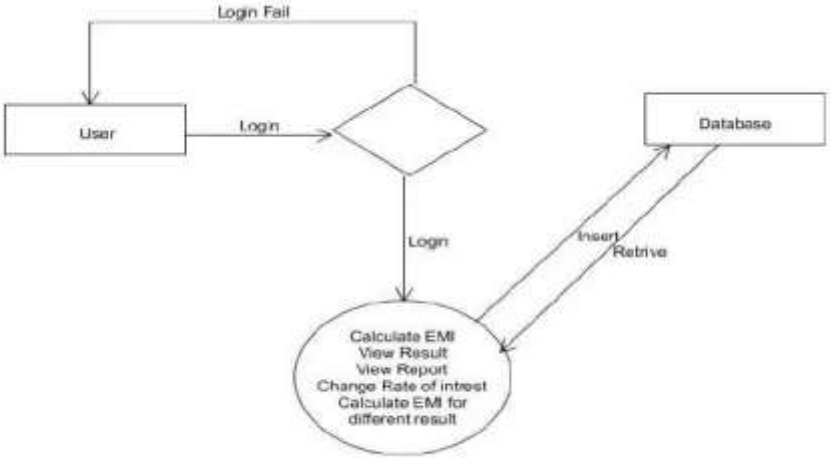
4.4 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



4.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step.



CHAPTER 5

TESTING

5.1 Test case for click Button

Input: process the given integer values.

Output : perform the mathematical operation and display the result.

Work Execution Exception

Failure Occurred while Executing com.andriod.bulid.gradle.internal.res

Resource Complier Runnable. **FAILURE!**

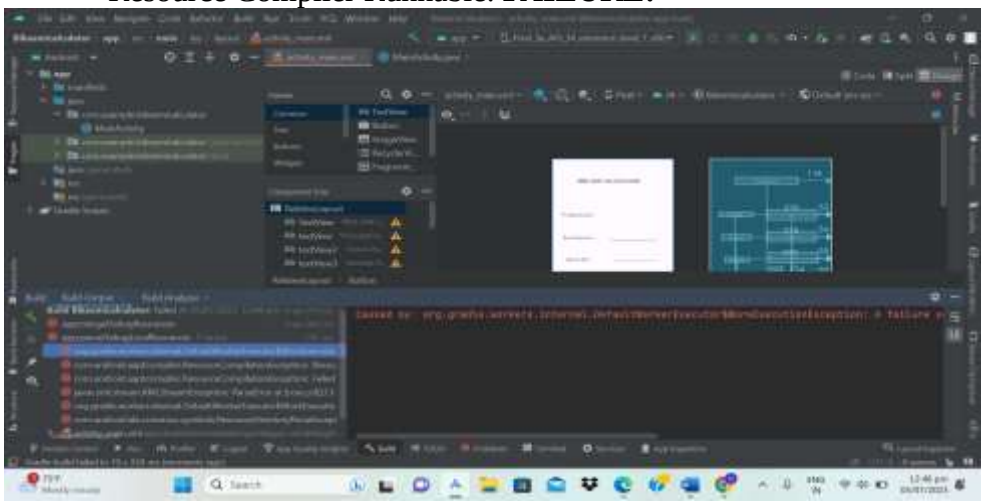


Fig 5.1Test case for button

5.2 Test case for Text view

Input : Accept the integer and fraction values[0-9]

Output : display exactly Emi amount to paid.

SUCCESSFUL!

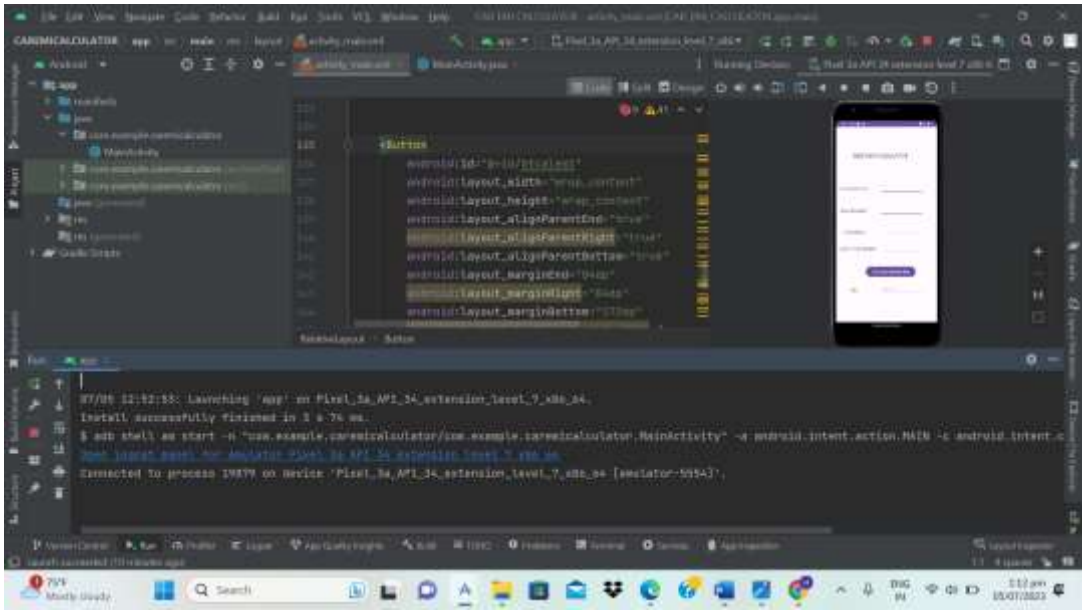


Fig 5.2 Test case for text view

CHAPTER 6

IMPLEMENTATION

Android .xml code

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="114dp"
        android:layout_marginRight="114dp"
        android:layout_marginBottom="634dp"
        android:text="CAR EMI CALCULATOR"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="270dp"
```

```
    android:layout_marginRight="270dp"
    android:layout_marginBottom="510dp"
    android:text="Principal Amount:" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="275dp"
    android:layout_marginRight="275dp"
    android:layout_marginBottom="424dp"
    android:text="Down Payment:" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="279dp"
    android:layout_marginRight="279dp"
    android:layout_marginBottom="341dp"
    android:text="Interest Rate:" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="234dp"
    android:layout_marginRight="234dp"
    android:layout_marginBottom="276dp"
```

```
    android:text="Loan Term (in months):" />

<EditText
    android:id="@+id/pa"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="52dp"
    android:layout_marginRight="52dp"
    android:layout_marginBottom="501dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    android:textSize="14sp" />

<EditText
    android:id="@+id/dnp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="54dp"
    android:layout_marginRight="54dp"
    android:layout_marginBottom="416dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    android:textSize="14sp" />

<EditText
    android:id="@+id/rt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
```

```
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginBottom="336dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    android:textSize="14sp" />

<EditText
    android:id="@+id/term"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginBottom="264dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    android:textSize="14sp" />


<Button
    android:id="@+id/btcalemi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="84dp"
    android:layout_marginRight="84dp"
    android:layout_marginBottom="172dp"
    android:text="Calculate Monthly EMI" />

<TextView
```

```
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="310dp"
        android:layout_marginRight="310dp"
        android:layout_marginBottom="115dp"
        android:text="EMI:" />

    <EditText
        android:id="@+id/res"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="47dp"
        android:layout_marginRight="47dp"
        android:layout_marginBottom="109dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="Result"
        android:text=""
        android:textSize="14sp" />

</RelativeLayout>
```

Android .java Code

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
Button calEMI;
TextView principal,downpay,rate,month,result;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        principal=findViewById(R.id.pa);
        downpay=findViewById(R.id.dnp);
        rate=findViewById(R.id.rt);
        month=findViewById(R.id.term);
        result=findViewById(R.id.res);
        calEMI=findViewById(R.id.btcalemi);
        calEMI.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String st1 = principal.getText().toString();
                String st2 = downpay.getText().toString();
                String st3 = rate.getText().toString();
```

```
            String st4 = month.getText().toString();
            float pr = Float.parseFloat(principal.getText().toString());
            float dp = Float.parseFloat(downpay.getText().toString());
            float interest = Float.parseFloat(rate.getText().toString());
            int n = Integer.parseInt(month.getText().toString());
            float p = pr-dp;
            float r = interest/100;
            float emi = calculateEMI(p,r,n);
            result.setText(""+emi);
        }
    });
  }

  public float calculateEMI(float p, float r, int n) {
    float k = p * r * ((power(r, n)) / ((power(r, n)) - 1));
    return (float)(k);
  }
  public float power (float r, int n){
    return (float)(Math.pow(1 + r, n));
  }
}
```

CHAPTER 7

SNAPSHOTS

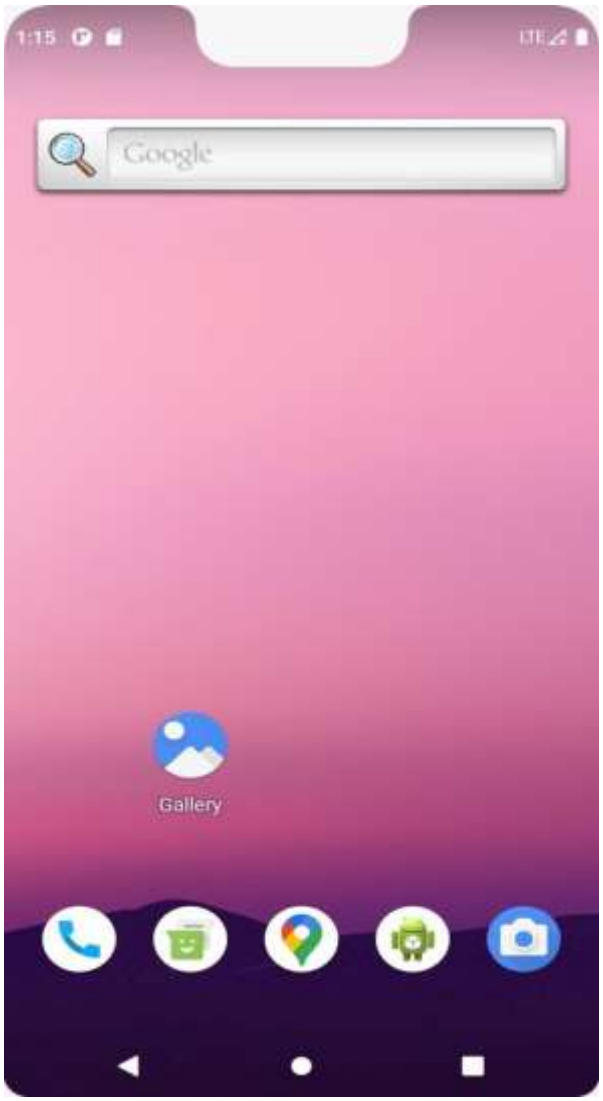


Fig 7.1 Display Page

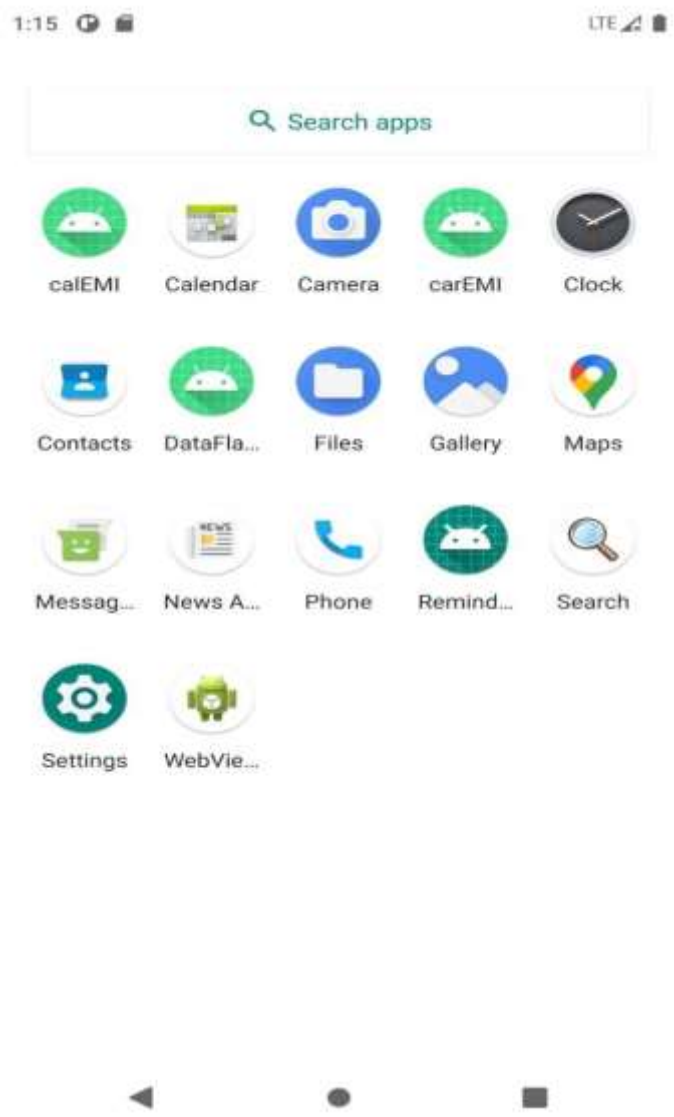


Fig 7.2 Application Page

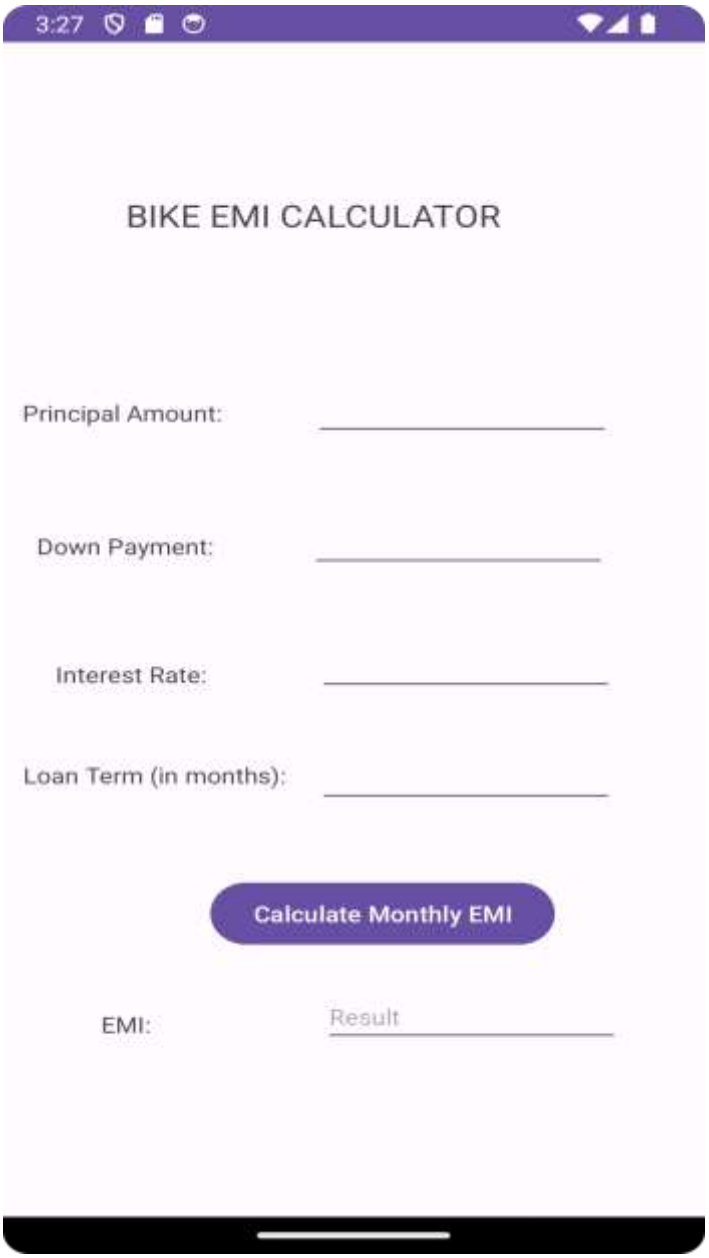


Fig 7.3 Bike Emi calculator

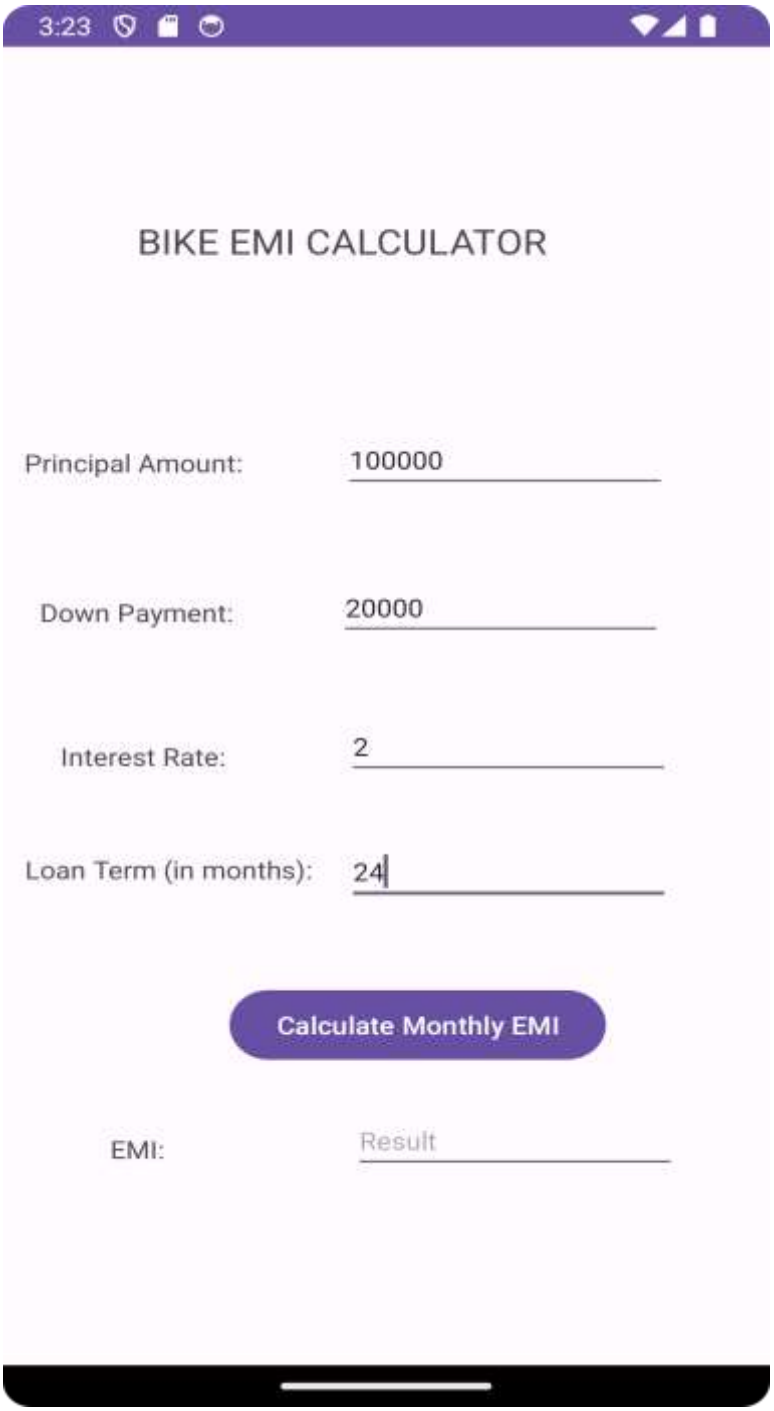


Fig 7.4 Bike EMI Calculator insert the values

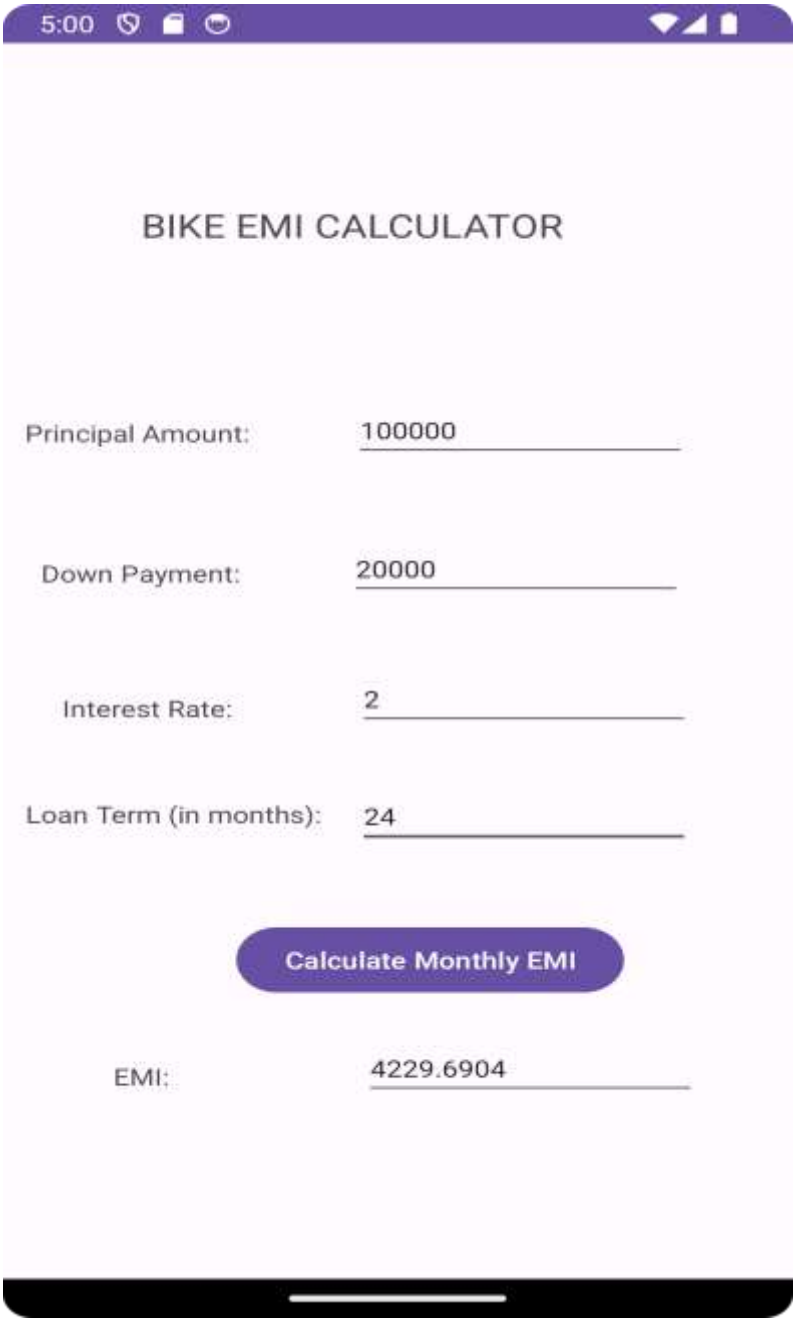


Fig 7.5 Bike EMI Calculator Display the result

CHAPTER 8

DISCUSSION

If we are talking about the meaning, it is easy to understand the EMI. As the name suggests, it is the amount to be paid to repay the outstanding loan inside a specified time span.

The EMI depends on some factors including the amount you have taken, interest rate, tenure, and more. But, do you know how the actually EMI has been calculated? The EMI is calculated by the formulae. But, sometimes the formulae may irritate you. So, it is better to have quite an easier tool to calculate EMI.

To calculate the perfect EMI, it asks you to fill data like Amount, Interest, and Tenure (in Years and Months). Then, tap on Calculate option and it shows you all the required details like EMI, Payable Interest, and more.

CHAPTER 9

CONCLUSION AND ENHANCEMENT

The complicated process of “EMI CALCULATOR” as simple as possible using Structured & Modular technique & Menu oriented interface. We have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. This Calculator could enhance effortless calculation of EMI and is most familiar among all the age groups. Intelligence can be a property of any purpose-driven decision maker. This basic idea has been suggested many times. An algorithm of calculating EMI has been presented and tested that works in efficient way. Overall the system works without any bugs.

BIBILIOGRAPHY

1. https://youtu.be/o5OivIH-wm8
2. https://developer.android.com/courses?gclid=CjwKCAjwzJmlBhBBEiwAEJyLu570Y8-UI1FnY838fJoG9bFIj7SCYQq6hYieJxd2PyY_02uCGI7TZxoCacoQAvD_BwE&gclsrc=aw.ds

TEXTBOOKS

1. https://www.academia.edu/36635920/Java_The_Complete_Reference_7th_Edition
2. https://www.w3schools.com/java/