NAME:SWETHA.V
REG NO:185002112

UIT1711-MOBILE APPLICATION DEVELOPMENT LAB

Swetha V

IT B

185002112

NAME:SWETHA.V
REG NO:185002112

EX NO:1,2 **Develop an application that uses GUI components, Font,Colours and**
DATE: **use Layout Managers and event listeners.**

AIM: To Develop an application that uses GUI components, Font and Colours and an application that uses Layout Managers and event listeners.

CODE:

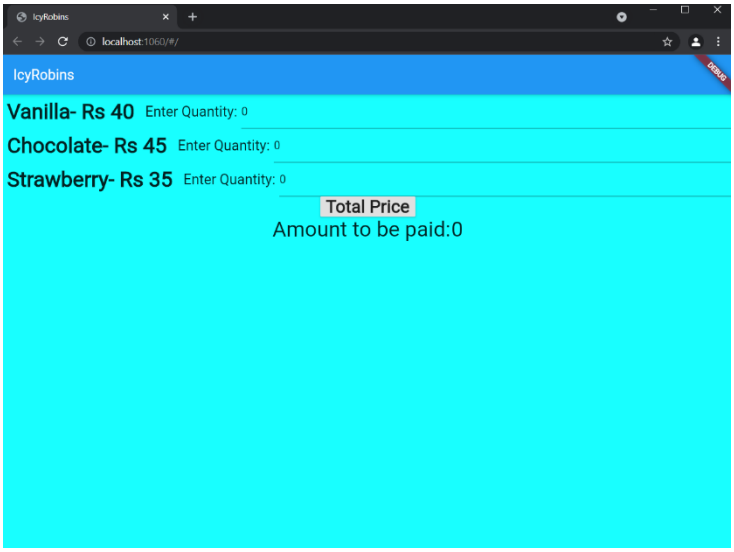
Main.dart

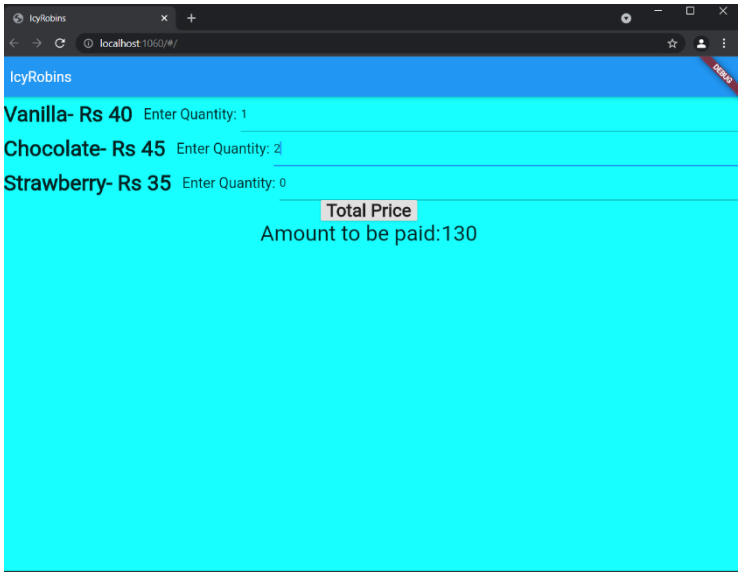
```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final appTitle = 'IcyRobins';

    return MaterialApp(
      title: appTitle,
      home: Scaffold(
        backgroundColor: Colors.cyanAccent,
        appBar: AppBar(
          title: Text(appTitle),
        ),
        body: AddTwoNumbers(),
      ),
    );
  }
}
class AddTwoNumbers extends StatefulWidget {
  @override
  _AddTwoNumbersState createState() => _AddTwoNumbersState();
}

class _AddTwoNumbersState extends State<AddTwoNumbers> {
  TextEditingController num1controller = new TextEditingController()..text="0";
  TextEditingController num2controller = new TextEditingController()..text="0";
  TextEditingController num3controller = new TextEditingController()..text="0";


  String result = "0";
  @override
  Widget build(BuildContext context) {
```

```
  return Container(
    child: Column(
      children: <Widget>[
        Row(
          children: <Widget>[

            Text(" Vanilla- Rs 40  ",style:TextStyle(fontWeight: FontWeight.bold,fontSize:
30)),
            Text("Enter Quantity: ",style: TextStyle(fontSize: 20),),
            new Flexible(
              child: new TextField(
                keyboardType: TextInputType.number,
                controller: num1controller,
              ),
            ),
          ],
        ),
        Row(
          children: <Widget>[
            Text(" Chocolate- Rs 45  ",style:TextStyle(fontWeight: FontWeight.bold,fontSize:
30)),
            Text("Enter Quantity: ",style: TextStyle(fontSize: 20),),
            new Flexible(
              child: new TextField(
                keyboardType: TextInputType.number,
                controller: num2controller,
              ),
            ),
          ],
        ),
        Row(
          children: <Widget>[
            Text(" Strawberry- Rs 35  ",style:TextStyle(fontWeight: FontWeight.bold,fontSize:
30)),
            Text("Enter Quantity: ",style: TextStyle(fontSize: 20),),
            new Flexible(
              child: new TextField(
                keyboardType: TextInputType.number,
                controller: num3controller,
              ),
            ),
          ],
        ),

        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
```

```
            RaisedButton(
              child: Text("Total Price",style: TextStyle(fontSize: 25,fontWeight:
FontWeight.bold),),
              onPressed : () {
                setState(() {
                  int sum = int.parse(num1controller.text)*40 + int.parse(num2controller.text)*45
+int.parse(num3controller.text)*35;
                  result = sum.toString();
                });
              },
            )
          ],
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text("Amount to be paid:",
              style: TextStyle(
                fontSize: 30,
              ),),
            Text(result,
              style: TextStyle(
                fontSize: 30,
              ),),
          ],
        ),
      ],
    ),
  );
 }

}
```

OUTPUT:





RESULT:

Thus the application that uses GUI components, Font and Colours and an application that uses Layout Managers and event listeners using flutter was created successfully.

EX NO:03 Develop a Native Calculator Application

DATE:

AIM: To develop a native calculator application.

CODE:

Main.dart

```
import 'package:flutter/material.dart';
import 'package:math_expressions/math_expressions.dart';

void main(){
  runApp(Calculator());
}
class Calculator extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Calculator',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: SimpleCalculator(),
    );
  }
}

class SimpleCalculator extends StatefulWidget {
  @override
  _SimpleCalculatorState createState() => _SimpleCalculatorState();
}

class _SimpleCalculatorState extends State<SimpleCalculator> {

  String equation = "0";
  String result = "0";
  String expression = "";
  double equationFontSize = 38.0;
  double resultFontSize = 48.0;

  buttonPressed(String buttonText){
    setState(() {
      if(buttonText == "C"){
        equation = "0";
        result = "0";
        equationFontSize = 38.0;
        resultFontSize = 48.0;
      }
```

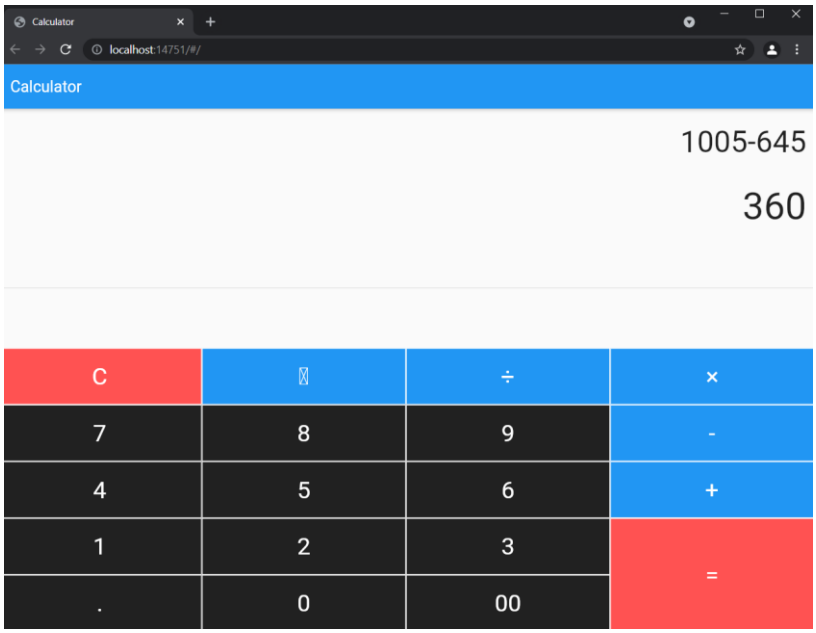
```
    else if(buttonText == "□"){
      equationFontSize = 48.0;
      resultFontSize = 38.0;
      equation = equation.substring(0, equation.length - 1);
      if(equation == ""){
        equation = "0";
      }
    }

    else if(buttonText == "="){
      equationFontSize = 38.0;
      resultFontSize = 48.0;

      expression = equation;
      expression = expression.replaceAll('×', '*');
      expression = expression.replaceAll('÷', '/');

      try{
        Parser p = Parser();
        Expression exp = p.parse(expression);

        ContextModel cm = ContextModel();
        result = '${exp.evaluate(EvaluationType.REAL, cm)}';
      }catch(e){
        result = "Error";
      }

    }

    else{
      equationFontSize = 48.0;
      resultFontSize = 38.0;
      if(equation == "0"){
        equation = buttonText;
      }else {
        equation = equation + buttonText;
      }
    }
  });
}

Widget buildButton(String buttonText, double buttonHeight, Color buttonColor){
  return Container(
    height: MediaQuery.of(context).size.height * 0.1 * buttonHeight,
    color: buttonColor,
    child: FlatButton(
```

```
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(0.0),
            side: BorderSide(
                color: Colors.white,
                width: 1,
                style: BorderStyle.solid
            )
        ),
        padding: EdgeInsets.all(16.0),
        onPressed: () => buttonPressed(buttonText),
        child: Text(
          buttonText,
          style: TextStyle(
              fontSize: 30.0,
              fontWeight: FontWeight.normal,
              color: Colors.white
          ),
        )
    ),
  );
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Calculator')),
    body: Column(
      children: <Widget>[
        Container(
          alignment: Alignment.centerRight,
          padding: EdgeInsets.fromLTRB(10, 20, 10, 0),
          child: Text(equation, style: TextStyle(fontSize: equationFontSize),),
        ),
        Container(
          alignment: Alignment.centerRight,
          padding: EdgeInsets.fromLTRB(10, 30, 10, 0),
          child: Text(result, style: TextStyle(fontSize: resultFontSize),),
        ),
        Expanded(
          child: Divider(),
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Container(
              width: MediaQuery.of(context).size.width * .75,
              child: Table(
                children: [
```

```
      TableRow(
        children: [
          buildButton("C", 1, Colors.redAccent),
          buildButton("□", 1, Colors.blue),
          buildButton("÷", 1, Colors.blue),
        ]
      ),

      TableRow(
        children: [
          buildButton("7", 1, Colors.black87),
          buildButton("8", 1, Colors.black87),
          buildButton("9", 1, Colors.black87),
        ]
      ),

      TableRow(
        children: [
          buildButton("4", 1, Colors.black87),
          buildButton("5", 1, Colors.black87),
          buildButton("6", 1, Colors.black87),
        ]
      ),

      TableRow(
        children: [
          buildButton("1", 1, Colors.black87),
          buildButton("2", 1, Colors.black87),
          buildButton("3", 1, Colors.black87),
        ]
      ),
      TableRow(
        children: [
          buildButton(".", 1, Colors.black87),
          buildButton("0", 1, Colors.black87),
          buildButton("00", 1, Colors.black87),
        ]
      ),
    ],
  ),
),
Container(
  width: MediaQuery.of(context).size.width * 0.25,
  child: Table(
    children: [
      TableRow(
        children: [
```

```
                buildButton("×", 1, Colors.blue),
              ]
            ),
            TableRow(
              children: [
                buildButton("-", 1, Colors.blue),
              ]
            ),
            TableRow(
              children: [
                buildButton("+", 1, Colors.blue),
              ]
            ),

            TableRow(
              children: [
                buildButton("=", 2, Colors.redAccent),
              ]
            ),
          ], ),     )   ],
      ),
    ],
  ),
 );
 }
}
```

OUTPUT:



RESULT:

Thus the native calculator application using flutter has been created successfully.

EX NO:04 Develop an Application that draws basic Graphical Primitives on the

DATE: screen

AIM: To write an application that draws basic graphical primitives on the screen.

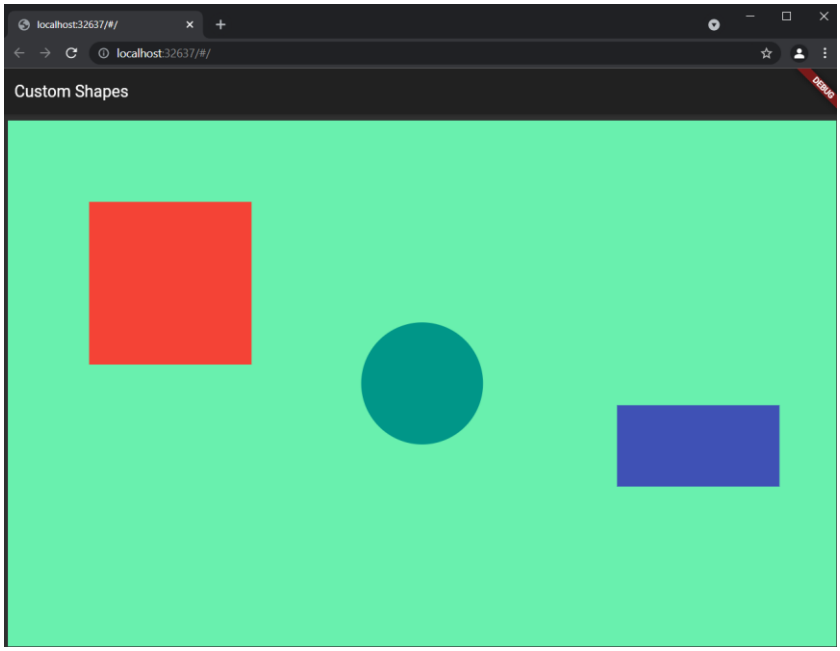
CODE:

Main.dart

```
import 'package:flutter/material.dart';

void main() => runApp(HomePage());

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        brightness: Brightness.dark,
        accentColor: Colors.teal,
      ),
      home: Scaffold(
        appBar: AppBar(
          title: Text('Custom Shapes'),
        ),
        body: Padding(
          padding: EdgeInsets.all(8.0),
          child: CustomPaint(
            painter: ShapesPainter(),
            child: Container(
              height: 700,
            ),
          ),
        ),
      ),
    );
  }
}

class ShapesPainter extends CustomPainter {
  @override
  void paint(Canvas canvas, Size size) {
    final paint = Paint();

    // set the paint color to be white
```

```
    paint.color = Colors.greenAccent;

    // Create a rectangle with size and width same as the canvas
    var rect = Rect.fromLTWH(0, 0, size.width, size.height);


    // draw the rectangle using the paint
    canvas.drawRect(rect, paint);

    paint.color = Colors.teal;

    // create a path
    var path = Path();
    path.lineTo(0, size.height);
    path.lineTo(size.width, 0);
    // close the path to form a bounded shape
    path.close();

    //canvas.drawPath(path, paint);

    // set the color property of the paint
    paint.color = Colors.teal;

    // center of the canvas is (x,y) => (width/2, height/2)
    var center = Offset(size.width / 2, size.height / 2);

    // draw the circle with center having radius 75.0
    canvas.drawCircle(center, 75.0, paint);

    paint.color = Colors.red;
    canvas.drawRect(Offset(100,100) & Size(200,200), paint);
    paint.color = Colors.indigo;
    canvas.drawRect(Offset(750,350) & Size(200,100), paint);

  }

  @override
  bool shouldRepaint(CustomPainter oldDelegate) => false;
}
```

OUTPUT:



RESULT:

Thus the flutter program to draw the basic primitives has been executed successfully.

EX NO:05 **Develop an application that makes use of database.**

DATE:

AIM: To develop an application that makes use of database

CODE:

Mainactivity.java

```
package com.example.ex05;

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();
    }
}
```

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"
    android:background="#97FAE3">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="95dp"
        android:layout_y="22dp"
        android:text="Student Details"
        android:textColor="#009688"
        android:textSize="30sp"
        android:textStyle="bold" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="110dp"
        android:text="Enter Rollno:"
        android:textColor="#000000"
        android:textSize="20sp" />
```

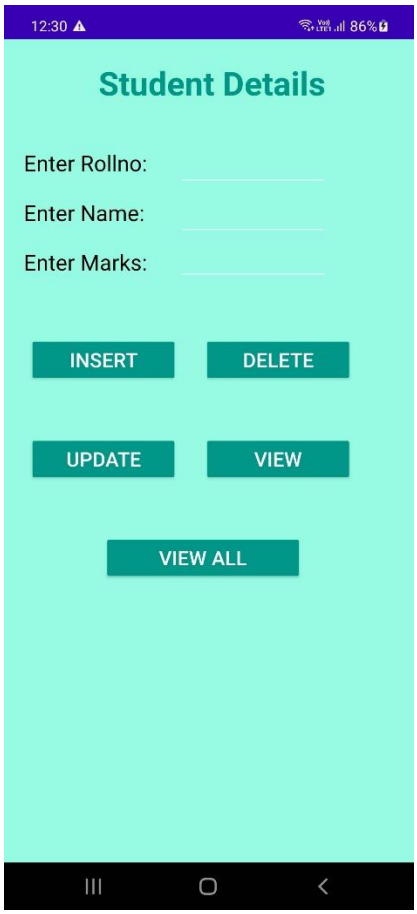
```
<EditText
    android:id="@+id/Rollno"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="100dp"
    android:inputType="number"
    android:textColor="#000000"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="160dp"
    android:text="Enter Name:"
    android:textColor="#000000"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Marks"
    android:layout_width="152dp"
    android:layout_height="wrap_content"
    android:layout_x="174dp"
    android:layout_y="198dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textColor="#000000" />

<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:textColor="#000000"
    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:textColor="#000000"
```

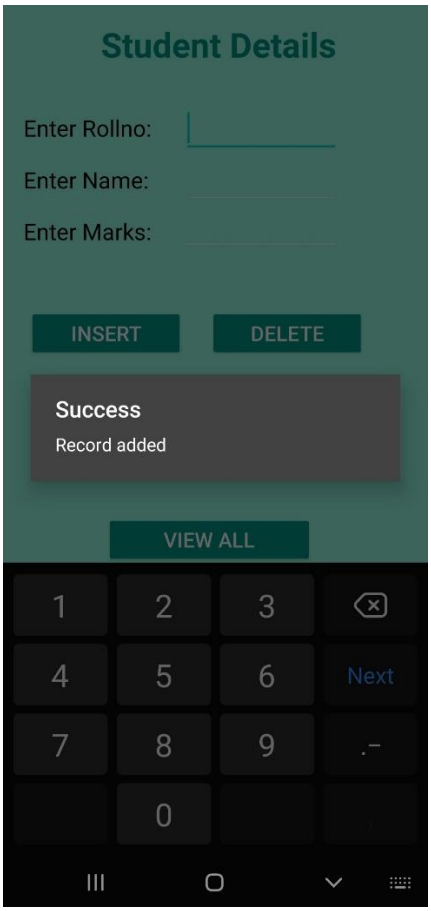
```
    android:textSize="20sp" />

<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:backgroundTint="#009688"
    android:text="Insert"
    android:textSize="20dp" />

<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:backgroundTint="#009688"
    android:text="Delete"
    android:textSize="20dp" />

<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:backgroundTint="#009688"
    android:text="Update"
    android:textSize="20dp" />

<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="400dp"
    android:backgroundTint="#009688"
    android:text="View"
    android:textSize="20dp" />

<Button
    android:id="@+id/ViewAll"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_x="100dp"
```

```
        android:layout_y="500dp"
        android:backgroundTint="#009688"
        android:text="View All"
        android:textSize="20dp" />

</AbsoluteLayout>
```

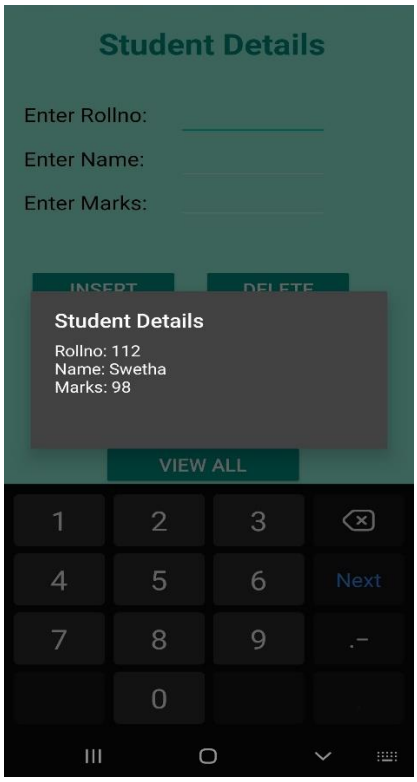
OUTPUT:

Insert

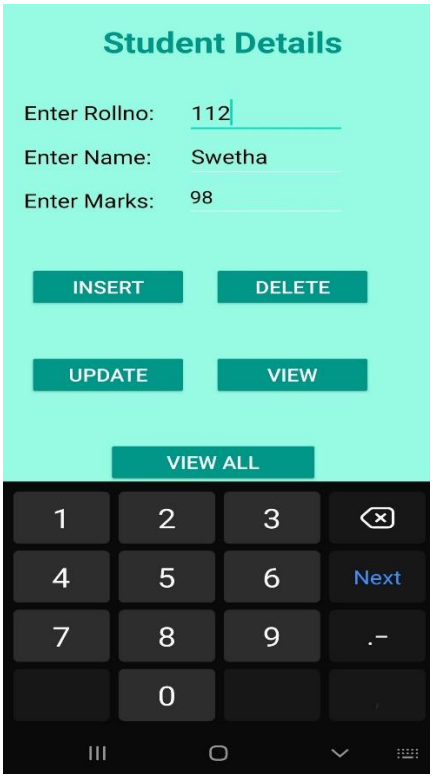




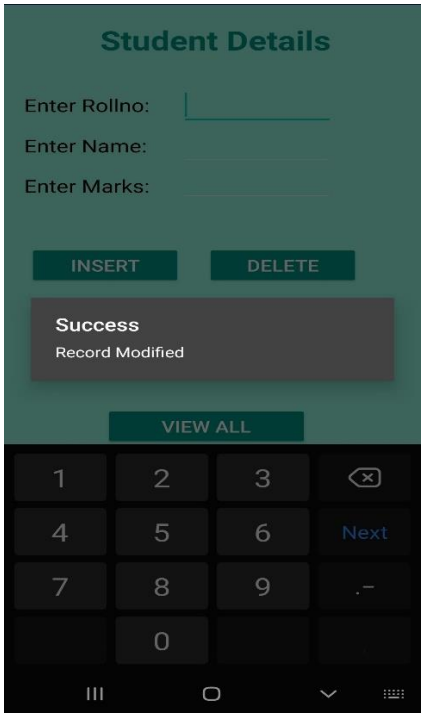
ViewAll

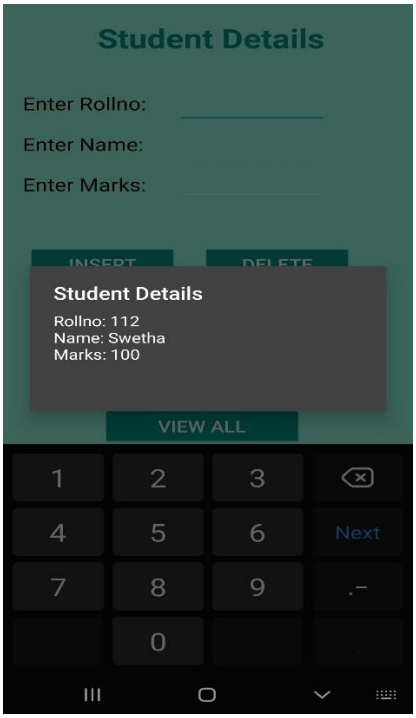


View

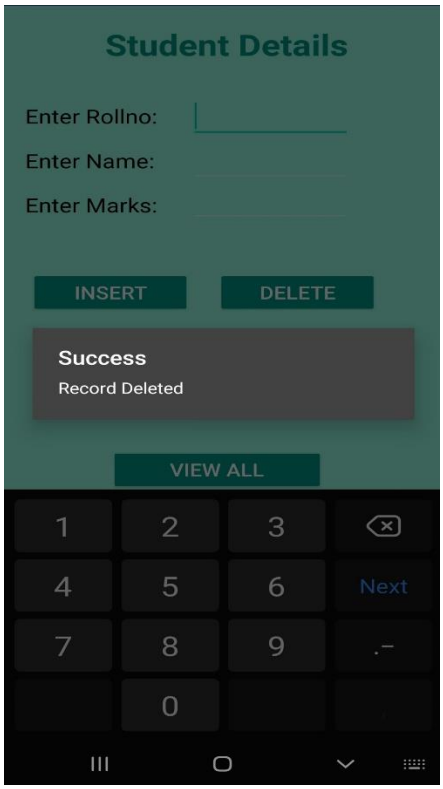


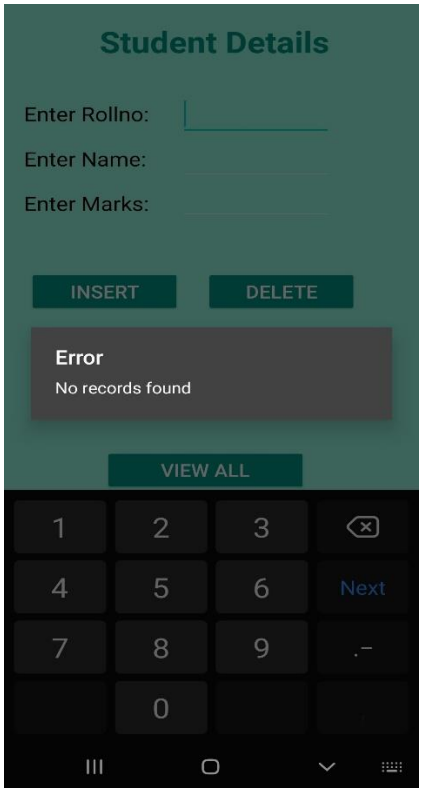
Update





Delete





RESULT:

Thus the application making use of database has been created successfully.

EX NO:06 Develop an application that makes use of RSS Feed.

DATE:

AIM: To develop an application that makes use of RSS Feed.

CODE:

Mainactivity.java

```
package com.example.ex06;

import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ListActivity
{
   List headlines;
   List links;

   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      new MyAsyncTask().execute();
   }

   class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
   {
      @Override
      protected ArrayAdapter doInBackground(Object[] params)
      {
         headlines = new ArrayList();
```

```
        links = new ArrayList();
        try
        {
           URL url = new URL("https://codingconnect.net/feed");
           XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
           factory.setNamespaceAware(false);
           XmlPullParser xpp = factory.newPullParser();

           // We will get the XML from an input stream
           xpp.setInput(getInputStream(url), "UTF_8");
           boolean insideItem = false;

           // Returns the type of current event: START_TAG, END_TAG, etc..
           int eventType = xpp.getEventType();
           while (eventType != XmlPullParser.END_DOCUMENT)
           {
              if (eventType == XmlPullParser.START_TAG)
              {
                 if (xpp.getName().equalsIgnoreCase("item"))
                 {
                    insideItem = true;
                 }
                 else if (xpp.getName().equalsIgnoreCase("title"))
                 {
                    if (insideItem)
                       headlines.add(xpp.nextText()); //extract the headline
                 }
                 else if (xpp.getName().equalsIgnoreCase("link"))
                 {
                    if (insideItem)
                       links.add(xpp.nextText()); //extract the link of article
                 }
              }
              else if(eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
              {
                 insideItem=false;
              }
              eventType = xpp.next(); //move to next element
           }

        }
        catch (MalformedURLException e)
        {
           e.printStackTrace();
        }
        catch (XmlPullParserException e)
```

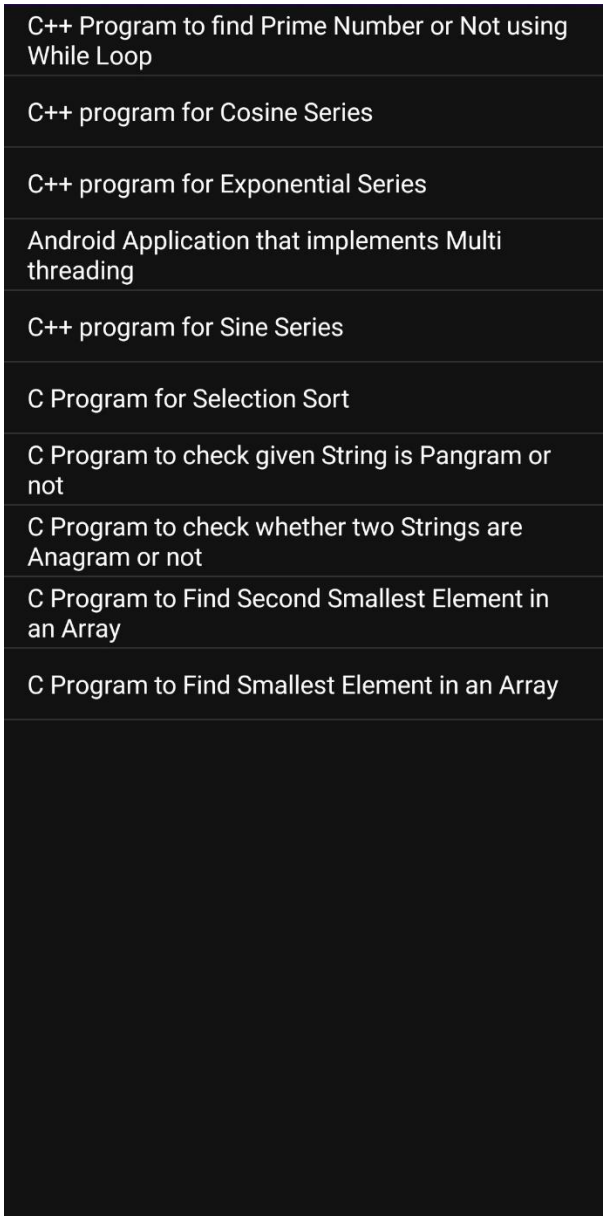
```
            {
               e.printStackTrace();
            }
            catch (IOException e)
            {
               e.printStackTrace();
            }
            return null;
         }
         protected void onPostExecute(ArrayAdapter adapter)
         {
            adapter = new ArrayAdapter(MainActivity.this, android.R.layout.simple_list_item_1,
headlines);
            setListAdapter(adapter);
         }
      }

      @Override
      protected void onListItemClick(ListView l, View v, int position, long id)
      {
         Uri uri = Uri.parse((links.get(position)).toString());
         Intent intent = new Intent(Intent.ACTION_VIEW, uri);
         startActivity(intent);
      }

      public InputStream getInputStream(URL url)
      {
         try
         {
            return url.openConnection().getInputStream();
         }
         catch (IOException e)
         {
            return null;
         }
      }
}
```

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" >
```

```
    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

OUTPUT:



RESULT:

Thus the application that uses RSS feed has been created successfully.