



# BMS

## INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru-54



RECORD OF PRACTICE

NAME	PRAGNA.M.V	UNIVERSITY SEAT NUMBER (USN)	1BY20S4505
PROGRAMME	M.Tech (ISE)	SEMESTER/SECTION	1
COURSE CODE	2050SL16	COURSE NAME	AI/ML



B.M.S. INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru - 560 064

# RECORD OF PRACTICAL WORK

NAME	PRAGNA-M-V	UNIVERSITY SEAT NUMBER (USN)	1BY20SC505
PROGRAMME	M.Tech (CSE)	SEMESTER/ SECTION	1
COURSE CODE	20SCSL17	COURSE NAME	



B.M.S. INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru - 560 064

**LABORATORY CERTIFICATE**

This is to Certify that Mr. / Ms. Pragna M.V.  
has Satisfactorily completed the course of experiments in Practical  
Algorithms and Advance Database Management Prescribed  
by the Visvesvaraya Technological University for.....  
Semester M.Tech..... Course in the Laboratory of the college  
in the year 2020- 2021

*[Signature]*

Head of the Department

*[Signature]*

Staff incharge of the Batch

Date : .....

Marks	
Maximum	Obtained
20	20

Name of the Candidate : Pragna M.V.

Roll No : ..... USN : 18Y20SC505

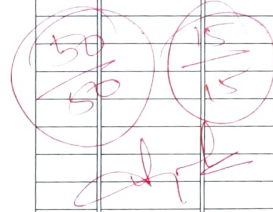
Pragna  
Signature of the Candidate

Rec 19/155 + Test 5/15 = Total 20/20

Particulars of the Experiments Performed  
CONTENTS

Expt No.	Date	Experiment	Marks Obtained	Page No
1.	28-01-21	INSERT AND RETRIEVAL OF BLOB OBJECT	10	01-08
2.	04-03-21	MULTIVALUED ATTRIBUTES & NESTED TABLES	10	09-18
3.	04-04-21	TRIGGER	10	19-22
4.	15-04-21	K-MEANS	10	23-27
5.	04-05-21	APRIORI ALGORITHM	10	28-32

DE  
7



## Particulars of the Experiments Performed

### CONTENTS

Expt No	Date	Experiment	Marks Obtained	Page No

Expt. No. 1

Page No. 1

Develop a database application to demonstrate entering and retrieving of BLOB and CLOB objects.

- a. Write a binary large object to a database as either binary or character data, depending on the type of the field in your data source. To write a BLOB value to database, issue the appropriate INSERT or UPDATE statement and pass the BLOB value as input parameter. If your BLOB is stored as text such as SQL server's text field, pass the BLOB as string parameter. If the BLOB is stored in binary format, such as SQL server's image field, pass an array type byte as a binary parameter.
- b. Once entering of BLOB and CLOB objects is done, retrieve them and display the results accordingly.

DATABASE CREATION

Create database LIBRARY ;

Use database LIBRARY ;

Teacher's Signature : \_\_\_\_\_

Database changed

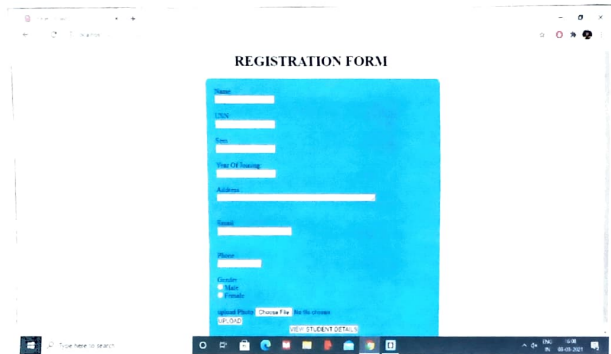
Table created

TABLE CREATION

```
create table STUDENT (Name varchar(20), USN  
varchar(10), Sem int, Year_of_Joining varchar(4),  
Address varchar(100), Email_id varchar(50), Phone  
varchar(10), Gender varchar(10), Photo longblob,  
Primary Key (USN));
```

Imageupload.php

```
<? php  
$dbName = "LIBRARY";  
$conn = mysql_connect ("localhost", "root", "",  
$dbName);  
  
if ($conn)  
{  
?  
?>  
<script >  
alert ("connection successfully");  
</script >  
<? php  
}  
else  
{  
?  
?>  
<script >
```



Expt. No. .... 1 .....

Date .....

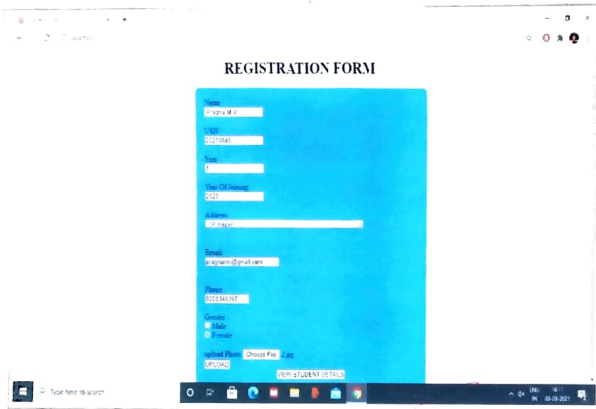
Page No. .... 03 .....

```

        alert ("Not connected successfully");
    </script >
<?php
?
mysql_connect_db ($conn, "UBRARY");
if (isset ($ _POST ['submit']))
?
    $name      = $ _POST ['Name'];
    $usn       = $ _POST ['USN'];
    $sem       = $ _POST ['Sem'];
    $email     = $ _POST ['Email-Id'];
    $year      = $ _POST ['Year-Of-Joining'];
    $address   = $ _POST ['Address'];
    $phone     = $ _POST ['Phone'];
    $gender    = $ _POST ['Gender'];
    $files     = $ _FILES ['file'];
    $filename  = $files ['name'];
    $fileext   = $files ['extension'];
    $filetmp   = $files ['tmp_name'];
    $filesize  = $files ['size'];
    $fileext   = explode ('.', $filename);
    $filecheck = strtolower (end ($fileext));
    $fileextarray = array ('jpg', 'jpeg', 'png');
    if (in_array ($filecheck, $fileextarray))
    ?
        $filedestination = 'upload/' . $filename ;

```

Teacher's Signature : \_\_\_\_\_



```
move_uploaded_file ($filetmp, $filedestination);
$insertquery = "INSERT INTO STUDENT (Name, USN,
Sem, Year of Joining, Address, Email Id, Phone,
Gender, Photo) VALUES (' $Name', '$USN', '$Sem',
'$Yoj', '$address', '$email', '$Phone', '$gender',
'$filedestination')";
```

```
$query = mysql_query ($conn, $insertquery);
if ($query)
{
```

```
}>
```

```
</script>
```

```
alert ("New record created successfully");
```

```
</script>
```

```
<?php
```

```
}
```

```
else
```

```
{
```

```
?>
```

```
</script>
```

```
alert ("Not inserted successfully");
```

```
</script>
```

```
<?php
```

```
}
```

```
?>
```

```
<!DOCTYPE html >
```

```
<html >
```

```
<head >
```



<h1> REGISTRATION FORM </h1>

<title> Image Upload </title>

</head>

<body>

<form method = "POST" action = "" enctype = "multipart/ form-data" >

Name <br>

<input type = "text" , name = "Name" , size = "15" />

USN <br>

<input type = "text" , name = "USN" , size = "15" />

Sem <br>

<input type = "text" , name = "Sem" size = "15" />

Year Of Joining <br>

<input type = "text" name = "Year Of Joining" size = "15" />

Address <br>

< text area cols = "50" rows = "1" id = "Address" >

name = "Address" value = "Address" > </ text area >

Email <br>

<input type = "email" id = "Email id" name = "Email id" />

Phone <br>

<input type = "text" name = "Phone" size = "10" />

Gender <br>

<input type = "radio" name = "Gender" value = "Male" />

Male <br>

<input type = "radio" name = "Gender" value = "Female" />

Female <br>

Upload Photo

```
<input type="file" name="file" value="" />
```

```
</div>
```

```
<button type="submit" name="submit">
```

```
    UPLOAD </button>
```

```
</div>
```

```
<button action="retrieve.php" type="submit">
```

```
    VIEW STUDENT DETAILS </button>
```

```
</form>
```

```
</body>
```

```
</html>
```

retrieve.php

```
<? php
```

```
    $dbName = "LIBRARY";
```

```
    $conn = mysqli_connect ("localhost", "root", "",
                                $dbName);
```

```
    if ($conn)
```

```
    {
```

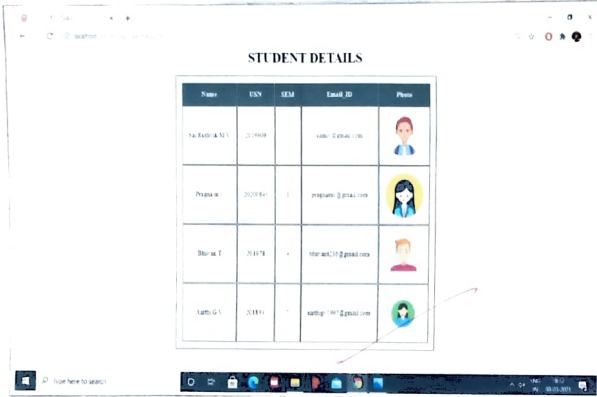
```
        echo "Net connected";
```

```
    }
```

```
>
```

```
<!DOCTYPE html >
```

```
<html>
```



```

<head>
  <title> Student Information </title>
</head>
<body>
  <h1> Student Information </h1>
  <table>
    <thead>
      <th> Name </th>
      <th> ENR </th>
      <th> Sem </th>
      <th> Email_ID </th>
      <th> Photo </th>
    </thead>
  </table>

```

```

<tbody>
  <? php
    mysql_select_db ($conn, "LIBRARY");
    if (isset ($_POST ['submit']))
    {
      $displayquery = "select * from STUDENT";
      $querydisplay = mysql_query ($conn,
        $displayquery);

      while ($result = mysql_fetch_array (
        $querydisplay))
      {
        ?>

```

```
<td>
```

```
<?php echo $result['name']; ?></td>
```

```
<td><?php echo $result['user']; ?></td>
```

```
<td><?php echo $result['sem']; ?></td>
```

```
<td><?php echo $result['email']; ?></td>
```

```
<td><img src = '<?php echo $result['photo']; ?>' height = '100px' width = '100px'></td>
```

```
</tr>
```

```
<?php
```

```
{
```

```
}
```

```
}
```

```
mysqli_close ($conn);
```

```
?>
```

```
</tbody>
```

```
</table>
```

```
</body>
```

```
</html>
```

$$7 + 3 = 10$$

Develop a database application to demonstrate the representation of multivalued attributes and the use of nested tables to represent complex objects. Write suitable queries to demonstrate their use.

Consider Purchase order Example: The example is based on typical business activity: managing customer orders. Need to demonstrate how the application might evolve from relational to object oriented-relational, and show you would reuse it from scratch using a pure object-oriented approach.

a. Show how to implement the schema - Implementing the Application under the Relational to object-oriented application on the top of this relational schema using object views.

```
create database purchaseorder;
use database purchaseorder;
```

```
create table customer (customer_id varchar(10),
customer_name varchar(100), customer_phone
int, primary key (customer_id));
```

```
create table (add_id varchar(10), add_cust_id varchar(10),
street varchar(100), landmark varchar(100),
city varchar(100), foreign key (add_cust_id) references
customer(customer_id));
```

Teacher's Signature: \_\_\_\_\_

DATABASE CREATED

DATABASE CHANGED

```
create table item (item_id varchar(10), item_name
varchar(100), item_price int, primary key (item_id));
```

```
create table order (order_id varchar(10), o_c_id varchar(10),
o_p_name varchar(500), ord_am int(10), ord_q int(10),
order_unit_p int(10), OR DATE DATE, OR D DATE Date,
primary key (order_id), foreign key (o_c_id)
references customer (customer_id));
```

```
create table purchase (purchase_id int(10) AUTO INCRE
MENT, pur_cust_id int(10), Product_name varchar(50),
unit_price int(10), quantity int(10), amount int(10),
P DATE date, primary key (purchase_id), foreign
key (pur_cust_id) references customer (cust_id));
```

Mainpage.java

```
package reviewer;
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JButton;
```



## MAIN PAGE

CUSTOMER DETAILS ENTRY

SHOPPING

ORDER DETAILS

```
MAINPAGE.java
```

```
package reviewer;
```

```
import java.awt.BorderLayout;  
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.border.EmptyBorder;  
import javax.swing.JLabel;  
import java.awt.Font;  
import javax.swing.JButton;  
import java.awt.Color;  
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;
```

```
public class MAINPAGE extends JFrame {  
  
    private JPanel contentPane;  
  
    /**  
     * Launch the application.  
     */  
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    MAINPAGE frame = new  
MAINPAGE();  
                    frame.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
    }  
  
    /**  
     * Create the frame.  
     */  
    public MAINPAGE() {  
        JFrame frame=new JFrame();  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setBounds(100, 100, 450, 346);  
        contentPane = new JPanel();  
        contentPane.setBackground(new Color(255, 255, 255));  
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  
        setContentPane(contentPane);  
        contentPane.setLayout(null);  
  
        JLabel lblNewLabel = new JLabel("MAIN PAGE");  
        lblNewLabel.setFont(new Font("Times New Roman",  
Font.BOLD, 23));  
        lblNewLabel.setBounds(133, 0, 145, 68);  
        contentPane.add(lblNewLabel);  
    }  
}
```

JButton btnNewButton = new JButton("CUSTOMER DETAILS

ENTRY");

```
btnNewButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0)
```

```
        frame.setVisible(false);  
        dispose();  
        new CUSTOMER().setVisible(true);
```

```
    });  
    btnNewButton.setForeground(new Color(255, 255, 255));  
    btnNewButton.setBackground(new Color(0, 128, 0));  
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD,
```

11));

```
    btnNewButton.setBounds(23, 109, 368, 34);  
    contentPane.add(btnNewButton);
```

```
JButton btnNewButton_1 = new JButton("SHOPPING");  
btnNewButton_1.addActionListener(new ActionListener()
```

```
    {  
        public void actionPerformed(ActionEvent e) {  
            frame.setVisible(false);  
            dispose();  
            new SHOPPING().setVisible(true);
```

```
        });  
        btnNewButton_1.setForeground(new Color(255, 255,
```

255));

```
        btnNewButton_1.setBackground(new Color(0, 128, 0));  
        btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD,
```

11));

```
        btnNewButton_1.setBounds(23, 168, 368, 34);  
        contentPane.add(btnNewButton_1);
```

```
JButton btnNewButton_2 = new JButton("ORDER DETAILS");  
btnNewButton_2.addActionListener(new ActionListener()
```

```
    {  
        public void actionPerformed(ActionEvent e) {  
            frame.setVisible(false);  
            dispose();  
            new ORDER().setVisible(true);
```

```
        });  
        btnNewButton_2.setForeground(new Color(255, 255,
```

255));

```
        btnNewButton_2.setBackground(new Color(0, 128, 0));  
        btnNewButton_2.setFont(new Font("Tahoma", Font.BOLD,
```

11));

```
        btnNewButton_2.setBounds(23, 226, 368, 34);  
        contentPane.add(btnNewButton_2);
```

Date .....

Expt. No. .... 2 .....

Page No. .... 11 .....

```
public class Mainpage extends JFrame
```

```
{
```

```
    private JPanel contentPane;
```

```
    public static void main (String args[])
```

```
{
```

```
        EventQueue.invokeLater(new Runnable() {
```

```
            {
```

```
                public void Run ()
```

```
                {
```

```
                    try
```

```
                    {
```

```
                        MAINPAGE frame = new MAINPAGE ();
```

```
                        frame.setVisible(true);
```

```
                    }
```

```
                catch (Exception e) {
```

```
                    e.printStackTrace();
```

```
                }
```

```
            }
```

```
        });
```

```
    }
```

```
    public Mainpage ()
```

```
    {
```

```
        JFrame frame = new JFrame ();
```

```
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
```

```
        setBounds (100, 100, 450, 346);
```

```
        contentPane = new JPanel ();
```

Teacher's Signature : \_\_\_\_\_



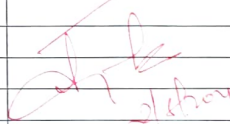
```

contentPane.setBackground(new Color(255, 255, 255));
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(new);
JLabel lblNewLabel = new JLabel("MainPage");
lblNewLabel.setFont(new Font("Times New Roman"));
lblNewLabel.setBounds(133, 0, 145, 65);
contentPane.add(lblNewLabel);
JButton btnNewButton = new JButton("CUSTOMER
                                DETAILS");
btnNewButton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        frame.setVisible(false);
        dispose();
        new CUSTOMER().setVisible(true);
    }
});
}
}

```

$$\frac{10}{10} \quad E + V$$

$$\frac{10}{10} = I + 3 = \frac{10}{10}$$



```
package reviewer;
```

```
import java.awt.BorderLayout;  
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.border.EmptyBorder;  
import javax.swing.JTextField;  
import javax.swing.JButton;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.awt.event.ActionEvent;  
import javax.swing.JLabel;  
import java.awt.Font;  
import java.awt.Component;
```

```
public class CUSTOMER extends JFrame {
```

```
    private JPanel contentPane;  
    private JTextField NAME;  
    private JLabel lblNewLabel;  
    private JLabel lblNewLabel_1;  
    private JLabel lblNewLabel_2;  
    private JTextField PHONE;  
    String names;  
    private JLabel lblNewLabel_3;  
    private JTextField STREET;  
    private JLabel lblNewLabel_4;  
    private JTextField LANDMARK;  
    private JLabel lblNewLabel_5;  
    private JTextField CITY;  
    private JLabel lblNewLabel_6;  
    private JTextField STATE;  
    private JLabel lblNewLabel_7;  
    private JTextField COUNTRY;  
    private JButton btnNewButton_1;
```

```
    /**  
     * Launch the application.  
     */
```

```
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {
```

```
                try {  
                    CUSTOMER frame = new  
CUSTOMER();  
                    frame.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        }  
    }  
}
```

```

}
/**
 * Create the frame.
 */
public CUSTOMER() {
    JFrame frame=new JFrame();
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 528);
    contentPane = new JPanel();
    contentPane.setFont(new Font("Times New Roman",
Font.PLAIN, 15));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    NAME = new JTextField();
    NAME.setBounds(147, 84, 255, 20);
    contentPane.add(NAME);
    NAME.setColumns(10);

    lblNewLabel = new JLabel("CUSTOMER DETAILS");
    lblNewLabel.setFont(new Font("Times New Roman",
Font.BOLD, 16));
    lblNewLabel.setBounds(113, 11, 200, 38);
    contentPane.add(lblNewLabel);

    lblNewLabel_1 = new JLabel("NAME");
    lblNewLabel_1.setFont(new Font("Times New Roman",
Font.PLAIN, 15));
    lblNewLabel_1.setBounds(65, 85, 72, 17);
    contentPane.add(lblNewLabel_1);

    lblNewLabel_2 = new JLabel("PHONE NO");
    lblNewLabel_2.setFont(new Font("Times New Roman",
Font.PLAIN, 15));
    lblNewLabel_2.setBounds(44, 133, 93, 14);
    contentPane.add(lblNewLabel_2);

    PHONE = new JTextField();
    PHONE.setBounds(147, 131, 255, 20);
    contentPane.add(PHONE);
    PHONE.setColumns(10);

    lblNewLabel_3 = new JLabel("STREET");
    lblNewLabel_3.setFont(new Font("Times New Roman",
Font.PLAIN, 15));
    lblNewLabel_3.setBounds(54, 175, 65, 20);
    contentPane.add(lblNewLabel_3);

    STREET = new JTextField();
    STREET.setBounds(149, 176, 253, 20);
    contentPane.add(STREET);
    STREET.setColumns(10);

    lblNewLabel_4 = new JLabel("LANDMARK");

```

CUSTOMER.java

```

public void actionPerformed(ActionEvent e)
{
    buy
    {
        Class jdbcName ("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection (
            "jdbc:mysql://localhost:3306/purchaseOrder",
            "root", "");
        Statement stmt = con.createStatement();
        String input = ("INSERT INTO 'customers' ('customerName',
            'customer_id', 'PHONE') VALUES ('" + NAME.getText() + ",
            ' NULL', '" + PHONE.getText() + "')");
        PreparedStatement ps = con.prepareStatement(input);
        ps.executeUpdate();
        Statement stmt = con.createStatement();
        String sql = "select * from 'customers' where
            PHONE = '" + PHONE.getText() + "'";
        ResultSet rs = stmt.executeQuery(sql);
        if (rs.next())
        {
            buy
            {
                names = rs.getString("customer_id");
                System.out.println(names);
            }
        }
    }
}

```

### CUSTOMER DETAILS

NAME	PRAGNA
PHONE NO	9008345385
STREET	CHANDRA LAYOUT
LANDMARK	TEMPLE
CITY	BELLARY
STATE	KARNATAKA
COUNTRY	INDIA
<input type="button" value="SUBMIT"/>	
<input type="button" value="BACK"/>	

```
catch (Exception d)
{
    System.out.println(d);
}
}
String input = (" INSERT INTO 'address' ('add id',
'add cb id', 'street', 'landmark', 'city', 'state', 'country',
values ( NULL, '"+ Names + "', " + STREET.getText () + "',
" + LANDMARK.getText () + "', " + CITY.getText () + "',
" + STATE.getText () + "', " + COUNTRY.getText () + "));");
PreparedStatement ps1 = con.prepareStatement (input),
ps1.executeUpdate ();
JOptionPane.showMessageDialog (null, SUBMITTED
SUCCESSFULLY.. ");
}
catch (Exception a)
{
    System.out.println (a);
JOptionPane.showMessageDialog (null, " Fill all details");
}
}
```

```
package reviewer;
```

```
import java.awt.BorderLayout;  
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.border.EmptyBorder;  
import java.awt.Color;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;
```

```
import java.awt.Font;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.awt.Canvas;  
import javax.swing.JTextField;  
import javax.swing.JButton;  
import java.awt.SystemColor;  
import java.awt.Button;  
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;
```

```
public class SHOPPING extends JFrame {
```

```
    private JPanel contentPane;  
    int i=0;  
    private JTextField PID;  
    private JTextField PQ;  
    /**  
     * Launch the application.  
     */
```

```
    String pname;  
    String aee;  
    String cid;  
    int amm;
```

```
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {
```

```
                try {  
                    SHOPPING frame = new  
SHOPPING();  
                    frame.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
    }
```

```
    /**  
     * Create the frame.  
     */
```

```

JFrame frame = new JFrame();
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(100, 100, 721, 449);
contentPane = new JPanel();
contentPane.setBackground(Color.WHITE);
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("ITEM LIST");
lblNewLabel.setFont(new Font("Times New Roman",
Font.BOLD, 23));
lblNewLabel.setBounds(78, 0, 150, 42);
contentPane.add(lblNewLabel);

Canvas canvas = new Canvas();
canvas.setBackground(Color.BLACK);
canvas.setBounds(338, 0, 15, 456);
contentPane.add(canvas);

JLabel lblNewLabel_1 = new JLabel("ITEM NAME");
lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD,
11));
lblNewLabel_1.setBounds(112, 53, 74, 23);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("ITEM UNIT PRICE");
lblNewLabel_2.setFont(new Font("Tahoma", Font.BOLD,
11));
lblNewLabel_2.setBounds(209, 53, 106, 23);
contentPane.add(lblNewLabel_2);

JLabel lblNewLabel_3 = new JLabel("ORDER");
lblNewLabel_3.setFont(new Font("Times New Roman",
Font.BOLD, 23));
lblNewLabel_3.setBounds(480, 0, 162, 42);
contentPane.add(lblNewLabel_3);

JLabel lblProductName = new JLabel("PRODUCT ID");
lblProductName.setBounds(359, 57, 74, 14);
contentPane.add(lblProductName);

JLabel lblNewLabel_4 = new JLabel("ITEM ID");
lblNewLabel_4.setFont(new Font("Tahoma", Font.BOLD,
11));
lblNewLabel_4.setBounds(27, 57, 46, 14);
contentPane.add(lblNewLabel_4);

JLabel id = new JLabel("");
id.setBounds(27, 86, 46, 14);
contentPane.add(id);

PID = new JTextField();
PID.setBounds(451, 54, 117, 20);
contentPane.add(PID);
PID.setColumns(10);

JLabel lblNewLabel_5 = new JLabel("PRODUCT NAME");
lblNewLabel_5.setBounds(359, 102, 96, 14);
contentPane.add(lblNewLabel_5);

JLabel lblNewLabel_6 = new JLabel("PRODUCT PRICE");
lblNewLabel_6.setBounds(359, 145, 96, 14);
contentPane.add(lblNewLabel_6);

JLabel lblNewLabel_7 = new JLabel("PRODUCT QUANTITY");
lblNewLabel_7.setBounds(359, 189, 122, 14);
contentPane.add(lblNewLabel_7);

JButton btnNewButton_2 = new JButton("BACK");
btnNewButton_2.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        frame.setVisible(false);
        dispose();
        new MAINPAGE().setVisible(true);
    }
});
btnNewButton_2.setBackground(new Color(0, 0, 0));
btnNewButton_2.setForeground(new Color(255, 255,
250));

btnNewButton_2.setBounds(490, 376, 89, 23);
contentPane.add(btnNewButton_2);

JLabel PNAME = new JLabel("");
PNAME.setBounds(494, 102, 129, 14);
contentPane.add(PNAME);

JLabel PPRICE = new JLabel("");
PPRICE.setBounds(490, 145, 78, 14);
contentPane.add(PPRICE);

JTextField phone = new JTextField();
phone.setBounds(490, 243, 158, 20);
contentPane.add(phone);
phone.setColumns(10);

PQ = new JTextField();
PQ.setBounds(491, 186, 151, 20);
contentPane.add(PQ);
PQ.setColumns(10);

JButton btnNewButton = new JButton("ADD CART");
btnNewButton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent arg0)
    {
        try
        {

```

```

class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/purchaseo
rder","root","");
Statement
String sql="Select * from item
WHERE item_id='"+PID.getText()+"'";
ResultSet
rs=stmt.executeQuery(sql);
if (rs.next())
{
try
{
PNAME.setText(rs.getString("item_name"));
PPRICE.setText(rs.getString("item_price"));
aee=rs.getString("item_price");
pname=rs.getString("item_name");
dd=PQ.getText();
aa=Integer.parseInt(aee);
ab=Integer.parseInt(dd);
amm=aa*ab;
}
catch(Exception e) {System.out.print(e);}
}
(System.out.print(e));
try
{
Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/purchaseo
rder","root","");
Statement
String sql="Select * from
CUSTOMER WHERE phone='"+phone.getText()+"'";
ResultSet
rs=stmt.executeQuery(sql);
if (rs.next())
{
try
{

```

```

ciders.getString("customer_id");
}
catch(Exception e) {System.out.print(e);}
}
catch(Exception e)
{System.out.print(e);}
}
});
btnNewButton.setForeground(SystemColor.text);
btnNewButton.setBackground(SystemColor.textHighlight);
btnNewButton.setBounds(589, 53, 106, 23);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("ORDER");
btnNewButton_1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/purchaseo
rder","root","");
Statement
stmt=con.createStatement();
String input=("INSERT INTO
'order' ('order_id', 'o_c_id', 'order_p_name', 'order_q', 'order_am',
'ordu_unit_p', 'OR_DATE', 'OR_D_DATE') VALUES (NULL, '"+cid+"',
 '"+pname+"', '"+PQ.getText()+"', '"+amm+'', '"+aee+'', NOW(),
NOW());");
PreparedStatement ps =con.prepareStatement(input);
ps.executeUpdate();
JOptionPane.showMessageDialog(null,
"SUBMITTED SUCCESSFULLY.....");
}
catch(Exception ee)
{System.out.print(ee);JOptionPane.showMessageDialog(null, "Fill all
the details.....");}
}
});
btnNewButton_1.setForeground(new Color(255, 255,
255));
btnNewButton_1.setBackground(new Color(0, 128, 0));
btnNewButton_1.setBounds(490, 342, 89, 23);
contentPane.add(btnNewButton_1);

JLabel aw3 = new JLabel("PHONE NUMBER");
aw3.setBounds(359, 246, 96, 14);
contentPane.add(aw3);

```



```

try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/purchaseo
rder","root","");
    Statement stmt=con.createStatement();
    String sql="Select * from item";
    ResultSet rs=stmt.executeQuery(sql);
    while (rs.next())
    {
        try
        {
            JLabel idS = new
JLabel("");
idS.setName("lblphoto"+rs.getString("item_id"));
idS.setBounds(33,
87+i, 46, 14);
idS.setText(rs.getString("item_id"));
contentPane.add(idS);
JLabel NAME = new
JLabel("");
NAME.setName("lblphoto"+rs.getString("item_name"));
NAME.setBounds(120,
87+i, 109, 14);
NAME.setText(rs.getString("item_name"));
contentPane.add(NAME);
JLabel PRICE = new
JLabel("");
PRICE.setName("lblphoto"+rs.getString("item_price"));
PRICE.setBounds(245,
87+i, 95, 14);
PRICE.setText(rs.getString("item_price"));
contentPane.add(PRICE);
i+=40;
}
catch(Exception e)
{System.out.print(e);}
}
}
catch(Exception e) {System.out.print(e);}
}
}

```

## Shopping.java

```

public void actionPerformed (new ActionListener ()
{
    try
    {
        class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:
mysql://localhost:3306/purchaseorder","root");
        Statement stmt = con.createStatement();
        String sql = "select * from item where item_id:
'" + PID.getText() + "'";
        ResultSet rs = stmt.executeQuery(sql);
        if (rs.next())
        {
            try
            {
                PName.setText(rs.getString("item_name"));
                Price.setText(rs.getString("item_price"));
                acc = rs.getString("item_price");
                pname = rs.getString("item_name");
                dd = PID.getText();
                aa = Integer.parseInt(acc);
                ab = Integer.parseInt(dd);
                amm = aa * ab;
            }
            catch (Exception e) {System.out.println(e);}
        }
    }
}

```

ITEM LIST			ORDER		
ITEM ID	ITEM NAME	ITEM UNIT PRICE	PRODUCT ID	<input type="text" value="211"/>	<input type="button" value="ADD CART"/>
1	MONITOR	5000	PRODUCT NAME	MOBILE	
211	MOBILE	12000	PRODUCT PRICE	12000	
212	TELEVISION	52000	PRODUCT QUANTITY	<input type="text" value="1"/>	
			PHONE NUMBER	<input type="text" value="9008345385"/>	
			<input type="button" value="ORDER"/>		
			<input type="button" value="BACK"/>		

```

public void actionPerformed (ActionEvent e)
{
    try
    {
        Class.forName ("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection ("jdbc:
mysql://localhost:3306/yourdatabase","root","");
        Statement stmt = con.createStatement ();
        String input = ("INSERT INTO 'orders' ('order_id',
'or_id', 'order_p_name', 'order_q', 'order am',
' order_unit_p', 'OR_DATE', 'OR_D DATE') values ('NULL',
'" + Pd + "', '" + pname + "', '" + PQ.getText () + "',
'" + amm + "', '" + tae + "', NOW (), NOW ())");
        PreparedStatement ps = con.prepareStatement (input);
        ps.executeUpdate ();
        JOptionPane.showMessageDialog (null, "Submitted
successfully ....");
    }
    catch (Exception ee)
    {
        System.out.println (ee);
        JOptionPane.showMessageDialog (null, "All the
details");
    }
}
}

```

```
package reviewer;
```

```
import java.awt.BorderLayout;  
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.border.EmptyBorder;  
import java.awt.Color;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;
```

```
import java.awt.Font;  
import java.awt.Image;
```

```
import javax.swing.JTextField;  
import javax.imageio.ImageIO;  
import javax.swing.ImageIcon;  
import javax.swing.JButton;  
import java.awt.event.ActionListener;  
import java.awt.image.BufferedImage;  
import java.io.File;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.awt.event.ActionEvent;
```

```
public class ORDER extends JFrame {
```

```
    private JPanel contentPane;  
    private JTextField textField;  
    String bb;
```

```
    /**  
     * Launch the application.  
     */
```

```
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {
```

```
            public void run() {
```

```
                try {
```

```
                    ORDER frame = new ORDER();
```

```
                    frame.setVisible(true);
```

```
                } catch (Exception e) {
```

```
                    e.printStackTrace();
```

```
                }
```

```
            }
```

```
        });
```

```
    }
```

```
    /**  
     * Create the frame.  
     */
```

```
    public ORDER() {
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        setBounds(100, 100, 450, 368);
```

```
        contentPane = new JPanel();
```

```

contentPane.setBackground(new Color(255, 255, 255));
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("TRACK ORDER");
lblNewLabel.setFont(new Font("Times New Roman",
Font.BOLD, 23));
lblNewLabel.setBounds(107, 11, 212, 23);
contentPane.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("ORDER ID");
lblNewLabel_1.setBounds(41, 60, 63, 14);
contentPane.add(lblNewLabel_1);

textField = new JTextField();
textField.setBounds(114, 57, 184, 20);
contentPane.add(textField);
textField.setColumns(10);

JLabel lblNewLabel_2 = new JLabel("PRODUCT NAME :");
lblNewLabel_2.setBounds(10, 104, 104, 14);
contentPane.add(lblNewLabel_2);

JLabel PNAME = new JLabel("");
PNAME.setBounds(124, 95, 169, 23);
contentPane.add(PNAME);

JLabel lblNewLabel_3 = new JLabel("ORDER AMOUNT :");
lblNewLabel_3.setBounds(10, 129, 104, 14);
contentPane.add(lblNewLabel_3);

JLabel OA = new JLabel("");
OA.setBounds(129, 129, 169, 14);
contentPane.add(OA);

JLabel lblNewLabel_4 = new JLabel("CUSTOMER NAME :");
lblNewLabel_4.setBounds(10, 208, 117, 14);
contentPane.add(lblNewLabel_4);

JLabel lblNewLabel_5 = new JLabel("ORDERED DATE :");
lblNewLabel_5.setBounds(10, 154, 104, 14);
contentPane.add(lblNewLabel_5);

JLabel lblNewLabel_6 = new JLabel("DELIVERY DATE :");
lblNewLabel_6.setBounds(10, 183, 104, 14);
contentPane.add(lblNewLabel_6);

JLabel lblNewLabel_7 = new JLabel("CUSTOMER ADDRESS :");
lblNewLabel_7.setBounds(10, 233, 130, 14);
contentPane.add(lblNewLabel_7);

JLabel OD = new JLabel("");
OD.setBounds(124, 154, 169, 14);
contentPane.add(OD);

```

```

JLabel ID = new JLabel("");
ID.setBounds(124, 183, 169, 14);
contentPane.add(ID);

JLabel CN = new JLabel("");
CN.setBounds(144, 208, 174, 14);
contentPane.add(CN);

JLabel ST = new JLabel("");
ST.setBounds(114, 27, 169, 14);
contentPane.add(ST);

JLabel LM = new JLabel("");
LM.setBounds(198, 233, 199, 14);
contentPane.add(LM);

JLabel CY = new JLabel("");
CY.setBounds(325, 23, 33, 14);
contentPane.add(CY);

JLabel SA = new JLabel("");
SA.setBounds(20, 272, 107, 14);
contentPane.add(SA);

JLabel CO = new JLabel("");
CO.setBounds(111, 272, 95, 14);
contentPane.add(CO);

JButton btnNewButton = new JButton("TRACK");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/purchases
            rder","root","");
            Statement stmt=con.createStatement();
            String sql="SELECT * FROM
            `order` WHERE order_id = "+textField.getText();
            ResultSet rs=stmt.executeQuery(sql);
            if(rs.next())
            {
                PNAME.setText(rs.getString("order_p_name"));
                OA.setText(rs.getString("order_am"));
                OD.setText(rs.getString("OR_DATE"));
                DD.setText(rs.getString("OR_D_DATE"));
            }
        }
    }
});

```

```
bb=rs.getString("o.c.id");
```

```
        }  
        catch (Exception e)  
{System.out.print(e); JOptionPane.showMessageDialog(null, "DATA NOT  
FOUND");}  
        try  
{  
Class.forName("com.mysql.jdbc.Driver");  
Connection  
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/purchaseo  
rder","root","");  
Statement  
stmt=con.createStatement();  
String sql="Select * from CUSTOMER  
c,address a where c.customer_id = "+bb+"&& a.add_cs_id="+bb;  
ResultSet rsl=stmt.executeQuery(sql);  
if(rsl.next())  
{  
CN.setText(rsl.getString("c.customer_name"));  
ST.setText(rsl.getString("a.Street"));  
LM.setText(rsl.getString("a.Landmark"));  
CY.setText(rsl.getString("a.city"));  
SA.setText(rsl.getString("a.state"));  
CO.setText(rsl.getString("a.country"));  
}  
}catch (Exception e)  
{System.out.print(e); JOptionPane.showMessageDialog(null, "DATA NOT  
FOUND");}  
}  
});  
btnNewButton.setForeground(new Color(255, 255, 255));  
btnNewButton.setBackground(new Color(0, 0, 128));  
btnNewButton.setBounds(308, 56, 89, 23);  
contentPane.add(btnNewButton);
```

Date .....

Expt. No. 2

Page No. 17

Order.java

```
public void actionPerformed (new ActionListener ()  
{  
try  
{  
Class.forName("com.mysql.jdbc.Driver");  
con = DriverManager.getConnection("jdbc:mysql://  
localhost:3306/purchaseorder","root","");  
Statement stmt = con.createStatement();  
String sql = "Select * from 'order' WHERE  
order_id = " + jTextField.getText();  
ResultSet rs = stmt.executeQuery(sql);  
if(rs.next())  
{  
PName.setText(rs.getString("order.p_name"));  
OA.setText(rs.getString("order.ora"));  
OD.setText(rs.getString("OR DATE"));  
DD.setText(rs.getString("OR D DATE"));  
bb = rs.getString("o.c.id");  
}  
}  
}  
catch (Exception e)  
{  
System.out.print(e);  
JOptionPane.showMessageDialog(null, "DATA NOT FOUND");  
}  
}
```

Teacher's Signature : \_\_\_\_\_

## TRACK ORDER

ORDER ID  TRACK

PRODUCT NAME : MOBILE  
 ORDER AMOUNT : 12000  
 ORDERED DATE : 2021-04-14  
 DELIVERY DATE : 2021-04-14  
 CUSTOMER NAME : PRAGNA  
 CUSTOMER ADDRESS : CHANDRA IEMPLE BANGLORE

KA                      IN

```

class.forName ("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/purchaseOrder", "root", "");
stmt = con.createStatement ();
String query = "select * from CUSTOMER c, address a
where c.customer_id = 'abb' && a.add_id = 'abb';
ResultSet rs1 = stmt.executeQuery (query);
if (rs1.next ())
{
    CN.setText (rs1.getString ("c.customer_name"));
    ST.setText (rs1.getString ("a.street"));
    LM.setText (rs1.getString ("a.landmark"));
    CY.setText (rs1.getString ("a.city"));
}
catch (Exception e)
{
    System.out.println (e); JOptionPane.showMessageDialog (Dialog
    (null, "Data Not Found"));
}
}

```

Design and develop a suitable student database application by considering appropriate attribute. couple of attributes to be maintained is the attendance of a student in each subject for which he/she has enrolled and Internal assessment using TRIGGER, write active rules to do the following

- whenever the attendance is updated check if the attendance is less than 25%, if so notify the head of the department concerned
- whenever the marks in an Internal assessment Test are entered, check if the marks are less than 40% if so, notify the head of the department concerned

```
create table STUDENT (S_Name varchar(100),
S_SUB_1_M INT, S_SUB_2_M INT, S_SUB_1_A INT,
S_SUB_2_A INT);
```

```
create table AVERAGE (S_Name varchar(100),
S_SUB_1_M INT, S_SUB_2_M INT, S_SUB_1_A INT,
S_SUB_2_A INT, AVG1_M INT, AVG1_A INT);
```

```
create Trigger TR_SH_AT on Student details
after Insert as
Insert into 'Student', 'average' (Name, 'AVG1_m', 'AVG1_A')
```

**STUDENT DETAILS**

STUDENT NAME	<input type="text" value="PRAGNA"/>
SUBJECT 1 MARKS	<input type="text" value="24"/>
SUBJECT 2 MARKS	<input type="text" value="34"/>
SUBJECT 1 Att	<input type="text" value="45"/>
SUBJECT 2 Att	<input type="text" value="45"/>

```

package 1119;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import prog1.Me.npage;

import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.Statement;
import java.awt.event.ActionEvent;

public class Student_Details extends JFrame {

    private JPanel contentPane;
    private JTextField s_NAME;
    private JTextField s_1_M;
    private JTextField s_2_M;
    private JTextField s_1_A;
    private JTextField s_2_A;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Student_Details frame = new
Student_Details();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Student_Details() {
        JFrame frame = new JFrame();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 484, 535);

```



```

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
contentPane.setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("STUDENT DETAILS");
lblNewLabel.setFont(new Font("Times New Roman",
Font.BOLD, 21));
lblNewLabel.setBounds(124, 11, 218, 47);
contentPane.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("STUDENT NAME");
lblNewLabel_1.setFont(new Font("Times New Roman",
Font.BOLD, 12));
lblNewLabel_1.setBounds(23, 98, 115, 14);
contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("SUBJECT 1 MARKS");
lblNewLabel_2.setFont(new Font("Times New Roman",
Font.BOLD, 12));
lblNewLabel_2.setBounds(23, 152, 133, 14);
contentPane.add(lblNewLabel_2);

JLabel lblNewLabel_3 = new JLabel("SUBJECT 2 MARKS");
lblNewLabel_3.setFont(new Font("Times New Roman",
Font.BOLD, 12));
lblNewLabel_3.setBounds(23, 208, 115, 14);
contentPane.add(lblNewLabel_3);

JLabel lblNewLabel_4 = new JLabel("SUBJECT 1 Att");
lblNewLabel_4.setFont(new Font("Times New Roman",
Font.BOLD, 12));
lblNewLabel_4.setBounds(23, 263, 115, 14);
contentPane.add(lblNewLabel_4);

JLabel lblNewLabel_5 = new JLabel("SUBJECT 2 Att");
lblNewLabel_5.setFont(new Font("Times New Roman",
Font.BOLD, 12));
lblNewLabel_5.setBounds(23, 313, 115, 14);
contentPane.add(lblNewLabel_5);

S_NAME = new JTextField();
S_NAME.setBounds(167, 95, 218, 20);
contentPane.add(S_NAME);
S_NAME.setColumns(10);

S_1_M = new JTextField();
S_1_M.setBounds(166, 149, 219, 20);
contentPane.add(S_1_M);
S_1_M.setColumns(10);

S_2_M = new JTextField();
S_2_M.setBounds(167, 205, 218, 20);
contentPane.add(S_2_M);
S_2_M.setColumns(10);

S_1_A = new JTextField();

```

```

S_1_A.setBounds(166, 260, 219, 20);
contentPane.add(S_1_A);
S_1_A.setColumns(10);

S_2_A = new JTextField();
S_2_A.setBounds(166, 310, 219, 20);
contentPane.add(S_2_A);
S_2_A.setColumns(10);

JButton btnNewButton = new JButton("SUBMIT");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student","r
oot","");
            Statement
stmt=con.createStatement();
            String input="INSERT INTO
'student_details' ('S_NAME', 'S_SUB_1_M', 'S_SUB_2_M',
'S_SUB_1_A', 'S_SUB_2_A') VALUES ('"+S_NAME.getText()+"",
 '"+S_1_M.getText()+"', '"+S_2_M.getText()+"', '"+S_1_A.getText()+"',
 '"+S_2_A.getText()+"');";
            //String input2="DELETE FROM
'student_details' where S_SUB_1_M>40 && S_SUB_2_M>40;";
            PreparedStatement
ps=con.prepareStatement(input);
            //PreparedStatement
ps1=con.prepareStatement(input2);
            ps.executeUpdate();
            //ps1.executeUpdate();

JOptionPane.showMessageDialog(null, "Submitted Successfully...");

        }catch(Exception a)
        {System.out.println(a);JOptionPane.showMessageDialog(null, "Fill all the
details ....");}
    }
});
btnNewButton.setBounds(10, 377, 435, 23);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("SHORTAGE
ATTENDANCE LIST");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        frame.setVisible(false);
        dispose();
        new Short_A().setVisible(true);
    }
});
btnNewButton_1.setBounds(10, 411, 435, 23);
contentPane.add(btnNewButton_1);

```

LIST):

```
JButton btnNewButton_2 = new JButton("SHORTAGE MARKS");  
btnNewButton_2.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        frame.setVisible(false);  
        dispose();  
        new Short_M().setVisible(true);  
    }  
});  
btnNewButton_2.setBounds(10, 446, 435, 23);  
contentPane.add(btnNewButton_2);
```

Date \_\_\_\_\_

Expt. No. 3

Page No. 20

```
values (New.S_Name, (New.S.SUB_1.M + New.S.SUB_2.M)  
        (New.S.SUB_1.A + New.S.SUB_2.A) / 2)
```

Student\_Details.java

```
public void actionPerformed(ActionEvent e)  
{  
    try  
    {  
        class.forName("com.mysql.jdbc.Driver");  
        Connection con = DriverManager.getConnection("jdbc:  
        mysql://localhost:3306/student","root","");  
        Statement stmt = con.createStatement();  
        String input = "INSERT INTO 'student' 'student detail'  
        ('S_Name', 'S.SUB_1.M', 'S.SUB_2.M', 'S.SUB_1.A',  
        'S.SUB_2.A') values (" + S_Name.getText() + ",  
        " + S_1.M.getText() + ", " + S_2.M.getText() + ",  
        " + S_1.A.getText() + ", " + S_2.A.getText() + ")";  
        Statement stmt2 = con.prepareStatement(input);  
        JOptionPane.showMessageDialog(null, "Submitted  
        successfully...");  
    }  
    catch (Exception e) { System.out.println(e);  
        JOptionPane.showMessageDialog(null, "Fill all details");  
    }  
}
```

Teacher's Signature: \_\_\_\_\_

MARKS SHORTAGE LIST

NAME	SUB 1 MAR...	SUB 2 MAR...	AVERAGE
DIVYA	23	23	23
SANJEEV	35	32	34
RITA	23	45	34

BACK

```

package UI9;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Short_M extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Short_M frame = new Short_M();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Short_M() {
        JFrame frame=new JFrame();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 539);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("MARKS SHORTAGE
LIST");
        lblNewLabel.setFont(new Font("Times New Roman",

```

```
Font.BOLD, 16));
```

```
lblNewLabel.setBounds(104, 11, 213, 43);  
contentPane.add(lblNewLabel);
```

```
JLabel lblNewLabel_1 = new JLabel("NAME");  
lblNewLabel_1.setBounds(26, 63, 46, 14);  
contentPane.add(lblNewLabel_1);
```

```
JLabel lblNewLabel_2 = new JLabel("SUB 1 MARKS");  
lblNewLabel_2.setBounds(104, 63, 78, 14);  
contentPane.add(lblNewLabel_2);
```

```
JLabel lblNewLabel_3 = new JLabel("SUB 2 MARKS");  
lblNewLabel_3.setBounds(217, 65, 78, 14);  
contentPane.add(lblNewLabel_3);
```

```
JLabel lblNewLabel_4 = new JLabel("AVERAGE");  
lblNewLabel_4.setBounds(339, 63, 63, 14);  
contentPane.add(lblNewLabel_4);
```

```
JButton btnNewButton = new JButton("BACK");  
btnNewButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        frame.setVisible(false);  
        dispose();  
        new
```

```
Student_Details().setVisible(true);  
    }  
});
```

```
btnNewButton.setBounds(156, 467, 89, 23);  
contentPane.add(btnNewButton);
```

```
try  
{
```

```
    Class.forName("com.mysql.cj.jdbc.Driver");  
    Connection  
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student",  
"root","");
```

```
    Statement stmt=con.createStatement();  
    String input=("SELECT * FROM `average` a  
, student_details s WHERE a.AVG_M<40 && a.NAME = s.S_NAME");  
    ResultSet rs=stmt.executeQuery(input);  
    while(rs.next())  
    {
```

```
        int i=0;  
        JLabel S_NAME = new  
JLabel(rs.getString("NAME"));
```

```
S_NAME.setBounds(36, 99+1, 78, 14);  
contentPane.add(S_NAME);
```

```
JLabel SUB1 = new  
JLabel(rs.getString("s.S_SUB_1_M"));  
SUB1.setBounds(104, 99+1, 46, 14);  
contentPane.add(SUB1);
```

```
JLabel SUB2 = new  
JLabel(rs.getString("s.S_SUB_2_M"));  
SUB2.setBounds(227, 99+1, 46, 14);  
contentPane.add(SUB2);
```

```
JLabel AM = new  
AM.setBounds(349, 99+1, 82, 14);  
contentPane.add(AM);  
i=i+40;
```

```
}
```

```
}catch(Exception a)
```

```
{System.out.println(a);JOptionPane.showMessageDialog(null, "DATA NOT  
FOUND");}
```

```
}
```

Short M. java

```
try
```

```
{
```

```
class.forName("com.mysql.jdbc.Driver");
```

```
Connection con = DriverManager.getConnection("jdbc:
mysql://localhost:3306/student", "root", "");
```

```
Statement stmt = con.createStatement();
```

```
String input = ("select * from 'average' a,
'student_details s where a.AVG-M < 40 &&
a.S-NAME = S.S-Name");
```

```
ResultSet rs = stmt.executeQuery(input);
```

```
while (rs.next())
```

```
{
```

```
int i = 0;
```

```
JLabel S-Name = new JLabel(rs.getString("Name"),
```

```
S-Name, setBounds(30, 99+i, 78, 14);
```

```
ContentPane.add(S-Name);
```

```
JLabel S-1 = new JLabel(rs.getString("S.S-S-1.m"),
```

```
S-1, setBounds(104, 99+i, 46, 14);
```

```
ContentPane.add(S-1);
```

```
i = i + 40;
```

```
}
```

```
}
```

```
catch (Exception a)
```

```
{
```

```
System.out.println(a);
```

```
}
```



### ATTENDANCE SHORTAGE LIST

NAME	SUB 1 ATT	SUB 2 ATT	AVERAGE
PURVI	64	59	62
DISHA	24	34	29
DIVYA	34	45	40

Back

```
package trig;
```

```
import java.awt.BorderLayout;  
import java.awt.EventQueue;  
import java.awt.Font;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;
```

```
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.border.EmptyBorder;  
import javax.swing.JButton;  
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;
```

```
public class Short_A extends JFrame {
```

```
    private JPanel contentPane;
```

```
    /**  
     * Launch the application.  
     */
```

```
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {
```

```
                    Short_A frame = new Short_A();  
                    frame.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });
```

```
    }  
  
    /**  
     * Create the frame.  
     */  
    public Short_A() {
```

```
        JFrame frame=new JFrame();  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setBounds(100, 100, 446, 472);  
        contentPane = new JPanel();  
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  
        setContentPane(contentPane);  
        contentPane.setLayout(null);
```

```
        JLabel lblNewLabel = new JLabel("ATTENDANCE SHORTAGE  
LIST");  
        lblNewLabel.setFont(new Font("Times New Roman",  
Font.BOLD, 16));  
        lblNewLabel.setBounds(70, 11, 247, 43);
```

```

        JLabel lblNewLabel_1 = new JLabel("NAME");
        lblNewLabel_1.setBounds(26, 63, 46, 14);
        contentPane.add(lblNewLabel_1);

        JLabel lblNewLabel_2 = new JLabel("SUB 1 ATT");
        lblNewLabel_2.setBounds(104, 63, 78, 14);
        contentPane.add(lblNewLabel_2);

        JLabel lblNewLabel_3 = new JLabel("SUB 2 ATT");
        lblNewLabel_3.setBounds(217, 65, 78, 14);
        contentPane.add(lblNewLabel_3);

        JLabel lblNewLabel_4 = new JLabel("AVERAGE");
        lblNewLabel_4.setBounds(339, 63, 63, 14);
        contentPane.add(lblNewLabel_4);

        JButton btnNewButton = new JButton("Back");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                frame.setVisible(false);
                dispose();
                new
                Student_Details().setVisible(true);
            }
        });
        btnNewButton.setBounds(172, 400, 89, 23);
        contentPane.add(btnNewButton);

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student",
            "root","");

            Statement stmt=con.createStatement();
            String input=("SELECT * FROM 'average' a
            ,student_details s WHERE a.AVG <85 && a.NAME = s.S_NAME");
            ResultSet rs=stmt.executeQuery(input);
            while(rs.next())
            {
                int i=0;
                JLabel S_NAME = new
                JLabel(rs.getString("NAME"));
                S_NAME.setBounds(30, 99+i, 78, 14);
                contentPane.add(S_NAME);

                JLabel SUB1 = new
                JLabel(rs.getString("s.S_SUB_1 ATT"));
                SUB1.setBounds(14, 99+i, 46, 14);
                contentPane.add(SUB1);

                JLabel SUB2 = new
                JLabel(rs.getString("s.S_SUB_2 ATT"));
                SUB2.setBounds(221, 99+i, 46, 14);
                contentPane.add(SUB2);

                JLabel AM = new
                JLabel(rs.getString("AVG_A"));
                AM.setBounds(349, 99+i, 92, 14);
                contentPane.add(AM);
                i=i+40;
            }
        } catch (Exception a)
        {
            System.out.println(a);JOptionPane.showMessageDialog(null, "DATA NOT
            FOUND");
        }
    }
}

```

Sheet A.java

```

public Sheet A ()
{
    try
    {
        class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:
            mysql://localhost:3306/student", "root", "");
        Statement stmt = con.createStatement();
        String input = "SELECT * from 'average' a,
            student details s where a.Avg1-A <= 82
            a.Name = s.S.Name";
        ResultSet rs = stmt.executeQuery(input);
        while (rs.next())
        {
            int p = 0;
            JLabel sName = new JLabel(rs.getString("Name"));
            sName.setBounds(30, 99 + p, 78, 14);
            contentPane.add(sName);
            JLabel sub1 = new JLabel(rs.getString("S.SUB1"));
            SUB1.setBounds(104, 99 + p, 46, 14);
            contentPane.add(sub1);
            p = p + 40;
        }
    }
    catch (Exception a) {
        System.out.println(a);
    }
}

```



Design, develop and execute a program to implement K-means clustering for finding association rules. Run the program against any image database available in public domain and discuss the results.

```
package KMeans;
import java.io.*;
import java.util.*;
import java.util.concurrent.ThreadLocalRandom;
```

```
public class KClusterer extends ReadDataset
```

```
{
```

```
    public KClusterer () { }
    public static void main (String args[])
```

```
{
```

```
    ReadDataset r1 = new ReadDataset ();
```

```
    r1.features.clear ();
```

```
    Scanner sc = new Scanner (System.in);
```

```
    System.out.println ("Enter the filename with path");
```

```
    String files = sc.next ();
```

```
    r1.read (files);
```

```
    int ex = 1;
```

```
    do {
```

```
        System.out.println ("Enter the no. of clusters");
```

```
        int k = sc.nextInt ();
```

```
        System.out.println ("Enter maximum iterations");
```

```
        int max_iterations = sc.nextInt ();
```

eclipse-workspace KMeans\KMeans\Clusterer.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Problems
  javadoc
  Declaration
  Console
  Terminal
  Console

K-Clusters (1) | Java Application | C:\Program Files\Java\jdk-10\bin\javaw.exe (08 Apr 2021, 4:13:51 pm)

Enter the filename with path

Enter the no. of clusters

Enter maximum iterations

Enter distance metric 1 or 2:

1. Euclidean  
2. Manhattan

Final Clustering of Data				
Feature1	Feature2	Feature3	Feature4	Cluster
5.5	4.2	1.4	0.2	2
7.0	3.2	4.7	1.4	3
6.0	3.0	4.8	1.8	3
6.3	3.3	6.0	2.5	3
6.4	3.2	4.5	1.5	3
6.5	3.0	5.8	2.2	3
5.1	3.5	1.4	0.2	2
5.4	3.9	1.7	0.4	0
5.4	3.4	1.5	0.4	0
5.8	2.8	5.1	2.4	3
6.7	3.0	5.8	1.7	3
6.3	2.9	5.6	1.8	3
4.9	3.1	1.5	0.1	1
4.6	3.1	1.5	0.2	1
5.7	2.5	5.0	2.0	3
6.9	3.2	5.7	2.3	3
7.7	2.8	6.7	2.8	3
6.0	2.7	5.1	1.6	3
5.8	3.6	1.4	0.2	2
5.6	2.8	4.9	2.0	3
5.7	3.0	4.2	2.0	3
4.8	3.1	1.6	0.2	1
6.3	2.3	4.4	1.3	3
5.2	4.1	1.5	0.1	2
6.7	3.1	4.7	1.5	3

Type here to search



Eclipse workspace: kMeans.src\KMeans\_K\_Clustering.java - Eclipse IDE  
 File Edit Source Refactor Navigate Search Project Run Window Help  
 Problems Javadoc Declaration Console Terminal  
 K\_Clustering (!) [Java Application] (C:\Program Files\Java\jdk-16\bin\javaw.exe (08-Apr-2021, 4:13:51 pm))  
 2. Manhattan

Final Clustering of Data

Feature1	Feature2	Feature3	Feature4	Cluster
5.6	4.2	1.4	0.2	2
7.0	3.2	4.7	1.4	3
6.0	3.0	4.8	1.8	3
6.3	3.3	6.0	2.5	3
6.4	3.2	4.5	1.5	3
6.5	3.0	5.8	2.2	3
5.1	3.5	1.4	0.2	2
5.4	3.9	1.7	0.4	0
5.4	3.4	1.5	0.4	0
5.8	2.8	5.1	2.4	3
6.7	3.0	5.0	1.7	3
6.3	2.9	5.6	1.8	3
4.9	3.1	1.5	0.1	1
4.6	2.1	1.5	0.2	1
5.7	2.5	5.0	2.0	3
6.9	3.2	5.7	2.3	3
7.7	2.8	6.7	2.0	3
6.0	2.7	5.1	1.6	3
5.0	3.6	1.4	0.2	2
5.6	2.8	4.9	2.0	3
5.7	3.0	4.2	1.2	3
4.8	3.1	1.6	0.2	1
6.3	2.3	4.4	1.3	3
5.2	4.1	1.5	0.1	2
6.7	3.1	4.7	1.5	3
5.8	2.7	3.9	1.2	3

Distance Metric: Euclidean  
 Iterations: 50  
 Number of Clusters: 4  
 HGSS: 17.988023529411767  
 Press 1 if you want to continue else press 0 to exit..

Type here to search

```

Map <Integer, double [] > centroids = new HashMap <> ();
double [] xi = new double [numberof features];
int xi=0;
for (int i=0; i<k; i++)
{
    xi = all_features.get(xi++);
    centroids.put(i, xi);
}

Map < double [], Integer > clusters = new HashMap <> ();
clusters = kmeans (all_features, distance, centroids, k);
double db [] = new double [numberof features];
int k=1;
for (int i=0; i<max_iterations; i++)
{
    for (int j=0; j<k; j++)
    {
        List < double [] > list = new ArrayList <> ();
        if (clusters.get (key) == j)
        {
            list.add (key);
        }
        db = centroidcalculator (list);
        centroids.put (j, db);
    }
}

```

```

double wcss = 0;
for (double [] key ; clusters.keySet ())
{
    if (clusters.get(key) == 2)
    {
        SSE += Math.pow (Distance.eclideanDistance
            (key, centroids.get (1)), 2);
    }
}

public static double [] centroidCalculator (List
    double [] > a)
{
    int count = 0;
    double sum = 0;
    double [] centroids = new double [ReadDataset.
        numberOfFeatures];
    for (int i = 0 ; i < ReadDataset.numberOfFeatures; i++)
    {
        sum = 0.0;
        count = 0;
        for (double [] x : a)
        {
            count++;
            sum = sum + x [i];
        }
        centroids [i] = sum / count;
    }
}

```

```

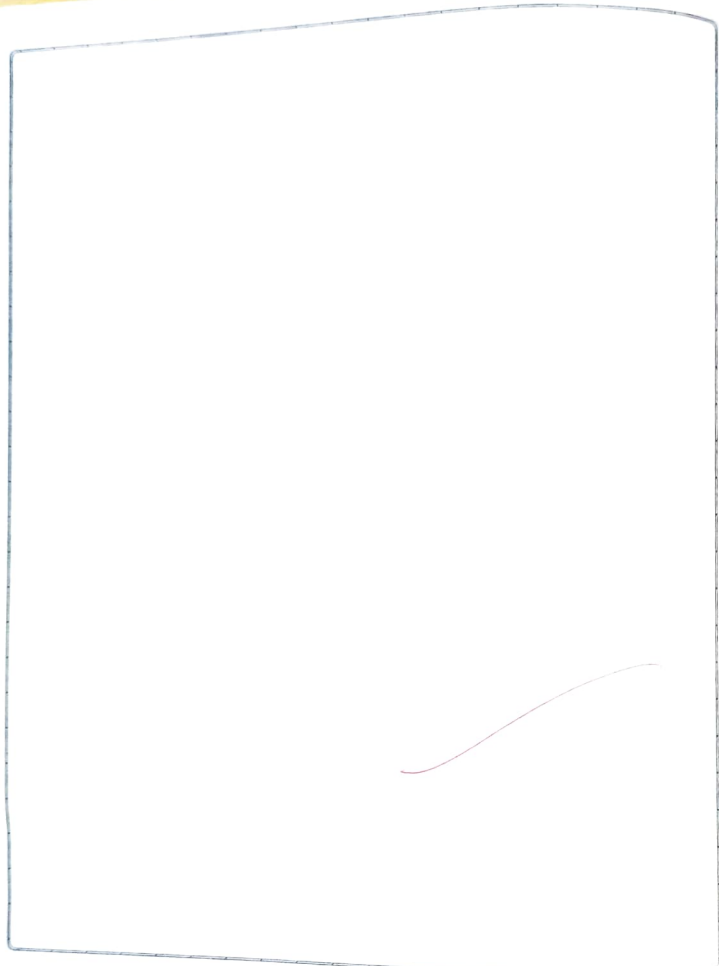
public static Map <double[], Integer> KMeans
    (List <double[] > features, int distance,
     Map <Integer> clusters = new HashMap<>(),
     int d=0
     double dist = 0
     for (double[] x: features)
     {
         for (int j=0; j<h; j++)
         {
             dist = distance.euclideanDistance(centers.get(j), x);
         }
         if (dist < minimum)
         {
             minimum = dist;
             k1 = j;
         }
     }
     clusters.put(x, k1);
 }
 return clusters;
 }
 }

```

$$E + V = \frac{10}{10}$$

$$= 1 + 3 = \frac{10}{10}$$

ad 1/2/2021



```
package KMeans;

import java.io.IOException;
import java.util.*;
import java.util.concurrent.ThreadLocalRandom;

public class K_Clusterer extends ReadDataset {

    public K_Clusterer() {
    }

    public static void main(String args[]) throws IOException {
        ReadDataset rl = new ReadDataset();
        rl.features.clear();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the filename with path");
        String file=sc.next();
        rl.read(file); //load data
        int ex=1;
        do{
            System.out.println("Enter the no. of clusters");
            int k = sc.nextInt();
            System.out.println("Enter maximum iterations");
            int max_iterations = sc.nextInt();
            System.out.println("Enter distance metric 1 or 2: 1.
Euclidean\n2. Manhattan");
            int distance = sc.nextInt();

            Map<Integer, double[]> centroids = new HashMap<>();

            double[] x1 = new double[numberOfFeatures];
            int r =0;
            for (int i = 0; i < k; i++) {

                x1=rl.features.get(r++);
                centroids.put(i, x1);

            }

            Map<double[], Integer> clusters = new HashMap<>();
            clusters = kmeans(rl.features, distance, centroids,
k);

            double db[] = new double[numberOfFeatures];

            for (int i = 0; i < max_iterations; i++) {
                for (int j = 0; j < k; j++) {
                    List<double[]> list = new ArrayList<>()
0;
                    for (double[] key : clusters.keySet())
                    {
                        if (clusters.get(key)==j) {
                            list.add(key);
                        }
                    }
                    db = centroidCalculator(list);
                }
            }
        }
    }
}
```

```

        centroids.put(j, db);
    }
    clusters.clear();
    clusters = kmeans(r1.features,distance,
centroids, k);
}

System.out.println("\nFinal Clustering of Data");

System.out.println("Feature1\tFeature2\tFeature3\tFeature4\tCluster");
for (double[] key : clusters.keySet()) {
    for (int i = 0; i < key.length; i++) {
        System.out.print(key[i] + "\t \t");
    }
    System.out.print(clusters.get(key) + "\n");
}

double wcss=0;

for(int i=0;i<k;i++){
    double sse=0;
    for (double[] key : clusters.keySet()) {
        if (clusters.get(key)==i) {

sse+=Math.pow(Distance.euclidianDistance(key, centroids.get(i)),2);
        }
    }
    wcss+=sse;
}
String dis="";
if(distance ==1)
    dis="Euclidean";
else
    dis="Manhattan";
System.out.println("\nDistance Metric: "+dis);
System.out.println("Iterations: "+max_iterations);
System.out.println("Number of Clusters: "+k);
System.out.println("WCSS: "+wcss);
System.out.println("Press 1 if you want to continue
else press 0 to
exit..");
ex=sc.nextInt();
}while(ex==1);
}

//method to calculate centroids
public static double[] centroidCalculator(List<double[]> a) {

    int count = 0;
    //double x[] = new
double[ReadDataset.numberofFeatures];
    double sum=0.0;
    double[] centroids = new
double[ReadDataset.numberofFeatures];

```

```

        for (int i = 0; i < ReadDataset.numberofFeatures; i++)
        {
            sum=0.0;
            count = 0;
            for(double[] x:a){
                count++;
                sum = sum + x[i];
            }
            centroids[i] = sum / count;
        }
        return centroids;
    }

}

//method for putting features to clusters and reassignment of
clusters.
public static Map<double[], Integer> kmeans(List<double[]>
features,int distance, Map<Integer, double[]> centroids, int k) {
    Map<double[], Integer> clusters = new HashMap<>();
    int k1 = 0;
    double dist=0.0;
    for(double[] x:features) {
        double minimum = 999999.0;
        for (int j = 0; j < x.length; j++) {
            if(distance==1){
                dist =
                Distance.euclidianDistance(centroids.get(j), x);
            }
            else if(distance==2){
                dist =
                Distance.manhattanDistance(centroids.get(j), x);
            }
            if (dist < minimum) {
                minimum = dist;
                k1 = j;
            }
        }
        clusters.put(x, k1);
    }
    return clusters;
}
}

```

```
package KMeans;

public class Distance {

    public Distance() {
        // TODO Auto-generated constructor stub
    }

    public static double euclidianDistance(double[] point1,
double[] point2) {
        double sum = 0.0;
        for(int i = 0; i < point1.length; i++) {
            //System.out.println(point1[i]+" "+point2[i]);
            sum += ((point1[i] - point2[i]) * (point1[i] -
point2[i]));
        }
        return Math.sqrt(sum);
    }

    public static double manhattanDistance(double point1[], double
point2[]){
        double sum = 0.0;
        for(int i = 0; i < point1.length; i++) {
            sum += (Math.abs(point1[i] - point2[i]));
        }
        return sum;
    }

}
```

```
package KMeans;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.Reader;
import java.nio.channels.ReadPendingException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;

public class ReadDataset {

    protected List<Double[]> features=new ArrayList<>();
    protected List<String> label=new ArrayList<>();
    protected static int numberOfFeatures;

    public List<Double[]> getFeatures()
    {
        return features;
    }

    public List<String> getLabel()
    {
        return label;
    }

    void read(String s) throws NumberFormatException, IOException
    {
        File file=new File(s);

        try {
            BufferedReader readFile=new BufferedReader(new
FileReader(file));
            String line;
            while((line=readFile.readLine()) != null)
            {
                String[] split = line.split(",");
                double[] feature = new double[split.length -
1];
                numberOfFeatures = split.length-1;
                for (int i = 0; i < split.length - 1; i++)
                    feature[i] =
Double.parseDouble(split[i]);
                features.add(feature);
                String labels = split[feature.length];
                label.add(labels);
            }
        } catch (FileNotFoundException e)
        {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```



```
void display()
```

```
Iterator<double[]> itr=features.iterator();  
Iterator<String> sitr=label.iterator();  
while(itr.hasNext())
```

```
    double db[]=itr.next();  
    for(int i=0; i<4;i++)
```

```
        System.out.print(db[i]+" ");
```

```
    String s=sitr.next();  
    System.out.println(s);  
    //System.out.println();  
}
```

Design, develop and execute a program to implement Apriori Algorithm for mining association rules. Run the program against any large database available in public domain and discuss the results.

```
import java.util.*;
```

```
public class Apriori
```

```
{
```

```
    public static void main (String args[])
```

```
{
```

```
    Scanner terminal = new Scanner (System.in);
```

```
    System.out.println (" Number of Transactions ");
```

```
    int NoOfTransactions = Integer.parseInt(terminal.  
                                            nextLine());
```

```
    System.out.println (" Enter the Transaction : ");
```

```
    ArrayList <ArrayList <String>> Transactions =
```

```
        new ArrayList <ArrayList <String>> ();
```

```
    ArrayList <ArrayList <String>> Transactions =
```

```
        new ArrayList <ArrayList <String>> ();
```

```
    for (int i=0 ; i < NoOfTransactions ; i++)
```

```
{
```

```
        ArrayList <String> Transaction = new ArrayList  
                                            <String> ();
```

```
        String str = terminal.nextLine();
```

```
        String arr[] = str.split (" ");
```

OUTPUT :-

Number of Transactions : 5

Enter Transactions :

a b c d

b c e f

a d e f

a e f

b d f

Minimum Support Count : 3

The Items that are most frequent Items : [c, f]

Date .....

Expt. No. ....

Page No. ...28...

```
for (int j=0; j< arr.length; j++)
    Transactions.add(arr[j]);
Transactions.add(Transactions);
Transactions.add(Transactions);
}

System.out.println("Minimum Support Count");
int minSupportCount = Integer.parseInt(JOptionPane.nextFloat());
ArrayList<String> Items = getUniqueItems(Transactions);
int x=0;
while (true)
{
    x++;
    ArrayList<Integer> supportCountList = new ArrayList<Integer>();
    for (ArrayList<String> Itemset : Itemsets)
    {
        int count = 0;
        for (ArrayList<String> Transaction : Transactions)
        {
            if (existsInTransaction(Itemset, Transaction))
                count++;
        }
        supportCountList.add(count);
    }
    ArrayList<ArrayList<String>> ItemsetsWithMinSupport =
        getItemsetsWithMinSupport(count, Itemsets);
}
```

Teacher's Signature : \_\_\_\_\_

```

if (itemSetWithMinSupportCount.size() == 0)
;
    System.out.println("The itemset that are most
        frequent itemset.");
    System.out.println("prev item set with min support count");
    break;
}
itemSet = getUniqueItems (itemSetWithMinSupportCount);
prevItemSetWithMinSupportCount = itemSetWithMinSupport
    Count;
}
}
private static ArrayList<String> getUniqueItems
    (ArrayList<ArrayList<String>> data)
{
    ArrayList<String> doReturn = new ArrayList<String>();
    for (ArrayList<String> transaction : data)
    {
        for (String : item - transaction)
        {
            if (!doReturn.contains(item)) doReturn.add(item);
        }
    }
    Collections.sort(doReturn);
    return doReturn;
}

```

```

private static ArrayList<ArrayList<String>> getItems2
(ArrayList<String> items, int number)
{
    if (number == 1)
    {
        ArrayList<ArrayList<String>> doReturn =
            new ArrayList<ArrayList<String>> ();
        for (String item : items)
        {
            ArrayList<String> alist = new ArrayList<String> ();
            alist.add(item);
            doReturn.add(alist);
        }
        return doReturn;
    }
    else
    {
        int size;
        ArrayList<ArrayList<String>> doReturn =
            new ArrayList<ArrayList<String>> ();
        for (int i = 0; i < size; i++)
        {
            ArrayList<String> _items = new ArrayList<String> ();
            for (String item : items)
            {
                _items.add(item);
            }
        }
    }
}

```

```
String thisItem = items.get(i);
```

```
for (int j=0; j<=i; j++)
```

```
{
```

```
    items.remove(0);
```

```
};
```

```
ArrayList <ArrayList <String >> permutationsBelow =  
    getItemsBelow(items, number-1);
```

```
for (ArrayList <String > alist: permutationsBelow)
```

```
{
```

```
    alist.add(thisItem);
```

```
    Collections.sort(alist);
```

```
    toReturn.add(alist);
```

```
};
```

```
}
```

```
return toReturn;
```

```
}
```

```
}
```

```
private static boolean existsInTransaction (ArrayList  
    <String > items, ArrayList <String > transaction)
```

```
{
```

```
    for (String item : items)
```

```
{
```

```
        if (!transaction.contains(item))
```

```
            return false;
```

```
};
```

```
return true;
```

```
}
```

Expt. No. \_\_\_\_\_

```

private static ArrayList<ArrayList<String>>
    getItemSetWithMinSupportCount
{
    ArrayList<ArrayList<String>> itemsets, ArrayList
    <Integer> count, int minSupportCount)
{
    ArrayList<ArrayList<String>> itemsets,
    ArrayList<Integer> count, int minSupportCount)
{
    ArrayList<ArrayList<String>> toReturn =
        new ArrayList<ArrayList<String>>();
    for (int i=0; i < count.size(); i++)
    {
        int c = count.get(i);
        if (c >= minSupportCount)
        {
            toReturn.add(itemsets.get(i));
        }
    }
    return toReturn;
}
}

```

$$\begin{array}{r}
 E + V \\
 7 + 3 = \frac{10}{10}
 \end{array}$$

step 1

Teacher's Signature : \_\_\_\_\_